

# The Cables.gl Book

A Comprehensive Guide to Visual Programming

Alexandre Rangel

2026

# Table of Contents

<b>1</b>	<b>Introduction to Cables.gl</b>	<b>30</b>
1.1	What is Cables.gl?	30
1.2	A (Brief) History of cables.gl	30
1.3	Why Use Cables.gl?	31
1.3.1	Visual Programming	31
1.3.2	Browser-Based	31
1.3.3	High Performance	31
1.3.4	Export Options	31
1.3.5	Extensible	31
1.4	Key Concepts	31
1.4.1	Operators (Ops)	31
1.4.2	Patches	31
1.4.3	Ports	31
1.4.4	Types of Ports	31
1.5	Featured Videos	31
1.5.1	Overview and Getting Started	31
1.5.2	More Resources	32
1.6	Getting Help	32
<b>2</b>	<b>Getting Started with Cables.gl</b>	<b>33</b>
2.1	Creating Your Account	33
2.2	The Interface	33
2.2.1	Main Areas	33
2.2.2	Key Interface Elements	33
2.3	Navigation Controls	33
2.4	Keyboard Shortcuts	33
2.4.1	Essential Shortcuts	33
2.4.2	Selection & Navigation	33
2.4.3	Organizing & Aligning	33
2.4.4	Working with Ops	34
2.4.5	Cables & Connections	34
2.4.6	View & Interface	34
2.4.7	Performance & Debugging	34
2.4.8	Pro Tips	34
2.5	Your First Patch	34
2.5.1	Step 1: Create the Render Pipeline	34
2.5.2	Step 2: Add a BasicMaterial	34
2.5.3	Step 3: Draw a Circle	34
2.5.4	Step 4: Add Color	35
2.5.5	Step 5: Animate It	35
2.6	Understanding the Flow	35
2.7	Saving Your Work	35
2.8	Tips for Beginners	35
2.9	Featured Videos	35
2.10	Common First-Patch Issues	35
2.10.1	"I don't see anything!"	35
2.10.2	"Colors aren't changing!"	35
2.10.3	"Animation isn't working!"	35
<b>3</b>	<b>2D Graphics in Cables.gl</b>	<b>36</b>
3.1	Introduction to 2D Drawing	36
3.2	Basic Shapes	36
3.2.1	Circle	36
3.2.2	Rectangle	36
3.2.3	RoundedRectangle	36
3.2.4	Polygon	36
3.2.5	Line / Lines	37
3.3	Color and Appearance	37
3.3.1	SetColor	37
3.3.2	SetAlpha	37
3.3.3	Gradients	37
3.4	Transformations	37
3.4.1	Transform	37
3.4.2	Transformation Order Matters!	37
3.4.3	Nested Transforms	38
3.5	Blending Modes	38
3.5.1	SetBlending	38
3.5.2	Depth Testing	38
3.6	Patterns and Repetition	39
3.6.1	IteratorLoop	39
3.6.2	ArrayIterator	39
3.7	Text Rendering	39
3.7.1	DrawText	39
3.7.2	TextTexture	39
3.8	Advanced Transformation Techniques	39
3.8.1	Matrix Transformations	39

3.8.2	Pivot Points . . . . .	39
3.8.3	Compound Transformations . . . . .	39
3.9	Interactive 2D Elements . . . . .	39
3.9.1	InteractiveRectangle . . . . .	39
3.9.2	Mouse Input . . . . .	40
3.9.3	Example: Interactive Color Picker . . . . .	40
3.10	Generative Art Techniques . . . . .	40
3.10.1	Feedback Loops . . . . .	40
3.10.2	Op Art and Moiré Patterns . . . . .	40
3.10.3	Procedural Pattern Generation . . . . .	41
3.11	Post-Processing Effects . . . . .	41
3.11.1	Image Composition . . . . .	41
3.11.2	TextureEffects for 2D . . . . .	41
3.11.3	Displacement Mapping . . . . .	41
3.12	Advanced Pattern Techniques . . . . .	41
3.12.1	Recursive Subdivision . . . . .	42
3.12.2	Particle Systems in 2D . . . . .	42
3.12.3	Grid-Based Automata . . . . .	42
3.13	Data Visualization . . . . .	42
3.13.1	Chart Generation . . . . .	42
3.13.2	Integration with ECharts . . . . .	42
3.13.3	Real-Time Data . . . . .	44
3.14	Complex Example Projects . . . . .	44
3.14.1	Example 4: Kaleidoscope Effect . . . . .	44
3.14.2	Example 5: Audio-Reactive Loading Animation . . . . .	44
3.14.3	Example 6: Data-Driven Weather Visualization . . . . .	44
3.14.4	Example 7: Feedback Tunnel Effect . . . . .	44
3.14.5	Example 8: Mouse Trail with Fade . . . . .	45
3.15	Performance Optimization . . . . .	45
3.15.1	Culling and Clipping . . . . .	45
3.15.2	Object Pooling . . . . .	45
3.15.3	Level of Detail (LOD) . . . . .	45
3.15.4	Batching Draw Calls . . . . .	45
3.16	Masking and Clipping . . . . .	46
3.16.1	Stencil Buffer Masking . . . . .	46
3.16.2	Alpha Mask Technique . . . . .	46
3.17	Color Theory in Practice . . . . .	46
3.17.1	Color Harmonies . . . . .	46
3.17.2	Gradient Creation . . . . .	46
3.18	Typography and Text Effects . . . . .	47
3.18.1	Dynamic Text . . . . .	47
3.18.2	Text as Texture . . . . .	47
3.18.3	Kinetic Typography . . . . .	47
3.19	Practical Examples . . . . .	47
3.19.1	Example 1: Pulsing Circle . . . . .	47
3.19.2	Example 2: Rotating Grid . . . . .	47
3.19.3	Example 3: Color Gradient Circle . . . . .	47
3.20	Debugging and Workflow Tips . . . . .	48
3.20.1	Visualizing Values . . . . .	48
3.20.2	Color Coding . . . . .	48
3.20.3	Naming Convention . . . . .	48
3.20.4	Comment Ops . . . . .	48
3.21	Common Patterns and Recipes . . . . .	48
3.21.1	Pattern: Circular Array . . . . .	48
3.21.2	Pattern: Wave Grid . . . . .	48
3.21.3	Pattern: Spiral . . . . .	48
3.21.4	Pattern: Responsive Grid . . . . .	49
3.22	Troubleshooting Common Issues . . . . .	49
3.22.1	"Shapes not appearing" . . . . .	49
3.22.2	"Performance is slow" . . . . .	49
3.22.3	"Colors look wrong" . . . . .	49
3.22.4	"Animation is jerky" . . . . .	49
3.23	Performance Tips . . . . .	49
3.24	Featured Videos . . . . .	49
3.24.1	Official Tutorials . . . . .	49
3.24.2	Additional Resources . . . . .	50
3.25	Exercises . . . . .	50
3.25.1	Beginner . . . . .	50
3.25.2	Intermediate . . . . .	50
3.25.3	Advanced . . . . .	50
3.26	Project Ideas . . . . .	50
4	<b>3D Graphics in Cables.gl</b>	51
4.1	Introduction to 3D . . . . .	51
4.2	The 3D Pipeline . . . . .	51
4.3	Cameras . . . . .	51
4.3.1	PerspectiveCamera . . . . .	51

4.3.2	OrthographicCamera . . . . .	51	4.7.1	BasicMaterial . . . . .	63
4.3.3	Orbit Controls . . . . .	51	4.7.2	LambertMaterial . . . . .	63
4.3.4	LookAt . . . . .	51	4.7.3	PhongMaterial . . . . .	64
4.3.5	Camera Animation . . . . .	52	4.7.4	PBRMaterial (Physically Based Rendering) . . . . .	64
4.3.6	Camera Shake Effect . . . . .	52	4.7.5	Material Blending . . . . .	64
4.3.7	First-Person Camera . . . . .	52	4.7.6	Animated Materials . . . . .	64
4.3.8	Camera Path Following . . . . .	52	4.7.7	Material Variants . . . . .	64
4.3.9	Camera Constraints . . . . .	52	4.7.8	Custom Shader Materials . . . . .	64
4.4	Lighting . . . . .	52	4.7.9	Material Instancing . . . . .	64
4.4.1	AmbientLight . . . . .	52	4.8	Transformations in 3D . . . . .	64
4.4.2	DirectionalLight . . . . .	52	4.8.1	Transform . . . . .	64
4.4.3	PointLight . . . . .	52	4.8.2	Matrix Operations . . . . .	65
4.4.4	SpotLight . . . . .	53	4.8.3	Hierarchical Transforms . . . . .	65
4.4.5	Shadow Mapping . . . . .	53	4.8.4	Constraint Systems . . . . .	65
4.4.6	Three-Point Lighting Setup . . . . .	53	4.8.5	IK (Inverse Kinematics) . . . . .	65
4.4.7	Image-Based Lighting (IBL) . . . . .	53	4.8.6	Physics-Based Transforms . . . . .	65
4.4.8	Light Probes . . . . .	53	4.8.7	Transform Caching . . . . .	65
4.4.9	Volumetric Lighting . . . . .	53	4.9	Rendering Techniques . . . . .	65
4.4.10	Light Animation . . . . .	53	4.9.1	Rendering Order . . . . .	65
4.5	Geometry and Meshes . . . . .	54	4.9.2	Multiple Render Passes . . . . .	66
4.5.1	Primitive Shapes . . . . .	54	4.9.3	Fog . . . . .	66
4.5.2	Loading 3D Models . . . . .	54	4.9.4	Screen-Space Ambient Occlusion (SSAO) . . . . .	66
4.5.3	Creating Custom Geometry . . . . .	54	4.9.5	Screen-Space Reflections (SSR) . . . . .	66
4.5.4	Procedural Geometry Generation . . . . .	54	4.9.6	Depth of Field . . . . .	66
4.5.5	Geometry Instancing . . . . .	55	4.9.7	Bloom . . . . .	66
4.5.6	Geometry Modifiers . . . . .	55	4.9.8	Motion Blur . . . . .	66
4.5.7	Boolean Operations . . . . .	55	4.9.9	Color Grading . . . . .	67
4.6	Real-Time Mesh Distortion . . . . .	55	4.9.10	Chromatic Aberration . . . . .	67
4.6.1	Understanding Vertex Manipulation . . . . .	55	4.9.11	Vignette . . . . .	67
4.6.2	Method 1: Node-Based Distortion . . . . .	56	4.9.12	Post-Processing Chain . . . . .	67
4.6.3	Method 2: JavaScript Custom Op for Mesh Distortion . . . . .	57	4.10	Scene Management . . . . .	67
4.6.4	Advanced: Animated Wall Distortion . . . . .	60	4.10.1	Scene Hierarchy . . . . .	67
4.6.5	Practical Example: Interactive Curved Wall . . . . .	61	4.10.2	Object Grouping . . . . .	68
4.6.6	Performance Optimization . . . . .	62	4.10.3	Layer System . . . . .	68
4.6.7	Advanced Techniques . . . . .	62	4.10.4	Culling and Optimization . . . . .	68
4.6.8	Common Use Cases . . . . .	63	4.10.5	LOD (Level of Detail) System . . . . .	68
4.6.9	Troubleshooting . . . . .	63	4.11	Practical Examples . . . . .	68
4.7	Materials . . . . .	63	4.11.1	Example 1: Rotating Cube . . . . .	68

4.11.2	Example 2: Lit Sphere with Orbit Controls . . . . .	68	4.16	Best Practices . . . . .	76																																																																																																																																																																														
4.11.3	Example 3: Loading a 3D Model . . . . .	68	4.17	Featured Videos . . . . .	76																																																																																																																																																																														
4.11.4	Example 4: Solar System . . . . .	69	4.18	Exercises . . . . .	76																																																																																																																																																																														
4.11.5	Example 5: Procedural Terrain . . . . .	69	4.18.1	Beginner . . . . .	76																																																																																																																																																																														
4.11.6	Example 6: Instanced Forest . . . . .	69	4.18.2	Intermediate . . . . .	76																																																																																																																																																																														
4.11.7	Example 7: Interactive 3D Scene . . . . .	69	4.18.3	Advanced . . . . .	76																																																																																																																																																																														
4.11.8	Example 8: Animated Character . . . . .	70	4.19	Project Ideas . . . . .	76																																																																																																																																																																														
4.11.9	Example 9: Particle System . . . . .	70																																																																																																																																																																																	
4.11.10	Example 10: Reflective Surface . . . . .	70																																																																																																																																																																																	
4.11.11	Example 11: Volumetric Fog . . . . .	70																																																																																																																																																																																	
4.11.12	Example 12: Dynamic Lighting Setup . . . . .	71	<b>5</b>	<b>Texturing in Cables.gl</b> . . . . .	<b>77</b>																																																																																																																																																																														
4.11.13	Example 13: Morphing Objects . . . . .	71	4.11.14	Example 14: Physics Simulation . . . . .	71	5.1	Introduction to Textures . . . . .	77	4.11.15	Example 15: Post-Processing Pipeline . . . . .	71	4.11.16	Example 16: Audio-Reactive 3D . . . . .	72	5.2	Loading Textures . . . . .	77	4.11.17	Example 17: Procedural City . . . . .	72	5.2.1	ImageTexture . . . . .	77	4.11.18	Example 18: Water Surface . . . . .	72	5.2.2	VideoTexture . . . . .	77	4.11.19	Example 19: Portal Effect . . . . .	72	5.2.3	WebcamTexture . . . . .	77	4.11.20	Example 20: Multi-Pass Rendering . . . . .	73	5.3	Texture Mapping . . . . .	77	4.12	Advanced Animation Techniques . . . . .	73	5.3.1	UV Coordinates . . . . .	77	4.12.1	Skeletal Animation . . . . .	73	5.3.2	UV Transform . . . . .	78	4.12.2	Morph Targets . . . . .	73	5.3.3	Tiling Textures . . . . .	78	4.12.3	Procedural Animation . . . . .	73	5.4	Advanced Texture Workflow (Production Mindset) . . . . .	78	4.12.4	Physics Animation . . . . .	73	5.4.1	Color Space: sRGB vs Linear (Why Your Colors Look "Off") . . . . .	78	4.12.5	Animation Blending . . . . .	74	5.4.2	Alpha (Transparency) Pitfalls . . . . .	79	4.13	Performance Optimization . . . . .	74	5.4.3	Filtering, Mipmaps, and Why Textures "Shimmer" . . . . .	79	4.13.1	General Tips . . . . .	74	5.4.4	Power-of-Two Sizes (and When It Matters) . . . . .	79	4.13.2	Advanced Optimization . . . . .	74	5.4.5	Aspect Ratio Correctness (Especially for Video) . . . . .	79	4.13.3	Performance Monitoring . . . . .	74	5.5	Advanced Techniques and Patch Recipes . . . . .	79	4.13.4	Adaptive Quality . . . . .	74	5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79	4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81
4.11.14	Example 14: Physics Simulation . . . . .	71	5.1	Introduction to Textures . . . . .	77																																																																																																																																																																														
4.11.15	Example 15: Post-Processing Pipeline . . . . .	71	4.11.16	Example 16: Audio-Reactive 3D . . . . .	72	5.2	Loading Textures . . . . .	77	4.11.17	Example 17: Procedural City . . . . .	72	5.2.1	ImageTexture . . . . .	77	4.11.18	Example 18: Water Surface . . . . .	72	5.2.2	VideoTexture . . . . .	77	4.11.19	Example 19: Portal Effect . . . . .	72	5.2.3	WebcamTexture . . . . .	77	4.11.20	Example 20: Multi-Pass Rendering . . . . .	73	5.3	Texture Mapping . . . . .	77	4.12	Advanced Animation Techniques . . . . .	73	5.3.1	UV Coordinates . . . . .	77	4.12.1	Skeletal Animation . . . . .	73	5.3.2	UV Transform . . . . .	78	4.12.2	Morph Targets . . . . .	73	5.3.3	Tiling Textures . . . . .	78	4.12.3	Procedural Animation . . . . .	73	5.4	Advanced Texture Workflow (Production Mindset) . . . . .	78	4.12.4	Physics Animation . . . . .	73	5.4.1	Color Space: sRGB vs Linear (Why Your Colors Look "Off") . . . . .	78	4.12.5	Animation Blending . . . . .	74	5.4.2	Alpha (Transparency) Pitfalls . . . . .	79	4.13	Performance Optimization . . . . .	74	5.4.3	Filtering, Mipmaps, and Why Textures "Shimmer" . . . . .	79	4.13.1	General Tips . . . . .	74	5.4.4	Power-of-Two Sizes (and When It Matters) . . . . .	79	4.13.2	Advanced Optimization . . . . .	74	5.4.5	Aspect Ratio Correctness (Especially for Video) . . . . .	79	4.13.3	Performance Monitoring . . . . .	74	5.5	Advanced Techniques and Patch Recipes . . . . .	79	4.13.4	Adaptive Quality . . . . .	74	5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79	4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81									
4.11.16	Example 16: Audio-Reactive 3D . . . . .	72	5.2	Loading Textures . . . . .	77																																																																																																																																																																														
4.11.17	Example 17: Procedural City . . . . .	72	5.2.1	ImageTexture . . . . .	77	4.11.18	Example 18: Water Surface . . . . .	72	5.2.2	VideoTexture . . . . .	77	4.11.19	Example 19: Portal Effect . . . . .	72	5.2.3	WebcamTexture . . . . .	77	4.11.20	Example 20: Multi-Pass Rendering . . . . .	73	5.3	Texture Mapping . . . . .	77	4.12	Advanced Animation Techniques . . . . .	73	5.3.1	UV Coordinates . . . . .	77	4.12.1	Skeletal Animation . . . . .	73	5.3.2	UV Transform . . . . .	78	4.12.2	Morph Targets . . . . .	73	5.3.3	Tiling Textures . . . . .	78	4.12.3	Procedural Animation . . . . .	73	5.4	Advanced Texture Workflow (Production Mindset) . . . . .	78	4.12.4	Physics Animation . . . . .	73	5.4.1	Color Space: sRGB vs Linear (Why Your Colors Look "Off") . . . . .	78	4.12.5	Animation Blending . . . . .	74	5.4.2	Alpha (Transparency) Pitfalls . . . . .	79	4.13	Performance Optimization . . . . .	74	5.4.3	Filtering, Mipmaps, and Why Textures "Shimmer" . . . . .	79	4.13.1	General Tips . . . . .	74	5.4.4	Power-of-Two Sizes (and When It Matters) . . . . .	79	4.13.2	Advanced Optimization . . . . .	74	5.4.5	Aspect Ratio Correctness (Especially for Video) . . . . .	79	4.13.3	Performance Monitoring . . . . .	74	5.5	Advanced Techniques and Patch Recipes . . . . .	79	4.13.4	Adaptive Quality . . . . .	74	5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79	4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																		
5.2.1	ImageTexture . . . . .	77																																																																																																																																																																																	
4.11.18	Example 18: Water Surface . . . . .	72	5.2.2	VideoTexture . . . . .	77	4.11.19	Example 19: Portal Effect . . . . .	72	5.2.3	WebcamTexture . . . . .	77	4.11.20	Example 20: Multi-Pass Rendering . . . . .	73	5.3	Texture Mapping . . . . .	77	4.12	Advanced Animation Techniques . . . . .	73	5.3.1	UV Coordinates . . . . .	77	4.12.1	Skeletal Animation . . . . .	73	5.3.2	UV Transform . . . . .	78	4.12.2	Morph Targets . . . . .	73	5.3.3	Tiling Textures . . . . .	78	4.12.3	Procedural Animation . . . . .	73	5.4	Advanced Texture Workflow (Production Mindset) . . . . .	78	4.12.4	Physics Animation . . . . .	73	5.4.1	Color Space: sRGB vs Linear (Why Your Colors Look "Off") . . . . .	78	4.12.5	Animation Blending . . . . .	74	5.4.2	Alpha (Transparency) Pitfalls . . . . .	79	4.13	Performance Optimization . . . . .	74	5.4.3	Filtering, Mipmaps, and Why Textures "Shimmer" . . . . .	79	4.13.1	General Tips . . . . .	74	5.4.4	Power-of-Two Sizes (and When It Matters) . . . . .	79	4.13.2	Advanced Optimization . . . . .	74	5.4.5	Aspect Ratio Correctness (Especially for Video) . . . . .	79	4.13.3	Performance Monitoring . . . . .	74	5.5	Advanced Techniques and Patch Recipes . . . . .	79	4.13.4	Adaptive Quality . . . . .	74	5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79	4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																								
5.2.2	VideoTexture . . . . .	77																																																																																																																																																																																	
4.11.19	Example 19: Portal Effect . . . . .	72	5.2.3	WebcamTexture . . . . .	77	4.11.20	Example 20: Multi-Pass Rendering . . . . .	73	5.3	Texture Mapping . . . . .	77	4.12	Advanced Animation Techniques . . . . .	73	5.3.1	UV Coordinates . . . . .	77	4.12.1	Skeletal Animation . . . . .	73	5.3.2	UV Transform . . . . .	78	4.12.2	Morph Targets . . . . .	73	5.3.3	Tiling Textures . . . . .	78	4.12.3	Procedural Animation . . . . .	73	5.4	Advanced Texture Workflow (Production Mindset) . . . . .	78	4.12.4	Physics Animation . . . . .	73	5.4.1	Color Space: sRGB vs Linear (Why Your Colors Look "Off") . . . . .	78	4.12.5	Animation Blending . . . . .	74	5.4.2	Alpha (Transparency) Pitfalls . . . . .	79	4.13	Performance Optimization . . . . .	74	5.4.3	Filtering, Mipmaps, and Why Textures "Shimmer" . . . . .	79	4.13.1	General Tips . . . . .	74	5.4.4	Power-of-Two Sizes (and When It Matters) . . . . .	79	4.13.2	Advanced Optimization . . . . .	74	5.4.5	Aspect Ratio Correctness (Especially for Video) . . . . .	79	4.13.3	Performance Monitoring . . . . .	74	5.5	Advanced Techniques and Patch Recipes . . . . .	79	4.13.4	Adaptive Quality . . . . .	74	5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79	4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																														
5.2.3	WebcamTexture . . . . .	77																																																																																																																																																																																	
4.11.20	Example 20: Multi-Pass Rendering . . . . .	73	5.3	Texture Mapping . . . . .	77	4.12	Advanced Animation Techniques . . . . .	73	5.3.1	UV Coordinates . . . . .	77	4.12.1	Skeletal Animation . . . . .	73	5.3.2	UV Transform . . . . .	78	4.12.2	Morph Targets . . . . .	73	5.3.3	Tiling Textures . . . . .	78	4.12.3	Procedural Animation . . . . .	73	5.4	Advanced Texture Workflow (Production Mindset) . . . . .	78	4.12.4	Physics Animation . . . . .	73	5.4.1	Color Space: sRGB vs Linear (Why Your Colors Look "Off") . . . . .	78	4.12.5	Animation Blending . . . . .	74	5.4.2	Alpha (Transparency) Pitfalls . . . . .	79	4.13	Performance Optimization . . . . .	74	5.4.3	Filtering, Mipmaps, and Why Textures "Shimmer" . . . . .	79	4.13.1	General Tips . . . . .	74	5.4.4	Power-of-Two Sizes (and When It Matters) . . . . .	79	4.13.2	Advanced Optimization . . . . .	74	5.4.5	Aspect Ratio Correctness (Especially for Video) . . . . .	79	4.13.3	Performance Monitoring . . . . .	74	5.5	Advanced Techniques and Patch Recipes . . . . .	79	4.13.4	Adaptive Quality . . . . .	74	5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79	4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																				
5.3	Texture Mapping . . . . .	77																																																																																																																																																																																	
4.12	Advanced Animation Techniques . . . . .	73	5.3.1	UV Coordinates . . . . .	77	4.12.1	Skeletal Animation . . . . .	73	5.3.2	UV Transform . . . . .	78	4.12.2	Morph Targets . . . . .	73	5.3.3	Tiling Textures . . . . .	78	4.12.3	Procedural Animation . . . . .	73	5.4	Advanced Texture Workflow (Production Mindset) . . . . .	78	4.12.4	Physics Animation . . . . .	73	5.4.1	Color Space: sRGB vs Linear (Why Your Colors Look "Off") . . . . .	78	4.12.5	Animation Blending . . . . .	74	5.4.2	Alpha (Transparency) Pitfalls . . . . .	79	4.13	Performance Optimization . . . . .	74	5.4.3	Filtering, Mipmaps, and Why Textures "Shimmer" . . . . .	79	4.13.1	General Tips . . . . .	74	5.4.4	Power-of-Two Sizes (and When It Matters) . . . . .	79	4.13.2	Advanced Optimization . . . . .	74	5.4.5	Aspect Ratio Correctness (Especially for Video) . . . . .	79	4.13.3	Performance Monitoring . . . . .	74	5.5	Advanced Techniques and Patch Recipes . . . . .	79	4.13.4	Adaptive Quality . . . . .	74	5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79	4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																										
5.3.1	UV Coordinates . . . . .	77																																																																																																																																																																																	
4.12.1	Skeletal Animation . . . . .	73	5.3.2	UV Transform . . . . .	78	4.12.2	Morph Targets . . . . .	73	5.3.3	Tiling Textures . . . . .	78	4.12.3	Procedural Animation . . . . .	73	5.4	Advanced Texture Workflow (Production Mindset) . . . . .	78	4.12.4	Physics Animation . . . . .	73	5.4.1	Color Space: sRGB vs Linear (Why Your Colors Look "Off") . . . . .	78	4.12.5	Animation Blending . . . . .	74	5.4.2	Alpha (Transparency) Pitfalls . . . . .	79	4.13	Performance Optimization . . . . .	74	5.4.3	Filtering, Mipmaps, and Why Textures "Shimmer" . . . . .	79	4.13.1	General Tips . . . . .	74	5.4.4	Power-of-Two Sizes (and When It Matters) . . . . .	79	4.13.2	Advanced Optimization . . . . .	74	5.4.5	Aspect Ratio Correctness (Especially for Video) . . . . .	79	4.13.3	Performance Monitoring . . . . .	74	5.5	Advanced Techniques and Patch Recipes . . . . .	79	4.13.4	Adaptive Quality . . . . .	74	5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79	4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																
5.3.2	UV Transform . . . . .	78																																																																																																																																																																																	
4.12.2	Morph Targets . . . . .	73	5.3.3	Tiling Textures . . . . .	78	4.12.3	Procedural Animation . . . . .	73	5.4	Advanced Texture Workflow (Production Mindset) . . . . .	78	4.12.4	Physics Animation . . . . .	73	5.4.1	Color Space: sRGB vs Linear (Why Your Colors Look "Off") . . . . .	78	4.12.5	Animation Blending . . . . .	74	5.4.2	Alpha (Transparency) Pitfalls . . . . .	79	4.13	Performance Optimization . . . . .	74	5.4.3	Filtering, Mipmaps, and Why Textures "Shimmer" . . . . .	79	4.13.1	General Tips . . . . .	74	5.4.4	Power-of-Two Sizes (and When It Matters) . . . . .	79	4.13.2	Advanced Optimization . . . . .	74	5.4.5	Aspect Ratio Correctness (Especially for Video) . . . . .	79	4.13.3	Performance Monitoring . . . . .	74	5.5	Advanced Techniques and Patch Recipes . . . . .	79	4.13.4	Adaptive Quality . . . . .	74	5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79	4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																						
5.3.3	Tiling Textures . . . . .	78																																																																																																																																																																																	
4.12.3	Procedural Animation . . . . .	73	5.4	Advanced Texture Workflow (Production Mindset) . . . . .	78	4.12.4	Physics Animation . . . . .	73	5.4.1	Color Space: sRGB vs Linear (Why Your Colors Look "Off") . . . . .	78	4.12.5	Animation Blending . . . . .	74	5.4.2	Alpha (Transparency) Pitfalls . . . . .	79	4.13	Performance Optimization . . . . .	74	5.4.3	Filtering, Mipmaps, and Why Textures "Shimmer" . . . . .	79	4.13.1	General Tips . . . . .	74	5.4.4	Power-of-Two Sizes (and When It Matters) . . . . .	79	4.13.2	Advanced Optimization . . . . .	74	5.4.5	Aspect Ratio Correctness (Especially for Video) . . . . .	79	4.13.3	Performance Monitoring . . . . .	74	5.5	Advanced Techniques and Patch Recipes . . . . .	79	4.13.4	Adaptive Quality . . . . .	74	5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79	4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																												
5.4	Advanced Texture Workflow (Production Mindset) . . . . .	78																																																																																																																																																																																	
4.12.4	Physics Animation . . . . .	73	5.4.1	Color Space: sRGB vs Linear (Why Your Colors Look "Off") . . . . .	78	4.12.5	Animation Blending . . . . .	74	5.4.2	Alpha (Transparency) Pitfalls . . . . .	79	4.13	Performance Optimization . . . . .	74	5.4.3	Filtering, Mipmaps, and Why Textures "Shimmer" . . . . .	79	4.13.1	General Tips . . . . .	74	5.4.4	Power-of-Two Sizes (and When It Matters) . . . . .	79	4.13.2	Advanced Optimization . . . . .	74	5.4.5	Aspect Ratio Correctness (Especially for Video) . . . . .	79	4.13.3	Performance Monitoring . . . . .	74	5.5	Advanced Techniques and Patch Recipes . . . . .	79	4.13.4	Adaptive Quality . . . . .	74	5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79	4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																		
5.4.1	Color Space: sRGB vs Linear (Why Your Colors Look "Off") . . . . .	78																																																																																																																																																																																	
4.12.5	Animation Blending . . . . .	74	5.4.2	Alpha (Transparency) Pitfalls . . . . .	79	4.13	Performance Optimization . . . . .	74	5.4.3	Filtering, Mipmaps, and Why Textures "Shimmer" . . . . .	79	4.13.1	General Tips . . . . .	74	5.4.4	Power-of-Two Sizes (and When It Matters) . . . . .	79	4.13.2	Advanced Optimization . . . . .	74	5.4.5	Aspect Ratio Correctness (Especially for Video) . . . . .	79	4.13.3	Performance Monitoring . . . . .	74	5.5	Advanced Techniques and Patch Recipes . . . . .	79	4.13.4	Adaptive Quality . . . . .	74	5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79	4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																								
5.4.2	Alpha (Transparency) Pitfalls . . . . .	79																																																																																																																																																																																	
4.13	Performance Optimization . . . . .	74	5.4.3	Filtering, Mipmaps, and Why Textures "Shimmer" . . . . .	79	4.13.1	General Tips . . . . .	74	5.4.4	Power-of-Two Sizes (and When It Matters) . . . . .	79	4.13.2	Advanced Optimization . . . . .	74	5.4.5	Aspect Ratio Correctness (Especially for Video) . . . . .	79	4.13.3	Performance Monitoring . . . . .	74	5.5	Advanced Techniques and Patch Recipes . . . . .	79	4.13.4	Adaptive Quality . . . . .	74	5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79	4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																														
5.4.3	Filtering, Mipmaps, and Why Textures "Shimmer" . . . . .	79																																																																																																																																																																																	
4.13.1	General Tips . . . . .	74	5.4.4	Power-of-Two Sizes (and When It Matters) . . . . .	79	4.13.2	Advanced Optimization . . . . .	74	5.4.5	Aspect Ratio Correctness (Especially for Video) . . . . .	79	4.13.3	Performance Monitoring . . . . .	74	5.5	Advanced Techniques and Patch Recipes . . . . .	79	4.13.4	Adaptive Quality . . . . .	74	5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79	4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																																				
5.4.4	Power-of-Two Sizes (and When It Matters) . . . . .	79																																																																																																																																																																																	
4.13.2	Advanced Optimization . . . . .	74	5.4.5	Aspect Ratio Correctness (Especially for Video) . . . . .	79	4.13.3	Performance Monitoring . . . . .	74	5.5	Advanced Techniques and Patch Recipes . . . . .	79	4.13.4	Adaptive Quality . . . . .	74	5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79	4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																																										
5.4.5	Aspect Ratio Correctness (Especially for Video) . . . . .	79																																																																																																																																																																																	
4.13.3	Performance Monitoring . . . . .	74	5.5	Advanced Techniques and Patch Recipes . . . . .	79	4.13.4	Adaptive Quality . . . . .	74	5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79	4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																																																
5.5	Advanced Techniques and Patch Recipes . . . . .	79																																																																																																																																																																																	
4.13.4	Adaptive Quality . . . . .	74	5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79	4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																																																						
5.5.1	Recipe: Masked Texture Blend (Two Textures + a Mask) . . . . .	79																																																																																																																																																																																	
4.14	Common Patterns and Workflows . . . . .	74	5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79	4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																																																												
5.5.2	Recipe: Animated UVs (Scrolling / Parallax) . . . . .	79																																																																																																																																																																																	
4.14.1	Pattern: Object Pooling . . . . .	74	5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80	4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																																																																		
5.5.3	Recipe: Render-to-Texture for Post-Processing . . . . .	80																																																																																																																																																																																	
4.14.2	Pattern: Component System . . . . .	75	5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80	4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																																																																								
5.5.4	Recipe: Feedback / Trails (Texture Feedback Loop) . . . . .	80																																																																																																																																																																																	
4.14.3	Pattern: Event System . . . . .	75	5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80	4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																																																																														
5.5.5	Recipe: Planar "Mirror" Reflection (Render-to-Texture) . . . . .	80																																																																																																																																																																																	
4.14.4	Pattern: State Machine . . . . .	75	5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80	4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																																																																																				
5.5.6	Recipe: Environment Reflections (Cubemap/HDRI) . . . . .	80																																																																																																																																																																																	
4.15	Debugging 3D Scenes . . . . .	75	5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80	4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																																																																																										
5.5.7	Recipe: Video Texture "Billboard" (Reliable Playback) . . . . .	80																																																																																																																																																																																	
4.15.1	Visual Debugging . . . . .	75	5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80	4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																																																																																																
5.5.8	Recipe: Webcam Texture (Permissions + UX) . . . . .	80																																																																																																																																																																																	
4.15.2	Common Issues . . . . .	75	5.6	Texture Types for PBR Materials . . . . .	81				5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																																																																																																						
5.6	Texture Types for PBR Materials . . . . .	81																																																																																																																																																																																	
			5.6.1	Albedo/Diffuse Map . . . . .	81				5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																																																																																																												
5.6.1	Albedo/Diffuse Map . . . . .	81																																																																																																																																																																																	
			5.6.2	Normal Map . . . . .	81				5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																																																																																																																		
5.6.2	Normal Map . . . . .	81																																																																																																																																																																																	
			5.6.3	Roughness Map . . . . .	81				5.6.4	Metalness Map . . . . .	81																																																																																																																																																																								
5.6.3	Roughness Map . . . . .	81																																																																																																																																																																																	
			5.6.4	Metalness Map . . . . .	81																																																																																																																																																																														
5.6.4	Metalness Map . . . . .	81																																																																																																																																																																																	

5.6.5	Ambient Occlusion Map . . . . .	81	6.7	Common Shader Patterns . . . . .	85
5.6.6	Height/Displacement Map . . . . .	81	6.7.1	Solid Color . . . . .	85
5.6.7	Emissive Map . . . . .	81	6.7.2	UV Gradient . . . . .	85
5.7	Procedural Textures . . . . .	81	6.7.3	Circle (SDF) . . . . .	85
5.7.1	Noise Textures . . . . .	81	6.7.4	Smooth Circle . . . . .	85
5.7.2	Gradient Textures . . . . .	81	6.7.5	Animated Pattern . . . . .	86
5.7.3	Pattern Generators . . . . .	81	6.8	Uniforms . . . . .	86
5.8	Render to Texture . . . . .	81	6.9	Advanced Shader Workflows in cables.gl . . . . .	86
5.8.1	Common Uses: . . . . .	82	6.9.1	A Practical Uniform “Contract” . . . . .	86
5.9	Texture Effects . . . . .	82	6.9.2	Coordinate Spaces: UV vs Screen Space . . . . .	86
5.9.1	TextureEffects Op . . . . .	82	6.9.3	Anti-Aliasing SDFs (Clean Edges) . . . . .	86
5.9.2	Custom Shader Effects . . . . .	82	6.9.4	Palette Mapping (Better Color Fast) . . . . .	87
5.10	Cubemaps and Environment Maps . . . . .	82	6.10	Advanced Examples (Copy-and-Adapt) . . . . .	87
5.10.1	CubemapTexture . . . . .	82	6.10.1	Example: Texture Distortion (UV Warp) . . . . .	87
5.10.2	HDRITexture . . . . .	82	6.10.2	Example: Simple Bloom-ish Glow (Threshold + Blur-ish) . . . . .	87
5.11	Texture Compression and Optimization . . . . .	82	6.10.3	Example: Domain Warping (More Organic Noise) . . . . .	87
5.11.1	File Size Tips: . . . . .	82	6.11	Debugging Shaders (In Practice) . . . . .	88
5.11.2	Memory Considerations: . . . . .	82	6.11.1	Common Gotchas . . . . .	88
5.12	Practical Examples . . . . .	82	6.12	Performance Guidelines (Real-Time Friendly) . . . . .	88
5.12.1	Example 1: Textured Rotating Cube . . . . .	82	6.13	Signed Distance Functions (SDFs) . . . . .	88
5.12.2	Example 2: Video on a Plane . . . . .	83	6.13.1	SDF Primitives . . . . .	88
5.12.3	Example 3: Animated Noise Background . . . . .	83	6.13.2	SDF Operations . . . . .	89
5.12.4	Example 4: PBR Textured Material . . . . .	83	6.14	Noise Functions . . . . .	89
5.13	Featured Videos . . . . .	83	6.14.1	Simple Value Noise . . . . .	89
5.14	Exercises . . . . .	83	6.14.2	Fractal Brownian Motion (FBM) . . . . .	89
<b>6</b>	<b>Shaders &amp; GLSL in Cables.gl</b>	<b>84</b>	6.15	Post-Processing Effects . . . . .	89
6.1	Introduction to Shaders . . . . .	84	6.15.1	Vignette . . . . .	89
6.2	What Are Shaders? . . . . .	84	6.15.2	Chromatic Aberration . . . . .	90
6.3	Why Use Custom Shaders? . . . . .	84	6.15.3	Blur (Box Blur) . . . . .	90
6.4	Shader Ops in Cables.gl . . . . .	84	6.15.4	Pixelation . . . . .	90
6.4.1	ShaderMaterial . . . . .	84	6.16	Vertex Shader Basics . . . . .	90
6.4.2	TextureEffect (Shader-based) . . . . .	84	6.17	Debugging Shaders . . . . .	91
6.4.3	CustomShader . . . . .	84	6.17.1	Visualize Values . . . . .	91
6.5	GLSL Basics . . . . .	84	6.18	Performance Tips . . . . .	91
6.5.1	Data Types . . . . .	84	6.19	Professional Video Projection Mapping in Cables.gl . . . . .	91
6.5.2	Swizzling . . . . .	84	6.19.1	Understanding Cables.gl Shader Context . . . . .	91
6.5.3	Built-in Functions . . . . .	84	6.19.2	Cables.gl Shader Compliance Checklist . . . . .	92
6.6	Your First Fragment Shader . . . . .	85	6.19.3	Troubleshooting Common Issues . . . . .	92

6.19.4	Using Shaders in Cables.gl: Two Approaches . . . . .	92
6.19.5	Geometric Distortion Correction . . . . .	93
6.19.6	Multi-Projector Setups . . . . .	96
6.19.7	Projector Stacking . . . . .	97
6.19.8	Gradient Blend Composition . . . . .	98
6.19.9	Color Correction for Projection Mapping . . . . .	100
6.19.10	Complete Projection Mapping Pipeline . . . . .	103
6.19.11	JavaScript Custom Op Examples . . . . .	105
6.19.12	Comparison: Built-in Shader Ops vs Custom JavaScript Ops . . . . .	108
6.19.13	Quick Reference: Using These Shaders . . . . .	108
6.19.14	Best Practices for Projection Mapping in Cables.gl . .	109
6.19.15	Debug Visualization Shaders . . . . .	109
6.19.16	Summary: Shader Compliance and Usage . . . . .	110
6.20	Featured Videos . . . . .	110
6.21	Resources . . . . .	110
6.22	Exercises . . . . .	111
<b>7</b>	<b>JavaScript &amp; Custom Ops in Cables.gl</b>	<b>111</b>
7.1	Introduction . . . . .	111
7.2	When to Use Custom Ops . . . . .	111
7.3	Creating Your First Op . . . . .	111
7.3.1	Step 1: Open the Op Editor . . . . .	111
7.3.2	Step 2: Understanding the Structure . . . . .	111
7.4	Port Types . . . . .	111
7.4.1	Input Ports . . . . .	112
7.4.2	Output Ports . . . . .	112
7.5	Handling Events . . . . .	112
7.5.1	Trigger Execution . . . . .	112
7.5.2	Value Changes . . . . .	112
7.5.3	Linking Ports . . . . .	112
7.6	Working with Arrays . . . . .	113
7.7	Working with Objects . . . . .	113
7.8	Render Loop Integration . . . . .	113
7.9	UI Port Groups . . . . .	113
7.10	Port UI Types . . . . .	113
7.11	Accessing Patch Resources . . . . .	114
7.11.1	Timer and Time . . . . .	114
7.11.2	Canvas and Context . . . . .	114
7.11.3	Loading External Resources . . . . .	114
7.12	Using External Libraries . . . . .	114
7.12.1	Including Libraries . . . . .	114
7.12.2	Or use op.patch.loading for proper load tracking: . .	114
7.13	Error Handling . . . . .	115
7.14	Example: Custom Math Op . . . . .	115
7.15	Example: Array Processor . . . . .	115
7.16	Example: API Fetcher . . . . .	116
7.17	Debugging Tips . . . . .	116
7.18	Advanced Patterns (How to Build "Good" Ops) . . . . .	116
7.18.1	Pattern: Separate "Compute" from "Trigger" . . . . .	116
7.18.2	Pattern: "Only Recompute When Dirty" . . . . .	117
7.18.3	Pattern: Debounce (Stabilize Noisy Inputs) . . . . .	117
7.18.4	Pattern: Rate-Limit (Prevent Flooding Downstream) .	118
7.18.5	Pattern: Stateful Ops (Resettable Systems) . . . . .	118
7.19	Async Ops (Fetching Data Safely) . . . . .	118
7.19.1	Example: Fetch JSON with Cancellation . . . . .	118
7.19.2	Loading Semantics (Patch-Friendly) . . . . .	119
7.20	Performance Tips for Custom Ops . . . . .	119
7.21	Featured Videos . . . . .	119
7.22	Exercises . . . . .	119
<b>8</b>	<b>Audio &amp; Sound in Cables.gl</b>	<b>120</b>
8.1	Introduction . . . . .	120
8.2	Audio Sources . . . . .	120
8.2.1	AudioFile . . . . .	120
8.2.2	Microphone . . . . .	120
8.2.3	AudioBuffer . . . . .	120
8.2.4	WebAudio Oscillator . . . . .	120
8.3	Audio Analysis . . . . .	120
8.3.1	AudioAnalyzer . . . . .	120
8.3.2	FFT (Fast Fourier Transform) . . . . .	120
8.3.3	Smoothing . . . . .	120
8.4	Common Audio-Reactive Patterns . . . . .	121
8.4.1	Volume-Based Scaling . . . . .	121
8.4.2	Frequency Band Visualization . . . . .	121
8.4.3	Color from Audio . . . . .	121
8.4.4	Beat Detection . . . . .	121
8.5	Audio Effects . . . . .	121

8.5.1	Gain . . . . .	121
8.5.2	Filter . . . . .	121
8.5.3	Delay . . . . .	121
8.5.4	Reverb . . . . .	121
8.5.5	Compressor . . . . .	122
8.6	Building a Visualizer . . . . .	122
8.6.1	Step 1: Set Up Audio . . . . .	122
8.6.2	Step 2: Create Base Render . . . . .	122
8.6.3	Step 3: Add Audio-Reactive Elements . . . . .	122
8.6.4	Step 4: Add Frequency Visualization . . . . .	122
8.7	Synchronizing to Music . . . . .	122
8.7.1	BPM and Beat Sync . . . . .	122
8.7.2	Timeline with Audio . . . . .	122
8.8	Advanced Audio Techniques (Make It Feel “Musical”) . . . . .	123
8.8.1	Technique: Energy vs Transient (Two-Signal Approach) .	123
8.8.2	Technique: Log Frequency Mapping (Better Spectra) .	123
8.8.3	Technique: Peak Hold (Readable Visuals) . . . . .	123
8.8.4	Technique: Band-Specific Control (Bass Drives Scale, High Drives Detail) . . . . .	123
8.8.5	Technique: Audio -> Shader (The “Pro” Move) . . . . .	123
8.9	Advanced Patch Recipes . . . . .	124
8.9.1	Recipe: Stable Beat Trigger (Avoid Double-Triggers) .	124
8.9.2	Recipe: Audio-Reactive Post-Processing . . . . .	124
8.9.3	Recipe: Audio-Reactive 3D Equalizer (Optimized) .	124
8.9.4	Recipe: Audio-Driven Palette . . . . .	124
8.10	Practical Examples . . . . .	124
8.10.1	Example 1: Bass-Reactive Background . . . . .	124
8.10.2	Example 2: Circular Spectrum . . . . .	125
8.10.3	Example 3: Waveform Display . . . . .	125
8.10.4	Example 4: 3D Audio Visualization . . . . .	125
8.11	Performance Considerations . . . . .	125
8.12	Browser Audio Policies . . . . .	125
8.13	Featured Videos . . . . .	125
8.14	Exercises . . . . .	126
<b>9</b>	<b>Animation &amp; Timeline in Cables.gl</b>	<b>126</b>
9.1	Introduction . . . . .	126
9.2	Types of Animation . . . . .	126
9.2.1	1. Procedural Animation . . . . .	126
9.2.2	2. Keyframe Animation . . . . .	126
9.2.3	3. Physics-Based Animation . . . . .	126
9.2.4	4. Data-Driven Animation . . . . .	126
9.3	Procedural Animation . . . . .	126
9.3.1	The Time Op . . . . .	126
9.3.2	Basic Movement Patterns . . . . .	127
9.3.3	Easing Functions . . . . .	127
9.3.4	The Smooth Op . . . . .	127
9.3.5	Spring Animation . . . . .	127
9.4	Timeline Animation . . . . .	127
9.4.1	Opening the Timeline . . . . .	127
9.4.2	Timeline Interface . . . . .	127
9.4.3	Adding Keyframes . . . . .	128
9.4.4	Keyframe Types . . . . .	128
9.4.5	Editing Keyframes . . . . .	128
9.4.6	Timeline Tracks . . . . .	128
9.5	Non-Linear Animation Clips (New Animation System - November 2025) . . . . .	128
9.5.1	What Are Animation Clips? . . . . .	128
9.5.2	Creating Animation Clips . . . . .	128
9.5.3	Clip Properties and Options . . . . .	129
9.5.4	Additive Animation (Layering Clips) . . . . .	129
9.5.5	Mixable Animation (Blending Clips) . . . . .	129
9.5.6	Clip Management . . . . .	130
9.5.7	Advanced Clip Techniques . . . . .	130
9.6	JavaScript Custom Op Integration with Animation System .	130
9.6.1	Accessing Animation Data from Custom Ops . . . . .	130
9.6.2	Controlling Timeline from Custom Ops . . . . .	131
9.6.3	Creating Animation Clips Programmatically . . . . .	131
9.6.4	Manipulating Animation Clips . . . . .	132
9.6.5	Advanced: Custom Easing Functions . . . . .	133
9.6.6	Real-Time Animation Modification . . . . .	134
9.6.7	Integration Example: Physics-Driven Animation . . . . .	134
9.6.8	Best Practices for Animation + Custom Ops . . . . .	135
9.6.9	Example: Complete Animation Controller Op . . . . .	135
9.7	Sequence and Timing Ops . . . . .	136
9.7.1	Sequence . . . . .	136
9.7.2	Delay . . . . .	136

9.7.3	Timer . . . . .	136	9.15	Debugging Animation . . . . .	142
9.7.4	Interval . . . . .	136	9.15.1	Slow Motion . . . . .	142
9.8	Animation Patterns . . . . .	136	9.15.2	Visualize Values . . . . .	142
9.8.1	Staggered Animation . . . . .	136	9.15.3	Pause at Problem . . . . .	142
9.8.2	Loop with Pause . . . . .	137	9.16	Featured Videos . . . . .	142
9.8.3	Ping-Pong (Back and Forth) . . . . .	137	9.17	Exercises . . . . .	142
9.8.4	One-Shot Animation . . . . .	137	9.17.1	Basic Animation . . . . .	142
9.9	State Machines . . . . .	137	9.17.2	Animation Clips . . . . .	142
9.9.1	Simple States . . . . .	137	9.17.3	JavaScript Integration . . . . .	143
9.9.2	Transition Between States . . . . .	137	9.17.4	Advanced . . . . .	143
9.10	Interactive Animation . . . . .	137	<b>10</b>	<b>Interfaces in Cables.gl</b> . . . . .	<b>143</b>
9.10.1	Mouse-Based . . . . .	137	10.1	Introduction . . . . .	143
9.10.2	Scroll-Based . . . . .	137	10.2	Interface Approaches Overview . . . . .	143
9.10.3	Click-Triggered . . . . .	137	10.3	HTML/CSS Interfaces . . . . .	143
9.11	Advanced Animation Systems (How to Build “Scenes”) . . . . .	137	10.3.1	Overview . . . . .	144
9.11.1	Layering: Timeline for Structure, Procedural for Life . . . . .	138	10.3.2	The HTML Op . . . . .	144
9.11.2	Shot-Based Timelines (Cinematic Organization) . . . . .	138	10.3.3	Connecting HTML to Patch Logic . . . . .	145
9.11.3	Animation Curves: Clamp Early, Map Late . . . . .	138	10.3.4	Advanced HTML Interface Patterns . . . . .	148
9.11.4	Reusable “Rig” Pattern . . . . .	138	10.4	Native Sidebar Interface Ops . . . . .	149
9.11.5	Avoiding Jitter in Interactive Animation . . . . .	138	10.4.1	Overview . . . . .	149
9.11.6	Choreographing Triggers Reliably . . . . .	138	10.4.2	Available Interface Ops . . . . .	149
9.12	Advanced Recipes . . . . .	139	10.4.3	Basic Interface Op Setup . . . . .	149
9.12.1	Recipe: Scroll-Driven Scene (Interactive Storytelling) . . . . .	139	10.4.4	Styling Native Sidebar with CSS . . . . .	150
9.12.2	Recipe: Beat-Synced Timeline Sections . . . . .	139	10.4.5	Complete Styling Example: Professional Control Panel . . . . .	155
9.12.3	Recipe: One-Shot “Punch” Animation (No Keyframes) . . . . .	139	10.4.6	Advanced CSS Techniques . . . . .	161
9.12.4	Recipe: Camera Rig (Orbit + Handheld Micro Motion) . . . . .	139	10.5	Combining HTML and Native Interfaces . . . . .	161
9.13	Practical Examples . . . . .	139	10.6	Best Practices . . . . .	162
9.13.1	Example 1: Bouncing Ball . . . . .	139	10.6.1	1. Performance . . . . .	162
9.13.2	Example 2: Rotating Carousel . . . . .	139	10.6.2	2. Accessibility . . . . .	162
9.13.3	Example 3: Fade In Sequence . . . . .	140	10.6.3	3. Responsive Design . . . . .	162
9.13.4	Example 4: Timeline-Based Scene . . . . .	140	10.6.4	4. Organization . . . . .	162
9.13.5	Example 5: Layered Animation Clips (Additive) . . . . .	140	10.7	Practical Examples . . . . .	162
9.13.6	Example 6: Blended Animation Clips (Mixable) . . . . .	140	10.7.1	Example 1: Animation Control Panel . . . . .	162
9.13.7	Example 7: Reusable Clip System . . . . .	140	10.7.2	Example 2: Game UI Overlay . . . . .	162
9.13.8	Example 8: JavaScript-Controlled Animation . . . . .	141	10.7.3	Example 3: Data Visualization Dashboard . . . . .	163
9.13.9	Example 9: Physics + Animation Clip Hybrid . . . . .	141	10.8	Debugging Interface Issues . . . . .	163
9.13.10	Example 10: Conditional Clip Playback . . . . .	142	10.8.1	Common Issues . . . . .	163
9.14	Performance Tips . . . . .	142	10.8.2	Debugging Tools . . . . .	163

10.9	Exercises . . . . .	163
<b>11</b>	<b>Export &amp; Deployment in Cables.gl</b>	<b>164</b>
11.1	Introduction . . . . .	164
11.2	Export Options . . . . .	164
11.2.1	1. Public Patch Link . . . . .	164
11.2.2	2. Embedded iframe . . . . .	164
11.2.3	3. Standalone Export . . . . .	164
11.2.4	4. npm Package Export . . . . .	164
11.3	Standalone Export Process . . . . .	164
11.3.1	Step 1: Prepare Your Patch . . . . .	164
11.3.2	Step 2: Export . . . . .	164
11.3.3	Step 3: Download . . . . .	164
11.3.4	Step 4: Test Locally . . . . .	165
11.4	Customizing the Export . . . . .	165
11.4.1	Custom HTML Template . . . . .	165
11.4.2	Configuration Options . . . . .	165
11.5	Communicating with Your Patch . . . . .	165
11.5.1	Setting Variables from JavaScript . . . . .	165
11.5.2	Getting Values from the Patch . . . . .	166
11.5.3	Triggering Events . . . . .	166
11.6	Advanced Embedding & Integration . . . . .	166
11.6.1	Responsive Canvas: Beyond Width/Height . . . . .	166
11.6.2	Pausing When Not Visible . . . . .	166
11.6.3	postMessage Integration (iframe Control) . . . . .	166
11.6.4	Environment-Specific Configuration (dev / test / prod) . . . . .	167
11.6.5	Asset Path Gotchas . . . . .	167
11.7	Hosting Options . . . . .	167
11.7.1	Static Hosting . . . . .	167
11.7.2	GitHub Pages Deployment . . . . .	167
11.7.3	Netlify Deployment . . . . .	167
11.8	Embedding in Existing Websites . . . . .	167
11.8.1	As a Background . . . . .	167
11.8.2	As a Hero Section . . . . .	168
11.8.3	Responsive Embedding . . . . .	168
11.9	Performance Optimization . . . . .	168
11.9.1	Before Export . . . . .	168
11.9.2	Asset Optimization . . . . .	168
11.9.3	Loading Optimization . . . . .	168
11.10	Deployment Checklist (The Stuff That Breaks at the Worst Time) . . . . .	169
11.10.1	Cache Busting and Versioning . . . . .	169
11.10.2	MIME Types (Especially for Wasm / Binary Assets) . . . . .	169
11.10.3	CORS (Cross-Origin Assets) . . . . .	169
11.10.4	Content Security Policy (CSP) . . . . .	169
11.11	CI/CD Ideas (Optional, But Great for Teams) . . . . .	169
11.12	Offline/PWA . . . . .	169
11.12.1	manifest.json . . . . .	169
11.12.2	Service Worker . . . . .	170
11.13	Electron Desktop Applications . . . . .	170
11.13.1	Why Electron? . . . . .	170
11.13.2	Getting Started with Electron . . . . .	170
11.13.3	Main Process (main.js) . . . . .	171
11.13.4	Preload Script (preload.js) . . . . .	176
11.13.5	Advanced Window Configuration . . . . .	176
11.13.6	Inter-Window Communication . . . . .	178
11.13.7	Splash Screen Implementation . . . . .	180
11.13.8	JSON File Operations . . . . .	182
11.13.9	Code Signing for Distribution . . . . .	186
11.13.10	Building and Distributing . . . . .	190
11.13.11	Advanced Electron Features . . . . .	190
11.13.12	Performance Optimization for Electron . . . . .	191
11.13.13	Security Best Practices . . . . .	191
11.13.14	Troubleshooting Electron Issues . . . . .	191
11.14	Troubleshooting . . . . .	192
11.14.1	Common Issues . . . . .	192
11.15	Featured Videos . . . . .	192
11.16	Exercises . . . . .	192
<b>12</b>	<b>Video Tutorials</b>	<b>193</b>
12.1	Getting Started & Overviews . . . . .	193
12.2	Core Concepts & Workflow . . . . .	197
12.3	3D / 3D Meshes . . . . .	200
12.4	Textures / Post-Processing . . . . .	201
12.5	Shaders / Shadertoy / GLSL . . . . .	202
12.6	Audio / Music / MIDI . . . . .	202
12.7	Physics . . . . .	204
12.8	Export / Deployment / Embedding . . . . .	205

12.9	Hardware / External Tools . . . . .	206
12.10	Talks / Meetups / Release Notes . . . . .	206
12.11	Showcases / Demos / Visualizers . . . . .	207
12.12	Unsorted (Still cables.gl-related) . . . . .	209
<b>13</b>	<b>Ops.Anim</b>	<b>213</b>
13.1	Ops.Anim . . . . .	213
13.1.1	AnimNumber . . . . .	213
13.1.2	Bang . . . . .	213
13.1.3	BoolAnim . . . . .	214
13.1.4	Crossfade . . . . .	214
13.1.5	FrameRangeAnim_v2 . . . . .	214
13.1.6	FrameRangeAnimSwitcher . . . . .	215
13.1.7	InOutInAnim . . . . .	215
13.1.8	LFO_v3 . . . . .	215
13.1.9	RandomAnim_v2 . . . . .	216
13.1.10	SimpleAnim . . . . .	216
13.1.11	SineAnim . . . . .	216
13.1.12	Smooth . . . . .	217
13.1.13	Snap . . . . .	217
13.1.14	Spring . . . . .	217
13.1.15	StringTypeAnimation_v2 . . . . .	218
13.1.16	TimeDelta . . . . .	218
13.1.17	Timer_v2 . . . . .	218
<b>14</b>	<b>Ops.Array</b>	<b>219</b>
14.1	Ops.Array . . . . .	219
14.1.1	AnglesBetweenPoints . . . . .	219
14.1.2	AnimArray_v2 . . . . .	219
14.1.3	Array1toX_v2 . . . . .	219
14.1.4	Array2To3 . . . . .	219
14.1.5	Array3 . . . . .	220
14.1.6	Array3GetAverage . . . . .	220
14.1.7	Array3GetNumbers . . . . .	220
14.1.8	Array3InterpolateDistributed . . . . .	220
14.1.9	Array3Iterator . . . . .	221
14.1.10	Array3Multiply . . . . .	221
14.1.11	Array3PushNumbers_v2 . . . . .	221
14.1.12	Array3RandomSelection . . . . .	222
14.1.13	Array3SetNumber . . . . .	222
14.1.14	Array3Sum . . . . .	222
14.1.15	Array3To2 . . . . .	222
14.1.16	Array3To4 . . . . .	223
14.1.17	Array3VectorLength . . . . .	223
14.1.18	Array4 . . . . .	223
14.1.19	Array4GetNumbers . . . . .	223
14.1.20	Array4SetNumber . . . . .	224
14.1.21	Array4ToArray3 . . . . .	224
14.1.22	Array_v3 . . . . .	224
14.1.23	ArrayAbs . . . . .	224
14.1.24	ArrayAppendArray . . . . .	225
14.1.25	ArrayBuffer . . . . .	225
14.1.26	ArrayBuffer3 . . . . .	225
14.1.27	ArrayCeil . . . . .	225
14.1.28	ArrayChunk . . . . .	226
14.1.29	ArrayChunkDuplicate . . . . .	226
14.1.30	ArrayClamp . . . . .	226
14.1.31	ArrayContains_v2 . . . . .	226
14.1.32	ArrayDivide . . . . .	227
14.1.33	ArrayFindStrings . . . . .	227
14.1.34	ArrayFloor . . . . .	227
14.1.35	ArrayFract . . . . .	227
14.1.36	ArrayFromNumbers . . . . .	228
14.1.37	ArrayGetArray . . . . .	228
14.1.38	ArrayGetNumber . . . . .	228
14.1.39	ArrayGetObject . . . . .	229
14.1.40	ArrayGetString_v2 . . . . .	229
14.1.41	ArrayGetTexture . . . . .	229
14.1.42	ArrayGetValuesByIndexArray . . . . .	229
14.1.43	ArrayIndexBetween . . . . .	230
14.1.44	ArrayIndexMinMax . . . . .	230
14.1.45	ArrayIteratorArray . . . . .	230
14.1.46	ArrayIteratorNumbers . . . . .	230
14.1.47	ArrayIteratorObjects . . . . .	231
14.1.48	ArrayIteratorStrings . . . . .	231
14.1.49	ArrayIteratorTextures . . . . .	231
14.1.50	ArrayLength_v2 . . . . .	231

14.1.51	ArrayLogic . . . . .	232
14.1.52	ArrayLogicArray . . . . .	232
14.1.53	ArrayLogicBetween_v2 . . . . .	232
14.1.54	ArrayLookup . . . . .	232
14.1.55	ArrayMath . . . . .	233
14.1.56	ArrayMathArray . . . . .	233
14.1.57	ArrayMathExpression . . . . .	233
14.1.58	ArrayMathExpressionTrigger . . . . .	233
14.1.59	ArrayMax . . . . .	234
14.1.60	ArrayMerge_v3 . . . . .	234
14.1.61	ArrayMergeTrigger . . . . .	234
14.1.62	ArrayMin . . . . .	235
14.1.63	ArrayModulo . . . . .	235
14.1.64	ArrayMultiply . . . . .	235
14.1.65	ArrayNumberRamp_v2 . . . . .	235
14.1.66	ArrayOfArrays . . . . .	236
14.1.67	ArrayOfObjectsFilterByKeyValue_v3 . . . . .	236
14.1.68	ArrayOfObjectsFilterKeys . . . . .	236
14.1.69	ArrayOfObjectsToString . . . . .	237
14.1.70	ArrayPack . . . . .	237
14.1.71	ArrayPack2 . . . . .	237
14.1.72	ArrayPack2Simple . . . . .	237
14.1.73	ArrayPack3 . . . . .	238
14.1.74	ArrayPack3Simple . . . . .	238
14.1.75	ArrayPack4 . . . . .	238
14.1.76	ArrayPack4Simple . . . . .	238
14.1.77	ArrayPow . . . . .	239
14.1.78	ArrayPushString . . . . .	239
14.1.79	ArrayQuantizer . . . . .	239
14.1.80	ArrayRandomSelection . . . . .	240
14.1.81	ArrayRemoveFalsy . . . . .	240
14.1.82	ArrayReverse . . . . .	240
14.1.83	ArrayRound . . . . .	240
14.1.84	ArraySetNumber_v3 . . . . .	241
14.1.85	ArraySetString . . . . .	241
14.1.86	ArraySin . . . . .	241
14.1.87	ArraySmoothStep . . . . .	241
14.1.88	ArraySqrt . . . . .	242
14.1.89	ArraysToArrayMultiPort . . . . .	242
14.1.90	ArraySubtract . . . . .	242
14.1.91	ArraySum . . . . .	242
14.1.92	ArraySumPrevious . . . . .	243
14.1.93	ArraySumUp . . . . .	243
14.1.94	ArraySwizzle . . . . .	243
14.1.95	ArrayToArrays . . . . .	243
14.1.96	ArrayToByteBuffer . . . . .	243
14.1.97	ArrayToString_v3 . . . . .	244
14.1.98	ArrayTrigger . . . . .	244
14.1.99	ArrayUnique . . . . .	244
14.1.100	ArrayUniqueItemInfo . . . . .	244
14.1.101	ArrayUnpack2 . . . . .	245
14.1.102	ArrayUnpack3 . . . . .	245
14.1.103	ArrayUnpack4 . . . . .	245
14.1.104	ArrayUnshiftString . . . . .	245
14.1.105	AverageArray . . . . .	246
14.1.106	BoolStateArray . . . . .	246
14.1.107	CopyArray . . . . .	246
14.1.108	CopyArraySimple . . . . .	246
14.1.109	CropArray . . . . .	247
14.1.110	CutArray . . . . .	247
14.1.111	EaseArray . . . . .	247
14.1.112	EmptyArray . . . . .	247
14.1.113	FillArrayRandomDuplicates_v2 . . . . .	248
14.1.114	FilterArray . . . . .	248
14.1.115	FilterValidArray . . . . .	248
14.1.116	FlattenArray . . . . .	248
14.1.117	FreezeArray . . . . .	249
14.1.118	GateArray_v2 . . . . .	249
14.1.119	GetValuesFromArrayOfObjects . . . . .	249
14.1.120	HSBtoRGBArray . . . . .	249
14.1.121	InfoArray . . . . .	250
14.1.122	InfoArray2 . . . . .	250
14.1.123	InfoArray3 . . . . .	250
14.1.124	InterpolateArrays . . . . .	250
14.1.125	InterpolateArraysRange . . . . .	251
14.1.126	InterpolateNumbersArray . . . . .	251

14.1.127 InterpolateNumbersArray3 . . . . .	251	15.1 Ops.Array.PointArray . . . . .	261
14.1.128 LissajouseSpline . . . . .	252	15.1.1 Array3AreaRemove . . . . .	261
14.1.129 LoopArray3 . . . . .	252	15.1.2 Array3PointEditor . . . . .	262
14.1.130 MapRangeArray . . . . .	252	15.1.3 Array3RepeatTransform . . . . .	262
14.1.131 NumbersToArrayMultiPort_v2 . . . . .	252	15.1.4 Array3VectorDistance . . . . .	262
14.1.132 ObjectIteratorKeys . . . . .	253	15.1.5 ArraySpray . . . . .	262
14.1.133 PaletteLibrary . . . . .	253	15.1.6 CircularPoints_v2 . . . . .	263
14.1.134 PerlinArray . . . . .	253	15.1.7 FillPointArrayDuplicates . . . . .	263
14.1.135 Phyllotaxis . . . . .	253	15.1.8 PointsCube . . . . .	264
14.1.136 RandomNumbersArray_v4 . . . . .	254	15.1.9 PointsHexagonGrid . . . . .	264
14.1.137 RandomWordsArray . . . . .	254	15.1.10 PointsPlane_v2 . . . . .	264
14.1.138 ReduceArray3_v3 . . . . .	254	15.1.11 PointsRectangle_v2 . . . . .	264
14.1.139 ReverseArray3 . . . . .	254	15.1.12 PointsRectangleRounded_v2 . . . . .	265
14.1.140 RingBuffer . . . . .	255	15.1.13 PointsSphereRandom . . . . .	265
14.1.141 RotateArray . . . . .	255	15.1.14 RedistributeSplinePoints . . . . .	265
14.1.142 RouteArray . . . . .	255	15.1.15 SortArray3ByDistance . . . . .	266
14.1.143 SetNumberArray . . . . .	256	15.1.16 SplinePositionAtDistanceArray3 . . . . .	266
14.1.144 SetNumbersArray3 . . . . .	256	15.1.17 SubdivideArray3_v2 . . . . .	266
14.1.145 ShuffleArray3_v3 . . . . .	256	15.1.18 TransformArray3 . . . . .	267
14.1.146 ShuffleArray_v3 . . . . .	256		
14.1.147 SimplexArray . . . . .	257		
14.1.148 SmoothArray . . . . .	257		
14.1.149 SortArray . . . . .	257		
14.1.150 SortArray3 . . . . .	257		
14.1.151 SortArrayOfObjects . . . . .	258		
14.1.152 SortArrayWithIndices_v2 . . . . .	258		
14.1.153 SplineLengthArray3 . . . . .	258		
14.1.154 SplinesToLineStripArray . . . . .	258		
14.1.155 StringToArray_v2 . . . . .	259		
14.1.156 StringToCharArray . . . . .	259		
14.1.157 StringToLineNumberArray . . . . .	259		
14.1.158 SubdivideArray1 . . . . .	259		
14.1.159 SwitchArray . . . . .	260		
14.1.160 SwitchArrayMultiPort_v2 . . . . .	260		
14.1.161 SwitchArrayOnTrigger . . . . .	260		
14.1.162 TextureBufferArray . . . . .	261		
14.1.163 WeaveArrays . . . . .	261		
<b>15 Ops.Array.PointArray</b>	<b>261</b>	<b>16 Ops.Audio</b>	<b>267</b>
		16.1 Ops.Audio . . . . .	267
		16.1.1 BpmTap . . . . .	267
		16.1.2 MidiJson . . . . .	267
		16.1.3 MidiJsonNote_v2 . . . . .	268
<b>17 Ops.Boolean</b>	<b>269</b>		
		17.1 Ops.Boolean . . . . .	269
		17.1.1 And . . . . .	269
		17.1.2 AndMultiPort_v2 . . . . .	269
		17.1.3 BoolByTrigger . . . . .	269
		17.1.4 Boolean . . . . .	269
		17.1.5 BoolToColor . . . . .	270
		17.1.6 BoolToNumber_v2 . . . . .	270
		17.1.7 BoolToString . . . . .	270
		17.1.8 DelayBooleanSimple . . . . .	270
		17.1.9 IfFalseThen . . . . .	271
		17.1.10 IfTrueThen_v2 . . . . .	271
		17.1.11 IsOne . . . . .	271

17.1.12	IsZero . . . . .	271	19.1.10	RGBLuminance . . . . .	281
17.1.13	MonoFlop . . . . .	272	19.1.11	RGBToCMYK . . . . .	281
17.1.14	Not . . . . .	272	19.1.12	RgbToHex . . . . .	281
17.1.15	Or . . . . .	272	19.1.13	RGBtoHSB . . . . .	281
17.1.16	OrNumber_v2 . . . . .	272	19.1.14	RGBtoHSL . . . . .	282
17.1.17	ParseBoolean_v2 . . . . .	273			
17.1.18	RouteBoolean . . . . .	273	<b>20</b>	<b>Ops.Data</b>	<b>282</b>
17.1.19	ToggleBool_v2 . . . . .	273	20.1	Ops.Data . . . . .	282
17.1.20	TriggerChangedFalse . . . . .	274	20.1.1	SpreadSheetArray . . . . .	282
17.1.21	TriggerChangedTrue . . . . .	274			
17.1.22	TriggerOnChangeBoolean_v2 . . . . .	274	<b>21</b>	<b>Ops.Data.Compose.Array</b>	<b>283</b>
<b>18</b>	<b>Ops.Cables</b>	<b>275</b>	21.1	Ops.Data.Compose.Array . . . . .	283
18.1	Ops.Cables . . . . .	275	21.1.1	CompArray . . . . .	283
18.1.1	AssetPathURL . . . . .	275	21.1.2	CompArrayPopNumber . . . . .	283
18.1.2	CablesInfo . . . . .	275	21.1.3	CompArrayPushArray . . . . .	283
18.1.3	CallBack_v2 . . . . .	275	21.1.4	CompArrayPushArrayItems . . . . .	283
18.1.4	FPS_v2 . . . . .	275	21.1.5	CompArrayPushNumber . . . . .	284
18.1.5	Function_v2 . . . . .	276	21.1.6	CompArrayPushNumbers . . . . .	284
18.1.6	GetOpName . . . . .	276	21.1.7	CompArrayPushObject . . . . .	284
18.1.7	GetSubPatchName . . . . .	276	21.1.8	CompArrayPushString . . . . .	284
18.1.8	LoadingJob . . . . .	276	21.1.9	CompArrayPushTexture . . . . .	285
18.1.9	LoadingStatus_v2 . . . . .	277	21.1.10	CompArraySetNumber . . . . .	285
18.1.10	PatchInfo_v2 . . . . .	277	21.1.11	CompArraySetObject . . . . .	285
18.1.11	UIMode . . . . .	277	21.1.12	CompArrayShiftNumber . . . . .	285
18.1.12	UploadAsset . . . . .	277	21.1.13	CompArraySnapshot . . . . .	286
18.1.13	UploadScreenshot . . . . .	278			
<b>19</b>	<b>Ops.Color</b>	<b>278</b>	<b>22</b>	<b>Ops.Data.Compose.Object</b>	<b>286</b>
19.1	Ops.Color . . . . .	278	22.1	Ops.Data.Compose.Object . . . . .	286
19.1.1	ColorArraySort . . . . .	278	22.1.1	CompObject . . . . .	286
19.1.2	ColorPalettes . . . . .	278	22.1.2	CompObjectFromObject . . . . .	286
19.1.3	ColorValue . . . . .	279	22.1.3	CompObjectSetArray . . . . .	287
19.1.4	EyeDropper . . . . .	279	22.1.4	CompObjectSetBool . . . . .	287
19.1.5	Gradient . . . . .	279	22.1.5	CompObjectSetNumber . . . . .	287
19.1.6	GradientColorArray . . . . .	279	22.1.6	CompObjectSetObject . . . . .	287
19.1.7	HexToRGB_v2 . . . . .	280	22.1.7	CompObjectSetString . . . . .	288
19.1.8	HSLtoRGB . . . . .	280			
19.1.9	LuminanceContrast . . . . .	280	<b>23</b>	<b>Ops.Data.Compose.String</b>	<b>288</b>
			23.1	Ops.Data.Compose.String . . . . .	288
			23.1.1	CompString . . . . .	288

23.1.2	CompStringAppend . . . . .	288	27.1.2	ConsoleLog . . . . .	296
23.1.3	CompStringShorten . . . . .	289	27.1.3	CrashOp . . . . .	296
<b>24</b>	<b>Ops.Data.JsonPath</b>	<b>289</b>	27.1.4	GlLogErrors . . . . .	296
24.1	Ops.Data.JsonPath . . . . .	289	27.1.5	GlStates . . . . .	297
24.1.1	ArrayGetArrayByPath . . . . .	289	27.1.6	ProfileGL . . . . .	297
24.1.2	ArrayGetArrayValuesByPath . . . . .	289	27.1.7	StopWatch . . . . .	297
24.1.3	ArrayGetNumberByPath . . . . .	290			
24.1.4	ArrayGetObjectByPath . . . . .	290			
24.1.5	ArrayGetStringByPath_v2 . . . . .	290			
24.1.6	ObjectGetArrayByPath . . . . .	290			
24.1.7	ObjectGetArrayValuesByPath . . . . .	291			
24.1.8	ObjectGetNumberByPath . . . . .	291			
24.1.9	ObjectGetObjectByPath . . . . .	291			
24.1.10	ObjectGetStringByPath_v2 . . . . .	291			
<b>25</b>	<b>Ops.Data.StackValues</b>	<b>292</b>			
25.1	Ops.Data.StackValues . . . . .	292			
25.1.1	StackGetArray . . . . .	292			
25.1.2	StackGetNumber . . . . .	292			
25.1.3	StackGetTexture . . . . .	292			
25.1.4	StackPushArray . . . . .	292			
25.1.5	StackPushNumber . . . . .	293			
25.1.6	StackPushTexture . . . . .	293			
<b>26</b>	<b>Ops.Date</b>	<b>293</b>			
26.1	Ops.Date . . . . .	293			
26.1.1	DateAndTime . . . . .	293			
26.1.2	DateCalc . . . . .	293			
26.1.3	DateDifference . . . . .	294			
26.1.4	DateFormatter . . . . .	294			
26.1.5	DatetoIsoTimestamp . . . . .	294			
26.1.6	DateTimestamp . . . . .	295			
26.1.7	Milliseconds . . . . .	295			
26.1.8	TimestampToIsoDate . . . . .	295			
<b>27</b>	<b>Ops.Debug</b>	<b>296</b>			
27.1	Ops.Debug . . . . .	296			
27.1.1	Console . . . . .	296			
<b>28</b>	<b>Ops.Devices</b>	<b>298</b>			
28.1	Ops.Devices . . . . .	298			
28.1.1	TouchGesture . . . . .	298			
28.1.2	TouchScreen . . . . .	298			
<b>29</b>	<b>Ops.Devices.Browser</b>	<b>299</b>			
29.1	Ops.Devices.Browser . . . . .	299			
29.1.1	BrowserInfo_v3 . . . . .	299			
29.1.2	ColorScheme . . . . .	299			
29.1.3	History . . . . .	299			
29.1.4	JsExpression . . . . .	300			
29.1.5	JsMemory . . . . .	300			
29.1.6	UserActivation . . . . .	300			
29.1.7	WebShare . . . . .	300			
<b>30</b>	<b>Ops.Devices.GamePad</b>	<b>301</b>			
30.1	Ops.Devices.GamePad . . . . .	301			
30.1.1	GamePad . . . . .	301			
30.1.2	GamePadJoystickAxis . . . . .	302			
30.1.3	GamePads . . . . .	302			
<b>31</b>	<b>Ops.Devices.Keyboard</b>	<b>303</b>			
31.1	Ops.Devices.Keyboard . . . . .	303			
31.1.1	CursorKeys . . . . .	303			
31.1.2	KeyPress_v2 . . . . .	303			
31.1.3	KeyPressLearn . . . . .	303			
31.1.4	PersonController . . . . .	304			
31.1.5	QWERTYtoMidi . . . . .	304			
<b>32</b>	<b>Ops.Devices.Midi</b>	<b>305</b>			
32.1	Ops.Devices.Midi . . . . .	305			
32.1.1	DeviceList . . . . .	305			

32.1.2	MidiCC_v3 . . . . .	305	36.1	Ops.Extension.Ai . . . . .	315
32.1.3	MidiCCOut_v2 . . . . .	305	36.1.1	LLmModelList . . . . .	315
32.1.4	MidiChord3 . . . . .	305	36.1.2	LlmPrompt . . . . .	315
32.1.5	MidiClock . . . . .	306	<b>37</b>	<b>Ops.Extension.AmmoPhysics</b>	<b>316</b>
32.1.6	MidiInputDevice_v2 . . . . .	306	37.1	Ops.Extension.AmmoPhysics . . . . .	316
32.1.7	MidiMonitor . . . . .	307	37.1.1	AmmoBody . . . . .	316
32.1.8	MidiNote . . . . .	307	37.1.2	AmmoBodyCollision . . . . .	316
32.1.9	MidiNoteFilter . . . . .	307	37.1.3	AmmoCharacter . . . . .	316
32.1.10	MidiNoteOut . . . . .	308	37.1.4	AmmoCharacterFpsCamera . . . . .	317
32.1.11	MidiNRPN . . . . .	308	37.1.5	AmmoDebugRenderer . . . . .	317
32.1.12	MidiNRPNOut . . . . .	308	37.1.6	AmmoEmitter . . . . .	318
32.1.13	MidiOutputDevice . . . . .	309	37.1.7	AmmoRaycast . . . . .	318
32.1.14	MidiTranspose . . . . .	309	37.1.8	AmmoWorld . . . . .	319
32.1.15	MidiValueToNote_v2 . . . . .	309	37.1.9	GltfAmmoBodies . . . . .	319
<b>33</b>	<b>Ops.Devices.Mobile</b>	<b>310</b>	<b>38</b>	<b>Ops.Extension.DetectGpu</b>	<b>320</b>
33.1	Ops.Devices.Mobile . . . . .	310	38.1	Ops.Extension.DetectGpu . . . . .	320
33.1.1	DeviceVibrate . . . . .	310	38.1.1	DetectGPU . . . . .	320
33.1.2	GeoLocation . . . . .	310	<b>39</b>	<b>Ops.Extension.ECharts</b>	<b>320</b>
33.1.3	LockOrientation . . . . .	310	39.1	Ops.Extension.ECharts . . . . .	320
33.1.4	MotionSensor_v2 . . . . .	310	39.1.1	ECharts . . . . .	320
33.1.5	Pinch . . . . .	311	39.1.2	EChartsEvent . . . . .	321
33.1.6	ScreenOrientation_v2 . . . . .	311	<b>40</b>	<b>Ops.Extension.FxHash</b>	<b>321</b>
33.1.7	ShakeGesture . . . . .	311	40.1	Ops.Extension.FxHash . . . . .	321
<b>34</b>	<b>Ops.Devices.Mouse</b>	<b>312</b>	40.1.1	FxHash . . . . .	321
34.1	Ops.Devices.Mouse . . . . .	312	<b>41</b>	<b>Ops.Extension.GlParticles</b>	<b>322</b>
34.1.1	Mouse_v4 . . . . .	312	41.1	Ops.Extension.GlParticles . . . . .	322
34.1.2	MouseButtons . . . . .	312	41.1.1	VelocityBoundaries . . . . .	322
34.1.3	MouseDrag . . . . .	312	<b>42</b>	<b>Ops.Extension.HtmlElementArray</b>	<b>323</b>
34.1.4	MouseWheel_v2 . . . . .	313	42.1	Ops.Extension.HtmlElementArray . . . . .	323
34.1.5	PointerLock . . . . .	313	42.1.1	DivElements . . . . .	323
<b>35</b>	<b>Ops.Devices.WebXr.Vr</b>	<b>314</b>	42.1.2	ElementArrayCssPropertyName . . . . .	323
35.1	Ops.Devices.WebXr.Vr . . . . .	314	42.1.3	ElementArrayCssPropertyNameString . . . . .	323
35.1.1	Vr . . . . .	314	<b>43</b>	<b>Ops.Extension.HtmlToTexture</b>	<b>324</b>
35.1.2	VrController . . . . .	314			
<b>36</b>	<b>Ops.Extension.Ai</b>	<b>315</b>			

43.1	Ops.Extension.HtmlToTexture . . . . .	324		
43.1.1	HtmlToTexture . . . . .	324		
<b>44</b>	<b>Ops.Extension.Lottie</b>	<b>324</b>		
44.1	Ops.Extension.Lottie . . . . .	324		
44.1.1	LottieSVGPlayer . . . . .	324		
44.1.2	LottieTexturePlayer_v2 . . . . .	324		
<b>45</b>	<b>Ops.Extension.LSystem</b>	<b>325</b>		
45.1	Ops.Extension.LSystem . . . . .	325		
45.1.1	Lsystem_v2 . . . . .	325		
<b>46</b>	<b>Ops.Extension.Mediapipe</b>	<b>326</b>		
46.1	Ops.Extension.Mediapipe . . . . .	326		
46.1.1	FaceMesh . . . . .	326		
46.1.2	MpFaceTracking . . . . .	326		
46.1.3	MpHand . . . . .	326		
46.1.4	MpHandCoordinate . . . . .	326		
46.1.5	MpHandTracking . . . . .	327		
46.1.6	MpPoseGetCoordinate . . . . .	327		
46.1.7	MpPoseTracking . . . . .	327		
<b>47</b>	<b>Ops.Extension.OpenType</b>	<b>328</b>		
47.1	Ops.Extension.OpenType . . . . .	328		
47.1.1	OpentypeFont . . . . .	328		
47.1.2	OpentypeToSvgPath . . . . .	328		
<b>48</b>	<b>Ops.Extension.Osc2Ws</b>	<b>329</b>		
48.1	Ops.Extension.Osc2Ws . . . . .	329		
48.1.1	Osc2WsArray . . . . .	329		
48.1.2	Osc2WsMessage . . . . .	329		
48.1.3	Osc2WsNumber . . . . .	329		
48.1.4	Osc2WsNumbers . . . . .	329		
<b>49</b>	<b>Ops.Extension.ReactionDiffusion</b>	<b>330</b>		
49.1	Ops.Extension.ReactionDiffusion . . . . .	330		
49.1.1	ReactionDiffusionSystem_v2 . . . . .	330		
<b>50</b>	<b>Ops.Extension.SocketCluster</b>	<b>331</b>		
50.1	Ops.Extension.SocketCluster . . . . .	331		
50.1.1	SocketClusterClient_v2 . . . . .	331		
50.1.2	SocketClusterReceiveObject . . . . .	332		
50.1.3	SocketClusterReceiveTrigger . . . . .	332		
50.1.4	SocketClusterSendObject . . . . .	332		
50.1.5	SocketClusterSendTrigger . . . . .	332		
50.1.6	SocketClusterTopicInfo_v2 . . . . .	333		
<b>51</b>	<b>Ops.Extension.SocketCluster.Deprecated</b>	<b>333</b>		
51.1	Ops.Extension.SocketCluster.Deprecated . . . . .	333		
51.1.1	SocketClusterReceiveArray . . . . .	333		
51.1.2	SocketClusterReceiveBoolean . . . . .	333		
51.1.3	SocketClusterReceiveNumber . . . . .	334		
51.1.4	SocketClusterReceiveString . . . . .	334		
51.1.5	SocketClusterSendArray . . . . .	334		
51.1.6	SocketClusterSendBoolean . . . . .	334		
51.1.7	SocketClusterSendNumber . . . . .	335		
51.1.8	SocketClusterSendString . . . . .	335		
<b>52</b>	<b>Ops.Extension.Standalone</b>	<b>336</b>		
52.1	Ops.Extension.Standalone . . . . .	336		
52.1.1	Ffmpeg . . . . .	336		
<b>53</b>	<b>Ops.Extension.Standalone.Files</b>	<b>336</b>		
53.1	Ops.Extension.Standalone.Files . . . . .	336		
53.1.1	CreateFile . . . . .	336		
53.1.2	Exist . . . . .	336		
53.1.3	FileUrlToPath . . . . .	337		
53.1.4	Makedir . . . . .	337		
53.1.5	OpenFileManager . . . . .	337		
53.1.6	PathToFileUrl . . . . .	337		
53.1.7	ReadDir . . . . .	338		
53.1.8	ResolvePath . . . . .	338		
53.1.9	SelectDir . . . . .	338		
53.1.10	SelectFile . . . . .	338		
53.1.11	Stat . . . . .	339		
53.1.12	SystemDirs . . . . .	339		
53.1.13	Watch . . . . .	339		
53.1.14	WriteBinaryFileFromBase64 . . . . .	339		
53.1.15	WriteTextFile . . . . .	340		

<b>54 Ops.Extension.Standalone.Net</b>	<b>340</b>	
54.1 Ops.Extension.Standalone.Net . . . . .	340	
54.1.1 HttpServer . . . . .	340	
54.1.2 HttpServerResponse . . . . .	340	
54.1.3 IpAddress . . . . .	341	
54.1.4 Osc_v2 . . . . .	341	
54.1.5 OscSend . . . . .	341	
54.1.6 ReadTextFile . . . . .	341	
54.1.7 SocketClusterServer . . . . .	342	
<b>55 Ops.Extension.SuperShapes</b>	<b>342</b>	
55.1 Ops.Extension.SuperShapes . . . . .	342	
55.1.1 SuperShapesCpu . . . . .	342	
55.1.2 SuperShapesGpu . . . . .	343	
<b>56 Ops.Extension.TeachableMachines</b>	<b>343</b>	
56.1 Ops.Extension.TeachableMachines . . . . .	343	
56.1.1 AudioClassifier . . . . .	343	
56.1.2 ImageClassifier_v2 . . . . .	343	
56.1.3 PoseDetection_v2 . . . . .	344	
<b>57 Ops.Extension.Trackingjs</b>	<b>344</b>	
57.1 Ops.Extension.Trackingjs . . . . .	344	
57.1.1 TrackWebcamColor . . . . .	344	
<b>58 Ops.Extension.Voice</b>	<b>345</b>	
58.1 Ops.Extension.Voice . . . . .	345	
58.1.1 MeSpeak . . . . .	345	
58.1.2 Say_v2 . . . . .	345	
58.1.3 SpeechRecognition . . . . .	345	
<b>59 Ops.Extension.WebGpu</b>	<b>346</b>	
59.1 Ops.Extension.WebGpu . . . . .	346	
59.1.1 ArrayToGpuBuffer . . . . .	346	
59.1.2 ArrayToTexture . . . . .	346	
59.1.3 AttributeAsColorMaterial . . . . .	347	
59.1.4 BasicMaterial . . . . .	347	
59.1.5 ColorTexture . . . . .	347	
59.1.6 CompCompute . . . . .	347	
59.1.7 ComputeStorageInput . . . . .	348	
59.1.8 ComputeStorageOutput . . . . .	348	
59.1.9 ComputeUniform . . . . .	348	
59.1.10 DefaultTextures . . . . .	349	
59.1.11 FaceCulling . . . . .	349	
59.1.12 FrontBacksideMaterial . . . . .	349	
59.1.13 FullScreenRect . . . . .	349	
59.1.14 GpuBufferToArray . . . . .	350	
59.1.15 MatCapMaterial . . . . .	350	
59.1.16 MeshInstancer . . . . .	350	
59.1.17 Pipeline . . . . .	351	
59.1.18 RenderToTexture . . . . .	351	
59.1.19 SaselHund . . . . .	351	
59.1.20 Texture . . . . .	351	
59.1.21 VizTexture . . . . .	352	
59.1.22 WebGpuCanvas . . . . .	352	
59.1.23 WebGpuInfo . . . . .	352	
<b>60 Ops.Gl</b>	<b>353</b>	
60.1 Ops.Gl . . . . .	353	
60.1.1 ArrayToTexture_v2 . . . . .	353	
60.1.2 BlendMode . . . . .	353	
60.1.3 CanvasFocus . . . . .	353	
60.1.4 CanvasInfo_v3 . . . . .	353	
60.1.5 CanvasToTexture . . . . .	354	
60.1.6ClearColor . . . . .	354	
60.1.7ClearDepth . . . . .	354	
60.1.8ColorMask . . . . .	355	
60.1.9ColorPick . . . . .	355	
60.1.10DirectionalTranslate . . . . .	355	
60.1.11DownloadTexture_v3 . . . . .	355	
60.1.12DrawTextureMapping . . . . .	356	
60.1.13ElementInBrowserViewport . . . . .	356	
60.1.14ExternalCanvas . . . . .	356	
60.1.15FaceCulling_v2 . . . . .	356	
60.1.16FontMSDF_v2 . . . . .	357	
60.1.17ForceCanvasSize . . . . .	357	
60.1.18GateTexture . . . . .	357	
60.1.19GlBlendFunc . . . . .	358	

60.1.20	GlInfo_v2 . . . . .	358	60.1.58	TextureArrayLoaderFromArray_v3 . . . . .	370
60.1.21	GlPrimitive . . . . .	358	60.1.59	TextureColorPick . . . . .	371
60.1.22	GradientTexture . . . . .	358	60.1.60	TextureToArray_v4 . . . . .	371
60.1.23	GridTransform . . . . .	359	60.1.61	TextureToPointArray3 . . . . .	371
60.1.24	Identity . . . . .	359	60.1.62	TextureToRandomPoints . . . . .	372
60.1.25	IdentityViewMatrix . . . . .	359	60.1.63	TriggerOnCanvasResize . . . . .	372
60.1.26	ImageSequenceAnim_v2 . . . . .	360	60.1.64	ValidTexture . . . . .	372
60.1.27	InteractiveRectangle_v2 . . . . .	360	60.1.65	ViewPortSize . . . . .	372
60.1.28	LayerSequence . . . . .	360			
60.1.29	LineFont_v2 . . . . .	361			
60.1.30	MainLoop_v2 . . . . .	361			
60.1.31	MediaRecorder_v2 . . . . .	361			
60.1.32	MeshInstancer_v4 . . . . .	362			
60.1.33	MeshMorph . . . . .	362			
60.1.34	NormalizeScreenCoordinates . . . . .	363			
60.1.35	OrTexture . . . . .	363			
60.1.36	Orthogonal_v2 . . . . .	363			
60.1.37	OverwriteViewportSize . . . . .	363			
60.1.38	Performance . . . . .	364			
60.1.39	PerformanceMeasure . . . . .	364			
60.1.40	Perspective . . . . .	364			
60.1.41	PixelProjection_v3 . . . . .	365			
60.1.42	PointCollector . . . . .	365			
60.1.43	PointCollectorCollect . . . . .	365			
60.1.44	PointCollectorScreenCoords . . . . .	365			
60.1.45	RandomCluster . . . . .	366			
60.1.46	RenderAnim_v2 . . . . .	366			
60.1.47	RenderGeometry_v2 . . . . .	366			
60.1.48	RenderToTexture_v3 . . . . .	367			
60.1.49	RenderToTextures_v3 . . . . .	367			
60.1.50	ResetTransform . . . . .	367			
60.1.51	SaveScreenShot_v3 . . . . .	368			
60.1.52	ShowNormals_v2 . . . . .	368			
60.1.53	SurfaceScatter_v2 . . . . .	368			
60.1.54	TextMeshMSDF_v2 . . . . .	369			
60.1.55	Texture_v2 . . . . .	369			
60.1.56	TextureArray . . . . .	370			
60.1.57	TextureArrayLoader_v2 . . . . .	370			
			<b>61</b>	<b>Ops.Gl.CubeMap</b>	<b>373</b>
			61.1	Ops.Gl.CubeMap . . . . .	373
			61.1.1	CubeMapFromTextures_v2 . . . . .	373
			61.1.2	CubeMapMaterial_v2 . . . . .	373
			61.1.3	CubemapToEquirectangularTexture_v2 . . . . .	374
			61.1.4	EquirectangularTextureToCubemap . . . . .	374
			61.1.5	RenderToCubemap_v3 . . . . .	374
			61.1.6	Skybox . . . . .	375
			<b>62</b>	<b>Ops.Gl.Geometry</b>	<b>375</b>
			62.1	Ops.Gl.Geometry . . . . .	375
			62.1.1	BoundingBoxVisible . . . . .	375
			62.1.2	GeometryBoundingBox . . . . .	375
			<b>63</b>	<b>Ops.Gl.GLTF</b>	<b>376</b>
			63.1	Ops.Gl.GLTF . . . . .	376
			63.1.1	GltfAnimationArray . . . . .	376
			63.1.2	GltfCameraViewMatrix . . . . .	376
			63.1.3	GltfDracoCompression . . . . .	377
			63.1.4	GltfGeometry . . . . .	377
			63.1.5	GltfHierarchy . . . . .	377
			63.1.6	GltfInfo . . . . .	377
			63.1.7	GltfMeshSequence_v2 . . . . .	378
			63.1.8	GltfMorphTargets . . . . .	378
			63.1.9	GltfNode_v2 . . . . .	378
			63.1.10	GltfNodeSineAnim . . . . .	379
			63.1.11	GltfNodeTransform_v2 . . . . .	379
			63.1.12	GltfNodeTransforms_v3 . . . . .	379
			63.1.13	GltfScene_v4 . . . . .	380
			63.1.14	GltfSetMaterial . . . . .	380

63.1.15	GltfSkin . . . . .	380
63.1.16	GltfTexture . . . . .	381
63.1.17	GltfTransformNode . . . . .	381
63.1.18	GltfVertexAnim . . . . .	381
<b>64</b>	<b>Ops.Gl.ImageCompose</b>	<b>382</b>
64.1	Ops.Gl.ImageCompose . . . . .	382
64.1.1	Alpha . . . . .	382
64.1.2	AlphaMask_v2 . . . . .	382
64.1.3	BarrelDistortion_v3 . . . . .	382
64.1.4	Blur . . . . .	382
64.1.5	Border_v2 . . . . .	383
64.1.6	BrightnessContrast . . . . .	383
64.1.7	BulgePinch . . . . .	383
64.1.8	CheckerBoard_v2 . . . . .	384
64.1.9	ChromaticAberration_v2 . . . . .	384
64.1.10	CircleTexture_v4 . . . . .	384
64.1.11	ClampTexture_v2 . . . . .	385
64.1.12	Clarity . . . . .	385
64.1.13	Color_v2 . . . . .	385
64.1.14	ColorBalance_v2 . . . . .	385
64.1.15	ColorChannel_v2 . . . . .	386
64.1.16	ColorMap_v2 . . . . .	386
64.1.17	Denoise . . . . .	386
64.1.18	DepthTexture_v2 . . . . .	387
64.1.19	DepthTextureFocus_v2 . . . . .	387
64.1.20	Desaturate . . . . .	387
64.1.21	Dither_v2 . . . . .	387
64.1.22	DrawImage_v3 . . . . .	388
64.1.23	EdgeDetection_v4 . . . . .	388
64.1.24	Emboss . . . . .	388
64.1.25	FastBlur_v2 . . . . .	389
64.1.26	Flip . . . . .	389
64.1.27	Float32ToRgbeTexture . . . . .	389
64.1.28	Fog_v4 . . . . .	390
64.1.29	FXAA . . . . .	390
64.1.30	GammaCorrection_v2 . . . . .	390
64.1.31	Gradient_v2 . . . . .	391
64.1.32	GridTexture_v2 . . . . .	391
64.1.33	GrowPixels_v2 . . . . .	391
64.1.34	Hue . . . . .	392
64.1.35	ImageCompose_v4 . . . . .	392
64.1.36	ImageComposeAspectRatio . . . . .	392
64.1.37	ImageComposeSnapshot . . . . .	393
64.1.38	Interlace . . . . .	393
64.1.39	Invert_v2 . . . . .	393
64.1.40	Kaleidoscope_v2 . . . . .	393
64.1.41	LensDirt_v2 . . . . .	394
64.1.42	LensScratches_v2 . . . . .	394
64.1.43	Levels_v2 . . . . .	395
64.1.44	LumaKey_v3 . . . . .	395
64.1.45	LUTMap . . . . .	395
64.1.46	Mirror . . . . .	395
64.1.47	Mix . . . . .	396
64.1.48	MultiDrawImage . . . . .	396
64.1.49	OnePassBlur . . . . .	397
64.1.50	PatternLookup . . . . .	397
64.1.51	Pixelate_v2 . . . . .	397
64.1.52	PixelColor . . . . .	398
64.1.53	PixelDifference . . . . .	398
64.1.54	PixelDisplacement_v4 . . . . .	398
64.1.55	Plasma_v2 . . . . .	398
64.1.56	PolarCoords . . . . .	399
64.1.57	Posterize_v2 . . . . .	399
64.1.58	PseudoLensFlares . . . . .	399
64.1.59	RandomNumberTexture . . . . .	400
64.1.60	RectangleTexture_v5 . . . . .	400
64.1.61	RemoveAlpha . . . . .	401
64.1.62	RepeatTexture_v2 . . . . .	401
64.1.63	RgbMultiply . . . . .	401
64.1.64	RGBOffset_v2 . . . . .	401
64.1.65	RgbToHsvTexture . . . . .	402
64.1.66	RotateTexture_v2 . . . . .	402
64.1.67	RoundCorners . . . . .	402
64.1.68	ScaleTexture_v3 . . . . .	403
64.1.69	ScrollTexture . . . . .	403
64.1.70	Shapes2d_v2 . . . . .	403

64.1.71	Sharpen . . . . .	404	66.1.13	TriangleNoise_v2 . . . . .	416
64.1.72	SkewStretchImage_v2 . . . . .	404	66.1.14	ValueNoise_v2 . . . . .	416
64.1.73	Stripes_v4 . . . . .	404	66.1.15	VoronoiNoise_v2 . . . . .	416
64.1.74	TexMathModulo . . . . .	405	66.1.16	WorleyNoise_v2 . . . . .	417
64.1.75	TextureDifference . . . . .	405	<b>67</b>	<b>Ops.Gl.Matrix</b>	<b>417</b>
64.1.76	ToNormalMap_v2 . . . . .	405	67.1	Ops.Gl.Matrix . . . . .	417
64.1.77	Twirl_v4 . . . . .	405	67.1.1	AnimMatrix . . . . .	417
64.1.78	Vibrance . . . . .	406	67.1.2	ArrayPathFollow . . . . .	417
64.1.79	Vignette_v3 . . . . .	406	67.1.3	ArrayPathFollowParticles_v2 . . . . .	418
64.1.80	Waveform_v3 . . . . .	406	67.1.4	Billboard . . . . .	418
64.1.81	WaveformGradient_v4 . . . . .	407	67.1.5	Camera_v2 . . . . .	418
64.1.82	Wobble_v2 . . . . .	407	67.1.6	CameraInfo . . . . .	419
64.1.83	ZoomBlur_v2 . . . . .	407	67.1.7	CameraPosition . . . . .	419
<b>65</b>	<b>Ops.Gl.ImageCompose.Math</b>	<b>408</b>	67.1.8	Coordinates . . . . .	420
65.1	Ops.Gl.ImageCompose.Math . . . . .	408	67.1.9	DeviceOrientationCamera . . . . .	420
65.1.1	ColorMapRange . . . . .	408	67.1.10	GetMatrixScaling . . . . .	420
65.1.2	Normalize . . . . .	408	67.1.11	GetModelMatrix . . . . .	420
65.1.3	RgbToFloat32Texture . . . . .	409	67.1.12	GetProjectionMatrix . . . . .	421
65.1.4	RgbMath . . . . .	409	67.1.13	GetViewMatrix . . . . .	421
65.1.5	RgbMathExpression . . . . .	409	67.1.14	InterpolateMatrix . . . . .	421
65.1.6	RgbTransform . . . . .	410	67.1.15	InvertMatrix . . . . .	421
65.1.7	Round . . . . .	410	67.1.16	LookatCamera . . . . .	422
65.1.8	TexMathCompare . . . . .	410	67.1.17	MatrixTranslation . . . . .	422
<b>66</b>	<b>Ops.Gl.ImageCompose.Noise</b>	<b>411</b>	67.1.18	MultiplyModelMatrix . . . . .	422
66.1	Ops.Gl.ImageCompose.Noise . . . . .	411	67.1.19	MulViewMatrix . . . . .	422
66.1.1	CellularNoise_v2 . . . . .	411	67.1.20	Quaternion . . . . .	423
66.1.2	FBMNoise_v2 . . . . .	412	67.1.21	QuaternionCamera . . . . .	423
66.1.3	GaborNoise . . . . .	412	67.1.22	RandomGridPlacement . . . . .	423
66.1.4	GlitchNoise_v2 . . . . .	412	67.1.23	RandomGridPlacementArrays . . . . .	424
66.1.5	HexagonNoise_v2 . . . . .	413	67.1.24	Scale . . . . .	424
66.1.6	LayerNoise_v3 . . . . .	413	67.1.25	ScaleXYZViewMatrix . . . . .	424
66.1.7	Noise_v2 . . . . .	413	67.1.26	ScreenCoordinates_v2 . . . . .	424
66.1.8	PerlinNoise_v2 . . . . .	414	67.1.27	ScreenPosTo3d_v3 . . . . .	425
66.1.9	PixelNoise_v3 . . . . .	414	67.1.28	SetProjectionMatrix . . . . .	425
66.1.10	PolkaDotNoise_v2 . . . . .	415	67.1.29	Shear . . . . .	425
66.1.11	Shardnoise . . . . .	415	67.1.30	TransformMatrix . . . . .	425
66.1.12	SimplexNoise_v2 . . . . .	415	67.1.31	TransformMul . . . . .	426
			67.1.32	Translate . . . . .	426

67.1.33	TranslateView . . . . .	426	69.1	Ops.Gl.Pbr . . . . .	438
67.1.34	VectorTranslate . . . . .	427	69.1.1	PbrEnvironmentLight . . . . .	438
67.1.35	WASDCamera_v2 . . . . .	427	69.1.2	PbrMaterial . . . . .	438

<b>68</b>	<b>Ops.Gl.Meshes</b>	<b>428</b>	<b>70</b>	<b>Ops.Gl.Phong</b>	<b>439</b>
68.1	Ops.Gl.Meshes . . . . .	428	70.1	Ops.Gl.Phong . . . . .	439
68.1.1	ComposingGridOverlay . . . . .	428	70.1.1	AmbientLight_v4 . . . . .	439
68.1.2	Cone . . . . .	428	70.1.2	DirectionalLight_v5 . . . . .	439
68.1.3	Corner . . . . .	428	70.1.3	LambertMaterial_v2 . . . . .	440
68.1.4	Cylinder_v2 . . . . .	428	70.1.4	PhongMaterial_v6 . . . . .	440
68.1.5	FloorGrid . . . . .	429	70.1.5	PointLight_v5 . . . . .	441
68.1.6	FreeFormPlane . . . . .	429	70.1.6	ResetLights . . . . .	442
68.1.7	FullscreenRectangle_v2 . . . . .	430	70.1.7	SpotLight_v5 . . . . .	442
68.1.8	GeometryToTexture_v3 . . . . .	430			
68.1.9	Grid . . . . .	430			
68.1.10	HeightMap . . . . .	430			
68.1.11	Helix . . . . .	431			
68.1.12	Icosahedron_v2 . . . . .	431			
68.1.13	Line . . . . .	431			
68.1.14	LinesArray . . . . .	432			
68.1.15	MeshInstancerFromTexture_v3 . . . . .	432			
68.1.16	ParametricSurface . . . . .	432			
68.1.17	PointCloudFromArray_v2 . . . . .	433			
68.1.18	PointCloudFromTexture . . . . .	433			
68.1.19	Polyhedron_v2 . . . . .	433			
68.1.20	Pyramid_v2 . . . . .	434			
68.1.21	QuadWarpTexture . . . . .	434			
68.1.22	Rectangle9Slice . . . . .	434			
68.1.23	RectangleFrame_v2 . . . . .	435			
68.1.24	RectangleRounded_v2 . . . . .	435			
68.1.25	SimpleSpline_v2 . . . . .	435			
68.1.26	SimpleWireframe . . . . .	436			
68.1.27	SplineMesh_v2 . . . . .	436			
68.1.28	SplineMeshMaterial_v2 . . . . .	436			
68.1.29	TextMesh_v2 . . . . .	436			
68.1.30	Torus_v3 . . . . .	437			
68.1.31	TriangleSphere . . . . .	437			
<b>69</b>	<b>Ops.Gl.Pbr</b>	<b>438</b>	<b>72</b>	<b>Ops.Gl.ShaderEffects</b>	<b>449</b>

72.1	<b>Ops.Gl.ShaderEffects</b>	449	73.1.2	<b>ColorTexture</b>	461
72.1.1	AreaDiscardPixel_v2	449	73.1.3	CombineTextures	461
72.1.2	AreaRotate_v2	450	73.1.4	CopyTexture_v3	462
72.1.3	AreaScaler_v3	450	73.1.5	EmptyTexture	462
72.1.4	AreaTranslateFBMNoise	450	73.1.6	ExrTexture	462
72.1.5	AreaTranslateMeshes_v3	451	73.1.7	GraphTexture	463
72.1.6	Bend_v2	451	73.1.8	Histogram	463
72.1.7	ClampVertexPosition_v2	451	73.1.9	MontageTextures_v2	463
72.1.8	ColorArea_v5	451	73.1.10	NoiseTexture	464
72.1.9	DeformArea	452	73.1.11	PaletteTexture	464
72.1.10	DiscardMaterialAlpha	452	73.1.12	SequenceTextures	464
72.1.11	ExplodeDividedMesh_v2	452	73.1.13	SSAO	465
72.1.12	FogEffect	453	73.1.14	SwitchTextureMultiPort_v2	465
72.1.13	FresnelGlow	453	73.1.15	SwitchTextures_v2	465
72.1.14	InstancedDisplacementMap_v2	453	73.1.16	TextTexture_v6	466
72.1.15	InstancedPerlinPosition_v2	454	73.1.17	TextureArrayInfo	467
72.1.16	InstancedTextureColorize	454	73.1.18	TextureInfo_v2	467
72.1.17	LimitMeshByTexCoord	454	73.1.19	TextureSVG_v2	467
72.1.18	MeshPixelNoise_v2	455	73.1.20	TextureToBase64_v5	467
72.1.19	ModuloVertexPosition	455	73.1.21	TextureToCoordinateGrid	468
72.1.20	PerlinAreaDeform_v4	455	73.1.22	VideoTexture_v3	468
72.1.21	ScaleByNormal_v2	456	73.1.23	WebcamTexture_v3	469
72.1.22	Shadow_v3	456			
72.1.23	SplineDeform_v2	456			
72.1.24	TextureProjection_v2	457			
72.1.25	TransformTextureCoordinates	457			
72.1.26	TransformVertex	457			
72.1.27	Twist_v3	458			
72.1.28	UseVertexColor	458			
72.1.29	VertexArea	458			
72.1.30	VertexColorAsAlpha	459			
72.1.31	VertexDisplacementMap_v5	459			
72.1.32	VertexNumberLimit_v2	459			
72.1.33	VertexPositionFromTexture_v2	459			
72.1.34	VertexWobble_v2	460			
<b>73</b>	<b>Ops.Gl.Textures</b>	<b>461</b>	<b>74</b>	<b>Ops.Graphics</b>	<b>469</b>
73.1	<b>Ops.Gl.Textures</b>	461	74.1	<b>Ops.Graphics</b>	469
73.1.1	Base64ToTexture	461	74.1.1	ArrayToExr	469
			74.1.2	DepthTest	469
			74.1.3	GeometryMergeSimple	470
			74.1.4	GetMaterialId	470
			74.1.5	GetObjectId	470
			74.1.6	OrbitControls_v3	470
			74.1.7	Transform	471
			74.1.8	TransformView	471
			<b>75</b>	<b>Ops.Graphics.Geometry</b>	<b>472</b>
			75.1	<b>Ops.Graphics.Geometry</b>	472
			75.1.1	AlignGeometry	472
			75.1.2	BoundingBox	472
			75.1.3	CalculateNormals	472

75.1.4	DivideGeometry . . . . .	472	77.1.7	Star_v2 . . . . .	482
75.1.5	FlipNormals . . . . .	473	77.1.8	Triangle_v2 . . . . .	482
75.1.6	FreezeMeshes . . . . .	473	<b>78</b>	<b>Ops.Html</b> . . . . .	<b>483</b>
75.1.7	GeometryAttributes . . . . .	473	78.1	Ops.Html . . . . .	483
75.1.8	GeometryExtrude . . . . .	473	78.1.1	ActiveElement . . . . .	483
75.1.9	GeometryFromArrays . . . . .	474	78.1.2	AlignElement . . . . .	483
75.1.10	GeometryInfo . . . . .	474	78.1.3	AppendChild_v2 . . . . .	484
75.1.11	GeometryMerge . . . . .	474	78.1.4	BrowserSpecificFile_v2 . . . . .	484
75.1.12	GeometryToObj . . . . .	474	78.1.5	CanvasToBase64 . . . . .	484
75.1.13	GeometryToWireframeArray3 . . . . .	475	78.1.6	CompareImages_v2 . . . . .	484
75.1.14	GeometryUnIndex . . . . .	475	78.1.7	DocumentBody . . . . .	485
75.1.15	ObjGeometry . . . . .	475	78.1.8	DraggableElement . . . . .	485
75.1.16	RandomizeTriangles . . . . .	475	78.1.9	ElementAsHtmlString . . . . .	485
75.1.17	ReverseVertices . . . . .	476	78.1.10	ElementChilds_v2 . . . . .	485
75.1.18	ScaleGeometry . . . . .	476	78.1.11	ElementChildsMultiPort_v2 . . . . .	486
75.1.19	SortGeometryAxis . . . . .	476	78.1.12	ElementClientRect . . . . .	486
75.1.20	SvgPathToGeometry_v2 . . . . .	476	78.1.13	ElementCssCursor_v3 . . . . .	486
75.1.21	TesselateGeometry . . . . .	477	78.1.14	ElementCssString . . . . .	487
75.1.22	TransformGeometry . . . . .	477	78.1.15	ElementDataSet . . . . .	487
75.1.23	TriangleArrayToGeometry_v2 . . . . .	477	78.1.16	ElementEquals . . . . .	487
75.1.24	Triangulate2dPath . . . . .	477	78.1.17	ElementFadeInOut_v2 . . . . .	487
<b>76</b>	<b>Ops.Graphics.Intersection</b> . . . . .	<b>478</b>	78.1.18	ElementGetClosest . . . . .	488
76.1	Ops.Graphics.Intersection . . . . .	478	78.1.19	ElementGradientBg . . . . .	488
76.1.1	FilterIntersections . . . . .	478	78.1.20	ElementHasClass . . . . .	488
76.1.2	IntersectBody . . . . .	478	78.1.21	ElementInfo . . . . .	488
76.1.3	IntersectTestBody . . . . .	479	78.1.22	ElementIsFocused . . . . .	489
76.1.4	IntersectTestPoint . . . . .	479	78.1.23	ElementsPositionsByClass . . . . .	489
76.1.5	IntersectTestRaycast . . . . .	479	78.1.24	FontFile_v2 . . . . .	489
76.1.6	IntersectWorld . . . . .	480	78.1.25	FontsLoaded . . . . .	489
<b>77</b>	<b>Ops.Graphics.Meshes</b> . . . . .	<b>480</b>	78.1.26	FullscreenMode . . . . .	490
77.1	Ops.Graphics.Meshes . . . . .	480	78.1.27	GlCopyToCanvas . . . . .	490
77.1.1	CablesLogo . . . . .	480	78.1.28	HyperLink_v3 . . . . .	490
77.1.2	Circle_v3 . . . . .	480	78.1.29	InnerHTML . . . . .	490
77.1.3	Cross . . . . .	481	78.1.30	InnerHtmlAppend . . . . .	491
77.1.4	Cube_v2 . . . . .	481	78.1.31	MailtoLink . . . . .	491
77.1.5	Rectangle_v4 . . . . .	481	78.1.32	MarkdownToHtml . . . . .	491
77.1.6	Sphere_v3 . . . . .	482	78.1.33	ModalOverlay . . . . .	491
			78.1.34	QuerySelector_v3 . . . . .	492

78.1.35	QuerySelectorAll_v2 . . . . .	492	<b>83</b>	<b>Ops.Html.Utils</b>	<b>501</b>
78.1.36	ReloadPage . . . . .	492	83.1	Ops.Html.Utils . . . . .	501
78.1.37	ScrollIntoView . . . . .	492	83.1.1	CablesLink . . . . .	501
78.1.38	ScrollPosition_v2 . . . . .	493	83.1.2	LoadingIndicator_v2 . . . . .	501
78.1.39	ScrollTo . . . . .	493	83.1.3	Notification . . . . .	502
78.1.40	WindowClose . . . . .	493	83.1.4	PlayButton . . . . .	502
78.1.41	WindowHasFocus . . . . .	493	83.1.5	PlayerControlPanel_v2 . . . . .	502
78.1.42	WindowInfo . . . . .	494	83.1.6	QrCode . . . . .	503
78.1.43	WindowScroll . . . . .	494	83.1.7	YoutubePlayer . . . . .	503
<b>79</b>	<b>Ops.Html.Attributes</b>	<b>494</b>	<b>84</b>	<b>Ops.Json</b>	<b>504</b>
79.1	Ops.Html.Attributes . . . . .	494	84.1	Ops.Json . . . . .	504
79.1.1	ElementAccessibility . . . . .	494	84.1.1	ArrayOfObjectsMultiPort_v2 . . . . .	504
79.1.2	ElementAttributes . . . . .	494	84.1.2	CopyObject . . . . .	504
79.1.3	ElementGetAttribute . . . . .	495	84.1.3	CsvArray . . . . .	504
79.1.4	ElementSetAttribute . . . . .	495	84.1.4	CsvColumnArray_v2 . . . . .	504
<b>80</b>	<b>Ops.Html.Css</b>	<b>496</b>	84.1.5	EmptyObject . . . . .	505
80.1	Ops.Html.Css . . . . .	496	84.1.6	FilterValidObject . . . . .	505
80.1.1	ElementBackDropFilter . . . . .	496	84.1.7	GateObject . . . . .	505
80.1.2	ElementMargin . . . . .	496	84.1.8	HttpFetchStream . . . . .	505
<b>81</b>	<b>Ops.Html.Elements</b>	<b>497</b>	84.1.9	HttpRequest_v4 . . . . .	505
81.1	Ops.Html.Elements . . . . .	497	84.1.10	Object . . . . .	506
81.1.1	AudioMediaElement . . . . .	497	84.1.11	ObjectDeleteKey . . . . .	506
81.1.2	Element_v2 . . . . .	497	84.1.12	ObjectFilterContentByKey . . . . .	506
81.1.3	IFrame_v3 . . . . .	497	84.1.13	ObjectFunnel . . . . .	507
81.1.4	ImageElement_v3 . . . . .	498	84.1.14	ObjectGetArray_v2 . . . . .	507
81.1.5	InputElement . . . . .	498	84.1.15	ObjectGetNumber_v2 . . . . .	507
81.1.6	VideoElement . . . . .	498	84.1.16	ObjectGetObject_v2 . . . . .	507
<b>82</b>	<b>Ops.Html.Event</b>	<b>499</b>	84.1.17	ObjectGetString_v2 . . . . .	508
82.1	Ops.Html.Event . . . . .	499	84.1.18	ObjectIsNull . . . . .	508
82.1.1	ElementEventListener_v2 . . . . .	499	84.1.19	ObjectKeys . . . . .	508
82.1.2	ElementPointerEvents . . . . .	499	84.1.20	ObjectMerge . . . . .	508
82.1.3	ElementsArrayEventListener . . . . .	500	84.1.21	ObjectOr . . . . .	509
82.1.4	PreventDefault . . . . .	500	84.1.22	ObjectRecorder . . . . .	509
82.1.5	StopPropagation . . . . .	500	84.1.23	ObjectSetArray_v2 . . . . .	509

84.1.28	ObjectSetString_v2 . . . . .	510	85.1.23	Ease . . . . .	520
84.1.29	ObjectStringify_v2 . . . . .	511	85.1.24	Exp . . . . .	520
84.1.30	ObjectToArray . . . . .	511	85.1.25	FlipSign . . . . .	521
84.1.31	ObjectValuesAsArray . . . . .	511	85.1.26	Floor . . . . .	521
84.1.32	ParseObject_v2 . . . . .	511	85.1.27	Fract . . . . .	521
84.1.33	RemoveDataUrlPrefix . . . . .	512	85.1.28	GaussianRandomArray . . . . .	521
84.1.34	RouteObject . . . . .	512	85.1.29	Incrementor . . . . .	522
84.1.35	SaveJsonFile . . . . .	512	85.1.30	IndexFraction . . . . .	522
84.1.36	SequenceObjects_v2 . . . . .	512	85.1.31	Interpolate . . . . .	522
84.1.37	SwitchObject . . . . .	513	85.1.32	IsNumberRising . . . . .	522
84.1.38	SwitchObjectMultiPort_v2 . . . . .	513	85.1.33	Log . . . . .	523
84.1.39	TriggerObject . . . . .	514	85.1.34	MapGeoCoordsSpherical . . . . .	523
84.1.40	TriggerObjectSetNumber . . . . .	514	85.1.35	MapRange . . . . .	523
84.1.41	TriggerObjectSetString . . . . .	514	85.1.36	Math . . . . .	523
<b>85</b>	<b>Ops.Math</b>	<b>515</b>	85.1.37	MathExpression . . . . .	524
85.1	Ops.Math . . . . .	515	85.1.38	Max . . . . .	524
85.1.1	Abs . . . . .	515	85.1.39	MaxSinceReset . . . . .	524
85.1.2	Accumulator . . . . .	515	85.1.40	MercatorCoord . . . . .	524
85.1.3	AddUp . . . . .	515	85.1.41	MercatorCoordsArray . . . . .	525
85.1.4	AngleBetweenPoints . . . . .	515	85.1.42	Min_v3 . . . . .	525
85.1.5	Array3MultiplyMatrix . . . . .	516	85.1.43	MinSinceReset . . . . .	525
85.1.6	Array3To2dProjection . . . . .	516	85.1.44	Modulo . . . . .	525
85.1.7	Atan2 . . . . .	516	85.1.45	MulMatrixXyz . . . . .	526
85.1.8	Average . . . . .	516	85.1.46	Multiply . . . . .	526
85.1.9	ButterflyCurve . . . . .	517	85.1.47	Multiply3Numbers . . . . .	526
85.1.10	Ceil . . . . .	517	85.1.48	Normalize . . . . .	526
85.1.11	CircleCoordinates . . . . .	517	85.1.49	NumberDivisible . . . . .	527
85.1.12	Clamp . . . . .	517	85.1.50	OneMinus . . . . .	527
85.1.13	Cosine . . . . .	518	85.1.51	PerlinNoise_v2 . . . . .	527
85.1.14	Cross . . . . .	518	85.1.52	Pi . . . . .	527
85.1.15	Degrees . . . . .	518	85.1.53	PointInRectangle2d . . . . .	528
85.1.16	DegreeToVector . . . . .	518	85.1.54	Pow . . . . .	528
85.1.17	Delta . . . . .	519	85.1.55	PowerOfTwoSize . . . . .	528
85.1.18	DeltaSum . . . . .	519	85.1.56	Radians . . . . .	528
85.1.19	Difference . . . . .	519	85.1.57	RandomCounter . . . . .	529
85.1.20	Distance2d . . . . .	519	85.1.58	RandomNumbers_v3 . . . . .	529
85.1.21	Distance3d_v2 . . . . .	520	85.1.59	RandomNumbersFromString . . . . .	529
85.1.22	Divide . . . . .	520	85.1.60	RotationFromNormal . . . . .	529

85.1.61	Round . . . . .	530	89.1.2	DelayNumberSimple . . . . .	537																																																																																																						
85.1.62	RoundEven . . . . .	530	89.1.3	FilterValidNumber . . . . .	538																																																																																																						
85.1.63	SchlickBias . . . . .	530	89.1.4	FreezeNumber . . . . .	538																																																																																																						
85.1.64	Sign . . . . .	530	89.1.5	GateNumber . . . . .	538																																																																																																						
85.1.65	SimpleMovingAverage . . . . .	531	89.1.6	Integer . . . . .	538																																																																																																						
85.1.66	Sine . . . . .	531	89.1.7	MaximumSafeInteger . . . . .	539																																																																																																						
85.1.67	Speed . . . . .	531	89.1.8	MinimumSafeInteger . . . . .	539																																																																																																						
85.1.68	Sqrt . . . . .	531	89.1.9	Number . . . . .	539																																																																																																						
85.1.69	Subtract . . . . .	531	89.1.10	NumberSequence . . . . .	539																																																																																																						
85.1.70	Sum . . . . .	532	89.1.11	Preset . . . . .	540																																																																																																						
85.1.71	Tangent . . . . .	532	89.1.12	PreviousNumberStore . . . . .	540																																																																																																						
85.1.72	TriggerMathExpression . . . . .	532	89.1.13	RouteNumber . . . . .	540																																																																																																						
85.1.73	TriggerRandomNumber_v3 . . . . .	533	89.1.14	SequenceNumbers . . . . .	541																																																																																																						
85.1.74	VectorLength . . . . .	533	89.1.15	SumMultiPort_v2 . . . . .	541																																																																																																						
<b>86</b>	<b>Ops.Math.Compare</b>	<b>533</b>	89.1.16	SwitchNumber . . . . .	542																																																																																																						
86.1	Ops.Math.Compare . . . . .	533	89.1.17	SwitchNumberMultiPort_v2 . . . . .	542																																																																																																						
86.1.1	Between . . . . .	533	89.1.18	SwitchNumberOnTrigger . . . . .	542																																																																																																						
86.1.2	BetweenEquals . . . . .	533	89.1.19	Trigger3Numbers . . . . .	543																																																																																																						
86.1.3	CompareNumbers . . . . .	534	89.1.20	TriggerOnChangeNumber_v2 . . . . .	543																																																																																																						
86.1.4	Equals . . . . .	534	<b>90</b>	<b>Ops Sidebar</b>	<b>544</b>																																																																																																						
86.1.5	GreaterOrEquals . . . . .	534	90.1	Ops Sidebar . . . . .	544	86.1.6	GreaterThan . . . . .	534	90.1.1	Button_v2 . . . . .	544	86.1.7	IfBetweenThen . . . . .	535	90.1.2	ColorPicker_v3 . . . . .	544	86.1.8	IsEven . . . . .	535	90.1.3	DisplayValue_v2 . . . . .	544	86.1.9	LessThan . . . . .	535	90.1.4	DropDown_v2 . . . . .	545	<b>87</b>	<b>Ops.Net</b>	<b>536</b>	90.1.5	Group . . . . .	545	87.1	Ops.Net . . . . .	536	90.1.6	Incrementor_v3 . . . . .	545	87.1.1	CorsProxy_v3 . . . . .	536	90.1.7	LocalFileToDataUrl . . . . .	546	<b>88</b>	<b>Ops.Net.WebSocket</b>	<b>536</b>	90.1.8	NumberInput_v2 . . . . .	546	88.1	Ops.Net.WebSocket . . . . .	536	90.1.9	Presets_v2 . . . . .	546	88.1.1	WebSocket_v2 . . . . .	536	90.1.10	Sidebar . . . . .	547	88.1.2	WebSocketSend . . . . .	536	90.1.11	SidebarDateTime . . . . .	547	<b>89</b>	<b>Ops.Number</b>	<b>537</b>	90.1.12	SidebarElement . . . . .	547	89.1	Ops.Number . . . . .	537	90.1.13	SideBarImage . . . . .	548	89.1.1	DelayedNumber . . . . .	537	90.1.14	SideBarStyle . . . . .	548				90.1.15	SideBarSwitch . . . . .	548				90.1.16	SidebarText_v3 . . . . .	548				90.1.17	SidebarVariables . . . . .	549
90.1	Ops Sidebar . . . . .	544																																																																																																									
86.1.6	GreaterThan . . . . .	534	90.1.1	Button_v2 . . . . .	544	86.1.7	IfBetweenThen . . . . .	535	90.1.2	ColorPicker_v3 . . . . .	544	86.1.8	IsEven . . . . .	535	90.1.3	DisplayValue_v2 . . . . .	544	86.1.9	LessThan . . . . .	535	90.1.4	DropDown_v2 . . . . .	545	<b>87</b>	<b>Ops.Net</b>	<b>536</b>	90.1.5	Group . . . . .	545	87.1	Ops.Net . . . . .	536	90.1.6	Incrementor_v3 . . . . .	545	87.1.1	CorsProxy_v3 . . . . .	536	90.1.7	LocalFileToDataUrl . . . . .	546	<b>88</b>	<b>Ops.Net.WebSocket</b>	<b>536</b>	90.1.8	NumberInput_v2 . . . . .	546	88.1	Ops.Net.WebSocket . . . . .	536	90.1.9	Presets_v2 . . . . .	546	88.1.1	WebSocket_v2 . . . . .	536	90.1.10	Sidebar . . . . .	547	88.1.2	WebSocketSend . . . . .	536	90.1.11	SidebarDateTime . . . . .	547	<b>89</b>	<b>Ops.Number</b>	<b>537</b>	90.1.12	SidebarElement . . . . .	547	89.1	Ops.Number . . . . .	537	90.1.13	SideBarImage . . . . .	548	89.1.1	DelayedNumber . . . . .	537	90.1.14	SideBarStyle . . . . .	548				90.1.15	SideBarSwitch . . . . .	548				90.1.16	SidebarText_v3 . . . . .	548				90.1.17	SidebarVariables . . . . .	549						
90.1.1	Button_v2 . . . . .	544																																																																																																									
86.1.7	IfBetweenThen . . . . .	535	90.1.2	ColorPicker_v3 . . . . .	544	86.1.8	IsEven . . . . .	535	90.1.3	DisplayValue_v2 . . . . .	544	86.1.9	LessThan . . . . .	535	90.1.4	DropDown_v2 . . . . .	545	<b>87</b>	<b>Ops.Net</b>	<b>536</b>	90.1.5	Group . . . . .	545	87.1	Ops.Net . . . . .	536	90.1.6	Incrementor_v3 . . . . .	545	87.1.1	CorsProxy_v3 . . . . .	536	90.1.7	LocalFileToDataUrl . . . . .	546	<b>88</b>	<b>Ops.Net.WebSocket</b>	<b>536</b>	90.1.8	NumberInput_v2 . . . . .	546	88.1	Ops.Net.WebSocket . . . . .	536	90.1.9	Presets_v2 . . . . .	546	88.1.1	WebSocket_v2 . . . . .	536	90.1.10	Sidebar . . . . .	547	88.1.2	WebSocketSend . . . . .	536	90.1.11	SidebarDateTime . . . . .	547	<b>89</b>	<b>Ops.Number</b>	<b>537</b>	90.1.12	SidebarElement . . . . .	547	89.1	Ops.Number . . . . .	537	90.1.13	SideBarImage . . . . .	548	89.1.1	DelayedNumber . . . . .	537	90.1.14	SideBarStyle . . . . .	548				90.1.15	SideBarSwitch . . . . .	548				90.1.16	SidebarText_v3 . . . . .	548				90.1.17	SidebarVariables . . . . .	549												
90.1.2	ColorPicker_v3 . . . . .	544																																																																																																									
86.1.8	IsEven . . . . .	535	90.1.3	DisplayValue_v2 . . . . .	544	86.1.9	LessThan . . . . .	535	90.1.4	DropDown_v2 . . . . .	545	<b>87</b>	<b>Ops.Net</b>	<b>536</b>	90.1.5	Group . . . . .	545	87.1	Ops.Net . . . . .	536	90.1.6	Incrementor_v3 . . . . .	545	87.1.1	CorsProxy_v3 . . . . .	536	90.1.7	LocalFileToDataUrl . . . . .	546	<b>88</b>	<b>Ops.Net.WebSocket</b>	<b>536</b>	90.1.8	NumberInput_v2 . . . . .	546	88.1	Ops.Net.WebSocket . . . . .	536	90.1.9	Presets_v2 . . . . .	546	88.1.1	WebSocket_v2 . . . . .	536	90.1.10	Sidebar . . . . .	547	88.1.2	WebSocketSend . . . . .	536	90.1.11	SidebarDateTime . . . . .	547	<b>89</b>	<b>Ops.Number</b>	<b>537</b>	90.1.12	SidebarElement . . . . .	547	89.1	Ops.Number . . . . .	537	90.1.13	SideBarImage . . . . .	548	89.1.1	DelayedNumber . . . . .	537	90.1.14	SideBarStyle . . . . .	548				90.1.15	SideBarSwitch . . . . .	548				90.1.16	SidebarText_v3 . . . . .	548				90.1.17	SidebarVariables . . . . .	549																		
90.1.3	DisplayValue_v2 . . . . .	544																																																																																																									
86.1.9	LessThan . . . . .	535	90.1.4	DropDown_v2 . . . . .	545	<b>87</b>	<b>Ops.Net</b>	<b>536</b>	90.1.5	Group . . . . .	545	87.1	Ops.Net . . . . .	536	90.1.6	Incrementor_v3 . . . . .	545	87.1.1	CorsProxy_v3 . . . . .	536	90.1.7	LocalFileToDataUrl . . . . .	546	<b>88</b>	<b>Ops.Net.WebSocket</b>	<b>536</b>	90.1.8	NumberInput_v2 . . . . .	546	88.1	Ops.Net.WebSocket . . . . .	536	90.1.9	Presets_v2 . . . . .	546	88.1.1	WebSocket_v2 . . . . .	536	90.1.10	Sidebar . . . . .	547	88.1.2	WebSocketSend . . . . .	536	90.1.11	SidebarDateTime . . . . .	547	<b>89</b>	<b>Ops.Number</b>	<b>537</b>	90.1.12	SidebarElement . . . . .	547	89.1	Ops.Number . . . . .	537	90.1.13	SideBarImage . . . . .	548	89.1.1	DelayedNumber . . . . .	537	90.1.14	SideBarStyle . . . . .	548				90.1.15	SideBarSwitch . . . . .	548				90.1.16	SidebarText_v3 . . . . .	548				90.1.17	SidebarVariables . . . . .	549																								
90.1.4	DropDown_v2 . . . . .	545																																																																																																									
<b>87</b>	<b>Ops.Net</b>	<b>536</b>	90.1.5	Group . . . . .	545	87.1	Ops.Net . . . . .	536	90.1.6	Incrementor_v3 . . . . .	545	87.1.1	CorsProxy_v3 . . . . .	536	90.1.7	LocalFileToDataUrl . . . . .	546	<b>88</b>	<b>Ops.Net.WebSocket</b>	<b>536</b>	90.1.8	NumberInput_v2 . . . . .	546	88.1	Ops.Net.WebSocket . . . . .	536	90.1.9	Presets_v2 . . . . .	546	88.1.1	WebSocket_v2 . . . . .	536	90.1.10	Sidebar . . . . .	547	88.1.2	WebSocketSend . . . . .	536	90.1.11	SidebarDateTime . . . . .	547	<b>89</b>	<b>Ops.Number</b>	<b>537</b>	90.1.12	SidebarElement . . . . .	547	89.1	Ops.Number . . . . .	537	90.1.13	SideBarImage . . . . .	548	89.1.1	DelayedNumber . . . . .	537	90.1.14	SideBarStyle . . . . .	548				90.1.15	SideBarSwitch . . . . .	548				90.1.16	SidebarText_v3 . . . . .	548				90.1.17	SidebarVariables . . . . .	549																														
90.1.5	Group . . . . .	545																																																																																																									
87.1	Ops.Net . . . . .	536	90.1.6	Incrementor_v3 . . . . .	545	87.1.1	CorsProxy_v3 . . . . .	536	90.1.7	LocalFileToDataUrl . . . . .	546	<b>88</b>	<b>Ops.Net.WebSocket</b>	<b>536</b>	90.1.8	NumberInput_v2 . . . . .	546	88.1	Ops.Net.WebSocket . . . . .	536	90.1.9	Presets_v2 . . . . .	546	88.1.1	WebSocket_v2 . . . . .	536	90.1.10	Sidebar . . . . .	547	88.1.2	WebSocketSend . . . . .	536	90.1.11	SidebarDateTime . . . . .	547	<b>89</b>	<b>Ops.Number</b>	<b>537</b>	90.1.12	SidebarElement . . . . .	547	89.1	Ops.Number . . . . .	537	90.1.13	SideBarImage . . . . .	548	89.1.1	DelayedNumber . . . . .	537	90.1.14	SideBarStyle . . . . .	548				90.1.15	SideBarSwitch . . . . .	548				90.1.16	SidebarText_v3 . . . . .	548				90.1.17	SidebarVariables . . . . .	549																																				
90.1.6	Incrementor_v3 . . . . .	545																																																																																																									
87.1.1	CorsProxy_v3 . . . . .	536	90.1.7	LocalFileToDataUrl . . . . .	546	<b>88</b>	<b>Ops.Net.WebSocket</b>	<b>536</b>	90.1.8	NumberInput_v2 . . . . .	546	88.1	Ops.Net.WebSocket . . . . .	536	90.1.9	Presets_v2 . . . . .	546	88.1.1	WebSocket_v2 . . . . .	536	90.1.10	Sidebar . . . . .	547	88.1.2	WebSocketSend . . . . .	536	90.1.11	SidebarDateTime . . . . .	547	<b>89</b>	<b>Ops.Number</b>	<b>537</b>	90.1.12	SidebarElement . . . . .	547	89.1	Ops.Number . . . . .	537	90.1.13	SideBarImage . . . . .	548	89.1.1	DelayedNumber . . . . .	537	90.1.14	SideBarStyle . . . . .	548				90.1.15	SideBarSwitch . . . . .	548				90.1.16	SidebarText_v3 . . . . .	548				90.1.17	SidebarVariables . . . . .	549																																										
90.1.7	LocalFileToDataUrl . . . . .	546																																																																																																									
<b>88</b>	<b>Ops.Net.WebSocket</b>	<b>536</b>	90.1.8	NumberInput_v2 . . . . .	546	88.1	Ops.Net.WebSocket . . . . .	536	90.1.9	Presets_v2 . . . . .	546	88.1.1	WebSocket_v2 . . . . .	536	90.1.10	Sidebar . . . . .	547	88.1.2	WebSocketSend . . . . .	536	90.1.11	SidebarDateTime . . . . .	547	<b>89</b>	<b>Ops.Number</b>	<b>537</b>	90.1.12	SidebarElement . . . . .	547	89.1	Ops.Number . . . . .	537	90.1.13	SideBarImage . . . . .	548	89.1.1	DelayedNumber . . . . .	537	90.1.14	SideBarStyle . . . . .	548				90.1.15	SideBarSwitch . . . . .	548				90.1.16	SidebarText_v3 . . . . .	548				90.1.17	SidebarVariables . . . . .	549																																																
90.1.8	NumberInput_v2 . . . . .	546																																																																																																									
88.1	Ops.Net.WebSocket . . . . .	536	90.1.9	Presets_v2 . . . . .	546	88.1.1	WebSocket_v2 . . . . .	536	90.1.10	Sidebar . . . . .	547	88.1.2	WebSocketSend . . . . .	536	90.1.11	SidebarDateTime . . . . .	547	<b>89</b>	<b>Ops.Number</b>	<b>537</b>	90.1.12	SidebarElement . . . . .	547	89.1	Ops.Number . . . . .	537	90.1.13	SideBarImage . . . . .	548	89.1.1	DelayedNumber . . . . .	537	90.1.14	SideBarStyle . . . . .	548				90.1.15	SideBarSwitch . . . . .	548				90.1.16	SidebarText_v3 . . . . .	548				90.1.17	SidebarVariables . . . . .	549																																																						
90.1.9	Presets_v2 . . . . .	546																																																																																																									
88.1.1	WebSocket_v2 . . . . .	536	90.1.10	Sidebar . . . . .	547	88.1.2	WebSocketSend . . . . .	536	90.1.11	SidebarDateTime . . . . .	547	<b>89</b>	<b>Ops.Number</b>	<b>537</b>	90.1.12	SidebarElement . . . . .	547	89.1	Ops.Number . . . . .	537	90.1.13	SideBarImage . . . . .	548	89.1.1	DelayedNumber . . . . .	537	90.1.14	SideBarStyle . . . . .	548				90.1.15	SideBarSwitch . . . . .	548				90.1.16	SidebarText_v3 . . . . .	548				90.1.17	SidebarVariables . . . . .	549																																																												
90.1.10	Sidebar . . . . .	547																																																																																																									
88.1.2	WebSocketSend . . . . .	536	90.1.11	SidebarDateTime . . . . .	547	<b>89</b>	<b>Ops.Number</b>	<b>537</b>	90.1.12	SidebarElement . . . . .	547	89.1	Ops.Number . . . . .	537	90.1.13	SideBarImage . . . . .	548	89.1.1	DelayedNumber . . . . .	537	90.1.14	SideBarStyle . . . . .	548				90.1.15	SideBarSwitch . . . . .	548				90.1.16	SidebarText_v3 . . . . .	548				90.1.17	SidebarVariables . . . . .	549																																																																		
90.1.11	SidebarDateTime . . . . .	547																																																																																																									
<b>89</b>	<b>Ops.Number</b>	<b>537</b>	90.1.12	SidebarElement . . . . .	547	89.1	Ops.Number . . . . .	537	90.1.13	SideBarImage . . . . .	548	89.1.1	DelayedNumber . . . . .	537	90.1.14	SideBarStyle . . . . .	548				90.1.15	SideBarSwitch . . . . .	548				90.1.16	SidebarText_v3 . . . . .	548				90.1.17	SidebarVariables . . . . .	549																																																																								
90.1.12	SidebarElement . . . . .	547																																																																																																									
89.1	Ops.Number . . . . .	537	90.1.13	SideBarImage . . . . .	548	89.1.1	DelayedNumber . . . . .	537	90.1.14	SideBarStyle . . . . .	548				90.1.15	SideBarSwitch . . . . .	548				90.1.16	SidebarText_v3 . . . . .	548				90.1.17	SidebarVariables . . . . .	549																																																																														
90.1.13	SideBarImage . . . . .	548																																																																																																									
89.1.1	DelayedNumber . . . . .	537	90.1.14	SideBarStyle . . . . .	548				90.1.15	SideBarSwitch . . . . .	548				90.1.16	SidebarText_v3 . . . . .	548				90.1.17	SidebarVariables . . . . .	549																																																																																				
90.1.14	SideBarStyle . . . . .	548																																																																																																									
			90.1.15	SideBarSwitch . . . . .	548				90.1.16	SidebarText_v3 . . . . .	548				90.1.17	SidebarVariables . . . . .	549																																																																																										
90.1.15	SideBarSwitch . . . . .	548																																																																																																									
			90.1.16	SidebarText_v3 . . . . .	548				90.1.17	SidebarVariables . . . . .	549																																																																																																
90.1.16	SidebarText_v3 . . . . .	548																																																																																																									
			90.1.17	SidebarVariables . . . . .	549																																																																																																						
90.1.17	SidebarVariables . . . . .	549																																																																																																									

90.1.18	Slider_v3 . . . . .	549
90.1.19	TextInput_v2 . . . . .	549
90.1.20	Toggle_v4 . . . . .	550
90.1.21	XYPad . . . . .	550

## 91 Ops.String 551

91.1	Ops.String . . . . .	551
91.1.1	AddLineBreaks_v2 . . . . .	551
91.1.2	ArrayContainsString . . . . .	551
91.1.3	ArrayOfStrings . . . . .	551
91.1.4	CharacterRotate . . . . .	551
91.1.5	Concat_v2 . . . . .	552
91.1.6	ConcatMulti_v2 . . . . .	552
91.1.7	ConcatMultiPort_v2 . . . . .	552
91.1.8	CopyToClipboard . . . . .	552
91.1.9	DelayStringSimple . . . . .	553
91.1.10	EndsWith . . . . .	553
91.1.11	FileUrlsToArrayMultiPort_v2 . . . . .	553
91.1.12	FilterValidString . . . . .	553
91.1.13	FreezeString . . . . .	554
91.1.14	GateString . . . . .	554
91.1.15	HandleBarsHtml_v2 . . . . .	554
91.1.16	HtmlDecode . . . . .	554
91.1.17	HtmlEncode . . . . .	555
91.1.18	LeftPad_v2 . . . . .	555
91.1.19	LimitLineBreaks_v2 . . . . .	555
91.1.20	LineBreak . . . . .	555
91.1.21	LineBreaksHtml . . . . .	556
91.1.22	LoremIpsum . . . . .	556
91.1.23	Lowercase_v2 . . . . .	556
91.1.24	Md5 . . . . .	556
91.1.25	NumberFormatter . . . . .	556
91.1.26	NumberSwitchByString . . . . .	557
91.1.27	NumberToString_v2 . . . . .	557
91.1.28	NumTotalLineBreaks . . . . .	558
91.1.29	OrString . . . . .	558
91.1.30	ParseInt_v2 . . . . .	558
91.1.31	RandomString_v3 . . . . .	558
91.1.32	RightPad_v2 . . . . .	559

91.1.33	RightPadNumber_v2 . . . . .	559
91.1.34	RouteString . . . . .	559
91.1.35	SaveTextFile . . . . .	559
91.1.36	SequenceStrings . . . . .	560
91.1.37	StartsWith . . . . .	560
91.1.38	String_v3 . . . . .	560
91.1.39	StringCompose_v3 . . . . .	561
91.1.40	StringContains_v2 . . . . .	561
91.1.41	StringEditor . . . . .	561
91.1.42	StringEquals_v2 . . . . .	561
91.1.43	StringGetLineNumAtIndex . . . . .	562
91.1.44	StringIterator_v2 . . . . .	562
91.1.45	StringLength_v2 . . . . .	562
91.1.46	StringRemoveCharacters . . . . .	562
91.1.47	StringReplace . . . . .	563
91.1.48	StringSortLines . . . . .	563
91.1.49	StringsToArrayMultiPort_v2 . . . . .	563
91.1.50	StringSwitchByString . . . . .	563
91.1.51	StringToNumber . . . . .	564
91.1.52	StringTrim_v2 . . . . .	564
91.1.53	StripHtml . . . . .	564
91.1.54	SubString_v2 . . . . .	564
91.1.55	SwitchString . . . . .	565
91.1.56	SwitchStringMultiPort_v2 . . . . .	565
91.1.57	Uppercase_v2 . . . . .	565
91.1.58	UUID . . . . .	566

## 92 Ops.String.Base64 566

92.1	Ops.String.Base64 . . . . .	566
92.1.1	Base64Decode_v2 . . . . .	566
92.1.2	Base64Encode_v3 . . . . .	566
92.1.3	DownloadBase64File . . . . .	567

## 93 Ops.String.File 567

93.1	Ops.String.File . . . . .	567
93.1.1	FileInput_v2 . . . . .	567
93.1.2	SwitchFile_v2 . . . . .	567

## 94 Ops.Templates 568

94.1	<b>Ops.Templates</b>	568	96.1.1	TimeLineBPM	576
94.1.1	ExampleVizOp	568	96.1.2	TimeLineImage	576
94.1.2	MinimalMaterial	568			
94.1.3	PortsArrayExample	569	<b>97</b>	<b>Ops.Trigger</b>	<b>577</b>
94.1.4	PortsBooleanExample	569	97.1	Ops.Trigger	577
94.1.5	PortsObjectExample	569	97.1.1	DelayedTrigger	577
94.1.6	PortsStringExample	569	97.1.2	GateTrigger	577
94.1.7	PortsTriggerExample	569	97.1.3	Interval	578
94.1.8	PortsValueExample	570	97.1.4	IsTriggered	578
94.1.9	ShaderEffectExample	570	97.1.5	NthTrigger_v2	578
94.1.10	UiTestOp	570	97.1.6	NumberByTrigger	578
<b>95</b>	<b>Ops.TimeLine</b>	<b>571</b>	97.1.7	NumberByTriggerMultiPort_v2	579
95.1	Ops.TimeLine	571	97.1.8	ProbabilityTrigger	579
95.1.1	Anim	571	97.1.9	RandomTrigger	579
95.1.2	AnimGetKey	571	97.1.10	Repeat2d	579
95.1.3	AnimGetValue	571	97.1.11	Repeat_v2	580
95.1.4	AnimInfo	571	97.1.12	RouteTrigger	580
95.1.5	AutoPlay	572	97.1.13	RouteTriggerAnimated	581
95.1.6	DemoPrerender	572	97.1.14	RouteTriggerMultiPort_v2	581
95.1.7	GotoFrame	572	97.1.15	RouteTriggerString_v2	581
95.1.8	PreRender	572	97.1.16	Sequence	582
95.1.9	TimelineConfig	573	97.1.17	SequenceMultiPort_v2	583
95.1.10	TimeLineControls	573	97.1.18	SwitchTrigger	583
95.1.11	TimelineDebug	573	97.1.19	Threshold	584
95.1.12	TimeLineFrame	573	97.1.20	TimedSequence	584
95.1.13	TimeLineLength	574	97.1.21	TimeSinceTrigger	584
95.1.14	TimeLineLoop	574	97.1.22	TriggerButton	585
95.1.15	TimeLineOverwrite	574	97.1.23	TriggerCounter	585
95.1.16	TimeLinePlay	574	97.1.24	TriggerCounterLoop	585
95.1.17	TimeLinePlayer	575	97.1.25	TriggerDistributeByValue	585
95.1.18	TimeLineRewind	575	97.1.26	TriggerExtender	586
95.1.19	TimeLineSetTime	575	97.1.27	TriggerIfDecreased	586
95.1.20	TimeLineTime	575	97.1.28	TriggerIfIncreased	586
95.1.21	TimeLineTogglePlay	576	97.1.29	TriggerLimiter	587
95.1.22	TimelineValue	576	97.1.30	TriggerNumber	587
<b>96</b>	<b>Ops.TimeLine.Viz</b>	<b>576</b>	97.1.31	TriggerOnce	587
96.1	Ops.TimeLine.Viz	576	97.1.32	TriggerOnChangeArray_v2	587
			97.1.33	TriggerOnChangeObject_v2	588
			97.1.34	TriggerOnChangeString_v2	588

97.1.35 TriggerOnChangeTexture . . . . .	588	99.1 Ops.Ui.Debug . . . . .	597
97.1.36 TriggerReceive . . . . .	588	99.1.1 PatchUiCoords . . . . .	597
97.1.37 TriggerReceiveFilter . . . . .	589	<b>100 Ops.Ui.Routing</b>	<b>597</b>
97.1.38 TriggerSend . . . . .	589	100.1 Ops.Ui.Routing . . . . .	597
97.1.39 TriggerSendNamed . . . . .	589	100.1.1 RouteArray . . . . .	597
97.1.40 TriggersPerSecond . . . . .	589	100.1.2 RouteNumber . . . . .	597
97.1.41 TriggerString . . . . .	590	100.1.3 RouteObject . . . . .	598
97.1.42 ValueBecameZeroTrigger . . . . .	590	100.1.4 RouteString . . . . .	598
<b>98 Ops.Ui</b>	<b>590</b>	100.1.5 RouteTrigger . . . . .	598
98.1 Ops.Ui . . . . .	590	<b>101 Ops.Vars</b>	<b>599</b>
98.1.1 Area . . . . .	590	101.1 Ops.Vars . . . . .	599
98.1.2 CablesEditorEvents . . . . .	590	101.1.1 TriggerVarGetArray . . . . .	599
98.1.3 Comment_v2 . . . . .	591	101.1.2 TriggerVarGetNumber . . . . .	599
98.1.4 GetCablesDefaultTheme . . . . .	591	101.1.3 TriggerVarGetObject . . . . .	599
98.1.5 MaximizeRenderer . . . . .	591	101.1.4 TriggerVarGetString . . . . .	599
98.1.6 PatchInput . . . . .	591	101.1.5 VarGetArray_v2 . . . . .	600
98.1.7 PatchOutput . . . . .	591	101.1.6 VarGetNumber_v2 . . . . .	600
98.1.8 SetCablesTheme . . . . .	592	101.1.7 VarGetObject_v2 . . . . .	600
98.1.9 SubPatch . . . . .	592	101.1.8 VarGetString . . . . .	600
98.1.10 Subpatch2Template . . . . .	592	101.1.9 VarGetTexture_v2 . . . . .	600
98.1.11 SubPatchInput . . . . .	592	101.1.10 VariablesAsObject . . . . .	601
98.1.12 SubPatchOutput . . . . .	592	101.1.11 VarSetArray_v2 . . . . .	601
98.1.13 VizArrayChart . . . . .	593	101.1.12 VarSetNumber_v2 . . . . .	601
98.1.14 VizArrayGraph . . . . .	593	101.1.13 VarSetObject_v2 . . . . .	601
98.1.15 VizArrayTable_v2 . . . . .	593	101.1.14 VarSetString_v2 . . . . .	601
98.1.16 VizBool . . . . .	593	101.1.15 VarSetTextTexture_v2 . . . . .	602
98.1.17 VizGraph . . . . .	594	101.1.16 VarTriggerArray . . . . .	602
98.1.18 VizImageUrl . . . . .	594	101.1.17 VarTriggerNumber . . . . .	602
98.1.19 VizLogger . . . . .	594	101.1.18 VarTriggerObject . . . . .	602
98.1.20 VizNumber . . . . .	594	101.1.19 VarTriggerString . . . . .	603
98.1.21 VizNumberBar . . . . .	595	101.1.20 VarTriggerTexture . . . . .	603
98.1.22 VizObject . . . . .	595	<b>102 Ops.WebAudio</b>	<b>603</b>
98.1.23 VizString . . . . .	595	102.1 Ops.WebAudio . . . . .	603
98.1.24 VizTexture . . . . .	596	102.1.1 AnalyzerTexture_v2 . . . . .	603
98.1.25 VizTextureTable . . . . .	596	102.1.2 AudioAnalyzer_v2 . . . . .	603
98.1.26 VizTrigger . . . . .	596	102.1.3 AudioBuffer_v3 . . . . .	604
<b>99 Ops.Ui.Debug</b>	<b>597</b>		

102.1.4	AudioBufferChannelRouter . . . . .	604
102.1.5	AudioBufferPlayer_v2 . . . . .	604
102.1.6	AudioBufferToSplineArray . . . . .	605
102.1.7	AudioPanner . . . . .	605
102.1.8	AudioRecorder . . . . .	605
102.1.9	BiquadFilter_v2 . . . . .	606
102.1.10	ClockSequencer . . . . .	606
102.1.11	ClockSequencerPattern . . . . .	606
102.1.12	Convolver_v2 . . . . .	607
102.1.13	CutFilter . . . . .	607
102.1.14	Delay . . . . .	607
102.1.15	FFTAreaAverage_v3 . . . . .	607
102.1.16	Gain . . . . .	608
102.1.17	KeyPiano . . . . .	608
102.1.18	MicrophoneIn_v2 . . . . .	609
102.1.19	MidiValueToFrequency . . . . .	609
102.1.20	Mixer . . . . .	609
102.1.21	MusicalScales . . . . .	610
102.1.22	Output_v2 . . . . .	610
102.1.23	ThreeBandEqualizer . . . . .	610
102.1.24	WaveformMesh . . . . .	611
102.1.25	Waveshaper . . . . .	611

<b>103</b>	<b>Ops.Website</b>	<b>612</b>
103.1	Ops.Website . . . . .	612
103.1.1	Cookie . . . . .	612
103.1.2	FilenameInfo . . . . .	612
103.1.3	ForceHttps . . . . .	612
103.1.4	InfoURL . . . . .	612
103.1.5	InIframe . . . . .	613
103.1.6	LocalStorageNumber . . . . .	613
103.1.7	LocalStorageString . . . . .	613
103.1.8	LocationHashRoute . . . . .	613
103.1.9	SetLocationHash . . . . .	614
103.1.10	UrlQueryParams_v2 . . . . .	614

# 1 Introduction to Cables.gl

## 1.1 What is Cables.gl?



**Cables.gl** is a powerful, browser-based visual programming environment for creating interactive 2D and 3D graphics using WebGL. It was created by **undev** in Berlin and has become a popular tool for creative coding, interactive installations, data visualization, and web-based visual experiences.

Unlike traditional coding environments, cables.gl uses a **node-based** (or “patch-based”) approach where you connect visual operators (ops) together to create your projects. This makes it accessible to artists and designers while still being powerful enough for developers.

## 1.2 A (Brief) History of cables.gl

cables.gl was created by **undev** (Berlin) with the goal of making **real-time WebGL** creation approachable through a node-based workflow—similar in spirit to visual programming environments used in motion design and interactive installations, but built for the browser.

Over time, cables.gl grew from a tool for quick experiments into a full ecosystem:

- **Early days:** a strong focus on rapid prototyping and sharing patches online.

- **Maturing platform:** a steadily growing op library for 2D, 3D, textures, audio, and interaction, plus better tooling (timeline, profiling/debugging utilities, export options).
- **Community-driven growth:** more public patches, tutorials, Discord knowledge-sharing, and reusable patterns (e.g., render-to-texture workflows, post-processing chains, audio-reactive setups).
- **Production use:** cables.gl exports make it viable for deployment in websites, installations, and client work—where performance, asset management, and reliable runtime behavior matter.

If you're coming from traditional code, it helps to think of cables.gl as a **visual runtime graph**: triggers define *when* things run; value connections define *what data flows*; and the patch as a whole becomes a web-ready app.

## 1.3 Why Use Cables.gl?

### 1.3.1 Visual Programming

- No coding required to get started
- Drag-and-drop interface
- See results in real-time as you build

### 1.3.2 Browser-Based

- No installation needed
- Works on any modern browser
- Collaborate and share easily

### 1.3.3 High Performance

- Built on WebGL for GPU-accelerated graphics
- Optimized for real-time rendering
- Handles complex 3D scenes smoothly

### 1.3.4 Export Options

- Standalone HTML/JS builds
- Embed in websites
- Create offline applications

## 1.3.5 Extensible

- Write custom operators (ops) in JavaScript
- GLSL shader support
- Import external libraries

## 1.4 Key Concepts

### 1.4.1 Operators (Ops)

The building blocks of cables.gl. Each op performs a specific function - from drawing shapes to processing audio to handling user input.

### 1.4.2 Patches

A patch is your complete project - a collection of ops connected together to create your visual experience.

### 1.4.3 Ports

Ops have input and output ports. You connect ports together with "cables" (hence the name!) to pass data between ops.

### 1.4.4 Types of Ports

- **Trigger** (grey) - Execution flow, like "when to do something"
- **Number** (green) - Numerical values
- **String** (yellow) - Text values
- **Object** (blue) - Complex data like meshes, textures, arrays
- **Array** (cyan) - Collections of data

## 1.5 Featured Videos

### 1.5.1 Overview and Getting Started



<https://youtu.be/hVxrxXhH7vQ>

**Cables.gl Standalone (Offline) Build: Create Without Limits!**  
by Decode GL



<https://youtu.be/goO3PhuenBI>

**First Steps in Cables.gl - Tutorial**  
by The Interactive & Immersive HQ



<https://youtu.be/xnObNRv8n9I>

**Introduction to cables.gl - Data-Driven Gradient from Geo-Located Weather - Part 0**  
by Kirell Benzi

## 1.5.2 More Resources

**Note:** There are limited intro-specific YouTube videos for cables.gl, but the platform has excellent resources: - Browse the cables.gl Public Patches to see

examples - Check the official cables.gl YouTube channel for official tutorials - The Decode GL channel has multiple cables.gl tutorials - Search for "cables.gl" on YouTube for the latest community content - Many cables.gl creators share their work on social media and personal channels

## 1.6 Getting Help

- **Official Documentation:** [cables.gl/docs](https://cables.gl/docs)
- **Example Patches:** Browse public patches for inspiration

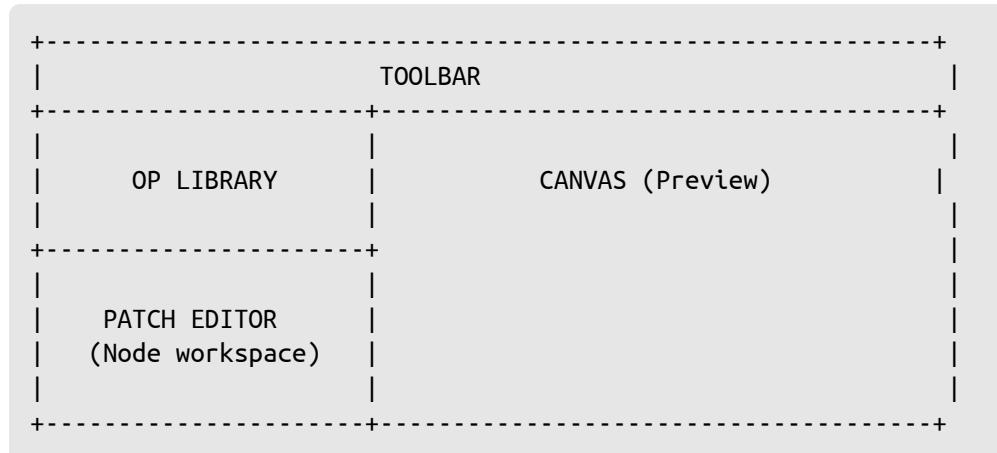
# 2 Getting Started with Cables.gl

## 2.1 Creating Your Account

1. Go to cables.gl
2. Click "Sign Up" to create a free account
3. Verify your email
4. You're ready to start creating!

## 2.2 The Interface

### 2.2.1 Main Areas



### 2.2.2 Key Interface Elements

1. **Canvas** - Live preview of your creation
2. **Patch Editor** - Where you place and connect ops
3. **Op Library** - Browse and search for operators
4. **Parameters Panel** - Adjust settings for selected ops
5. **Timeline** - For animation keyframes

## 2.3 Navigation Controls

- **Pan the view:** Middle mouse drag or Space + drag
- **Zoom in/out:** Mouse scroll wheel

- **Select op:** Left click
- **Multi-select:** Shift + click or drag box
- **Delete selected:** Delete or Backspace
- **Add new op:** Double-click or Tab
- **Connect ports:** Drag from output to input

## 2.4 Keyboard Shortcuts

Mastering keyboard shortcuts will significantly speed up your workflow in cables.gl.

### 2.4.1 Essential Shortcuts

- **Tab or Double-click:** Add new op (opens search)
- **Delete or Backspace:** Delete selected op(s)
- **Ctrl + C / Cmd + C:** Copy selected op(s)
- **Ctrl + V / Cmd + V:** Paste selected op(s)
- **Ctrl + X / Cmd + X:** Cut selected op(s)
- **Ctrl + D / Cmd + D:** Duplicate selected op(s)
- **Ctrl + Z / Cmd + Z:** Undo
- **Ctrl + Shift + Z / Cmd + Shift + Z:** Redo

### 2.4.2 Selection & Navigation

- **Ctrl + A / Cmd + A:** Select all ops
- **Shift + Click:** Add to selection
- **Ctrl + Click / Cmd + Click:** Toggle selection
- **Escape:** Deselect all
- **Space + Drag:** Pan the patch view
- **Mouse Wheel:** Zoom in/out
- **Ctrl + 0 / Cmd + 0:** Zoom to fit all ops
- **F:** Focus/frame selected op(s)

### 2.4.3 Organizing & Aligning

- **Ctrl + Shift + A / Cmd + Shift + A:** Align selected ops horizontally
- **Ctrl + Shift + D / Cmd + Shift + D:** Distribute selected ops evenly
- **Ctrl + G / Cmd + G:** Group selected ops

- **Arrow Keys:** Nudge selected op(s) by small amount
- **Shift + Arrow Keys:** Nudge selected op(s) by larger amount

## 2.4.4 Working with Ops

- **Enter:** Open/edit selected op's parameters
- **Ctrl + E / Cmd + E:** Enable/disable selected op
- **Ctrl + M / Cmd + M:** Mute selected op
- **R:** Rename selected op
- **C:** Add comment node
- **Ctrl + F / Cmd + F:** Find/search ops in patch

## 2.4.5 Cables & Connections

- **Drag from port:** Create connection
- **Click connection:** Select cable
- **Alt + Click connection:** Delete cable
- **Shift + Drag port:** Create cable with search

## 2.4.6 View & Interface

- **T:** Toggle timeline
- **Ctrl + / / Cmd + /:** Toggle op library
- **H:** Toggle patch editor visibility
- **P:** Toggle parameters panel
- **Ctrl + S / Cmd + S:** Save patch
- **Ctrl + Shift + S / Cmd + Shift + S:** Save as...

## 2.4.7 Performance & Debugging

- **Ctrl + Shift + P / Cmd + Shift + P:** Performance monitor
- **Ctrl + Shift + L / Cmd + Shift + L:** Show patch loading info
- **Alt + Click op:** View op documentation

## 2.4.8 Pro Tips

- **Hold Shift while connecting:** Automatically opens op search to insert an op in the connection

- **Hold Alt while dragging:** Duplicate op while moving
- **Double-click a connection:** Insert a new op in that cable
- **Right-click an op:** Quick access to op menu (rename, mute, group, etc.)
- **Click and drag in empty space:** Selection box for multiple ops

## 2.5 Your First Patch

Let's create a simple animated shape!

### 2.5.1 Step 1: Create the Render Pipeline

1. Double-click in the patch editor to open the op search
2. Search for MainLoop and add it
3. The MainLoop is the heartbeat of your patch - it triggers every frame

### 2.5.2 Step 2: Add a BasicMaterial

1. Add a BasicMaterial op
2. Connect MainLoop's trigger output to BasicMaterial's trigger input
3. You should see a black canvas appear

### 2.5.3 Step 3: Draw a Circle

1. Add a Circle op
2. Connect BasicMaterial -> Circle
3. A white circle appears!

Here's what your patch should look like:

#### Visualization Options

##### Option 1: Screenshot from Real Cables.gl (Most Authentic)

See Screenshot Guide for instructions on capturing real cables.gl patches.

##### Option 3: HTML/CSS Interactive

Open HTML Version in your browser for an interactive view.

##### Option 4: Mermaid Diagram (Simple Flow)

```

graph TD
    MainLoop[MainLoop] -->|trigger| BasicMaterial[BasicMaterial]
    BasicMaterial -->|trigger| Circle[Circle]

    style MainLoop fill:#2d2d2d,stroke:#4a4a4a,color:#e0e0e0
    style BasicMaterial fill:#2d2d2d,stroke:#4a4a4a,color:#e0e0e0
    style Circle fill:#2d2d2d,stroke:#4a4a4a,color:#e0e0e0

```

*The basic render chain: MainLoop triggers the BasicMaterial, which then draws the Circle*

## 2.5.4 Step 4: Add Color

1. Select the BasicMaterial op
2. Adjust the color values (r, g, b) in the parameters panel
3. Or connect a SetColor op's output to BasicMaterial's color input ports
4. The circle will display with your chosen color

## 2.5.5 Step 5: Animate It

1. Add a Time op (outputs current time)
2. Add a Math op (for calculations)
3. Add a Sin op (sine wave)
4. Connect: Time -> Sin -> Circle's Scale input
5. Watch your circle pulse!

## 2.6 Understanding the Flow

Data flows from **top to bottom** and **left to right**:

```

MainLoop (starts the frame)
|
BasicMaterial (defines appearance and color)
|
Circle (draws the shape)

```

The **trigger** connection (grey) determines WHEN things happen. The **value** connections (colored) determine WHAT values are used.

## 2.7 Saving Your Work

- Patches auto-save regularly
- Click the save icon to force a save
- Use “Save As” to create copies
- Export for standalone deployment

## 2.8 Tips for Beginners

1. **Start Simple** - Begin with basic shapes before complex 3D
2. **Explore Examples** - Study public patches to learn patterns
3. **Use Comments** - Add comment ops to document your work
4. **Name Your Ops** - Rename ops for clarity in complex patches
5. **Save Often** - And use versioning for major changes

## 2.9 Featured Videos

## 2.10 Common First-Patch Issues

### 2.10.1 “I don’t see anything!”

- Make sure MainLoop is connected to BasicMaterial
- Check that your shape ops are connected in the chain
- Verify the canvas is visible (not minimized)

### 2.10.2 “Colors aren’t changing!”

- Check BasicMaterial’s color values (r, g, b) in the parameters panel
- Make sure RGB values aren’t all 0 (black)
- If using SetColor, connect it to BasicMaterial’s color input ports

### 2.10.3 “Animation isn’t working!”

- Ensure Time op is connected
- Check that the animated value is actually changing (view the port value)

---

## 3 2D Graphics in Cables.gl

### 3.1 Introduction to 2D Drawing

Cables.gl excels at creating stunning 2D graphics and animations, from simple shapes to complex generative art. This comprehensive chapter covers fundamental 2D drawing operations, advanced transformations, interactive elements, feedback loops, post-processing effects, and professional techniques for creating production-ready 2D visuals.

Whether you're creating data visualizations, interactive installations, or generative art, this chapter will give you the tools and knowledge to master 2D graphics in cables.gl.

### 3.2 Basic Shapes

#### 3.2.1 Circle

The Circle op is one of the most common 2D primitives.

**Key Parameters:** - Radius - Size of the circle - Segments - Smoothness (more segments = smoother circle) - Inner Radius - Creates a ring when > 0

#### 3.2.2 Rectangle

The Rectangle op draws rectangular shapes.

**Key Parameters:** - Width - Horizontal size - Height - Vertical size - Pivot - Origin point for positioning

#### 3.2.3 RoundedRectangle

A rectangle with smooth corners.

**Key Parameters:** - Width / Height - Dimensions - Corner Radius - How rounded the corners are

#### 3.2.4 Polygon

Create regular polygons (triangles, pentagons, etc.)

**Key Parameters:** - Sides - Number of sides (3 = triangle, 5 = pentagon, etc.) - Radius - Size of the polygon

## 3.2.5 Line / Lines

Draw single or multiple lines.

**Key Parameters:** - Start and End coordinates - Line width - Line style (solid, dashed)

## 3.3 Color and Appearance

### 3.3.1 SetColor

Changes the drawing color for subsequent shapes.

```
MainLoop -> BasicMaterial -> Circle
```

Connect SetColor output to BasicMaterial's color input ports (r, g, b, a) to set the color.

**Color Modes:** - RGB (Red, Green, Blue) - HSB (Hue, Saturation, Brightness) - Hex values

### 3.3.2 SetAlpha

Controls transparency.

```
MainLoop -> BasicMaterial -> Shape
```

Connect SetAlpha output to BasicMaterial's alpha (a) input port to control transparency.

Values range from 0 (invisible) to 1 (fully opaque).

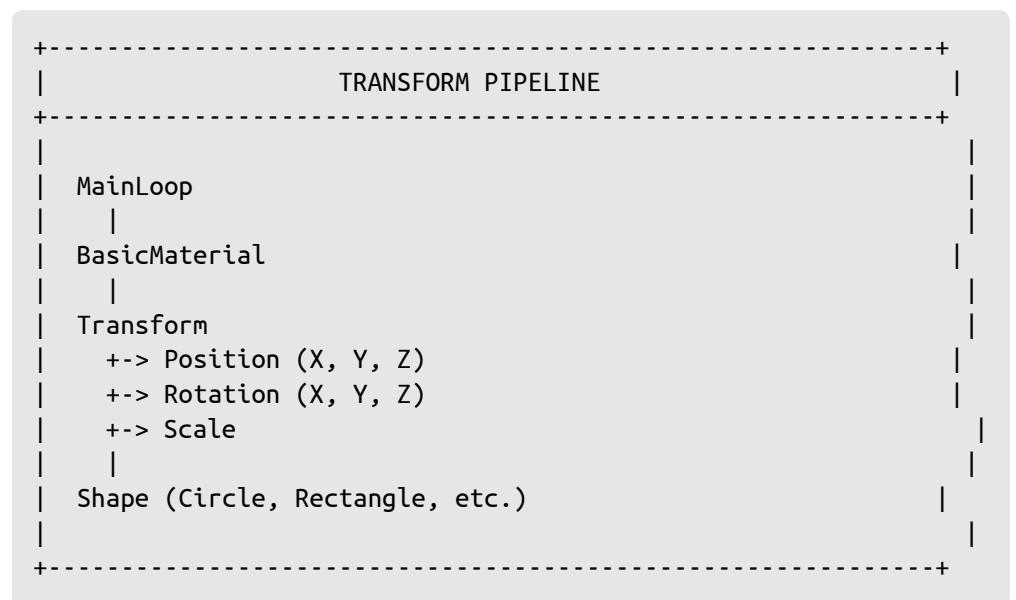
## 3.3.3 Gradients

Use texture-based gradients or shader-generated gradients for smooth color transitions.

## 3.4 Transformations

### 3.4.1 Transform

The Transform op modifies position, rotation, and scale of all following shapes.



**Parameters:** - TranslateX, TranslateY, TranslateZ - Position - RotateX, RotateY, RotateZ - Rotation (degrees) - Scale - Uniform scaling

### 3.4.2 Transformation Order Matters!

Transformations are applied in order. These produce different results:

**Rotate then Translate:**

```
+-----+
| ROTATE THEN TRANSLATE |
+-----+
|
| Original Position
|   o
|
| Step 1: Rotate 45 degrees
|   o
|   (rotate)
|
| Step 2: Translate Right
|   -> o
|
| Result: Object rotates around origin, then moves
+-----+
```

Transform (rotate) -> Transform (translate) -> Shape

### Translate then Rotate:

```
+-----+
| TRANSLATE THEN ROTATE |
+-----+
|
| Original Position
|   o
|
| Step 1: Translate Right
|   -> o
|
| Step 2: Rotate 45 degrees (around new position)
|   (rotate) o
+-----+
```

```
| Result: Object moves first, then rotates around new origin |
|                                                               |
+-----+
```

Transform (translate) -> Transform (rotate) -> Shape

### 3.4.3 Nested Transforms

Create hierarchies by chaining transforms:

```
Transform (parent)
  |
  Transform (child)
  |
  Shape
```

The child inherits and adds to the parent's transformations.

## 3.5 Blending Modes

### 3.5.1 SetBlending

Controls how colors combine when shapes overlap.

**Common Modes:** - Normal - Standard opacity blending - Add - Colors add together (great for glow effects) - Multiply - Colors multiply (darkening effect)

### 3.5.2 Depth Testing

For 2D, you often want to disable depth testing:

MainLoop -> BasicMaterial -> DepthTest (disabled) -> Your 2D Content

This ensures draw order matches your connection order.

## 3.6 Patterns and Repetition

### 3.6.1 IteratorLoop

Create patterns by repeating shapes:

```
MainLoop -> IteratorLoop -> [Your Shape Setup]
```

Use the iterator index to offset position, color, or other properties.

### 3.6.2 ArrayIterator

Iterate over data arrays to position multiple shapes.

## 3.7 Text Rendering

### 3.7.1 DrawText

Display text in your patches.

**Key Parameters:** - Text - The string to display - Font - Font family - Size - Text size - Alignment - Left, center, right

### 3.7.2 TextTexture

Create textures from text for more advanced effects.

## 3.8 Advanced Transformation Techniques

### 3.8.1 Matrix Transformations

For precise control, work directly with transformation matrices:

```
MatrixMultiply -> Combine multiple transformations  
MatrixInvert -> Reverse a transformation
```

### 3.8.2 Pivot Points

Control the center of rotation and scaling:

```
Transform (set pivot) -> Transform (rotate) -> Shape
```

**Common Pivot Values:** -0, 0 - Bottom left corner -0.5, 0.5 - Center (default)  
-1, 1 - Top right corner

### 3.8.3 Compound Transformations

Build complex motion by layering transforms:

#### Example: Orbital Motion

```
Transform (parent orbit)
  |
  Transform (child rotation)
  |
  Transform (child offset)
  |
  Shape
```

This creates a shape that orbits while rotating on its own axis.

## 3.9 Interactive 2D Elements

### 3.9.1 InteractiveRectangle

Create draggable, clickable UI elements:

```
InteractiveRectangle  
| (outputs X, Y, Width, Height on interaction)  
Control other ops with mouse input
```

**Use Cases:** - On-screen sliders - Draggable controllers - Interactive buttons - Touch-enabled interfaces

### 3.9.2 Mouse Input

Capture and use mouse position:

```
Mouse -> Map (screen to world coords) -> Visual property
```

**Mouse Ops:** - MouseX / MouseY - Cursor position - MouseButton - Click detection - MouseWheel - Scroll input

### 3.9.3 Example: Interactive Color Picker

```
MainLoop  
|  
MouseX -> Map (0 to 1) -> Hue  
MouseY -> Map (0 to 1) -> Brightness  
|  
HSBtoRGB -> BasicMaterial (color input)  
|  
FullscreenRectangle
```

## 3.10 Generative Art Techniques

### 3.10.1 Feedback Loops

Create evolving, self-referential visuals by feeding output back as input:

**Basic Feedback Setup:**

```
MainLoop  
|  
RenderToTexture (previous frame)  
|  
ImageCompose (blend with new content)  
|  
Transform (slight scale/rotate)  
|  
TextureEffects (blur, fade)  
|  
Draw new shapes  
|  
Output (becomes next frame's input)
```

**Parameters to Experiment With:** - Feedback decay (fade amount) - Transformation amount (scale, rotation) - Blend mode (add, multiply, screen) - Blur intensity

**Result:** Trails, echoes, and organic growth patterns

### 3.10.2 Op Art and Moiré Patterns

Create optical illusions with overlapping patterns:

```
IteratorLoop (creates grid)  
|  
Time -> Sin -> Rotation angle  
|  
IteratorLoop (nested for lines)  
|  
Rectangle (thin line)
```

Vary parameters like: - Line spacing - Rotation speed - Line thickness - Pattern density

### 3.10.3 Procedural Pattern Generation

Use noise and math to create endless variations:

**Perlin Noise-Based Patterns:**

```
IteratorLoop  
|  
Position -> NoiseTexture sample  
|  
Noise value -> Circle size  
|  
Noise value -> Color
```

**Grid Distortion:**

```
IteratorLoop (grid)  
|  
Position + (Noise * distortion amount)  
|  
Shape
```

## 3.11 Post-Processing Effects

### 3.11.1 Image Composition

Layer multiple render passes for rich effects:

```
RenderTarget (Pass 1: Shapes)  
RenderTarget (Pass 2: Glow)  
RenderTarget (Pass 3: Noise)  
|  
ImageCompose (blend all layers)  
|  
Final Output
```

### 3.11.2 TextureEffects for 2D

Apply effects to your rendered 2D scene:

**Blur:**

```
RenderTarget -> TextureEffects (Blur) -> Output
```

**Color Grading:**

```
RenderTarget -> ColorCorrection  
| (adjust hue, saturation, brightness, contrast)  
Output
```

**Glow Effect:**

```
Original scene  
|  
RenderTarget (bright pass)  
|  
Blur (large radius)  
|  
ImageCompose (add to original)
```

### 3.11.3 Displacement Mapping

Distort shapes using textures:

```
NoiseTexture -> DisplacementMap -> Shape rendering
```

Creates wavy, distorted effects on 2D graphics.

## 3.12 Advanced Pattern Techniques

### 3.12.1 Recursive Subdivision

Create fractal-like patterns:

```
// Custom op: Recursive shape division
for (depth = 0; depth < maxDepth; depth++) {
    // Draw shape
    // Divide into smaller shapes
    // Recursively apply
}
```

### 3.12.2 Particle Systems in 2D

Simple particle engine structure:

```
ArrayLoop (particle count)
|
Particle data (position, velocity, life)
|
Physics update (gravity, friction)
|
Transform -> Circle (particle visual)
```

### 3.12.3 Grid-Based Automata

Cellular automata and Game of Life patterns:

```
ArrayIterator (grid cells)
|
Cell state + neighbor count
|
Update rules (Conway's rules, etc.)
|
```

Visual representation

## 3.13 Data Visualization

### 3.13.1 Chart Generation

Create custom charts and graphs:

**Bar Chart:**

```
ArrayIterator (data values)
|
Index -> X position
Value -> Rectangle height
|
Rectangle (bar)
```

**Line Chart:**

```
ArrayIterator (data points)
|
Connect points with Lines op
|
Add circles for data points
```

### 3.13.2 Integration with ECharts

Apache ECharts is a powerful open-source charting library that integrates seamlessly with cables.gl. This combination lets you create professional-grade data visualizations with interactive 3D effects and real-time updates.

**Why ECharts + cables.gl?**

- **Rich Chart Types:** Bar, line, pie, scatter, radar, candlestick, heatmap, treemap, sunburst, and more
- **Interactive Features:** Tooltips, zooming, panning, data selection
- **Real-Time Updates:** Stream live data into animated charts

- **3D Enhancement:** Apply cables.gl effects to chart outputs

### Setup and Integration:

1. Load the **ECharts Extension** in cables.gl using the `Ops.Extension.ECharts.ECharts` op
2. **Configure Chart Options** using JSON format (same as standard ECharts)
3. **Connect Data Sources** from other cables.gl ops (JSON fetch, WebSocket, etc.)
4. **Apply Visual Effects** using cables.gl post-processing

### Basic ECharts Patch Structure:

```

MainLoop
|
ECharts Op
  +-> Option (JSON configuration)
  +-> Width / Height
  +-> Data inputs
  |
ECharts Instance -> Use in other ops

```

### Example: Simple Bar Chart Configuration:

```
{
  "xAxis": {
    "type": "category",
    "data": ["Mon", "Tue", "Wed", "Thu", "Fri"]
  },
  "yAxis": {
    "type": "value"
  },
  "series": [
    {
      "data": [120, 200, 150, 80, 70],
      "type": "bar",
      "color": "#5470c6"
    }
  ]
}
```

### Example: Real-Time Line Chart:

```

WebSocket (data stream)
  |
ParseJSON -> Extract values
  |
Array (rolling buffer of last N values)
  |
ECharts Op (line chart config)
  |
Render to texture
  |
Apply glow effect

```

### Example: Interactive Pie Chart with Events:

```

ECharts Op (pie chart)
  |
EChartsEvent Op
  +-> Click event -> Trigger actions
  +-> Hover event -> Show details
  |
Update other visuals based on selection

```

### Combining Charts with 3D:

```

ECharts Op -> Render to texture
  |
Plane3D (apply texture)
  |
Transform (rotate in 3D space)
  |
Post-processing (glow, bloom)

```

### Advanced Techniques:

- **Multi-Chart Dashboards:** Use multiple ECharts ops with different configurations
- **Animated Transitions:** ECharts handles smooth data transitions automatically
- **Custom Themes:** Define color palettes that match your cables.gl aesthetic
- **Responsive Charts:** Connect viewport size to chart dimensions

#### Performance Tips:

- Limit data points for smooth animation (< 1000 for real-time)
- Use `notMerge: true` for complete data replacement
- Disable animations for very high-frequency updates
- Cache chart instances when possible

#### Resources:

- Apache ECharts Documentation
- ECharts Examples Gallery
- cables.gl ECharts Integration Tutorial

### 3.13.3 Real-Time Data

Visualize live data streams:

```
WebSocket/API -> Parse data
|
ArrayIterator -> Visualize each value
|
Smooth/Interpolate for fluid animation
```

## 3.14 Complex Example Projects

### 3.14.1 Example 4: Kaleidoscope Effect

```
MainLoop
|
BasicMaterial
```

```
|  
IteratorLoop (6 segments)  
|  
Transform (rotate by index * 60°)  
|  
Transform (mirror flip alternating)  
|  
Your content (shapes, webcam, etc.)
```

### 3.14.2 Example 5: Audio-Reactive Loading Animation

```
AudioAnalyzer (beat detection)
|
IteratorLoop (circle of dots)
|
Index + Time -> Rotation
Beat amplitude -> Scale pulse
|
SetColor (beat changes color)
|
Circle (dot)
```

### 3.14.3 Example 6: Data-Driven Weather Visualization

```
API -> Fetch weather data
|
Parse JSON -> Extract values
|
Temperature -> Background color
Humidity -> Particle density
Wind -> Animation speed
|
Animated scene reflecting weather
```

### 3.14.4 Example 7: Feedback Tunnel Effect

```
RenderToTexture (previous frame)
|
Transform (scale 1.05, center pivot)
|
SetAlpha (0.98 for fade)
|
Draw to screen
|
Add new circles at edges
|
Feed back into texture
```

Creates an infinite tunnel effect.

### 3.14.5 Example 8: Mouse Trail with Fade

```
MousePosition
|
RenderToTexture (with feedback)
|
ColorCorrection (reduce brightness)
|
Draw circle at mouse position
|
Blend with previous frame
```

Creates smooth, fading trails following the cursor.

## 3.15 Performance Optimization

### 3.15.1 Culling and Clipping

Only draw what's visible:

```
If (shape position in viewport bounds)
-> Draw shape
```

```
Else
-> Skip
```

### 3.15.2 Object Pooling

Reuse shape instances instead of creating new ones:

```
// Maintain pool of inactive shapes
// Activate/deactivate as needed
// Prevents GC thrashing
```

### 3.15.3 Level of Detail (LOD)

Simplify distant or small shapes:

```
If (shape size < threshold)
-> Use simple circle
Else
-> Use detailed polygon
```

### 3.15.4 Batching Draw Calls

Group similar operations:

```
SetColor once
|
Draw all shapes of same color
|
SetColor again
|
Draw next batch
```

Reduces state changes and improves performance.

## 3.16 Masking and Clipping

### 3.16.1 Stencil Buffer Masking

Use shapes as masks for other shapes:

```
EnableStencil  
|  
Draw mask shape (Circle)  
|  
SetStencilMode (draw only inside)  
|  
Draw content (Rectangle)  
|  
DisableStencil
```

### 3.16.2 Alpha Mask Technique

Use texture alpha for complex masks:

```
MaskTexture -> AlphaMask  
|  
Your content (masked by texture)
```

## 3.17 Color Theory in Practice

### 3.17.1 Color Harmonies

Generate pleasing color palettes:

**Complementary:**

```
BaseHue -> SetColor (shape 1)  
BaseHue + 180° -> SetColor (shape 2)
```

**Triadic:**

```
BaseHue -> Color 1  
BaseHue + 120° -> Color 2  
BaseHue + 240° -> Color 3
```

**Analogous:**

```
BaseHue -> Color 1  
BaseHue + 30° -> Color 2  
BaseHue - 30° -> Color 3
```

### 3.17.2 Gradient Creation

Smooth color transitions:

**Linear Gradient:**

```
IteratorLoop (steps)  
|  
Index / TotalSteps -> Mix (Color1, Color2, t)  
|  
SetColor -> Rectangle strip
```

**Radial Gradient:**

```
Distance from center -> Mix (Inner, Outer, t)
```

## 3.18 Typography and Text Effects

### 3.18.1 Dynamic Text

Animate text properties:

```
Time -> Character spacing  
MouseX -> Font size  
AudioLevel -> Text opacity
```

### 3.18.2 Text as Texture

Use text rendering for effects:

```
TextTexture (render text to texture)  
|  
Apply shader effects  
|  
Use as sprite or background
```

### 3.18.3 Kinetic Typography

Animate individual letters:

```
TextArray (split into chars)  
|  
ArrayIterator  
|  
Transform (unique per character)  
|  
DrawText (single char)
```

## 3.19 Practical Examples

### 3.19.1 Example 1: Pulsing Circle

```
MainLoop  
|  
BasicMaterial (set your color)  
|  
Time -> Sin -> Scale input  
|  
Circle
```

### 3.19.2 Example 2: Rotating Grid

```
MainLoop  
|  
BasicMaterial  
|  
IteratorLoop (10x10)  
|  
Transform (position from iterator)  
|  
Transform (rotation from Time)  
|  
Rectangle
```

### 3.19.3 Example 3: Color Gradient Circle

```
MainLoop  
|  
IteratorLoop (for each ring)  
|  
IteratorIndex -> Map to Hue -> HSBtoRGB ->  
> BasicMaterial (color input)  
|  
BasicMaterial  
|  
Circle (radius from iterator index)
```

## 3.20 Debugging and Workflow Tips

### 3.20.1 Visualizing Values

See what your ops are outputting:

```
Value -> NumberDisplay  
Value -> DrawNumber (on screen)
```

### 3.20.2 Color Coding

Use consistent colors to identify different element types:  
- Structural elements: Blue  
- Interactive elements: Green  
- Data elements: Yellow  
- Background: Dark grey

### 3.20.3 Naming Convention

Name ops descriptively:  
- TransformRotation\_MainShape  
- Color\_Background  
- Iterator\_ParticleGrid

### 3.20.4 Comment Ops

Document complex sections:

```
Comment ("This section creates the feedback loop")  
|  
Your complex patch area
```

## 3.21 Common Patterns and Recipes

### 3.21.1 Pattern: Circular Array

Arrange shapes in a circle:

```
IteratorLoop (count)  
|  
Index * (360 / count) -> Angle  
Angle -> Cos -> X position  
Angle -> Sin -> Y position  
|  
Transform -> Shape
```

### 3.21.2 Pattern: Wave Grid

Create wave motion across a grid:

```
IteratorLoop (rows)  
|  
IteratorLoop (columns)  
|  
(X + Time) -> Sin -> Y offset  
|  
Transform -> Shape
```

### 3.21.3 Pattern: Spiral

Generate spiral patterns:

```
IteratorLoop  
|  
Index -> Angle (index * goldenAngle)  
Index -> Radius (sqrt(index) * spacing)  
|  
Polar to Cartesian  
|  
Transform -> Shape
```

### 3.21.4 Pattern: Responsive Grid

Grid that adapts to screen size:

```
ViewportWidth / CellSize -> Columns  
ViewportHeight / CellSize -> Rows  
|  
IteratorLoop (columns * rows)  
|  
Grid positioning logic
```

## 3.22 Troubleshooting Common Issues

### 3.22.1 “Shapes not appearing”

- Check trigger connections (grey ports)
- Verify MainLoop is connected to BasicMaterial
- Check BasicMaterial alpha isn't 0
- Verify camera/viewport settings

### 3.22.2 “Performance is slow”

- Reduce segment count on circles
- Lower particle/iterator counts
- Disable antialiasing if not needed
- Use simpler blend modes
- Check for unnecessary texture reads

### 3.22.3 “Colors look wrong”

- Verify color space (RGB vs HSB)
- Check SetColor is before shapes
- Verify alpha values
- Check blend modes

### 3.22.4 “Animation is jerky”

- Use Smooth op for value transitions
- Check frame rate in performance monitor
- Reduce complexity during motion
- Pre-calculate expensive operations

## 3.23 Performance Tips

1. **Reduce Segments** - Circles don't need 100 segments if they're small
2. **Batch Similar Shapes** - Group similar operations together
3. **Use Instancing** - For many identical shapes, use instanced drawing
4. **Limit Transparency** - Overlapping transparent shapes are expensive
5. **Cache Calculations** - Don't recalculate same values each frame
6. **Cull Off-Screen** - Don't draw what's not visible
7. **Simplify Blending** - Complex blend modes are expensive
8. **Optimize Textures** - Use appropriate texture sizes
9. **Limit Feedback Depth** - Don't keep too many feedback history frames
10. **Profile Regularly** - Use performance monitor to identify bottlenecks

## 3.24 Featured Videos

### 3.24.1 Official Tutorials



<https://youtu.be/goO3PhuenBI>  
**First Steps in Cables.gl - Kaleidoscope Webcam Effect**  
by The Interactive & Immersive HQ



<https://youtu.be/xnObNRv8n9I>

**Introduction to cables.gl - Data-Driven Gradient from Geo-Located Weather**  
by Kirell Benzi

## 3.24.2 Additional Resources

- **Generative Op Art Tutorial:** Class Central Course - Learn feedback loops and Op Art
- **Interactive Rectangle Tutorial:** Blog Post - Create on-screen sliders
- **Post-Processing Guide:** Official Docs - Apply effects to scenes
- **Data Visualization:** Apache ECharts Integration - Combine with charting libraries
- **Cables.gl Examples:** Official Examples - Browse community creations
- **Coding with Cables:** GitHub Repo - Code examples and custom ops

## 3.25 Exercises

### 3.25.1 Beginner

1. Create a colorful loading spinner using rotating circles
2. Build a grid of squares that change color based on mouse position
3. Make a simple particle system with random positions and sizes

### 3.25.2 Intermediate

4. Create a kaleidoscope effect with 8 mirrored segments
5. Build an interactive color picker using mouse position
6. Implement a feedback tunnel with infinite zoom effect
7. Create a data visualization showing time-series data as animated bars

### 3.25.3 Advanced

8. Build a generative Op Art piece using feedback loops
9. Create a particle system with physics (gravity, collision)
10. Implement a cellular automaton (Game of Life or similar)
11. Create an audio-reactive geometric pattern generator
12. Build a real-time weather visualization using API data

## 3.26 Project Ideas

1. **Abstract Clock** - Time visualization with geometric shapes
2. **Music Visualizer** - Frequency bands displayed as 2D patterns
3. **Generative Logo** - Company logo with parametric variations
4. **Loading Animations** - Collection of animated loaders
5. **Data Dashboard** - Real-time data display with charts
6. **Interactive Art Installation** - Touch/camera-driven visuals
7. **Typography Animation** - Kinetic text effects
8. **Pattern Generator** - Infinite procedural pattern variations
9. **Mouse-Driven Drawing Tool** - Paint with code
10. **Meditation Visual** - Calming, slowly evolving patterns

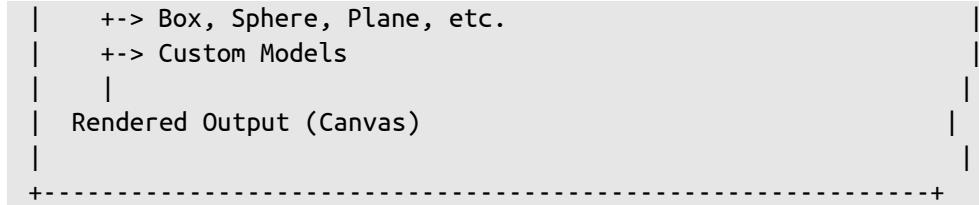
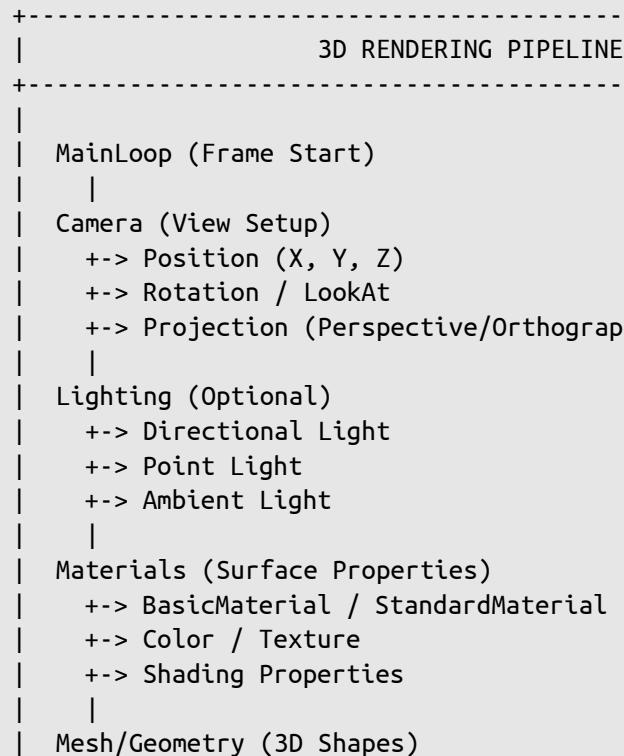
# 4 3D Graphics in Cables.gl

## 4.1 Introduction to 3D

Cables.gl provides powerful tools for creating real-time 3D graphics using WebGL. This chapter covers everything from basic 3D concepts to advanced rendering techniques, scene management, and performance optimization. Whether you're creating simple 3D visualizations or complex interactive experiences, this guide will give you the knowledge to master 3D graphics in cables.gl.

## 4.2 The 3D Pipeline

A basic 3D setup requires:



## 4.3 Cameras

Cameras define how we view the 3D scene.

### 4.3.1 PerspectiveCamera

The most common camera type - mimics human vision with perspective distortion.

**Key Parameters:** - FOV (Field of View) - How wide the view is (typically 45-90 degrees) - Near / Far - Clipping planes (objects outside this range aren't rendered) - Position X/Y/Z - Camera location

### 4.3.2 OrthographicCamera

No perspective distortion - useful for UI, 2D-style 3D, or technical views.

**Key Parameters:** - Zoom - Scale of the view - Near / Far - Clipping planes

### 4.3.3 Orbit Controls

Add interactive camera controls:

```
Camera -> OrbitControls
```

Allows users to rotate, zoom, and pan the view.

### 4.3.4 LookAt

Point the camera at a specific location or object.

```
Camera -> LookAt (target position)
```

**Use Cases:** - Follow a moving object - Create cinematic camera movements - Focus on specific scene elements

### 4.3.5 Camera Animation

Animate camera movement for cinematic effects:

```
Time -> Sin -> Camera Position X  
Time -> Cos -> Camera Position Z  
Time -> Camera Rotation Y (orbit)
```

### 4.3.6 Camera Shake Effect

Add dynamic camera shake:

```
Random -> Multiply (shake intensity) -> Add to Camera Position
```

### 4.3.7 First-Person Camera

Create FPS-style camera controls:

```
MouseX -> Camera Rotation Y  
MouseY -> Camera Rotation X  
WASD Keys -> Camera Position
```

### 4.3.8 Camera Path Following

Follow a predefined path:

```
ArrayIterator (path points)  
|  
Smooth interpolation between points  
|  
Camera Position
```

### 4.3.9 Camera Constraints

Limit camera movement:

```
Camera Position -> Clamp (min, max) -> Constrained Position
```

## 4.4 Lighting

Lighting brings depth and realism to 3D scenes.

### 4.4.1 AmbientLight

Uniform light that illuminates everything equally.

```
MainLoop -> Camera -> AmbientLight -> [Rest of scene]
```

**Tip:** Use subtle ambient light to prevent completely black shadows.

### 4.4.2 DirectionalLight

Light from a specific direction (like the sun).

**Key Parameters:** - Direction (X, Y, Z) - Color - Intensity

### 4.4.3 PointLight

Light emanating from a point in space (like a light bulb).

**Key Parameters:** - Position (X, Y, Z) - Color - Intensity - Falloff radius

#### 4.4.4 SpotLight

Focused beam of light (like a flashlight or stage light).

**Key Parameters:** - Position and direction - Cone angle - Falloff

#### 4.4.5 Shadow Mapping

Enable shadows for more realism:

```
DirectionalLight (shadows enabled) -> ShadowMap -> Scene
```

**Shadow Parameters:** - Shadow Map Size - Resolution (higher = sharper, slower) - Shadow Bias - Prevents shadow acne - Shadow Radius - Softness of shadow edges

**Tip:** Use lower shadow map sizes for better performance. 1024x1024 is usually sufficient.

#### 4.4.6 Three-Point Lighting Setup

Professional lighting arrangement:

```
MainLoop -> Camera  
|  
AmbientLight (subtle, 0.2 intensity) - Fill light  
|  
DirectionalLight (main, from top-left) - Key light  
|  
PointLight (weaker, opposite side) - Rim light  
|[Your scene]
```

**Key Light:** Main illumination (brightest) **Fill Light:** Reduces harsh shadows (ambient or weak directional) **Rim Light:** Creates edge highlights (back/side lighting)

#### 4.4.7 Image-Based Lighting (IBL)

Use environment maps for realistic lighting:

```
HDRITexture -> Environment Map -> PBRMaterial
```

Creates reflections and lighting based on real-world environments.

#### 4.4.8 Light Probes

Place light probes in your scene for accurate local lighting:

```
LightProbe -> Sample nearby lights -> Apply to objects
```

#### 4.4.9 Volumetric Lighting

Create god rays and atmospheric lighting:

```
DirectionalLight -> VolumetricScattering -> [Scene]
```

#### 4.4.10 Light Animation

Animate lights for dynamic scenes:

```
Time -> Sin -> Light Intensity (pulsing)  
Time -> Rotate -> Light Direction (rotating sun)
```

AudioAnalyzer -> Light Color (audio-reactive)

## 4.5 Geometry and Meshes

### 4.5.1 Primitive Shapes

**Cube** - Basic box shape

Parameters: Width, Height, Depth

**Sphere** - Perfect sphere

Parameters: Radius, Segments (horizontal/vertical)

**Cylinder** - Tube shape

Parameters: Radius Top/Bottom, Height, Segments

**Plane** - Flat surface

Parameters: Width, Height

**Torus** - Donut shape

Parameters: Radius, Tube Radius, Segments

### 4.5.2 Loading 3D Models

**OBJLoader** - Load .obj format models

**OBJLoader** -> Mesh

**GLTFLoader** - Load .gltf/.glb models (recommended)

**GLTFLoader** -> Scene/Mesh

**FBXLoader** - Load .fbx models

### 4.5.3 Creating Custom Geometry

Use **PointCloud** or **CustomGeometry** ops to build meshes from data.

**PointCloud**:

ArrayIterator (positions) -> PointCloud

**CustomGeometry**:

Vertices Array -> Normals Array -> UVs Array -> CustomGeometry

### 4.5.4 Procedural Geometry Generation

Create geometry programmatically:

**Example: Procedural Terrain**

```
IteratorLoop (grid)
  |
  NoiseTexture (sample at position) -> Height
  |
  Calculate vertex positions
  |
```

```
Generate normals  
|  
CustomGeometry
```

### Example: Parametric Surfaces

```
U/V parameters -> Math functions -> Vertex positions  
|  
CustomGeometry
```

## 4.5.5 Geometry Instancing

Render many copies efficiently:

```
Mesh -> InstanceTransform (array of transforms) -> InstancedMesh
```

**Use Cases:** - Forests of trees - Crowds of characters - Particle systems - Repeating architectural elements

## 4.5.6 Geometry Modifiers

Modify existing geometry:

### Subdivision:

```
Mesh -> Subdivide -> Smoother surface
```

### Displacement:

```
Mesh -> DisplacementMap -> Deformed geometry
```

### Morphing:

```
Mesh1 -> Morph -> Mesh2 (blend between shapes)
```

## 4.5.7 Boolean Operations

Combine geometries:

```
Mesh1 -> BooleanUnion -> Mesh2  
Mesh1 -> BooleanSubtract -> Mesh2  
Mesh1 -> BooleanIntersect -> Mesh2
```

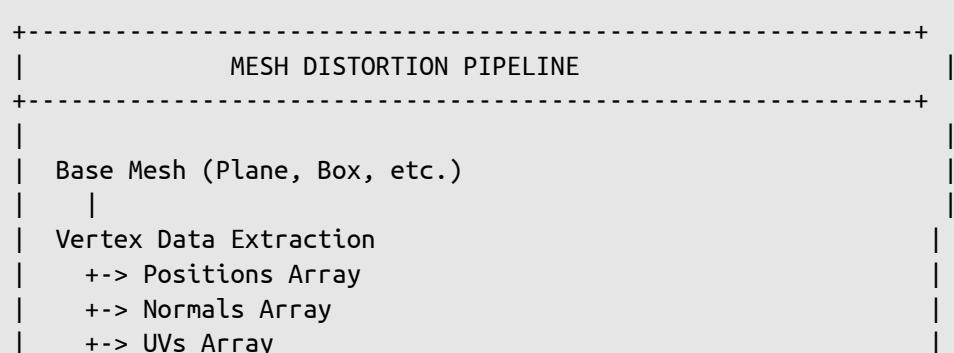
## 4.6 Real-Time Mesh Distortion

Real-time mesh distortion allows you to dynamically modify geometry vertices during rendering, creating effects like bending walls, scaling surfaces, and warping shapes. This is essential for architectural visualization, interactive installations, and dynamic 3D effects.

### 4.6.1 Understanding Vertex Manipulation

Mesh distortion works by modifying vertex positions in real-time. Each vertex has:

- **Position** (X, Y, Z) - Where the vertex is located
- **Normal** (NX, NY, NZ) - Which direction the surface faces
- **UV Coordinates** (U, V) - Texture mapping coordinates



```

+-----+
|   |
| Distortion Function
|     +--> Calculate new positions
|     +--> Update normals (if needed)
|     +--> Preserve UVs
|   |
| CustomGeometry (with distorted vertices)
|   |
| Material -> Render
|
+-----+

```

```

+-----+
|   |
| Material -> Render
|   |
+-----+

```

### Step-by-Step Node Setup:

#### 1. Create Base Plane:

- Add Plane op
- Set Width: 10, Height: 5
- Set Segments Width: 20, Segments Height: 10 (for smooth distortion)

#### 2. Extract Vertex Data:

- Add GetVertices op
- Connect Plane -> GetVertices
- Output: Array of vertex positions

#### 3. Create Scale Controls:

- Add Slider ops for X, Y, Z scale
- Or use Number ops with values

#### 4. Apply Scaling:

- Use ArrayMap or ArrayIterator to multiply each vertex
- For each vertex:  $[x, y, z] * [scaleX, scaleY, scaleZ]$

#### 5. Rebuild Geometry:

- Get original normals and UVs from Plane
- Add CustomGeometry op
- Connect: Scaled Positions -> CustomGeometry
- Connect: Original Normals -> CustomGeometry
- Connect: Original UVs -> CustomGeometry

## 4.6.2 Method 1: Node-Based Distortion

Using built-in cables.gl ops to distort meshes.

### Example 1: Scaling a Wall (Size Transformation)

Transform a plain wall into different sizes using procedural scaling:

```

+-----+
|       WALL SCALING SETUP
+-----+
|   |
| Plane (Base Wall)
|   |
| GetVertices -> Positions Array
|   |
| Scale Factor (X, Y, Z)
|   |
| ArrayMap (multiply each vertex by scale)
|   |
| GetNormals -> Normals Array
| GetUVs -> UVs Array
|   |
| CustomGeometry (new positions, normals, UVs)
|
+-----+

```

### Example 2: Bending a Wall (Curved Distortion)

Bend a plain wall into a curved wall with controllable angle:

```

+-----+
|       WALL BENDING SETUP
+-----+
|   |
| Plane (Base Wall)
|   |
+-----+

```

```

| GetVertices -> Positions Array
|
| Bend Angle (Slider/Number)
| Bend Center (X position where bend occurs)
|
| ArrayMap (apply bend transformation)
|
| Calculate New Normals (for proper lighting)
|
| CustomGeometry -> Material -> Render
|
+-----+

```

### Bending Algorithm (Node-Based):

For each vertex: 1. Calculate distance from bend center 2. Calculate angle based on distance and bend amount 3. Rotate vertex around bend axis 4. Update position

### Node Setup for Bending:

```

Plane
|
GetVertices -> ArrayIterator
|
For each vertex:
|
Vertex X -> Subtract (Bend Center) -> Distance from center
|
Distance -> Multiply (Bend Angle) -> Rotation angle
|
Vertex Y -> Sin(Rotation) -> New Y position
Vertex Z -> Cos(Rotation) -> New Z position
Vertex X -> Keep original
|
Combine -> New Vertex Position
|
ArrayCollect -> All Distorted Vertices
|

```

### CustomGeometry

## 4.6.3 Method 2: JavaScript Custom Op for Mesh Distortion

For more control and performance, use a JavaScript custom op to handle distortion.

### Custom Op: Wall Distorter

Create a custom op that handles both scaling and bending:

```

// Custom Op: WallDistorter
// Distorts a plane mesh with scaling and bending

const inVertices = op.inArray("Input Vertices");
const inNormals = op.inArray("Input Normals");
const inUVs = op.inArray("Input UVs");

// Scale parameters
const inScaleX = op.inFloat("Scale X", 1.0);
const inScaleY = op.inFloat("Scale Y", 1.0);
const inScaleZ = op.inFloat("Scale Z", 1.0);

// Bend parameters
const inBendAngle = op.inFloat("Bend Angle", 0.0); // in radians
const inBendCenter = op.inFloat("Bend Center X", 0.0); // X position of bend
const inBendAxis = op.inSwitch("Bend Axis", ["X", "Y", "Z"], "X");

// Outputs
const outVertices = op.outArray("Distorted Vertices");
const outNormals = op.outArray("Distorted Normals");
const outUVs = op.outArray("Output UVs");

function distortVertices() {
    const vertices = inVertices.get();
    const normals = inNormals.get();

```

```

const uvs = inUVs.get();

if (!vertices || vertices.length === 0) {
    outVertices.set([]);
    outNormals.set([]);
    outUVs.set([]);
    return;
}

const scaleX = inScaleX.get();
const scaleY = inScaleY.get();
const scaleZ = inScaleZ.get();
const bendAngle = inBendAngle.get();
const bendCenter = inBendCenter.get();
const bendAxis = inBendAxis.get();

const distortedVertices = [];
const distortedNormals = [];

for (let i = 0; i < vertices.length; i += 3) {
    let x = vertices[i];
    let y = vertices[i + 1];
    let z = vertices[i + 2];

    // Apply scaling first
    x *= scaleX;
    y *= scaleY;
    z *= scaleZ;

    // Apply bending
    if (Math.abs(bendAngle) > 0.001) {
        if (bendAxis === "X") {
            // Bend along X axis (curves in Y-Z plane)
            const distanceFromCenter = x - bendCenter;
            const angle = distanceFromCenter * bendAngle;

            // Rotate around X axis
            const cosA = Math.cos(angle);
            const sinA = Math.sin(angle);
        }
    }
}

```

```

const newY = y * cosA - z * sinA;
const newZ = y * sinA + z * cosA;
y = newY;
z = newZ;

// Update normals
if (normals && normals.length > i + 2) {
    const nx = normals[i];
    const ny = normals[i + 1];
    const nz = normals[i + 2];
    distortedNormals.push(
        nx,
        ny * cosA - nz * sinA,
        ny * sinA + nz * cosA
    );
}

} else if (bendAxis === "Y") {
    // Bend along Y axis (curves in X-Z plane)
    const distanceFromCenter = y - bendCenter;
    const angle = distanceFromCenter * bendAngle;

    const cosA = Math.cos(angle);
    const sinA = Math.sin(angle);
    const newX = x * cosA - z * sinA;
    const newZ = x * sinA + z * cosA;
    x = newX;
    z = newZ;

    if (normals && normals.length > i + 2) {
        const nx = normals[i];
        const ny = normals[i + 1];
        const nz = normals[i + 2];
        distortedNormals.push(
            nx * cosA - nz * sinA,
            ny,
            nx * sinA + nz * cosA
        );
    }
}

} else if (bendAxis === "Z") {
}

```

```

// Bend along Z axis (curves in X-Y plane)
const distanceFromCenter = z - bendCenter;
const angle = distanceFromCenter * bendAngle;

const cosA = Math.cos(angle);
const sinA = Math.sin(angle);
const newX = x * cosA - y * sinA;
const newY = x * sinA + y * cosA;
x = newX;
y = newY;

if (normals && normals.length > i + 2) {
    const nx = normals[i];
    const ny = normals[i + 1];
    const nz = normals[i + 2];
    distortedNormals.push(
        nx * cosA - ny * sinA,
        nx * sinA + ny * cosA,
        nz
    );
}
} else {
    // No bending, just copy normals
    if (normals && normals.length > i + 2) {
        distortedNormals.push(
            normals[i],
            normals[i + 1],
            normals[i + 2]
        );
    }
}

distortedVertices.push(x, y, z);
}

outVertices.set(distortedVertices);
if (distortedNormals.length > 0) {
    outNormals.set(distortedNormals);
}

```

```

} else if (normals) {
    outNormals.set(normals);
}
if (uvs) {
    outUVs.set(uvs);
}
}

// Update when inputs change
inVertices.onChange = distortVertices;
inNormals.onChange = distortVertices;
inUVs.onChange = distortVertices;
inScaleX.onChange = distortVertices;
inScaleY.onChange = distortVertices;
inScaleZ.onChange = distortVertices;
inBendAngle.onChange = distortVertices;
inBendCenter.onChange = distortVertices;
inBendAxis.onChange = distortVertices;

```

## Using the Wall Distorter Op

### Setup:

```

Plane (Base Wall)
|
GetVertices -> WallDistorter (Input Vertices)
GetNormals -> WallDistorter (Input Normals)
GetUVs -> WallDistorter (Input UVs)
|
WallDistorter (Distorted Vertices) -> CustomGeometry
WallDistorter (Distorted Normals) -> CustomGeometry
WallDistorter (Output UVs) -> CustomGeometry
|
Material -> Render

```

**Controls:** - **Scale X/Y/Z:** Resize the wall - **Bend Angle:** Curvature amount (in radians, use `Math.PI/4` for 45°) - **Bend Center X:** Where the bend occurs along the wall - **Bend Axis:** Which axis to bend around

#### 4.6.4 Advanced: Animated Wall Distortion

Combine distortion with animation for dynamic effects:

```
// Custom Op: AnimatedWallDistorter
// Adds time-based animation to distortion

const inVertices = op.inArray("Input Vertices");
const inNormals = op.inArray("Input Normals");
const inUVs = op.inArray("Input UVs");

// Animation parameters
const inTime = op.inFloat("Time", 0.0);
const inAnimationSpeed = op.inFloat("Animation Speed", 1.0);
const inAnimationType = op.inSwitch("Animation Type",
    ["None", "Pulse", "Wave", "Oscillate"], "None");

// Distortion parameters (same as before)
const inScaleX = op.inFloat("Scale X", 1.0);
const inScaleY = op.inFloat("Scale Y", 1.0);
const inScaleZ = op.inFloat("Scale Z", 1.0);
const inBendAngle = op.inFloat("Bend Angle", 0.0);
const inBendCenter = op.inFloat("Bend Center X", 0.0);

// Outputs
const outVertices = op.outArray("Distorted Vertices");
const outNormals = op.outArray("Distorted Normals");
const outUVs = op.outArray("Output UVs");

function getAnimatedBendAngle() {
    const baseAngle = inBendAngle.get();
    const time = inTime.get();
    const speed = inAnimationSpeed.get();
    const type = inAnimationType.get();
```

```
if (type === "None") {
    return baseAngle;
} else if (type === "Pulse") {
    // Pulse between 0 and baseAngle
    const pulse = (Math.sin(time * speed) + 1) / 2; // 0 to 1
    return baseAngle * pulse;
} else if (type === "Wave") {
    // Wave effect
    return baseAngle * Math.sin(time * speed);
} else if (type === "Oscillate") {
    // Oscillate around baseAngle
    return baseAngle + Math.sin(time * speed) * (baseAngle * 0.5);
}

return baseAngle;
}

function distortVertices() {
    const vertices = inVertices.get();
    const normals = inNormals.get();
    const uvs = inUVs.get();

    if (!vertices || vertices.length === 0) {
        outVertices.set([]);
        outNormals.set([]);
        outUVs.set([]);
        return;
    }

    const scaleX = inScaleX.get();
    const scaleY = inScaleY.get();
    const scaleZ = inScaleZ.get();
    const bendAngle = getAnimatedBendAngle();
    const bendCenter = inBendCenter.get();

    const distortedVertices = [];
    const distortedNormals = [];
```

```

for (let i = 0; i < vertices.length; i += 3) {
    let x = vertices[i];
    let y = vertices[i + 1];
    let z = vertices[i + 2];

    // Apply scaling
    x *= scaleX;
    y *= scaleY;
    z *= scaleZ;

    // Apply animated bending
    if (Math.abs(bendAngle) > 0.001) {
        const distanceFromCenter = x - bendCenter;
        const angle = distanceFromCenter * bendAngle;

        const cosA = Math.cos(angle);
        const sinA = Math.sin(angle);
        const newY = y * cosA - z * sinA;
        const newZ = y * sinA + z * cosA;
        y = newY;
        z = newZ;

        // Update normals
        if (normals && normals.length > i + 2) {
            const nx = normals[i];
            const ny = normals[i + 1];
            const nz = normals[i + 2];
            distortedNormals.push(
                nx,
                ny * cosA - nz * sinA,
                ny * sinA + z * cosA
            );
        }
    } else {
        if (normals && normals.length > i + 2) {
            distortedNormals.push(
                normals[i],
                normals[i + 1],
                normals[i + 2]
            );
        }
    }
}

// Distort vertices
function distortVertices() {
    const distortedVertices = [];

    for (let i = 0; i < vertices.length; i += 3) {
        let x = vertices[i];
        let y = vertices[i + 1];
        let z = vertices[i + 2];

        // Apply scaling
        x *= scaleX;
        y *= scaleY;
        z *= scaleZ;

        // Apply animated bending
        if (Math.abs(bendAngle) > 0.001) {
            const distanceFromCenter = x - bendCenter;
            const angle = distanceFromCenter * bendAngle;

            const cosA = Math.cos(angle);
            const sinA = Math.sin(angle);
            const newY = y * cosA - z * sinA;
            const newZ = y * sinA + z * cosA;
            y = newY;
            z = newZ;

            // Update normals
            if (normals && normals.length > i + 2) {
                const nx = normals[i];
                const ny = normals[i + 1];
                const nz = normals[i + 2];
                distortedNormals.push(
                    nx,
                    ny * cosA - nz * sinA,
                    ny * sinA + z * cosA
                );
            }
        } else {
            if (normals && normals.length > i + 2) {
                distortedNormals.push(
                    normals[i],
                    normals[i + 1],
                    normals[i + 2]
                );
            }
        }

        distortedVertices.push(x, y, z);
    }
}

outVertices.set(distortedVertices);
if (distortedNormals.length > 0) {
    outNormals.set(distortedNormals);
} else if (normals) {
    outNormals.set(normals);
}
if (uvs) {
    outUVs.set(uvs);
}
}

// Update on input changes
inVertices.onChange = distortVertices;
inNormals.onChange = distortVertices;
inUVs.onChange = distortVertices;
inTime.onChange = distortVertices;
inAnimationSpeed.onChange = distortVertices;
inAnimationType.onChange = distortVertices;
inScaleX.onChange = distortVertices;
inScaleY.onChange = distortVertices;
inScaleZ.onChange = distortVertices;
inBendAngle.onChange = distortVertices;
inBendCenter.onChange = distortVertices;

```

## 4.6.5 Practical Example: Interactive Curved Wall

Complete setup for an interactive curved wall with real-time controls:

+-----+	INTERACTIVE CURVED WALL SETUP	
---------	-------------------------------	--

```

+-----+
| MainLoop
|   |
| Plane (Base Wall)
|     Width: 10, Height: 5
|     Segments: 30x15 (for smooth curves)
|   |
| GetVertices -> WallDistorter
| GetNormals -> WallDistorter
| GetUVs -> WallDistorter
|   |
| Slider (Bend Angle: 0 to PI/2) -> WallDistorter
| Slider (Bend Center: -5 to 5) -> WallDistorter
| Slider (Scale X: 0.5 to 2.0) -> WallDistorter
| Slider (Scale Y: 0.5 to 2.0) -> WallDistorter
|   |
| WallDistorter -> CustomGeometry
|   |
| StandardMaterial -> Render
|   |
| Camera -> OrbitControls
+-----+

```

## 4.6.6 Performance Optimization

For real-time distortion, optimize your setup:

### 1. Reduce Vertex Count When Possible:

- Use fewer segments for static walls
- Increase segments only where distortion is visible

### 2. Cache Calculations:

```

let cachedVertices = null;
let cachedBendAngle = null;

```

```

let cachedScale = null;

function distortVertices() {
    const bendAngle = inBendAngle.get();
    const scale = inScaleX.get();

    // Only recalculate if inputs changed
    if (cachedVertices &&
        cachedBendAngle === bendAngle &&
        cachedScale === scale) {
        return; // Use cached result
    }

    // Recalculate...
    cachedVertices = distortedVertices;
    cachedBendAngle = bendAngle;
    cachedScale = scale;
}

```

### 3. Use Instancing for Multiple Walls:

- Create one distorted wall
- Use InstancedMesh to duplicate it
- Much faster than distorting each wall separately

### 4. Update Only When Needed:

```

// Only update on frame if animation is active
const inRender = op.inTrigger("Render");
inRender.onTriggered = function() {
    if (inAnimationType.get() !== "None") {
        distortVertices();
    }
};

```

## 4.6.7 Advanced Techniques

## Multi-Axis Bending

Bend along multiple axes simultaneously:

```
// Bend along both X and Y axes
const bendX = distanceFromCenterX * bendAngleX;
const bendY = distanceFromCenterY * bendAngleY;

// Apply rotations in sequence
// First rotate around X, then around Y
```

## Non-Linear Distortion

Use easing functions for smooth transitions:

```
function easeInOutCubic(t) {
    return t < 0.5
        ? 4 * t * t * t
        : 1 - Math.pow(-2 * t + 2, 3) / 2;
}

const easedAngle = baseAngle * easeInOutCubic(progress);
```

## Texture Coordinate Preservation

When distorting, UVs should remain unchanged for proper texturing:

```
// Always preserve original UVs
outUVs.set(inUVs.get()); // Don't modify UVs during distortion
```

## 4.6.8 Common Use Cases

### 1. Architectural Visualization:

- Bend walls to show different room layouts
- Scale walls to demonstrate space variations

### 2. Interactive Installations:

- User-controlled wall distortion
- Audio-reactive bending

### 3. Animation:

- Morphing between straight and curved walls
- Dynamic space transformations

### 4. Game Mechanics:

- Procedural level generation
- Dynamic environment changes

## 4.6.9 Troubleshooting

**Problem: Normals look wrong after distortion** - Solution: Recalculate normals after distortion - Use `CalculateNormals` op or compute in JavaScript

**Problem: Texture stretches or distorts** - Solution: Don't modify UV coordinates - Keep original UVs from the base mesh

**Problem: Performance is slow** - Solution: Reduce vertex count - Cache calculations - Only update when parameters change

**Problem: Bending looks jagged** - Solution: Increase mesh segments - Use smoother interpolation

## 4.7 Materials

Materials define how surfaces appear when lit.

### 4.7.1 BasicMaterial

Simple colored material, not affected by lighting.

### 4.7.2 LambertMaterial

Matte material with diffuse lighting.

### 4.7.3 PhongMaterial

Shiny material with specular highlights.

**Key Parameters:** - Diffuse Color - Base color - Specular Color - Highlight color - Shininess - How sharp the highlights are

### 4.7.4 PBRMaterial (Physically Based Rendering)

Most realistic material option.

**Key Parameters:** - Albedo - Base color - Metalness - How metallic (0 = plastic, 1 = metal) - Roughness - Surface smoothness (0 = mirror, 1 = rough) - Normal Map - Surface detail - Ambient Occlusion - Crevice shadows - Emissive - Self-illumination - Clearcoat - Additional glossy layer (for car paint, etc.)

**PBR Workflow Tips:** - Use real-world material values for best results - Metalness and Roughness are inverse - metals are usually smooth (low roughness) - Combine texture maps for realistic surfaces - Use HDR environment maps for accurate reflections

### 4.7.5 Material Blending

Blend between materials:

```
Material1 -> Mix -> Material2 (blend factor) -> BlendedMaterial
```

### 4.7.6 Animated Materials

Animate material properties:

```
Time -> Sin -> Material Color (pulsing)  
Time -> Material Roughness (shimmer effect)  
MouseX -> Material Metalness (interactive)
```

### 4.7.7 Material Variants

Create material variations:

```
BaseMaterial -> Multiply Color -> Variant1  
BaseMaterial -> Multiply Color -> Variant2
```

### 4.7.8 Custom Shader Materials

Use custom GLSL shaders (see Shaders chapter):

```
ShaderMaterial (custom GLSL) -> Mesh
```

### 4.7.9 Material Instancing

Apply same material to multiple objects efficiently:

```
Material -> Apply to multiple meshes
```

## 4.8 Transformations in 3D

### 4.8.1 Transform

Same as 2D but with full 3D control:

```
Transform  
+-- TranslateX, TranslateY, TranslateZ  
+-- RotateX, RotateY, RotateZ  
+-- ScaleX, ScaleY, ScaleZ (or uniform Scale)
```

## 4.8.2 Matrix Operations

For advanced control, use matrix ops:

- MatrixMultiply - Combine transformations
- LookAt - Point object at target
- Billboard - Always face camera
- MatrixInvert - Reverse transformation
- MatrixDecompose - Extract position/rotation/scale

## 4.8.3 Hierarchical Transforms

Create parent-child relationships:

```
Transform (parent)
  |
Transform (child) - inherits parent's transform
  |
  Mesh
```

**Use Cases:**

- Character rigging (body -> arm -> hand)
- Vehicle systems (car -> wheel -> tire)
- Solar systems (sun -> planet -> moon)

## 4.8.4 Constraint Systems

Constrain object movement:

### Distance Constraint:

```
Object1 Position -> Distance -> Object2 Position (maintain distance)
```

### Look-At Constraint:

```
Object -> LookAt -> Target (always face target)
```

### Path Constraint:

```
Object -> Follow Path -> Constrained movement
```

## 4.8.5 IK (Inverse Kinematics)

Control chains of objects:

```
End Effector Position -> IK Solver -> Joint Angles
  |
Transform chain
```

## 4.8.6 Physics-Based Transforms

Use physics for natural movement:

```
PhysicsBody -> Transform (position/rotation from physics)
```

## 4.8.7 Transform Caching

Cache expensive transformations:

```
Transform -> Cache -> Reuse for multiple objects
```

## 4.9 Rendering Techniques

### 4.9.1 Rendering Order

Opaque objects should render before transparent ones:

```
MainLoop -> Camera  
|  
[Opaque objects]  
|  
EnableBlending  
|  
[Transparent objects]
```

## 4.9.2 Multiple Render Passes

Create effects like glow, depth of field, or reflections:

```
MainLoop -> Camera -> RenderToTexture -> [Scene]  
|  
TextureEffect  
|  
RenderToScreen
```

## 4.9.3 Fog

Add atmospheric depth:

```
MainLoop -> Camera -> Fog -> [Scene]
```

**Types:** - Linear fog - Constant density - Exponential fog - Density increases with distance - Height fog - Fog based on Y position

## 4.9.4 Screen-Space Ambient Occlusion (SSAO)

Add depth and realism:

```
MainLoop -> Camera -> RenderToTexture (depth)  
|  
SSAO Effect  
|  
Apply to scene
```

## 4.9.5 Screen-Space Reflections (SSR)

Realistic reflections without reflection probes:

```
Scene -> RenderToTexture -> SSR Effect -> Reflections
```

## 4.9.6 Depth of Field

Focus blur effect:

```
Camera -> DepthOfField -> Focus distance -> Blur amount
```

## 4.9.7 Bloom

Glowing highlights:

```
Scene -> Brightness threshold -> Blur -> Add back -> Bloom
```

## 4.9.8 Motion Blur

Blur moving objects:

```
Previous frame -> Current frame -> Blend -> Motion blur
```

## 4.9.9 Color Grading

Post-process color adjustments:

```
Scene -> ColorCorrection
    +-+ Exposure
    +-+ Contrast
    +-+ Saturation
    +-+ Color temperature
    +-+ Tint
```

```
SSAO
  |
  Bloom
  |
  ColorGrading
  |
  ChromaticAberration
  |
  Vignette
  |
  Final Output
```

## 4.9.10 Chromatic Aberration

Color separation effect:

```
Scene -> ChromaticAberration -> Distorted colors
```

## 4.9.11 Vignette

Darken edges:

```
Scene -> Vignette -> Darkened corners
```

## 4.9.12 Post-Processing Chain

Combine multiple effects:

```
Scene
  |
  RenderToTexture
  |
```

## 4.10 Scene Management

### 4.10.1 Scene Hierarchy

Organize complex scenes:

```
MainLoop -> Camera
  |
  Scene (root)
    +-+ Environment
    |   +-+ Skybox
    |   +-+ Fog
    +-+ Lighting
    |   +-+ AmbientLight
    |   +-+ DirectionalLight
    |   +-+ PointLights (array)
    +-+ Static Objects
    |   +-+ [Buildings, terrain, etc.]
    +-+ Dynamic Objects
    |   +-+ [Characters, vehicles, etc.]
    +-+ Effects
      +-+ Particles
      +-+ Post-processing
```

## 4.10.2 Object Grouping

Group related objects:

```
Group (name: "Characters")
  +- Character1
  +- Character2
  +- Character3
```

## 4.10.3 Layer System

Use layers for organization:

```
Layer 0: Background
Layer 1: Environment
Layer 2: Characters
Layer 3: Effects
Layer 4: UI
```

## 4.10.4 Culling and Optimization

Hide objects outside view:

```
Object Position -> FrustumCull -> Only render if visible
```

## 4.10.5 LOD (Level of Detail) System

Use simpler models at distance:

```
Distance from camera -> If > threshold -> Use LOD model
```

## 4.11 Practical Examples

### 4.11.1 Example 1: Rotating Cube

```
MainLoop
  |
  PerspectiveCamera
  |
  DirectionalLight
  |
  Time -> RotateY input
  |
  PhongMaterial
  |
  Cube
```

### 4.11.2 Example 2: Lit Sphere with Orbit Controls

```
MainLoop
  |
  PerspectiveCamera -> OrbitControls
  |
  AmbientLight (subtle)
  |
  PointLight
  |
  PBRMaterial (metalness: 1, roughness: 0.2)
  |
  Sphere
```

### 4.11.3 Example 3: Loading a 3D Model

```
MainLoop
  |
  PerspectiveCamera
```

```
DirectionalLight  
|  
GLTFLoader (your model.glb)  
|  
Transform (scale/position)
```

#### 4.11.4 Example 4: Solar System

```
MainLoop  
|  
PerspectiveCamera -> OrbitControls  
|  
AmbientLight (space ambient)  
|  
DirectionalLight (sun)  
|  
[Sun] - Static sphere with emissive material  
|  
[Planet1] - Transform (orbit around sun)  
|   +- Time -> RotateY (orbit)  
|   +- Time -> RotateY (self-rotation)  
|   +- Sphere  
|  
[Planet2] - Different orbit speed  
+-- [Moon] - Orbits planet
```

#### 4.11.5 Example 5: Procedural Terrain

```
MainLoop  
|  
PerspectiveCamera -> OrbitControls  
|  
DirectionalLight  
|  
IteratorLoop (grid: 100x100)  
|
```

```
Position -> NoiseTexture (3D noise) -> Height  
|  
Calculate vertex (X, height, Z)  
|  
Calculate normal from neighbors  
|  
CustomGeometry  
|  
PBRMaterial (terrain textures)
```

#### 4.11.6 Example 6: Instanced Forest

```
MainLoop  
|  
PerspectiveCamera  
|  
DirectionalLight  
|  
TreeModel (loaded GLTF)  
|  
ArrayIterator (1000 positions)  
|  
Random -> Scale variation  
Random -> Rotation variation  
|  
InstanceTransform  
|  
InstancedMesh
```

#### 4.11.7 Example 7: Interactive 3D Scene

```
MainLoop  
|  
PerspectiveCamera -> OrbitControls  
|  
MouseX -> Map -> Light Direction X
```

```
MouseY -> Map -> Light Direction Y  
|  
DirectionalLight  
|  
MouseClick -> Toggle -> Object visibility  
|  
[Scene objects]
```

#### 4.11.8 Example 8: Animated Character

```
MainLoop  
|  
PerspectiveCamera  
|  
DirectionalLight  
|  
CharacterModel  
|  
Timeline  
  +-- Frame 0: Idle pose  
  +-- Frame 30: Walk cycle start  
  +-- Frame 60: Walk cycle end  
  +-- [Loop]  
|  
Apply to skeleton  
|  
AnimatedMesh
```

#### 4.11.9 Example 9: Particle System

```
MainLoop  
|  
PerspectiveCamera  
|  
ArrayIterator (particles)  
|
```

```
Particle Data  
  +-- Position (update with velocity)  
  +-- Velocity (update with forces)  
  +-- Life (decrease over time)  
  +-- Size (scale with life)  
|  
Transform (position, scale)  
|  
BasicMaterial (color from life)  
|  
Sphere (small)
```

#### 4.11.10 Example 10: Reflective Surface

```
MainLoop  
|  
PerspectiveCamera  
|  
[Scene to reflect]  
|  
RenderTargetTexture (reflection view)  
|  
CubemapTexture  
|  
PBRMaterial (reflection map)  
|  
Plane (mirror surface)
```

#### 4.11.11 Example 11: Volumetric Fog

```
MainLoop  
|  
PerspectiveCamera  
|  
Scene
```

```

RenderTarget (depth)
|
VolumetricFog
  +-+ Depth texture
  +-+ Noise texture (for variation)
  +-+ Light direction
  |
Blend with scene

```

#### 4.11.12 Example 12: Dynamic Lighting Setup

```

MainLoop
|
PerspectiveCamera
|
Time -> Sin -> Sun angle
|
Sun angle -> Calculate direction
|
DirectionalLight (sun)
  +-+ Color (warm -> cool based on angle)
  +-+ Intensity (day -> night)
|
AmbientLight
  +-+ Intensity (complement sun)
|
[Scene]

```

#### 4.11.13 Example 13: Morphing Objects

```

MainLoop
|
PerspectiveCamera
|
Time -> Sin -> Morph factor (0 to 1)
|

```

```

Mesh1 -> Morph -> Mesh2
|
Material

```

#### 4.11.14 Example 14: Physics Simulation

```

MainLoop
|
PerspectiveCamera
|
PhysicsWorld
  +-+ Gravity
  +-+ Colliders
  |
PhysicsBody (rigid body)
  +-+ Mass
  +-+ Forces
  +-+ Collisions
  |
Transform (from physics)
|
Mesh

```

#### 4.11.15 Example 15: Post-Processing Pipeline

```

MainLoop
|
PerspectiveCamera
|
[Render scene]
|
RenderTarget
|
SSAO
|
Bloom (extract bright areas)

```

```
|  
Blur (bloom)  
|  
Add bloom back  
|  
ColorGrading  
  +-- Exposure  
  +-- Contrast  
  +-- Saturation  
|  
ChromaticAberration  
|  
Vignette  
|  
Final output
```

#### 4.11.16 Example 16: Audio-Reactive 3D

```
MainLoop  
|  
PerspectiveCamera  
|  
AudioAnalyzer -> FFTArray  
|  
ArrayIterator (frequency bands)  
|  
FFT Value -> Scale Y  
|  
Transform (position from index, scale from FFT)  
|  
Cube (bar visualization)
```

#### 4.11.17 Example 17: Procedural City

```
MainLoop  
|
```

```
PerspectiveCamera -> OrbitControls  
|  
DirectionalLight  
|  
IteratorLoop (grid: city blocks)  
|  
NoiseTexture -> Building height  
Random -> Building type  
|  
Transform (position, height)  
|  
Cube (building)  
|  
PBRMaterial (building texture)
```

#### 4.11.18 Example 18: Water Surface

```
MainLoop  
|  
PerspectiveCamera  
|  
Time -> Sin -> Wave offset  
|  
Plane (subdivided)  
|  
Vertex shader (displace vertices)  
|  
WaterMaterial  
  +-- Normal map (animated)  
  +-- Reflection (scene)  
  +-- Refraction  
  +-- Foam (at edges)
```

#### 4.11.19 Example 19: Portal Effect

```

MainLoop
|
PerspectiveCamera
|
[Main scene]
|
PortalCamera (different view)
|
RenderToTexture (portal view)
|
Plane (portal frame)
|
Material (portal texture)
|
Stencil buffer (mask to portal shape)

```

#### 4.11.20 Example 20: Multi-Pass Rendering

```

MainLoop
|
PerspectiveCamera
|
[Pass 1: Opaque objects]
|
RenderToTexture
|
[Pass 2: Transparent objects]
|
Blend with Pass 1
|
[Pass 3: Effects]
|
Blend all passes
|
Post-processing

```

## 4.12 Advanced Animation Techniques

### 4.12.1 Skeletal Animation

Animate characters with bones:

```

Skeleton (bone hierarchy)
|
Animation data (keyframes)
|
Skin weights (vertex -> bone influence)
|
AnimatedMesh

```

### 4.12.2 Morph Targets

Blend between shape variations:

```
BaseMesh -> MorphTarget1 (blend factor) -> MorphTarget2
```

**Use Cases:** - Facial expressions - Shape variations - Smooth transitions

### 4.12.3 Procedural Animation

Generate animation with code:

```

Time -> Math functions -> Transform values
|
Apply to objects

```

### 4.12.4 Physics Animation

Use physics for natural movement:

PhysicsBody -> Forces -> Motion -> Transform

## 4.12.5 Animation Blending

Smoothly transition between animations:

Animation1 -> Blend -> Animation2 (blend factor)

## 4.13 Performance Optimization

### 4.13.1 General Tips

1. **Reduce Polygon Count** - Use lower-poly models when possible
2. **Texture Atlas** - Combine textures to reduce draw calls
3. **Level of Detail (LOD)** - Use simpler models for distant objects
4. **Frustum Culling** - Built-in, but organize scenes efficiently
5. **Bake Lighting** - Pre-calculate lighting for static scenes

### 4.13.2 Advanced Optimization

Occlusion Culling:

Object -> Check if occluded -> Skip rendering

Batching:

Similar objects -> Batch -> Single draw call

**Texture Compression:** - Use compressed texture formats (DXT, ETC) - Reduce texture resolution when possible - Use mipmaps for distant objects

**Geometry Optimization:** - Remove unnecessary vertices - Use indexed geometry - Optimize UV mapping

**Shader Optimization:** - Minimize texture samples - Use simpler shaders when possible - Avoid branching in shaders

**Render Target Optimization:** - Use appropriate render target sizes - Don't render at higher resolution than display - Use half-precision floats when possible

### 4.13.3 Performance Monitoring

Track performance metrics:

PerformanceMonitor

- +-- FPS
- +-- Draw calls
- +-- Triangle count
- +-- Texture memory
- +-- Shader compilation time

### 4.13.4 Adaptive Quality

Adjust quality based on performance:

FPS -> If < 30 -> Reduce quality

- +-- Lower LOD
- +-- Disable effects
- +-- Reduce particle count

## 4.14 Common Patterns and Workflows

### 4.14.1 Pattern: Object Pooling

Reuse objects instead of creating/destroying:

```
Pool of inactive objects
  |
Activate when needed
  |
Deactivate when done
  |
Return to pool
```

## 4.14.2 Pattern: Component System

Organize object behavior:

```
GameObject
  +-+ Transform component
  +-+ Render component
  +-+ Physics component
  +-+ Script component
```

## 4.14.3 Pattern: Event System

Decouple object interactions:

```
EventEmitter
  +-+ Subscribe (listener)
  +-+ Emit (event)
  |
Objects react to events
```

## 4.14.4 Pattern: State Machine

Manage object states:

```
StateMachine
  +-+ Idle state
  +-+ Active state
  +-+ Transition conditions
```

## 4.15 Debugging 3D Scenes

### 4.15.1 Visual Debugging

Show Normals:

```
Mesh -> DebugNormals -> Visualize normals
```

Show Bounding Boxes:

```
Mesh -> DebugBounds -> Show bounding boxes
```

Show Wireframe:

```
Material -> Wireframe mode -> See geometry
```

Show Grid:

```
GridHelper -> Visual reference
```

### 4.15.2 Common Issues

**"Objects not visible"** - Check camera position and direction - Verify objects are within near/far planes - Check material alpha values - Verify lighting setup

**"Shadows look wrong"** - Adjust shadow bias - Increase shadow map resolution - Check light shadow settings - Verify shadow receiving objects

**"Performance is slow"** - Reduce polygon count - Lower texture resolutions - Disable expensive effects - Use LOD system - Optimize shaders

**"Materials look incorrect"** - Verify texture UV mapping - Check normal map orientation - Verify PBR material values - Check lighting setup

## 4.16 Best Practices

1. **Start Simple** - Build complexity gradually
2. **Optimize Early** - Consider performance from the start
3. **Use Instancing** - For repeated objects
4. **Organize Scenes** - Use hierarchies and groups
5. **Test on Target Hardware** - Performance varies by device
6. **Use Appropriate Formats** - GLTF for models, compressed textures
7. **Profile Regularly** - Use performance tools
8. **Document Complex Setups** - Add comments to patches
9. **Version Control** - Save iterations of complex scenes
10. **Reuse Assets** - Don't duplicate unnecessarily

## 4.17 Featured Videos

## 4.18 Exercises

### 4.18.1 Beginner

1. Create a solar system with orbiting planets
2. Build a simple room with multiple light sources
3. Load a 3D model and add interactive rotation controls
4. Create a rotating cube with different materials
5. Build a simple scene with fog

### 4.18.2 Intermediate

6. Create a procedural terrain with noise
7. Build an instanced forest with 100+ trees
8. Implement a three-point lighting setup
9. Create a water surface with animated waves
10. Build a particle system with physics
11. Create a portal effect with dual cameras
12. Implement post-processing effects (bloom, SSAO)

13. Build an audio-reactive 3D visualization
14. Create a morphing object animation
15. Implement a character with skeletal animation

### 4.18.3 Advanced

16. Build a complete scene with LOD system
17. Create a volumetric fog effect
18. Implement screen-space reflections
19. Build a physics-based simulation
20. Create a procedural city generator
21. Implement a multi-pass rendering pipeline
22. Build an interactive 3D game scene
23. Create advanced post-processing chain
24. Implement custom shader materials
25. Build a complex scene with optimization techniques

## 4.19 Project Ideas

1. **3D Product Viewer** - Interactive product showcase
2. **Architectural Visualization** - Building walkthrough
3. **Game Prototype** - Simple 3D game mechanics
4. **Data Visualization** - 3D charts and graphs
5. **Virtual Gallery** - 3D art exhibition
6. **Interactive Installation** - Museum or event display
7. **Music Visualizer** - 3D audio-reactive visuals
8. **Procedural World** - Generated landscape exploration
9. **Character Animation** - Animated character showcase
10. **Physics Sandbox** - Interactive physics playground

# 5 Texturing in Cables.gl

## 5.1 Introduction to Textures

Textures add detail, color, and realism to your visuals. In cables.gl, textures can come from images, videos, webcams, or be generated procedurally.

## 5.2 Loading Textures

### 5.2.1 ImageTexture

Load images from files or URLs:

ImageTexture -> Material (texture input)

**Supported Formats:** - PNG (with transparency) - JPG - WebP - GIF (first frame or animated)

**Key Parameters:** - URL - Path to image - Filter - Nearest (pixelated) or Linear (smooth) - Wrap - Repeat, Clamp, Mirror

### 5.2.2 VideoTexture

Use video as a texture:

VideoTexture -> Material (texture input)

**Key Parameters:** - URL - Path to video file - Loop - Whether to loop playback - Playback Rate - Speed control - Volume - Audio volume

**Supported Formats:** - MP4 (H.264) - WebM

### 5.2.3 WebcamTexture

Live webcam input as a texture:

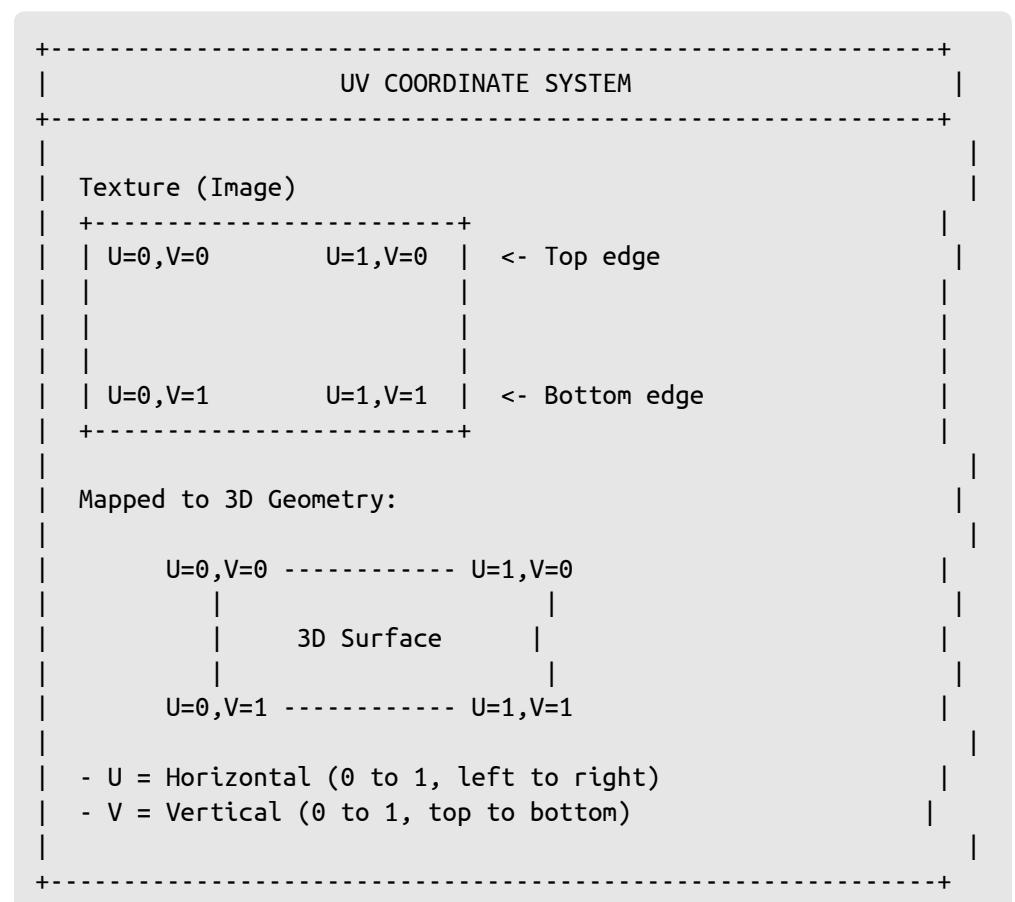
WebcamTexture -> Material (texture input)

**Tip:** Great for interactive installations!

## 5.3 Texture Mapping

### 5.3.1 UV Coordinates

UV coordinates define how textures wrap onto geometry:



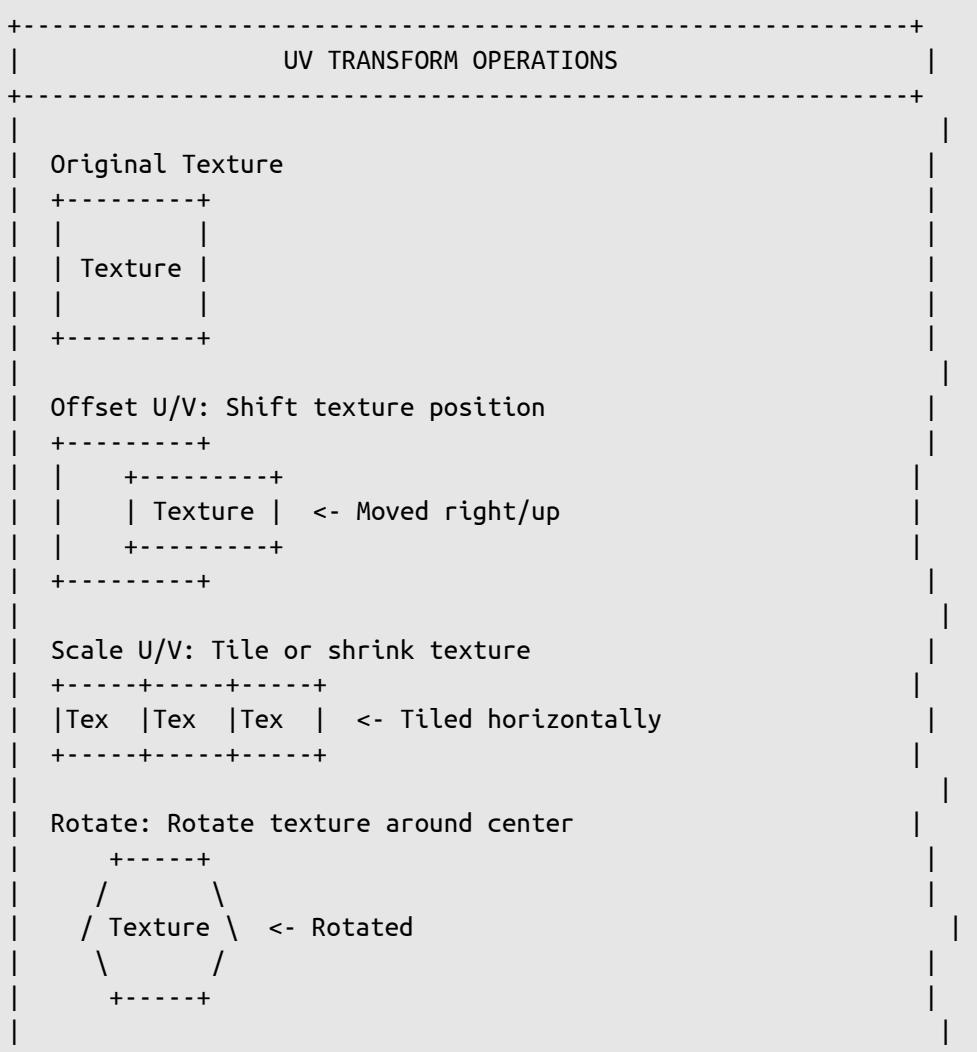
- U = Horizontal position (0 to 1)

- **V** = Vertical position (0 to 1)

Most primitive shapes have automatic UV mapping.

## 5.3.2 UV Transform

Modify texture coordinates:



+-----+  
+-----+

TextureTransform -> Before texture application

**Parameters:** -Offset U/V -Shift the texture -Scale U/V -Tile or shrink -Rotate  
- Rotate the texture

## 5.3.3 Tiling Textures

For seamless repeating:

1. Set wrap mode to Repeat
2. Scale UV coordinates > 1

## 5.4 Advanced Texture Workflow (Production Mindset)

Texturing is where many cables.gl projects move from “cool prototype” to “polished piece”. The two recurring themes are:

- **Correctness:** color space, alpha handling, UVs, aspect ratios, and predictable sampling.
- **Performance:** texture sizes, filtering, mipmaps, compression, and “how many textures are you sampling per pixel”.

### 5.4.1 Color Space: sRGB vs Linear (Why Your Colors Look “Off”)

Most images you download (JPG/PNG/WebP) are authored in **sRGB** (gamma corrected). Most lighting and shading math expects **linear** values. If your project mixes lit materials (e.g., PBR) with UI-like textures, you can run into:

- washed-out or too-dark textures
- incorrect blending
- “metal looks wrong” in PBR

**Practical rule of thumb:**

- **Color/albedo** textures are usually **sRGB**.

- Data textures (normal maps, roughness/metalness/AO, masks) are usually **linear**.

If a texture looks wrong, verify you're not treating a data map like a color map (or vice versa).

## 5.4.2 Alpha (Transparency) Pitfalls

If you see dark/bright halos around transparent textures (logos, sprites), you're likely looking at one of these issues:

- The texture was exported with a bad matte color (common in PNGs).
- The pipeline expects **premultiplied alpha** but you provided straight alpha (or the other way around).
- Filtering/mipmaps sample transparent pixels and "bleed" colors into edges.

**Fix strategies:**

- Add padding/bleed around sprites in your source image.
- Prefer power-of-two textures with mipmaps for distant rendering.
- If you have control over asset export, re-export with correct alpha handling.

## 5.4.3 Filtering, Mipmaps, and Why Textures "Shimmer"

When a textured surface gets small on screen, the GPU needs mipmaps to avoid shimmer and crawling.

- **Nearest** filtering: crisp pixels, great for pixel-art, terrible for most 3D.
- **Linear** filtering: smoother sampling, better for general use.
- **Mipmaps**: essential for 3D surfaces viewed at varying distances.

If a ground texture "crawls" when the camera moves, you typically need mipmaps and (if available) anisotropic filtering.

## 5.4.4 Power-of-Two Sizes (and When It Matters)

Power-of-two textures (256/512/1024/2048/4096) generally behave better for:

- mipmaps

- repeating wrap modes
- GPU compatibility/performance

Non-power-of-two often still works in modern WebGL, but when things behave oddly, returning to power-of-two sizes is a reliable fix.

## 5.4.5 Aspect Ratio Correctness (Especially for Video)

Video textures are a frequent source of "why is it stretched?" issues.

- Match the **Plane** aspect ratio to the video's aspect ratio.
- If you use Fullscreen rectangles, make sure you're compensating for screen aspect.

## 5.5 Advanced Techniques and Patch Recipes

These are "building block" patterns you can reuse across many projects.

### 5.5.1 Recipe: Masked Texture Blend (Two Textures + a Mask)

Use a mask texture (black/white) to blend between two images.

**Conceptual chain:**

```
ImageTexture (A) -+
                  --> (blend using mask) -> Material -> Mesh
ImageTexture (B) -+
ImageTexture (Mask)
```

**Notes:** - The mask should be treated as a data texture (linear). - Great for dirt overlays, decals, and transitions.

### 5.5.2 Recipe: Animated UVs (Scrolling / Parallax)

Scrolling textures are perfect for conveyor belts, moving backgrounds, water normals, etc.

```
Time -> (speed multiply) -> TextureTransform (Offset U/V)
ImageTexture -> Material (texture input)
Material -> Mesh
```

### 5.5.3 Recipe: Render-to-Texture for Post-Processing

Render your scene to a texture, apply effects, then output.

```
MainLoop -> Camera -> RenderToTexture
|
[Scene]
|
TextureEffects -> Output
```

**Use Cases:** - blur/glow chains - color grading - stylized distortion - feedback trails (see next recipe)

### 5.5.4 Recipe: Feedback / Trails (Texture Feedback Loop)

Feedback is a signature look in real-time visuals.

High-level structure:

```
Previous Frame Texture
|
TextureEffects (fade/blur)
|
Combine with New Frame Content
|
RenderToTexture (becomes "previous frame" next tick)
```

**Tip:** Keep feedback subtle (small fade each frame). Large blur + high persistence can become very expensive.

### 5.5.5 Recipe: Planar "Mirror" Reflection (Render-to-Texture)

To fake a mirror floor:

- Render the scene from a reflected camera to a texture.
- Apply that texture onto a plane.

```
MainLoop
+-> Camera (main) -> [Scene]
+-> Camera (reflected) -> RenderToTexture -> Plane Material -> Mirror Plane
```

### 5.5.6 Recipe: Environment Reflections (Cubemap/HDRI)

Use an environment texture for reflections and more believable PBR materials.

```
HDRTexure or CubemapTexture -> (environment input) -> PBRMaterial -> Mesh
```

**Tip:** Even simple objects look dramatically better with good environment lighting.

### 5.5.7 Recipe: Video Texture "Billboard" (Reliable Playback)

```
VideoTexture -> BasicMaterial -> Plane
```

**Checklist:** - Use a browser-served URL (avoid file:// in production). - Make sure autoplay policies are satisfied (user interaction may be required). - Use a fallback poster image if video takes time to load.

### 5.5.8 Recipe: Webcam Texture (Permissions + UX)

WebcamTexture -> BasicMaterial -> Plane

**Checklist:** - Provide a UI prompt ("Click to enable camera"). - Handle denied permissions gracefully (fallback texture). - Keep resolution reasonable for performance.

## 5.6 Texture Types for PBR Materials

### 5.6.1 Albedo/Diffuse Map

The base color of the surface.

### 5.6.2 Normal Map

Adds surface detail without extra geometry.

NormalMap -> PBRMaterial (normal input)

**Tip:** Use tangent-space normal maps (blue-purple appearance).

### 5.6.3 Roughness Map

Controls surface smoothness per-pixel.

- White = rough
- Black = smooth/shiny

### 5.6.4 Metalness Map

Defines metallic vs. non-metallic regions.

- White = metal
- Black = non-metal (dielectric)

### 5.6.5 Ambient Occlusion Map

Pre-baked shadow information for crevices.

### 5.6.6 Height/Displacement Map

Actual geometry displacement (more expensive).

### 5.6.7 Emissive Map

Self-illuminating regions of the surface.

## 5.7 Procedural Textures

Generate textures with code/nodes:

### 5.7.1 Noise Textures

NoiseTexture -> Creates Perlin/Simplex noise

**Types:** - Perlin noise - Simplex noise - Voronoi - Fractal/FBM

### 5.7.2 Gradient Textures

GradientTexture -> Creates color gradients

### 5.7.3 Pattern Generators

- Checkerboard
- Stripes
- Dots
- Custom math-based patterns

## 5.8 Render to Texture

Capture your scene as a texture for post-processing or effects:

```
MainLoop -> Camera -> RenderToTexture  
|  
[Scene to capture]  
|  
TextureOutput -> Use elsewhere
```

### 5.8.1 Common Uses:

1. **Post-processing effects** - Apply shaders to the entire scene
2. **Mirrors/Reflections** - Render from reflection viewpoint
3. **Dynamic textures** - Use one patch's output in another
4. **Feedback effects** - Feed output back as input

## 5.9 Texture Effects

### 5.9.1 TextureEffects Op

Chain of image processing effects:

```
ImageTexture -> TextureEffects -> Output
```

**Available Effects:** - Blur - Sharpen - Color correction - Distortion - Edge detection - Pixelation

### 5.9.2 Custom Shader Effects

Write GLSL for custom texture processing (see Shaders chapter).

## 5.10 Cubemaps and Environment Maps

### 5.10.1 CubemapTexture

Six images forming a surrounding environment:

```
CubemapTexture -> Environment lighting
```

**Uses:** - Skyboxes - Reflections - Image-based lighting (IBL)

### 5.10.2 HDRITexture

High Dynamic Range images for realistic lighting:

```
HDRITexture -> IBL/Environment
```

## 5.11 Texture Compression and Optimization

### 5.11.1 File Size Tips:

1. **Use appropriate formats:**
  - PNG for transparency
  - JPG for photos (no transparency)
  - WebP for best compression
2. **Power of 2 sizes:** 256, 512, 1024, 2048, 4096 pixels
3. **Mipmaps:** Enable for textures viewed at varying distances
4. **Compress textures:** Use tools like TinyPNG, Squoosh

### 5.11.2 Memory Considerations:

- 512x512: ~1 MB
- 1024x1024: ~4 MB
- 2048x2048: ~16 MB
- 4096x4096: ~64 MB

## 5.12 Practical Examples

### 5.12.1 Example 1: Textured Rotating Cube

```
MainLoop  
|  
PerspectiveCamera  
|  
DirectionalLight  
|  
Time -> RotateY  
|  
ImageTexture -> PhongMaterial (texture input)  
|  
Cube
```

```
ImageTexture (roughness)  
ImageTexture (metalness)  
| (all connected to PBRMaterial)  
PBRMaterial  
|  
Mesh
```

## 5.12.2 Example 2: Video on a Plane

```
MainLoop  
|  
VideoTexture -> BasicMaterial  
|  
Plane (aspect ratio matching video)
```

## 5.12.3 Example 3: Animated Noise Background

```
MainLoop  
|  
Time -> NoiseTexture (animate offset)  
|  
BasicMaterial  
|  
FullscreenRectangle
```

## 5.12.4 Example 4: PBR Textured Material

```
ImageTexture (albedo)  
ImageTexture (normal)
```

## 5.13 Featured Videos

## 5.14 Exercises

1. Create a textured cube that rotates and displays different images on each face
2. Build a video wall with multiple video textures
3. Create a procedural noise-based animated background
4. Apply PBR textures to a loaded 3D model

# 6 Shaders & GLSL in Cables.gl

## 6.1 Introduction to Shaders

Shaders are programs that run on the GPU, enabling custom visual effects and rendering techniques. Cables.gl provides powerful tools for writing and using GLSL (OpenGL Shading Language) shaders.

## 6.2 What Are Shaders?

Shaders are small programs that determine how graphics are rendered:

- **Vertex Shaders** - Transform vertex positions
- **Fragment Shaders** - Determine pixel colors

Together, they control everything you see on screen.

## 6.3 Why Use Custom Shaders?

- Create unique visual effects
- Achieve effects impossible with built-in ops
- Optimize performance for specific use cases
- Learn the fundamentals of graphics programming

## 6.4 Shader Ops in Cables.gl

### 6.4.1 ShaderMaterial

Apply custom GLSL code as a material:

ShaderMaterial -> Mesh

### 6.4.2 TextureEffect (Shader-based)

Process textures with custom fragment shaders.

### 6.4.3 CustomShader

Full control over vertex and fragment shaders.

## 6.5 GLSL Basics

### 6.5.1 Data Types

```
// Scalars
float a = 1.0;
int b = 5;
bool c = true;

// Vectors
vec2 uv = vec2(0.5, 0.5);
vec3 color = vec3(1.0, 0.0, 0.0); // RGB
vec4 rgba = vec4(1.0, 1.0, 1.0, 1.0);

// Matrices
mat4 transform;

// Samplers (textures)
sampler2D myTexture;
```

### 6.5.2 Swizzling

Access vector components in any order:

```
vec4 color = vec4(1.0, 0.5, 0.25, 1.0);
vec3 rgb = color.rgb;    // (1.0, 0.5, 0.25)
vec2 rg = color.rg;    // (1.0, 0.5)
float r = color.r;      // 1.0
vec3 bgr = color.bgr;  // (0.25, 0.5, 1.0) - reversed!
```

### 6.5.3 Built-in Functions

```

// Math
sin(x), cos(x), tan(x)
pow(x, y)
sqrt(x)
abs(x)
min(a, b), max(a, b)
clamp(x, min, max)

// Interpolation
mix(a, b, t)           // Linear interpolation
smoothstep(edge0, edge1, x)

// Vector operations
length(v)
normalize(v)
dot(a, b)
cross(a, b)
reflect(incident, normal)

// Texture sampling
texture(sampler, uv)

```

## 6.6 Your First Fragment Shader

A simple color gradient:

```

// Fragment Shader
precision mediump float;

varying vec2 vUV; // UV coordinates from vertex shader

void main() {
    // Create gradient based on UV
    vec3 color = vec3(vUV.x, vUV.y, 0.5);

    gl_FragColor = vec4(color, 1.0);
}

```

```

}
```

## 6.7 Common Shader Patterns

### 6.7.1 Solid Color

```

void main() {
    gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0); // Red
}

```

### 6.7.2 UV Gradient

```

void main() {
    gl_FragColor = vec4(vUV, 0.0, 1.0);
}

```

### 6.7.3 Circle (SDF)

```

void main() {
    vec2 center = vec2(0.5, 0.5);
    float dist = length(vUV - center);
    float circle = step(dist, 0.3);

    gl_FragColor = vec4(vec3(circle), 1.0);
}

```

### 6.7.4 Smooth Circle

```

void main() {
    vec2 center = vec2(0.5, 0.5);
    float dist = length(vUV - center);
    float circle = smoothstep(0.3, 0.28, dist);
}

```

```
    gl_FragColor = vec4(vec3(circle), 1.0);
}
```

## 6.7.5 Animated Pattern

```
uniform float time;

void main() {
    float wave = sin(vUV.x * 10.0 + time) * 0.5 + 0.5;
    gl_FragColor = vec4(vec3(wave), 1.0);
}
```

## 6.8 Uniforms

Uniforms are values passed from cables.gl to your shader:

```
uniform float time;      // Current time
uniform vec2 resolution; // Canvas size
uniform sampler2D tex;   // Texture
uniform vec3 color;      // Custom color
```

In cables.gl, connect ops to shader uniform inputs.

## 6.9 Advanced Shader Workflows in cables.gl

The biggest jump in quality comes from treating shaders like reusable “modules”:

- a **clear input contract** (uniforms you expect: time, resolution, textures, parameters)
- predictable **coordinate conventions** (UV vs screen space vs world space)
- a **debug strategy** (visualize intermediate values)
- performance awareness (texture samples, loops, precision)

### 6.9.1 A Practical Uniform “Contract”

In most patches you’ll end up with a small set of recurring uniforms:

- **time** (float): animation driver
- **resolution** (vec2): coordinate normalization
- **tex / tex0 / tex1** (sampler2D): one or more textures
- **amount / strength** (float): effect intensity
- **colorA / colorB** (vec3): palette endpoints

**Tip:** name your uniforms consistently so you can reuse the same patch wiring across multiple shader materials/effects.

### 6.9.2 Coordinate Spaces: UV vs Screen Space

- **UV space** (vUV) is normalized 0..1 per surface.
- **Screen space** is often derived from UV + resolution when you need pixel-sized offsets.

Example helper:

```
vec2 pixel(vec2 uv, vec2 resolution) {
    return 1.0 / resolution;
}
```

### 6.9.3 Anti-Aliasing SDFs (Clean Edges)

Hard step() edges often look jagged. A common pattern is to use smooth-step() with a small “feather”:

```
float aa(float dist, float radius) {
    float edge = 0.002; // tweak for your resolution / style
    return 1.0 - smoothstep(radius - edge, radius + edge, dist);
}
```

When available, fwidth() can provide adaptive edge widths, but keep in mind WebGL precision/derivative constraints in some contexts.

## 6.9.4 Palette Mapping (Better Color Fast)

Instead of picking random RGB values, map a scalar to a palette:

```
vec3 palette(float t, vec3 a, vec3 b, vec3 c, vec3 d) {  
    return a + b * cos(6.28318 * (c * t + d));  
}
```

This gives you rich gradients with a tiny amount of code.

## 6.10 Advanced Examples (Copy-and-Adapt)

These examples are written so you can drop them into a ShaderMaterial/TextureEffect-style fragment shader and then wire the uniforms from your patch.

### 6.10.1 Example: Texture Distortion (UV Warp)

```
precision mediump float;  
varying vec2 vUV;  
uniform sampler2D tex;  
uniform float time;  
uniform float amount;  
  
void main() {  
    vec2 uv = vUV;  
    uv.x += sin(uv.y * 10.0 + time) * amount;  
    uv.y += cos(uv.x * 10.0 + time) * amount;  
    gl_FragColor = texture2D(tex, uv);  
}
```

**Patch wiring idea:** - Time -> time - a slider (0..0.05) -> amount - input texture -> tex

### 6.10.2 Example: Simple Bloom-ish Glow (Threshold + Blur-ish)

This isn't a full separable blur, but it demonstrates the "sample neighbors" pattern.

```
precision mediump float;  
varying vec2 vUV;  
uniform sampler2D tex;  
uniform vec2 resolution;  
uniform float threshold;  
uniform float strength;  
  
void main() {  
    vec2 px = 1.0 / resolution;  
    vec3 c = texture2D(tex, vUV).rgb;  
  
    // crude 5-tap blur  
    vec3 b = vec3(0.0);  
    b += texture2D(tex, vUV + vec2( 1.0, 0.0) * px).rgb;  
    b += texture2D(tex, vUV + vec2(-1.0, 0.0) * px).rgb;  
    b += texture2D(tex, vUV + vec2( 0.0, 1.0) * px).rgb;  
    b += texture2D(tex, vUV + vec2( 0.0,-1.0) * px).rgb;  
    b *= 0.25;  
  
    float luma = dot(c, vec3(0.299, 0.587, 0.114));  
    vec3 glow = (luma > threshold) ? b : vec3(0.0);  
  
    gl_FragColor = vec4(c + glow * strength, 1.0);  
}
```

### 6.10.3 Example: Domain Warping (More Organic Noise)

Domain warping is a standard "make it look expensive" trick: distort the coordinates before sampling noise.

```

precision medium float;
varying vec2 vUV;
uniform float time;

float hash(vec2 p) {
    return fract(sin(dot(p, vec2(127.1, 311.7))) * 43758.5453);
}

float noise(vec2 p) {
    vec2 i = floor(p);
    vec2 f = fract(p);
    float a = hash(i);
    float b = hash(i + vec2(1.0, 0.0));
    float c = hash(i + vec2(0.0, 1.0));
    float d = hash(i + vec2(1.0, 1.0));
    vec2 u = f * f * (3.0 - 2.0 * f);
    return mix(a, b, u.x) + (c - a) * u.y * (1.0 - u.x) + (d -
b) * u.x * u.y;
}

void main() {
    vec2 uv = vUV * 4.0;
    vec2 warp = vec2(
        noise(uv + time * 0.2),
        noise(uv + vec2(5.2, 1.3) - time * 0.2)
    );
    float n = noise(uv + warp * 2.0);
    gl_FragColor = vec4(vec3(n), 1.0);
}

```

## 6.11 Debugging Shaders (In Practice)

When something is wrong, render the intermediate value:

- visualize UVs: `gl_FragColor = vec4(vUV, 0.0, 1.0);`
- visualize a scalar: `gl_FragColor = vec4(vec3(val), 1.0);`
- isolate channels: `gl_FragColor = vec4(texture2D(tex, vUV).rrrr, 1.0);`

### 6.11.1 Common Gotchas

- **Black output:** your shader compiles but outputs 0 (check uniform wiring; check ranges).
- **Solid color:** UVs are constant or your sampling coord is wrong.
- **Stretching:** you're using UVs but expect square pixels; incorporate resolution.
- **Banding:** precision too low; consider `highp` where supported, or dither slightly.

## 6.12 Performance Guidelines (Real-Time Friendly)

- **Texture samples are expensive:** keep them minimal and reuse results.
- **Avoid nested loops:** especially dynamic loops in fragment shaders.
- **Prefer simple math over heavy branching:** GPUs dislike divergent branches.
- **Keep effects modular:** multiple simpler passes can be easier to tune than one huge shader.

## 6.13 Signed Distance Functions (SDFs)

SDFs define shapes mathematically:

### 6.13.1 SDF Primitives

```

// Circle
float sdCircle(vec2 p, float r) {
    return length(p) - r;
}

// Box
float sdBox(vec2 p, vec2 b) {
    vec2 d = abs(p) - b;
    return length(max(d, 0.0)) + min(max(d.x, d.y), 0.0);
}

// Line segment

```

```

float sdSegment(vec2 p, vec2 a, vec2 b) {
    vec2 pa = p - a, ba = b - a;
    float h = clamp(dot(pa, ba) / dot(ba, ba), 0.0, 1.0);
    return length(pa - ba * h);
}

```

## 6.13.2 SDF Operations

```

// Union (combine shapes)
float opUnion(float d1, float d2) {
    return min(d1, d2);
}

// Subtraction (cut one from another)
float opSubtract(float d1, float d2) {
    return max(-d1, d2);
}

// Intersection (overlap only)
float opIntersect(float d1, float d2) {
    return max(d1, d2);
}

// Smooth union
float opSmoothUnion(float d1, float d2, float k) {
    float h = clamp(0.5 + 0.5 * (d2 - d1) / k, 0.0, 1.0);
    return mix(d2, d1, h) - k * h * (1.0 - h);
}

```

# 6.14 Noise Functions

## 6.14.1 Simple Value Noise

```

float random(vec2 st) {
    return fract(sin(dot(st.xy, vec2(12.9898, 78.233))) * 43758.5453);
}

```

```

float noise(vec2 st) {
    vec2 i = floor(st);
    vec2 f = fract(st);

    float a = random(i);
    float b = random(i + vec2(1.0, 0.0));
    float c = random(i + vec2(0.0, 1.0));
    float d = random(i + vec2(1.0, 1.0));

    vec2 u = f * f * (3.0 - 2.0 * f);

    return mix(a, b, u.x) + (c - a) * u.y * (1.0 - u.x) + (d - b) * u.x * u.y;
}

```

## 6.14.2 Fractal Brownian Motion (FBM)

```

float fbm(vec2 st) {
    float value = 0.0;
    float amplitude = 0.5;

    for (int i = 0; i < 5; i++) {
        value += amplitude * noise(st);
        st *= 2.0;
        amplitude *= 0.5;
    }

    return value;
}

```

# 6.15 Post-Processing Effects

## 6.15.1 Vignette

```

float vignette = 1.0 - length(vUV - 0.5) * 1.5;
color *= vignette;

```

## 6.15.2 Chromatic Aberration

```
precision mediump float;
varying vec2 vUV;
uniform sampler2D tex;

void main() {
    vec2 offset = (vUV - 0.5) * 0.01;
    float r = texture2D(tex, vUV + offset).r;
    float g = texture2D(tex, vUV).g;
    float b = texture2D(tex, vUV - offset).b;
    vec3 color = vec3(r, g, b);
    gl_FragColor = vec4(color, 1.0);
}
```

## 6.15.3 Blur (Box Blur)

```
precision mediump float;
varying vec2 vUV;
uniform sampler2D tex;

void main() {
    vec3 blur = vec3(0.0);
    float samples = 9.0;
    float offset = 0.005;

    for (float x = -1.0; x <= 1.0; x++) {
        for (float y = -1.0; y <= 1.0; y++) {
            blur += texture2D(tex, vUV + vec2(x, y) * offset).rgb;
        }
    }
    blur /= samples;
    gl_FragColor = vec4(blur, 1.0);
}
```

## 6.15.4 Pixelation

```
precision mediump float;
varying vec2 vUV;
uniform sampler2D tex;

void main() {
    float pixels = 100.0;
    vec2 pixelUV = floor(vUV * pixels) / pixels;
    vec3 color = texture2D(tex, pixelUV).rgb;
    gl_FragColor = vec4(color, 1.0);
}
```

## 6.16 Vertex Shader Basics

Modify geometry positions:

```
// Vertex Shader
attribute vec3 position;
attribute vec2 uv;

uniform mat4 modelViewMatrix;
uniform mat4 projectionMatrix;
uniform float time;

varying vec2 vUV;

void main() {
    vUV = uv;

    vec3 pos = position;
    // Wave deformation
    pos.z += sin(pos.x * 5.0 + time) * 0.2;

    gl_Position = projectionMatrix * modelViewMatrix * vec4(pos, 1.0);
}
```

## 6.17 Debugging Shaders

### 6.17.1 Visualize Values

```
// Show UV coordinates  
gl_FragColor = vec4(vUV, 0.0, 1.0);  
  
// Show a value as grayscale  
gl_FragColor = vec4(vec3(someValue), 1.0);  
  
// Show negative values in red  
float val = someCalculation;  
if (val < 0.0) {  
    gl_FragColor = vec4(-val, 0.0, 0.0, 1.0);  
} else {  
    gl_FragColor = vec4(0.0, val, 0.0, 1.0);  
}
```

## 6.18 Performance Tips

1. **Avoid branching** - GPUs don't like if/else
2. **Use built-in functions** - They're optimized
3. **Minimize texture samples** - Each sample has cost
4. **Precision matters** - Use mediump when possible
5. **Precompute values** - Do math in JavaScript when possible

## 6.19 Professional Video Projection Mapping in Cables.gl

Projection mapping (also called video mapping or spatial augmented reality) involves projecting images onto real-world surfaces, often requiring geometric correction, multi-projector blending, and specialized color correction. This section provides professional-grade shaders for simulating and preparing projection mapping content within cables.gl.

All shaders in this section are designed for use with cables.gl's built-in `TextureEffect` or `ShaderMaterial` ops - simply paste the shader code into the fragment shader field and connect your inputs. For JavaScript custom op implementations, see the "JavaScript Custom Op Examples" section below.

### 6.19.1 Understanding Cables.gl Shader Context

#### Critical Notes for Cables.gl Shaders:

1. **Resolution Handling:** In cables.gl, resolution uniform is typically `vec2(width, height)` in pixels. When working with UV coordinates (`vUV`), remember:
  - `vUV` ranges from 0.0 to 1.0
  - Screen space =  $vUV * \text{resolution}$
  - Pixel size =  $1.0 / \text{resolution}$
  - **Important:** resolution is NOT automatically provided - you must connect a `CanvasInfo` or `GetResolution` op to the `resolution` port
2. **Texture Sampling:** Always use `texture2D()` (WebGL 1.0 style) in cables.gl, not `texture()`.
3. **Coordinate Systems:**
  - UV space: `vUV` (0.0 to 1.0) - automatically provided
  - Screen space: `vUV * resolution`
  - Normalized screen space:  $(vUV - 0.5) * 2.0$  (ranges -1.0 to 1.0)
4. **Shader Headers:** Always include precision declaration at the top:

```
precision mediump float;
```

#### 5. Uniform Types:

- `float`, `vec2`, `vec3`, `vec4` - Fully supported, become Number/Vector ports
- `sampler2D` - Fully supported, becomes Texture port
- `mat3`, `mat4` - Supported, but verify with Matrix ops in your cables.gl version
- `int` - **Not recommended** - Use `float` instead and compare with `< 0.5` patterns

#### 6. Auto-Provided Variables:

- `varying vec2 vUV` - Always available (no need to declare in vertex shader for `TextureEffect`)
- `uniform float time` - Available if you connect a `Time` op

- uniform vec2 resolution - **NOT auto-provided** - must connect manually

## 6.19.2 Cables.gl Shader Compliance Checklist

Before using any shader in cables.gl, verify:

- Shader starts with precision mediump float;
- Uses texture2D() not texture() for sampling
- Uses varying vec2 vUV (auto-provided, don't declare in vertex shader for TextureEffect)
- No uniform int - converted to uniform float with float comparisons
- All uniforms are properly typed (float, vec2, vec3, vec4, sampler2D)
- Resolution uniform is documented as requiring manual connection
- Shader compiles without errors
- All texture samples are within 0.0-1.0 UV bounds (or clamped)
- No WebGL 2.0 specific features (use WebGL 1.0 compatible code)

## 6.19.3 Troubleshooting Common Issues

**Issue: "Shader won't compile"** - Check for precision mediump float; at the top - Verify all texture() calls are texture2D() - Ensure no WebGL 2.0 features are used - Check for syntax errors (missing semicolons, etc.)

**Issue: "Black screen or no output"** - Verify texture is connected to tex (or appropriate sampler2D) port - Check UV coordinates are in 0.0-1.0 range - Ensure resolution is connected if shader uses it - Check if shader is sampling outside texture bounds

**Issue: "Resolution uniform not working"** - resolution is NOT automatically provided - Connect CanvasInfo op or GetResolution op to resolution port - Verify resolution values are correct (width, height in pixels)

**Issue: "Integer uniforms not working"** - Cables.gl may not support uniform int reliably - Convert to uniform float and use float comparisons: - if (direction == 0) -> if (direction < 0.5) - if (direction == 1) -> if (direction > 0.5 && direction < 1.5)

**Issue: "Matrix uniforms not working"** - Verify your cables.gl version supports mat3/mat4 - Use Matrix ops to create matrix values - Consider using vec4 arrays or separate vec2/vec3 values if matrices aren't supported

**Issue: "Performance is poor"** - Reduce texture samples per pixel - Use mediump precision (already done) - Avoid branching in shaders when possible - Consider breaking into multiple passes - Check if using custom JavaScript ops (adds overhead)

**Issue: "Ports not appearing"** - Ensure uniform declarations match exactly (case-sensitive) - Check uniform types are supported - Verify shader compiles successfully - Try recompiling the shader in TextureEffect

## 6.19.4 Using Shaders in Cables.gl: Two Approaches

Cables.gl offers two ways to use custom shaders:

### Approach 1: Built-in Shader Ops (Recommended for Most Cases)

**ShaderMaterial** and **TextureEffect** ops automatically: - Create input ports for each uniform declaration - Provide varying vec2 vUV automatically - Handle shader compilation and execution on GPU - Require no JavaScript wrapper code

**How to Use:** 1. Add a TextureEffect op to your patch 2. Paste the shader code into the "Fragment Shader" field 3. Connect your textures and values to the automatically created ports 4. The shader runs directly on the GPU

**Auto-Provided Uniforms:** - varying vec2 vUV - Always available (0.0 to 1.0) - uniform float time - Available if you connect a Time op - uniform vec2 resolution - Available if you connect a Resolution/CanvasInfo op

**Manual Uniforms:** - All other uniform declarations become input ports automatically - Connect Texture ops for sampler2D uniforms - Connect Number/Vector ops for float, vec2, vec3, vec4 uniforms - Connect Matrix ops for mat3, mat4 uniforms (if supported)

### Example Patch Wiring for Keystone Correction:

```
ImageTexture -> TextureEffect (tex port)
CanvasInfo -> TextureEffect (resolution port)
Vector2 (topLeft) -> TextureEffect (topLeft port)
Vector2 (topRight) -> TextureEffect (topRight port)
Vector2 (bottomLeft) -> TextureEffect (bottomLeft port)
```

```
Vector2 (bottomRight) -> TextureEffect (bottomRight port)
```

## Approach 2: Custom JavaScript Ops (For Advanced Control)

JavaScript custom ops allow you to:

- Wrap shader code with additional logic
- Dynamically modify shader uniforms
- Create reusable, parameterized shader ops
- Add custom UI and port organization
- Handle complex texture management

**Trade-offs:**

- More setup required (JavaScript wrapper code)
- Potential JavaScript overhead
- More control over execution flow
- Better for reusable, packaged ops

See the “JavaScript Custom Op Examples” section below for implementation details.

## 6.19.5 Geometric Distortion Correction

Geometric distortion occurs when projectors are not perpendicular to the projection surface. Common types include keystone distortion, barrel distortion, and pincushion distortion.

### Keystone Correction (Perspective Distortion)

**Built-in Shader Op Ready** - Paste into TextureEffect

Keystone distortion creates a trapezoidal shape. This shader corrects it by applying inverse perspective transformation:

```
precision mediump float;

varying vec2 vUV;
uniform sampler2D tex;
uniform vec2 resolution;

// Keystone correction parameters
// topLeft, topRight, bottomLeft, bottomRight corners in UV space (0-1)
```

```
uniform vec2 topLeft;
uniform vec2 topRight;
uniform vec2 bottomLeft;
uniform vec2 bottomRight;

// Helper function: bilinear interpolation for perspective correction
vec2 perspectiveTransform(vec2 uv, vec2 tl, vec2 tr, vec2 bl, vec2 br) {
    // Convert UV to normalized coordinates (-1 to 1)
    vec2 nuv = (uv - 0.5) * 2.0;

    // Perspective correction using bilinear interpolation
    vec2 top = mix(tl, tr, uv.x);
    vec2 bottom = mix(bl, br, uv.x);
    vec2 corrected = mix(bottom, top, uv.y);

    return corrected;
}

void main() {
    vec2 correctedUV = perspectiveTransform(vUV, topLeft, topRight, bottomLeft, bottomRight);

    // Clamp to prevent sampling outside texture
    correctedUV = clamp(correctedUV, 0.0, 1.0);

    vec3 color = texture2D(tex, correctedUV).rgb;
    gl_FragColor = vec4(color, 1.0);
}
```

### Usage with TextureEffect (Built-in Shader Op):

1. Add a TextureEffect op to your patch
2. Paste the shader code above into the “Fragment Shader” field
3. Connect your inputs:
  - Input texture -> tex port (automatically created)
  - CanvasInfo op -> resolution port (or use GetResolution op)
  - Four Vector2 ops for corners -> topLeft, topRight, bottomLeft, bottomRight ports
4. The output texture will have keystone correction applied

**Note:** The resolution uniform is not automatically provided. You must connect a Resolution or CanvasInfo op to the resolution port.

## Advanced Keystone with Homography Matrix

For more precise control, use a 3x3 homography matrix:

**Note:** mat3 support may vary in cables.gl versions. Verify with Matrix ops or use the corner-based approach above if matrices aren't supported.

```
precision mediump float;

varying vec2 vUV;
uniform sampler2D tex;
uniform mat3 homographyMatrix; // 3x3 transformation matrix - verify Matrix op support in your cables.gl version

vec2 applyHomography(mat3 H, vec2 uv) {
    vec3 p = vec3(uv, 1.0);
    vec3 result = H * p;
    return result.xy / result.z;
}

void main() {
    vec2 correctedUV = applyHomography(homographyMatrix, vUV);

    // Check if point is within bounds
    if (correctedUV.x < 0.0 || correctedUV.x > 1.0 ||
        correctedUV.y < 0.0 || correctedUV.y > 1.0) {
        gl_FragColor = vec4(0.0, 0.0, 0.0, 1.0); // Black outside bounds
    } else {
        vec3 color = texture2D(tex, correctedUV).rgb;
        gl_FragColor = vec4(color, 1.0);
    }
}
```

## Barrel Distortion Correction

**Built-in Shader Op Ready** - Paste into TextureEffect

Barrel distortion creates a "bulging" effect. This shader corrects it:

```
precision mediump float;

varying vec2 vUV;
uniform sampler2D tex;
uniform vec2 resolution;
uniform float barrelStrength; // Typically -0.1 to -0.3 for correction

vec2 barrelDistortion(vec2 uv, float strength) {
    vec2 center = vec2(0.5, 0.5);
    vec2 coord = uv - center;
    float dist = length(coord);

    // Barrel distortion formula
    float factor = 1.0 + strength * dist * dist;
    vec2 corrected = center + coord * factor;

    return corrected;
}

void main() {
    vec2 correctedUV = barrelDistortion(vUV, barrelStrength);

    // Only sample if within bounds
    if (correctedUV.x < 0.0 || correctedUV.x > 1.0 ||
        correctedUV.y < 0.0 || correctedUV.y > 1.0) {
        gl_FragColor = vec4(0.0, 0.0, 0.0, 1.0);
    } else {
        vec3 color = texture2D(tex, correctedUV).rgb;
        gl_FragColor = vec4(color, 1.0);
    }
}
```

## Pincushion Distortion Correction

**Built-in Shader Op Ready** - Paste into TextureEffect

Pincushion distortion creates a "pinched" effect. This shader corrects it:

```

precision mediump float;

varying vec2 vUV;
uniform sampler2D tex;
uniform vec2 resolution;
uniform float pincushionStrength; // Typically 0.1 to 0.3 for correction

vec2 pincushionDistortion(vec2 uv, float strength) {
    vec2 center = vec2(0.5, 0.5);
    vec2 coord = uv - center;
    float dist = length(coord);

    // Pincushion distortion formula (opposite of barrel)
    float factor = 1.0 - strength * dist * dist;
    vec2 corrected = center + coord * factor;

    return corrected;
}

void main() {
    vec2 correctedUV = pincushionDistortion(vUV, pincushionStrength);

    if (correctedUV.x < 0.0 || correctedUV.x > 1.0 ||
        correctedUV.y < 0.0 || correctedUV.y > 1.0) {
        gl_FragColor = vec4(0.0, 0.0, 0.0, 1.0);
    } else {
        vec3 color = texture2D(tex, correctedUV).rgb;
        gl_FragColor = vec4(color, 1.0);
    }
}

```

## Combined Geometric Correction

A comprehensive shader combining multiple distortion types:

```
precision mediump float;
```

```

varying vec2 vUV;
uniform sampler2D tex;
uniform vec2 resolution;

// Keystone corners
uniform vec2 topLeft;
uniform vec2 topRight;
uniform vec2 bottomLeft;
uniform vec2 bottomRight;

// Distortion parameters
uniform float barrelAmount;
uniform float pincushionAmount;
uniform float rotation; // Rotation in radians

vec2 rotateUV(vec2 uv, float angle) {
    vec2 center = vec2(0.5, 0.5);
    vec2 coord = uv - center;
    float c = cos(angle);
    float s = sin(angle);
    mat2 rot = mat2(c, -s, s, c);
    return center + rot * coord;
}

vec2 applyDistortion(vec2 uv, float barrel, float pincushion) {
    vec2 center = vec2(0.5, 0.5);
    vec2 coord = uv - center;
    float dist = length(coord);

    float factor = 1.0 + (barrel + pincushion) * dist * dist;
    return center + coord * factor;
}

vec2 perspectiveTransform(vec2 uv, vec2 tl, vec2 tr, vec2 bl, vec2 br) {
    vec2 top = mix(tl, tr, uv.x);
    vec2 bottom = mix(bl, br, uv.x);
    return mix(bottom, top, uv.y);
}

```

```

void main() {
    vec2 uv = vUV;

    // Apply transformations in order: rotation -> distortion -> keystone
    uv = rotateUV(uv, rotation);
    uv = applyDistortion(uv, barrelAmount, pincushionAmount);
    uv = perspectiveTransform(uv, topLeft, topRight, bottomLeft, bottomRight);

    if (uv.x < 0.0 || uv.x > 1.0 || uv.y < 0.0 || uv.y > 1.0) {
        gl_FragColor = vec4(0.0, 0.0, 0.0, 1.0);
    } else {
        vec3 color = texture2D(tex, uv).rgb;
        gl_FragColor = vec4(color, 1.0);
    }
}

```

## 6.19.6 Multi-Projector Setups

When using multiple projectors, you need to define projection zones and blend overlapping areas.

### Projection Zone Mask

Define which projector covers which area:

```

precision mediump float;

varying vec2 vUV;
uniform sampler2D tex;
uniform vec2 resolution;

// Projection zone definition (in UV space, 0-1)
uniform vec4 zoneRect; // x, y, width, height of this projector's zone
uniform float feather; // Edge feathering amount

```

```

float getZoneMask(vec2 uv, vec4 zone) {
    vec2 zoneMin = zone.xy;
    vec2 zoneMax = zone.xy + zone.zw;

    // Distance to zone edges
    vec2 distToMin = uv - zoneMin;
    vec2 distToMax = zoneMax - uv;
    vec2 distToEdge = min(distToMin, distToMax);

    // Create mask with feathering
    float mask = 1.0;
    if (distToEdge.x < feather) {
        mask *= smoothstep(0.0, feather, distToEdge.x);
    }
    if (distToEdge.y < feather) {
        mask *= smoothstep(0.0, feather, distToEdge.y);
    }

    // Check if outside zone
    if (uv.x < zoneMin.x || uv.x > zoneMax.x ||
        uv.y < zoneMin.y || uv.y > zoneMax.y) {
        mask = 0.0;
    }

    return mask;
}

void main() {
    float mask = getZoneMask(vUV, zoneRect);
    vec3 color = texture2D(tex, vUV).rgb;

    gl_FragColor = vec4(color * mask, mask);
}

```

### Multi-Projector Blending

Blend multiple projector outputs with smooth transitions:

```

precision mediump float;

varying vec2 vUV;
uniform sampler2D tex;
uniform vec2 resolution;

// Blend zone definition
uniform vec4 blendZone; // x, y, width, height of blend area
uniform float blendWidth; // Width of blend transition
uniform float blendDirection; // 0.0=horizontal, 1.0=vertical, 2.0=both (1.5)
uniform float blendRadius; // Radius of blend transition (0.0=0.5)

float getBlendMask(vec2 uv, vec4 zone, float width, float direction) {
    vec2 zoneMin = zone.xy;
    vec2 zoneMax = zone.xy + zone.zw;
    vec2 zoneCenter = (zoneMin + zoneMax) * 0.5;

    float mask = 1.0;

    // Use float comparisons instead of int (cables.gl compatibility)
    if (direction < 0.5 || direction > 1.5) {
        // Horizontal blend (direction == 0.0 or 2.0)
        float distToCenter = abs(uv.x - zoneCenter.x);
        float zoneWidth = zone.z;
        if (distToCenter < zoneWidth * 0.5) {
            float blendDist = (zoneWidth * 0.5 - distToCenter) / width;
            mask *= smoothstep(0.0, 1.0, blendDist);
        }
    }

    if (direction > 0.5 && direction < 1.5 || direction > 1.5) {
        // Vertical blend (direction == 1.0 or 2.0)
        float distToCenter = abs(uv.y - zoneCenter.y);
        float zoneHeight = zone.w;
        if (distToCenter < zoneHeight * 0.5) {
            float blendDist = (zoneHeight * 0.5 - distToCenter) / width;
            mask *= smoothstep(0.0, 1.0, blendDist);
        }
    }
}

```

```

    return clamp(mask, 0.0, 1.0);
}

void main() {
    float blendMask = getBlendMask(vUV, blendZone, blendWidth, blendDirection);
    vec3 color = texture2D(tex, vUV).rgb;

    gl_FragColor = vec4(color * blendMask, blendMask);
}

```

## 6.19.7 Projector Stacking

Projector stacking involves overlapping multiple projectors to increase brightness and redundancy. This shader combines multiple inputs:

```

precision mediump float;

varying vec2 vUV;
uniform sampler2D tex1; // First projector
uniform sampler2D tex2; // Second projector
uniform sampler2D tex3; // Optional third projector
uniform sampler2D tex4; // Optional fourth projector

uniform float stackCount; // Number of active projectors (1-4)
uniform float blendMode; // 0=additive, 1=average, 2=max

vec3 blendStacked(vec3 c1, vec3 c2, vec3 c3, vec3 c4, float count, float mode)
{
    vec3 result = vec3(0.0);

    if (mode < 0.5) {
        // Additive blending (brightest, but can clip)
        if (count > 0.5) result += c1;
        if (count > 1.5) result += c2;
        if (count > 2.5) result += c3;
        if (count > 3.5) result += c4;
        result = clamp(result, 0.0, 1.0);
    }
}

```

```

} else if (mode < 1.5) {
    // Average blending (natural, reduces brightness)
    float sum = 0.0;
    if (count > 0.5) { result += c1; sum += 1.0; }
    if (count > 1.5) { result += c2; sum += 1.0; }
    if (count > 2.5) { result += c3; sum += 1.0; }
    if (count > 3.5) { result += c4; sum += 1.0; }
    result /= max(sum, 1.0);
} else {
    // Maximum blending (preserves highlights)
    result = c1;
    if (count > 1.5) result = max(result, c2);
    if (count > 2.5) result = max(result, c3);
    if (count > 3.5) result = max(result, c4);
}

return result;
}

void main() {
    vec3 c1 = texture2D(tex1, vUV).rgb;
    vec3 c2 = texture2D(tex2, vUV).rgb;
    vec3 c3 = texture2D(tex3, vUV).rgb;
    vec3 c4 = texture2D(tex4, vUV).rgb;

    vec3 result = blendStacked(c1, c2, c3, c4, stackCount, blendMode);

    gl_FragColor = vec4(result, 1.0);
}

```

## 6.19.8 Gradient Blend Composition

Gradient blends create smooth transitions between overlapping projectors. This is essential for seamless multi-projector setups.

### Linear Gradient Blend

```

precision mediump float;

varying vec2 vUV;
uniform sampler2D tex;
uniform vec2 resolution;

// Blend parameters
uniform float blendStart; // Where blend starts (0-1)
uniform float blendEnd; // Where blend ends (0-1)
uniform float blendAxis; // 0.0=horizontal, 1.0=vertical (use float instead of int)
uniform float blendPower; // Blend curve (1.0=linear, 2.0=smooth)

float getLinearBlend(vec2 uv, float start, float end, float axis, float power)
{
    float pos = axis < 0.5 ? uv.x : uv.y; // Use float comparison

    // Calculate blend factor
    float blendFactor = 0.0;
    if (pos < start) {
        blendFactor = 0.0;
    } else if (pos > end) {
        blendFactor = 1.0;
    } else {
        // Normalize to 0-1 range
        float t = (pos - start) / (end - start);
        // Apply power curve
        blendFactor = pow(t, power);
    }

    return blendFactor;
}

void main() {
    float blend = getLinearBlend(vUV, blendStart, blendEnd, blendAxis, blendPower);
    vec3 color = texture2D(tex, vUV).rgb;

    gl_FragColor = vec4(color * blend, blend);
}

```

## Radial Gradient Blend

For circular or elliptical blend zones:

```
precision mediump float;

varying vec2 vUV;
uniform sampler2D tex;
uniform vec2 resolution;

// Radial blend parameters
uniform vec2 center; // Blend center in UV space
uniform float innerRadius; // Inner radius (full opacity)
uniform float outerRadius; // Outer radius (zero opacity)
uniform float aspectRatio; // Aspect ratio correction
uniform float blendPower; // Blend curve

float getRadialBlend(vec2 uv, vec2 center, float innerR, float outerR, float aspect, float power) {
    vec2 offset = (uv - center) * vec2(aspect, 1.0);
    float dist = length(offset);

    float blendFactor = 0.0;
    if (dist < innerR) {
        blendFactor = 1.0;
    } else if (dist > outerR) {
        blendFactor = 0.0;
    } else {
        float t = (dist - innerR) / (outerR - innerR);
        blendFactor = 1.0 - pow(t, power);
    }

    return blendFactor;
}

void main() {
    float blend = getRadialBlend(vUV, center, innerRadius, outerRadius, aspectRatio, blendPower);
    vec3 color = texture2D(tex, vUV).rgb;
    gl_FragColor = vec4(color * blend, blend);
}
```

## Advanced Feather Blend with Soft Edges

Professional-grade blend with multiple falloff curves:

```
precision mediump float;

varying vec2 vUV;
uniform sampler2D tex;
uniform vec2 resolution;

uniform vec4 blendRect; // x, y, width, height
uniform float featherSize; // Feather size in UV units
uniform float featherCurve; // 0.0=linear, 1.0=smooth, 2.0=very smooth

float getFeatherBlend(vec2 uv, vec4 rect, float feather, float curve) {
    vec2 rectMin = rect.xy;
    vec2 rectMax = rect.xy + rect.zw;

    // Calculate distance to each edge
    float distLeft = uv.x - rectMin.x;
    float distRight = rectMax.x - uv.x;
    float distBottom = uv.y - rectMin.y;
    float distTop = rectMax.y - uv.y;

    // Find minimum distance to any edge
    float minDist = min(min(distLeft, distRight), min(distBottom, distTop));

    // Create feather mask
    float mask = 1.0;
    if (minDist < feather) {
        float t = minDist / feather;
        // Apply curve
        if (curve < 0.5) {
            // Linear
```

```

        mask = t;
    } else if (curve < 1.5) {
        // Smoothstep
        mask = smoothstep(0.0, 1.0, t);
    } else {
        // Custom smooth curve
        mask = t * t * (3.0 - 2.0 * t);
        mask = pow(mask, 1.0 / (curve - 0.5));
    }
}

// Check if outside rectangle
if (uv.x < rectMin.x || uv.x > rectMax.x ||
    uv.y < rectMin.y || uv.y > rectMax.y) {
    mask = 0.0;
}

return mask;
}

void main() {
    float blend = getFeatherBlend(vUV, blendRect, featherSize, featherCurve);
    vec3 color = texture2D(tex, vUV).rgb;

    gl_FragColor = vec4(color * blend, blend);
}

```

## 6.19.9 Color Correction for Projection Mapping

Projection mapping requires specialized color correction to account for surface colors, ambient light, and projector characteristics.

### Basic Color Correction

**Built-in Shader Op Ready** - Paste into TextureEffect

```

precision mediump float;

varying vec2 vUV;
uniform sampler2D tex;
uniform vec2 resolution;

// Color correction parameters
uniform float brightness; // -1.0 to 1.0
uniform float contrast; // -1.0 to 1.0
uniform float saturation; // -1.0 to 1.0
uniform float gamma; // Typically 0.5 to 3.0

vec3 applyColorCorrection(vec3 color, float bright, float cont, float sat, float gamma) {
    // Brightness
    color += bright;

    // Contrast
    color = (color - 0.5) * (1.0 + cont) + 0.5;

    // Saturation
    float luma = dot(color, vec3(0.299, 0.587, 0.114));
    color = mix(vec3(luma), color, 1.0 + sat);

    // Gamma
    color = pow(max(color, 0.0), vec3(1.0 / max(gamma, 0.01)));

    return clamp(color, 0.0, 1.0);
}

void main() {
    vec3 color = texture2D(tex, vUV).rgb;
    color = applyColorCorrection(color, brightness, contrast, saturation, gamma);

    gl_FragColor = vec4(color, 1.0);
}

```

## Advanced Color Correction with Color Temperature

```

precision mediump float;

varying vec2 vUV;
uniform sampler2D tex;
uniform vec2 resolution;

uniform float brightness;
uniform float contrast;
uniform float saturation;
uniform float gamma;
uniform float colorTemperature; // - 1.0 (cool/blue) to 1.0 (warm/orange)

// Color temperature adjustment
vec3 adjustColorTemperature(vec3 color, float temp) {
    // Convert to warmer (orange) or cooler (blue)
    if (temp > 0.0) {
        // Warmer: increase red/orange, decrease blue
        color.r += temp * 0.2;
        color.b -= temp * 0.1;
    } else {
        // Cooler: increase blue, decrease red
        color.r += temp * 0.1;
        color.b -= temp * 0.2;
    }
    return color;
}

vec3 applyColorCorrection(vec3 color, float bright, float cont, float sat,
    // Brightness
    color += bright;

    // Contrast
    color = (color - 0.5) * (1.0 + cont) + 0.5;

    // Saturation
    float luma = dot(color, vec3(0.299, 0.587, 0.114));
    color = mix(vec3(luma), color, 1.0 + sat);
}

```

```

    // Color temperature
    color = adjustColorTemperature(color, temp);

    // Gamma
    color = pow(max(color, 0.0), vec3(1.0 / max(gam, 0.01)));

    return clamp(color, 0.0, 1.0);
}

void main() {
    vec3 color = texture2D(tex, vUV).rgb;
    color = applyColorCorrection(color, brightness, contrast, saturation, gamma);

    gl_FragColor = vec4(color, 1.0);
}

```

## Per-Channel Color Correction

Independent control over RGB channels:

```

precision mediump float;

varying vec2 vUV;
uniform sampler2D tex;
uniform vec2 resolution;

// Per-channel brightness and contrast
uniform vec3 channelBrightness; // R, G, B
uniform vec3 channelContrast; // R, G, B
uniform vec3 channelGamma; // R, G, B

vec3 applyPerChannelCorrection(vec3 color, vec3 bright, vec3 cont, vec3 gam)
    // Apply per-channel brightness
    color += bright;

    // Apply per-channel contrast

```

```

color = (color - 0.5) * (1.0 + cont) + 0.5;

// Apply per-channel gamma
color = pow(max(color, 0.0), vec3(1.0 / max(gam, vec3(0.01))));

return clamp(color, 0.0, 1.0);
}

void main() {
    vec3 color = texture2D(tex, vUV).rgb;
    color = applyPerChannelCorrection(color, channelBrightness, channelContrast, channelGamma);

    gl_FragColor = vec4(color, 1.0);
}

```

## Surface Color Compensation

Compensate for colored projection surfaces (e.g., projecting on a red wall):

```

precision mediump float;

varying vec2 vUV;
uniform sampler2D tex;
uniform vec2 resolution;

// Surface color (what color the surface appears)
uniform vec3 surfaceColor;
uniform float compensationStrength; // 0.0 to 1.0

vec3 compensateSurfaceColor(vec3 color, vec3 surface, float strength) {
    // Calculate inverse of surface color
    vec3 inverseSurface = vec3(1.0) - surface;

    // Blend between original and compensated
    vec3 compensated = color / max(surface, vec3(0.01)); // Prevent division by zero
    compensated = clamp(compensated, 0.0, 1.0);
}

```

```

        return mix(color, compensated, strength);
    }

    void main() {
        vec3 color = texture2D(tex, vUV).rgb;
        color = compensateSurfaceColor(color, surfaceColor, compensationStrength);

        gl_FragColor = vec4(color, 1.0);
    }

```

## Advanced LUT-Based Color Correction

Use a 3D Look-Up Table (LUT) for professional color grading:

```

precision mediump float;

varying vec2 vUV;
uniform sampler2D tex;
uniform sampler2D lutTexture; // 3D LUT as 2D texture (typically 64x64 or 32x32)
uniform vec2 resolution;
uniform float lutStrength; // 0.0 to 1.0

// Sample 3D LUT (stored as 2D texture)
vec3 sampleLUT(sampler2D lut, vec3 color, float lutSize) {
    // Assume LUT is organized as a grid
    // For a 64x64 LUT, we have 8x8 grid of 8x8 color cubes

    float cellSize = 1.0 / 8.0; // 8x8 grid
    float cellPixelSize = 1.0 / 64.0; // 64 pixels per cell

    // Find which cell we're in
    vec3 cell = floor(color * 7.0);
    vec3 cellPos = fract(color * 7.0);

    // Calculate UV coordinates in LUT texture
    float cellIndex = cell.b * 8.0 + cell.r;
    vec2 lutUV = vec2(

```

```

        (cellIndex * cellSize) + (cellPos.r * cellPixelSize * 8.0),
        cell.g * cellSize + cellPos.g * cellPixelSize * 8.0
    );

    // Sample LUT
    vec3 lutColor = texture2D(lut, lutUV).rgb;

    return lutColor;
}

void main() {
    vec3 originalColor = texture2D(tex, vUV).rgb;
    vec3 lutColor = sampleLUT(lutTexture, originalColor, 64.0);

    vec3 finalColor = mix(originalColor, lutColor, lutStrength);

    gl_FragColor = vec4(finalColor, 1.0);
}

```

**Note:** For LUT textures, you'll need to create or load a 3D LUT texture. Common formats include 64x64 (8x8 grid) or 32x32 (4x4 grid) textures.

## Shadow and Highlight Recovery

Recover details in shadows and highlights:

```

precision mediump float;

varying vec2 vUV;
uniform sampler2D tex;
uniform vec2 resolution;

uniform float shadowRecovery; // 0.0 to 1.0
uniform float highlightRecovery; // 0.0 to 1.0
uniform float shadowPoint; // Where shadows start (0.0 to 1.0)
uniform float highlightPoint; // Where highlights start (0.0 to 1.0)

vec3 recoverShadowsHighlights(vec3 color, float shadowRec, float highlightRec)
{
    float luma = dot(color, vec3(0.299, 0.587, 0.114));

    // Shadow recovery
    float shadowMask = smoothstep(shadowPt - 0.1, shadowPt, luma);
    color += shadowMask * shadowRec * (1.0 - luma) * 0.5;

    // Highlight recovery (compress highlights)
    float highlightMask = smoothstep(highlightPt, highlightPt + 0.1, luma);
    color = mix(color, vec3(1.0) - (vec3(1.0) - color) * (1.0 - highlightRec), highlightMask);

    return clamp(color, 0.0, 1.0);
}

```

```

void main() {
    vec3 color = texture2D(tex, vUV).rgb;
    color = recoverShadowsHighlights(color, shadowRecovery, highlightRecovery);

    gl_FragColor = vec4(color, 1.0);
}

```

## 6.19.10 Complete Projection Mapping Pipeline

**Built-in Shader Op Ready** - Paste into TextureEffect (Note: This is a complex shader with many uniforms - consider breaking into multiple passes for easier management)

A comprehensive shader combining all projection mapping features:

```

precision mediump float;

varying vec2 vUV;
uniform sampler2D tex;
uniform vec2 resolution;

// Geometric correction
uniform vec2 topLeft;

```

```

uniform vec2 topRight;
uniform vec2 bottomLeft;
uniform vec2 bottomRight;
uniform float barrelAmount;
uniform float rotation;

// Blend parameters
uniform vec4 blendZone;
uniform float blendWidth;
uniform float blendPower;

// Color correction
uniform float brightness;
uniform float contrast;
uniform float saturation;
uniform float gamma;
uniform float colorTemperature;
uniform vec3 surfaceColor;
uniform float surfaceCompensation;

// Helper functions (include all from above)
vec2 perspectiveTransform(vec2 uv, vec2 tl, vec2 tr, vec2 bl, vec2 br) {
    vec2 top = mix(tl, tr, uv.x);
    vec2 bottom = mix(bl, br, uv.x);
    return mix(bottom, top, uv.y);
}

vec2 applyDistortion(vec2 uv, float barrel) {
    vec2 center = vec2(0.5, 0.5);
    vec2 coord = uv - center;
    float dist = length(coord);
    float factor = 1.0 + barrel * dist * dist;
    return center + coord * factor;
}

vec2 rotateUV(vec2 uv, float angle) {
    vec2 center = vec2(0.5, 0.5);
    vec2 coord = uv - center;
    float c = cos(angle);
}

```

```

float s = sin(angle);
mat2 rot = mat2(c, -s, s, c);
return center + rot * coord;
}

float getBlendMask(vec2 uv, vec4 zone, float width, float power) {
    vec2 zoneMin = zone.xy;
    vec2 zoneMax = zone.xy + zone.zw;
    vec2 zoneCenter = (zoneMin + zoneMax) * 0.5;

    float distToCenter = length(uv - zoneCenter);
    float maxDist = length(zoneMax - zoneCenter);

    if (distToCenter > maxDist) return 0.0;

    float blendDist = (maxDist - distToCenter) / width;
    return pow(clamp(blendDist, 0.0, 1.0), power);
}

vec3 applyColorCorrection(vec3 color, float bright, float cont, float sat, fl
    color += bright;
    color = (color - 0.5) * (1.0 + cont) + 0.5;

    float luma = dot(color, vec3(0.299, 0.587, 0.114));
    color = mix(vec3(luma), color, 1.0 + sat);

    if (temp > 0.0) {
        color.r += temp * 0.2;
        color.b -= temp * 0.1;
    } else {
        color.r += temp * 0.1;
        color.b -= temp * 0.2;
    }

    vec3 compensated = color / max(surface, vec3(0.01));
    color = mix(color, clamp(compensated, 0.0, 1.0), comp);

    color = pow(max(color, 0.0), vec3(1.0 / max(gam, 0.01)));
}

```

```

        return clamp(color, 0.0, 1.0);
    }

void main() {
    // Step 1: Geometric correction
    vec2 uv = vUV;
    uv = rotateUV(uv, rotation);
    uv = applyDistortion(uv, barrelAmount);
    uv = perspectiveTransform(uv, topLeft, topRight, bottomLeft, bottomRight);

    // Step 2: Sample texture
    if (uv.x < 0.0 || uv.x > 1.0 || uv.y < 0.0 || uv.y > 1.0) {
        gl_FragColor = vec4(0.0, 0.0, 0.0, 0.0);
        return;
    }

    vec3 color = texture2D(tex, uv).rgb;

    // Step 3: Color correction
    color = applyColorCorrection(color, brightness, contrast, saturation, gamma);

    // Step 4: Apply blend mask
    float blend = getBlendMask(vUV, blendZone, blendWidth, blendPower);
    color *= blend;

    gl_FragColor = vec4(color, blend);
}

```

## 6.19.11 JavaScript Custom Op Examples

For cases where you need more control, reusable components, or dynamic shader management, you can wrap shaders in JavaScript custom ops. Here are examples for key projection mapping features:

### Keystone Correction Custom Op

```
// Custom Op: KeystoneCorrection
```

```

// Name: Ops.User.ProjectionMapping.KeystoneCorrection

const inTexture = op.inTexture("Input Texture");
const inTopLeft = op.inVec2("Top Left", [0.0, 1.0]);
const inTopRight = op.inVec2("Top Right", [1.0, 1.0]);
const inBottomLeft = op.inVec2("Bottom Left", [0.0, 0.0]);
const inBottomRight = op.inVec2("Bottom Right", [1.0, 0.0]);
const inResolution = op.inVec2("Resolution", [1920.0, 1080.0]);
const outTexture = op.outTexture("Output");

// Shader code as string
const shaderCode = `

precision mediump float;
varying vec2 vUV;
uniform sampler2D tex;
uniform vec2 resolution;
uniform vec2 topLeft;
uniform vec2 topRight;
uniform vec2 bottomLeft;
uniform vec2 bottomRight;

vec2 perspectiveTransform(vec2 uv, vec2 tl, vec2 tr, vec2 bl, vec2 br) {
    vec2 top = mix(tl, tr, uv.x);
    vec2 bottom = mix(bl, br, uv.x);
    return mix(bottom, top, uv.y);
}

void main() {
    vec2 correctedUV = perspectiveTransform(vUV, topLeft, topRight, bottomLeft, bottomRight);
    correctedUV = clamp(correctedUV, 0.0, 1.0);
    vec3 color = texture2D(tex, correctedUV).rgb;
    gl_FragColor = vec4(color, 1.0);
}

let shaderMaterial = null;

function updateShader() {
    const tex = inTexture.get();

```

```

if (!tex) return;

// Create or update shader material
if (!shaderMaterial) {
    shaderMaterial = new op.patch.cgl.ShaderMaterial({
        fragmentShader: shaderCode,
        uniforms: {}
    });
}

// Update uniforms
shaderMaterial.uniforms.tex = { value: tex };
shaderMaterial.uniforms.resolution = { value: inResolution.get() };
shaderMaterial.uniforms.topLeft = { value: inTopLeft.get() };
shaderMaterial.uniforms.topRight = { value: inTopRight.get() };
shaderMaterial.uniforms.bottomLeft = { value: inBottomLeft.get() };
shaderMaterial.uniforms.bottomRight = { value: inBottomRight.get() };

// Render to texture
const renderTarget = op.patch.cgl.createRenderTarget(
    inResolution.get()[0],
    inResolution.get()[1]
);

// Apply shader and render
op.patch.cgl.render(renderTarget, shaderMaterial);

outTexture.set(renderTarget.texture);
}

inTexture.onChange = updateShader;
inTopLeft.onChange = updateShader;
inTopRight.onChange = updateShader;
inBottomLeft.onChange = updateShader;
inBottomRight.onChange = updateShader;
inResolution.onChange = updateShader;

```

**Note:** The above example shows the concept, but cables.gl's actual API may differ. In practice, you might use TextureEffect programmatically or create a render pass.

## Color Correction Custom Op

```

// Custom Op: ColorCorrection
// Name: Ops.User.ProjectionMapping.ColorCorrection

const inTexture = op.inTexture("Input Texture");
const inBrightness = op.inFloat("Brightness", 0.0);
const inContrast = op.inFloat("Contrast", 0.0);
const inSaturation = op.inFloat("Saturation", 0.0);
const inGamma = op.inFloat("Gamma", 1.0);
const inColorTemperature = op.inFloat("Color Temperature", 0.0);
const outTexture = op.outTexture("Output");

const shaderCode = `
precision mediump float;
varying vec2 vUV;
uniform sampler2D tex;
uniform float brightness;
uniform float contrast;
uniform float saturation;
uniform float gamma;
uniform float colorTemperature;

vec3 adjustColorTemperature(vec3 color, float temp) {
    if (temp > 0.0) {
        color.r += temp * 0.2;
        color.b -= temp * 0.1;
    } else {
        color.r += temp * 0.1;
        color.b -= temp * 0.2;
    }
    return color;
}

vec3 applyColorCorrection(vec3 color, float bright, float cont, float sat, fl
    color += bright;
    color = (color - 0.5) * (1.0 + cont) + 0.5;

    float luma = dot(color, vec3(0.299, 0.587, 0.114));

```

```

color = mix(vec3(luma), color, 1.0 + sat);

color = adjustColorTemperature(color, temp);
color = pow(max(color, 0.0), vec3(1.0 / max(gam, 0.01)));

return clamp(color, 0.0, 1.0);
}

void main() {
    vec3 color = texture2D(tex, vUV).rgb;
    color = applyColorCorrection(color, brightness, contrast, saturation, gamma);
    gl_FragColor = vec4(color, 1.0);
}
`;

// Implementation similar to keystone op above
// (Actual implementation depends on cables.gl's rendering API)

```

## Blend Composition Custom Op

```

// Custom Op: BlendComposition
// Name: Ops.User.ProjectionMapping.BlendComposition

const inTexture = op.inTexture("Input Texture");
const inBlendStart = op.inFloat("Blend Start", 0.0);
const inBlendEnd = op.inFloat("Blend End", 1.0);
const inBlendAxis = op.inFloat("Blend Axis", 0.0); // 0.0=horizontal, 1.0=vertical
const inBlendPower = op.inFloat("Blend Power", 1.0);
const inResolution = op.inVec2("Resolution", [1920.0, 1080.0]);
const outTexture = op.outTexture("Output");
const outAlpha = op.outNumber("Alpha Mask"); // For compositing

const shaderCode = `precision mediump float;
varying vec2 vUV;
uniform sampler2D tex;
uniform vec2 resolution;
uniform float blendStart;

```

```

uniform float blendEnd;
uniform float blendAxis;
uniform float blendPower;

float getLinearBlend(vec2 uv, float start, float end, float axis, float power) {
    float pos = axis < 0.5 ? uv.x : uv.y;
    float blendFactor = 0.0;

    if (pos < start) {
        blendFactor = 0.0;
    } else if (pos > end) {
        blendFactor = 1.0;
    } else {
        float t = (pos - start) / (end - start);
        blendFactor = pow(t, power);
    }

    return blendFactor;
}

void main() {
    float blend = getLinearBlend(vUV, blendStart, blendEnd, blendAxis, blendPower);
    vec3 color = texture2D(tex, vUV).rgb;
    gl_FragColor = vec4(color * blend, blend);
}
`;

// Implementation with uniform updates
// Note: This is a conceptual example - actual cables.gl API may vary

```

## Important Notes for JavaScript Custom Ops:

- Texture Handling:** You need to manage texture creation, rendering, and cleanup
- Render Targets:** May need to create render targets for shader output
- Performance:** JavaScript overhead can impact real-time performance
- API Differences:** Cables.gl's internal API may differ from these examples
- Best Practice:** Use built-in TextureEffect when possible; use custom ops for complex logic or reusable components

## 6.19.12 Comparison: Built-in Shader Ops vs Custom JavaScript Ops

### Code Cleanliness

**Built-in Shader Ops (TextureEffect/ShaderMaterial):** - Pure GLSL code - no wrapper needed - Minimal boilerplate - Easy to read and maintain - Direct shader editing in cables.gl UI - No JavaScript knowledge required

**Custom JavaScript Ops:** - [!] Requires JavaScript wrapper code - [!] Shader code stored as string (less readable) - [!] More complex file structure - [!] Requires understanding of both GLSL and JavaScript - Can organize shader code in separate files - Can add pre/post processing logic

**Winner:** Built-in Shader Ops - cleaner, more maintainable for pure shader effects

### Integration Ease

**Built-in Shader Ops:** - Paste shader code directly into TextureEffect - Ports created automatically from uniforms - Immediate visual feedback - No compilation step - Works out of the box - [!] Limited customization of port UI - [!] Can't add custom logic around shader

**Custom JavaScript Ops:** - [!] Must create op, write wrapper code - [!] Must manually create and configure ports - [!] More setup time - [!] Requires testing and debugging - Full control over port organization - Can add port groups, custom UI - Can add validation, error handling - Reusable across patches

**Winner:** Built-in Shader Ops - significantly easier to get started

### Performance

**Built-in Shader Ops:** - Direct GPU execution - Minimal overhead - Optimized by cables.gl - No JavaScript execution per frame - Efficient texture passing - Automatic shader compilation caching

**Custom JavaScript Ops:** - [!] Potential JavaScript overhead per frame - [!] Texture copying may be required - [!] Render target management overhead - [!] Uniform updates in JavaScript (CPU work) - Can optimize with dirty flags - Can batch operations - Can cache render targets

**Performance Comparison:** - Built-in: ~0.1-0.5ms overhead (shader execution only) - Custom: ~1-5ms overhead (JavaScript + shader execution) - **Winner:** Built-in Shader Ops - better performance for real-time applications

### When to Use Each Approach

**Use Built-in Shader Ops (TextureEffect/ShaderMaterial) when:** - You have pure shader effects (no complex logic) - You want quick prototyping - Performance is critical - You're learning shaders - You need immediate visual feedback - You don't need custom port organization

**Use Custom JavaScript Ops when:** - You need reusable, packaged shader components - You need complex pre/post processing logic - You need dynamic shader generation - You want custom port UI and organization - You're building a library of shader ops - You need conditional shader selection - You need to manage multiple render passes

**Hybrid Approach:** - Use built-in shader ops for individual effects - Use custom JavaScript ops to orchestrate multiple shader passes - Use custom ops for complex parameter management - Use built-in ops for simple, one-off effects

## 6.19.13 Quick Reference: Using These Shaders

### Step-by-Step Guide:

#### 1. Add TextureEffect Op:

- Click "+" in your patch
- Search for "TextureEffect"
- Add it to your patch

#### 2. Paste Shader Code:

- Click on the TextureEffect op
- Find the "Fragment Shader" field
- Paste the shader code (including precision mediump float; and varying vec2 vUV;)

#### 3. Connect Inputs:

- Input texture -> tex port (or tex0, tex1, etc. for multi-texture shaders)
- CanvasInfo or GetResolution -> resolution port (if shader uses it)
- Number/Vector ops -> parameter ports (brightness, contrast, corners, etc.)

#### 4. Get Output:

- Connect TextureEffect output to your render target or next effect

**Common Port Types:** - sampler2D tex -> Texture port (connect ImageTexture, VideoTexture, etc.) - vec2 resolution -> Vec2 port (connect CanvasInfo or GetResolution) - float brightness -> Number port (connect Number op or slider) - vec2 topLeft -> Vec2 port (connect Vector2 op) - vec3 color -> Vec3 port (connect Vector3 op or Color op)

## 6.19.14 Best Practices for Projection Mapping in Cables.gl

1. **Resolution Handling:** Always use resolution uniform for pixel-perfect calculations. Convert between UV space and screen space as needed. **Remember:** resolution is NOT auto-provided - connect it manually.

2. **Performance:** Projection mapping shaders can be expensive. Consider:

- Using lower precision where possible (mediump instead of highp)
- Minimizing texture samples
- Pre-computing values in JavaScript ops when possible

3. **Modular Approach:** Break complex setups into multiple shader passes:

- First pass: Geometric correction
- Second pass: Color correction
- Third pass: Blending

4. **Testing:** Always test with actual projection surfaces when possible. Screen simulation can differ from real-world results.

5. **Calibration:** Use test patterns (grids, color bars) to calibrate geometric and color corrections.

6. **Masking:** Use alpha channel output for blend masks to composite multiple projectors correctly.

## 6.19.15 Debug Visualization Shaders

Helpful shaders for debugging projection mapping setups:

### Grid Overlay

```
precision mediump float;

varying vec2 vUV;
uniform vec2 resolution;
uniform float gridSize; // Grid divisions
uniform vec3 gridColor;
uniform float gridOpacity;

void main() {
    vec2 gridUV = vUV * gridSize;
    // Use manual derivative calculation instead of fwidth() for better WebGL 1 compatibility
    vec2 grid = abs(fract(gridUV - 0.5) - 0.5);
    // Approximate derivative using step function
    float line = min(grid.x, grid.y) * gridSize * 100.0; // Scale factor for visibility
    float gridMask = 1.0 - min(line, 1.0);

    vec3 color = mix(vec3(0.0), gridColor, gridMask * gridOpacity);
    gl_FragColor = vec4(color, 1.0);
}
```

### Corner Pin Visualization

```
precision mediump float;

varying vec2 vUV;
uniform vec2 resolution;
uniform vec2 topLeft;
uniform vec2 topRight;
uniform vec2 bottomLeft;
uniform vec2 bottomRight;
uniform vec3 cornerColor;

void main() {
    vec3 color = vec3(0.0);

    // Draw corner points
    float cornerSize = 0.02;
```

```

float dist1 = length(vUV - topLeft);
float dist2 = length(vUV - topRight);
float dist3 = length(vUV - bottomLeft);
float dist4 = length(vUV - bottomRight);

float minDist = min(min(dist1, dist2), min(dist3, dist4));
if (minDist < cornerSize) {
    color = cornerColor;
}

// Draw lines between corners
// (Simplified - you'd use line SDF for proper lines)

gl_FragColor = vec4(color, 1.0);
}

```

## 6.19.16 Summary: Shader Compliance and Usage

All shaders in this projection mapping section are:

**Compliant with cables.gl's built-in shader ops** (TextureEffect/ShaderMaterial) **Ready to paste directly** into the fragment shader field **WebGL 1.0 compatible** (using texture2D(), mediump precision) **Properly formatted** with required headers and declarations **Uniform types verified** (float instead of int, proper vector types)

**Key Compliance Features:** - All shaders start with precision mediump float;  
 - All use texture2D() for texture sampling - All use varying vec2 vUV (auto-provided by cables.gl) - Integer uniforms converted to float with float comparisons - Resolution handling documented (requires manual connection) - Matrix uniforms noted with version compatibility warnings

**Usage Pattern:** 1. Copy shader code 2. Paste into TextureEffect op's fragment shader field 3. Connect inputs to automatically created ports 4. Get output texture

**For Advanced Use Cases:** - See "JavaScript Custom Op Examples" section for wrapper implementations - See "Comparison" section for when to use each approach - See "Troubleshooting" section for common issues

## 6.20 Featured Videos



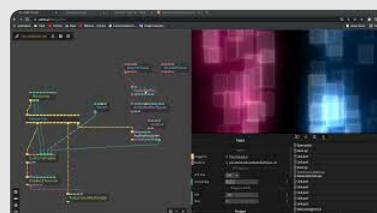
**Coding with cables**  
**Custom shader op**

<https://youtu.be/Zfhn8xSM0SE>  
**Coding with cables - custom shader op**  
 by cables\_gl



**TUTORIAL**  
**Shadertoy to cables - 01**

[https://youtu.be/j\\_ins4RW0c8](https://youtu.be/j_ins4RW0c8)  
**Shadertoy to cables - part 01**  
 by cables\_gl



<https://youtu.be/nil-HkZgNZ8>  
**Programmation d'un shadertoy avec Cables.gl Partie 8.**  
 by Meletou1

## 6.21 Resources

- The Book of Shaders - Excellent GLSL learning resource

- Shadertoy - Shader examples and inspiration
- GLSL Sandbox - More shader experiments

## 6.22 Exercises

1. Create a animated gradient that shifts colors over time
2. Build a kaleidoscope effect using UV manipulation
3. Write an SDF shader that draws a morphing shape
4. Create a post-processing glow effect
5. **Projection Mapping:** Implement keystone correction for a trapezoidal projection
6. **Projection Mapping:** Create a multi-projector blend setup with gradient transitions
7. **Projection Mapping:** Build a color correction shader that compensates for a colored projection surface
8. **Projection Mapping:** Combine geometric correction, color correction, and blending in a single shader pipeline
9. **Projection Mapping:** Create a debug visualization shader showing projection zones and blend areas
10. **Projection Mapping:** Implement projector stacking with additive and average blend modes

# 7 JavaScript & Custom Ops in Cables.gl

## 7.1 Introduction

While cables.gl's visual node system is powerful, sometimes you need custom functionality. JavaScript allows you to create your own operators (ops) and extend cables.gl's capabilities.

## 7.2 When to Use Custom Ops

- Processing data in ways built-in ops don't support
- Integrating external APIs or libraries
- Creating reusable custom functionality
- Performance optimization for specific tasks
- Complex mathematical operations

## 7.3 Creating Your First Op

### 7.3.1 Step 1: Open the Op Editor

1. In your patch, click the "+" button
2. Select "Create Op"
3. Choose a name (e.g., Ops.User.YourName.MyFirstOp)

### 7.3.2 Step 2: Understanding the Structure

```
// Ports (inputs and outputs)
const inValue = op.inFloat("Input Value", 0);
const outResult = op.outNumber("Result");

// When input changes, recalculate
inValue.onChange = function() {
    outResult.set(inValue.get() * 2);
};
```

## 7.4 Port Types

## 7.4.1 Input Ports

```
// Trigger (execution flow)
const inTrigger = op.inTrigger("Trigger");

// Numbers
const inFloat = op.inFloat("Float Value", 0.0);
const inInt = op.inInt("Integer", 0);
const inValue = op.inValue("Value", 0);

// Boolean
const inBool = op.inBool("Enabled", true);

// String
const inString = op.inString("Text", "default");

// Objects (textures, arrays, etc.)
const inObject = op.inObject("Object");
const inTexture = op.inTexture("Texture");
const inArray = op.inArray("Array");
```

## 7.4.2 Output Ports

```
// Trigger
const outTrigger = op.outTrigger("Trigger Out");

// Numbers
const outNumber = op.outNumber("Number Out");
const outValue = op.outValue("Value Out");

// Boolean
const outBool = op.outBool("Bool Out");

// String
const outString = op.outString("String Out");

// Objects
const outObject = op.outObject("Object Out");
```

```
const outTexture = op.outTexture("Texture Out");
const outArray = op.outArray("Array Out");
```

## 7.5 Handling Events

### 7.5.1 Trigger Execution

```
const inTrigger = op.inTrigger("Execute");
const outNext = op.outTrigger("Next");

inTrigger.onTriggered = function() {
    // Do something when triggered
    console.log("Op was triggered!");

    // Continue the chain
    outNext.trigger();
};
```

### 7.5.2 Value Changes

```
const inValue = op.inFloat("Value", 0);
const outDouble = op.outNumber("Double");

inValue.onChange = function() {
    const val = inValue.get();
    outDouble.set(val * 2);
};
```

### 7.5.3 Linking Ports

```
// Automatically update output when input changes
const inValue = op.inFloat("Value", 0);
const outValue = op.outNumber("Value Out");

inValue.onChange = outValue.setRef.bind(outValue, inValue);
```

```
// or simply:  
// inValue.onChange = () => outValue.set(inValue.get());
```

## 7.6 Working with Arrays

```
const inArray = op.inArray("Input Array");  
const outArray = op.outArray("Output Array");  
  
inArray.onChange = function() {  
    const arr = inArray.get();  
    if (!arr) return;  
  
    // Process array  
    const result = arr.map(x => x * 2);  
  
    outArray.set(result);  
};
```

## 7.7 Working with Objects

```
const inObject = op.inObject("Input");  
const outObject = op.outObject("Output");  
  
inObject.onChange = function() {  
    const obj = inObject.get();  
    if (!obj) return;  
  
    // Process or wrap the object  
    const processed = {  
        ...obj,  
        processed: true  
    };  
  
    outObject.set(processed);  
};
```

## 7.8 Render Loop Integration

For ops that need to run every frame:

```
const inTrigger = op.inTrigger("Render");  
const outNext = op.outTrigger("Next");  
  
let time = 0;  
  
inTrigger.onTriggered = function() {  
    time += op.patch.timer.getDelta();  
  
    // Do per-frame calculations  
  
    outNext.trigger();  
};
```

## 7.9 UI Port Groups

Organize your ports into collapsible groups:

```
// Create ports  
const inX = op.inFloat("X", 0);  
const inY = op.inFloat("Y", 0);  
const inZ = op.inFloat("Z", 0);  
  
// Group them  
op.setPortGroup("Position", [inX, inY, inZ]);
```

## 7.10 Port UI Types

Change how ports appear in the UI:

```
// Slider
const inValue = op.inFloat("Value", 0.5);
op.setUiAttrib({ "type": "slider", "min": 0, "max": 1 });

// Color picker
const inR = op.inFloat("R", 1);
const inG = op.inFloat("G", 1);
const inB = op.inFloat("B", 1);
op.setPortGroup("Color", [inR, inG, inB]);
inR.setUiAttribs({ colorPick: true });

// Dropdown
const inMode = op.inSwitch("Mode", ["Option1", "Option2", "Option3"], "Option1");
inMode.setUiAttribs({ dropdown: true });

```

## 7.11 Accessing Patch Resources

### 7.11.1 Timer and Time

```
// Current time
const time = op.patch.timer.getTime();

// Delta time (time since last frame)
const delta = op.patch.timer.getDelta();

// FPS
const fps = op.patch.timer.getFPS();
```

### 7.11.2 Canvas and Context

```
// Canvas element
const canvas = op.patch.cgl.canvas;

// WebGL context
const gl = op.patch.cgl.gl;
```

### 7.11.3 Loading External Resources

```
const inUrl = op.inString("URL", "");
const outData = op.outObject("Data");

inUrl.onChange = function() {
    const url = inUrl.get();
    if (!url) return;

    fetch(url)
        .then(response => response.json())
        .then(data => {
            outData.set(data);
        })
        .catch(error => {
            op.LogError("Failed to load:", error);
        });
};
```

## 7.12 Using External Libraries

### 7.12.1 Including Libraries

```
// In op's code, load an external script
const script = document.createElement("script");
script.src = "https://cdn.example.com/library.js";
script.onload = function() {
    // Library is ready
    initLibrary();
};
document.head.appendChild(script);
```

### 7.12.2 Or use op.patch.loading for proper load tracking:

```
op.patch.loading.start();
```

```

const script = document.createElement("script");
script.src = "https://cdn.example.com/library.js";
script.onload = function() {
    op.patch.loading.finished();
    initLibrary();
};
script.onerror = function() {
    op.patch.loading.finished();
    op.LogError("Failed to load library");
};
document.head.appendChild(script);

```

## 7.13 Error Handling

```

try {
    // Risky operation
    const result = riskyFunction();
    outResult.set(result);
} catch (error) {
    op.LogError("Operation failed:", error);
    op.setUiError("error", error.message);
}

// Clear error when fixed
op.setUiError("error", null);

```

## 7.14 Example: Custom Math Op

```

// Custom clamp with smoothing

const inValue = op.inFloat("Value", 0);
const inMin = op.inFloat("Min", 0);
const inMax = op.inFloat("Max", 1);
const inSmoothing = op.inFloat("Smoothing", 0);
const outValue = op.outNumber("Result");

```

```

let currentValue = 0;

function update() {
    let val = inValue.get();
    const min = inMin.get();
    const max = inMax.get();
    const smooth = inSmoothing.get();

    // Clamp
    val = Math.max(min, Math.min(max, val));

    // Smooth
    if (smooth > 0) {
        currentValue += (val - currentValue) * (1 - smooth);
    } else {
        currentValue = val;
    }

    outValue.set(currentValue);
}

inValue.onChange = update;
inMin.onChange = update;
inMax.onChange = update;
inSmoothing.onChange = update;

```

## 7.15 Example: Array Processor

```

// Sum all values in an array

const inArray = op.inArray("Values");
const outSum = op.outNumber("Sum");
const outAverage = op.outNumber("Average");
const outCount = op.outNumber("Count");

inArray.onChange = function() {
    const arr = inArray.get();

```

```

if (!arr || arr.length === 0) {
  outSum.set(0);
  outAverage.set(0);
  outCount.set(0);
  return;
}

const sum = arr.reduce((a, b) => a + b, 0);
const count = arr.length;
const average = sum / count;

outSum.set(sum);
outAverage.set(average);
outCount.set(count);
};

```

## 7.16 Example: API Fetcher

```

// Fetch data from an API

const inUrl = op.inString("API URL", "");
const inFetch = op.inTriggerButton("Fetch");
const outData = op.outObject("Data");
const outLoading = op.outBool("Loading");
const outError = op.outString("Error");

inFetch.onTriggered = async function() {
  const url = inUrl.get();
  if (!url) return;

  outLoading.set(true);
  outError.set("");

  try {
    const response = await fetch(url);
    const data = await response.json();
    outData.set(data);
  } catch (error) {

```

```

    outError.set(error.message);
    outData.set(null);
  } finally {
    outLoading.set(false);
  }
};

```

## 7.17 Debugging Tips

```

// Log to console
console.log("Value:", inValue.get());

// Op-specific logging (shows in cables UI)
op.log("This is a log message");
op.logWarn("This is a warning");
op.LogError("This is an error");

// Visual debugging
op.setUiAttrib({ "error": "Something went wrong" });

```

## 7.18 Advanced Patterns (How to Build “Good” Ops)

Once you start writing more than a couple custom ops, quality becomes less about JavaScript syntax and more about **behavior**:

- **Determinism**: given the same inputs, the op produces the same outputs.
- **Clear execution model**: value changes vs trigger-based evaluation are intentional.
- **Performance**: avoid unnecessary allocations and expensive work per frame.
- **Good UI/UX**: errors are visible, defaults are sane, ports are grouped and labeled.

### 7.18.1 Pattern: Separate “Compute” from “Trigger”

A clean approach is:

- collect values in `onChange`
- do the heavy compute in one `update()` function
- call `update()` from whichever events are relevant

```
const inTrigger = op.inTrigger("Update");
const inA = op.inFloat("A", 0);
const inB = op.inFloat("B", 0);
const outResult = op.outNumber("Result");
const outNext = op.outTrigger("Next");

function update() {
  outResult.set(inA.get() + inB.get());
}

inA.onChange = update;
inB.onChange = update;

inTrigger.onTriggered = function () {
  update();
  outNext.trigger();
};
```

```
function recompute() {
  const v = inValue.get();
  // pretend this is expensive:
  cached = Math.sin(v) * Math.cos(v) * 1000;
  outProcessed.set(cached);
  dirty = false;
}

inValue.onChange = function () {
  dirty = true;
};

inTrigger.onTriggered = function () {
  if (dirty) recompute();
  outNext.trigger();
};
```

## 7.18.2 Pattern: “Only Recompute When Dirty”

If an op gets triggered every frame but its inputs rarely change, cache the result:

```
const inTrigger = op.inTrigger("Render");
const outNext = op.outTrigger("Next");

const inValue = op.inFloat("Value", 0);
const outProcessed = op.outNumber("Processed");

let dirty = true;
let cached = 0;
```

## 7.18.3 Pattern: Debounce (Stabilize Noisy Inputs)

Useful for sliders, mouse input, or network-driven values.

```
const inValue = op.inFloat("Value", 0);
const inDelayMs = op.inInt("Delay (ms)", 200);
const outValue = op.outNumber("Debounced");

let t = null;

inValue.onChange = function () {
  if (t) clearTimeout(t);
  t = setTimeout(() => outValue.set(inValue.get()), inDelayMs);
```

## 7.18.4 Pattern: Rate-Limit (Prevent Flooding Downstream)

Useful when sending values to other systems (e.g., API calls, heavy compute, UI).

```
const inTrigger = op.inTrigger("Trigger");
const inMinIntervalMs = op.inInt("Min Interval (ms)", 100);
const outNext = op.outTrigger("Next");

let last = 0;

inTrigger.onTriggered = function () {
  const now = performance.now();
  if (now - last >= inMinIntervalMs.get()) {
    last = now;
    outNext.trigger();
  }
};
```

## 7.18.5 Pattern: Stateful Ops (Resettable Systems)

Any op that accumulates state should expose a reset trigger.

```
const inAdd = op.inTrigger("Add");
const inReset = op.inTrigger("Reset");
const inValue = op.inFloat("Value", 1);
const outSum = op.outNumber("Sum");

let sum = 0;

function emit() {
  outSum.set(sum);
}

inAdd.onTriggered = function () {
```

```
  sum += inValue.get();
  emit();
};

inReset.onTriggered = function () {
  sum = 0;
  emit();
};
```

## 7.19 Async Ops (Fetching Data Safely)

When you talk to the network, the two most important qualities are:

- **cancellation**: don't keep old requests alive if the user changes the URL
- **loading/error UX**: surface the state to the patch (and optionally the UI)

### 7.19.1 Example: Fetch JSON with Cancellation

```
const inUrl = op.inString("URL", "");
const inFetch = op.inTriggerButton("Fetch");

const outData = op.outObject("Data");
const outLoading = op.outBool("Loading");
const outError = op.outString("Error");

let controller = null;

inFetch.onTriggered = async function () {
  const url = inUrl.get();
  if (!url) return;

  // cancel previous request
  if (controller) controller.abort();
  controller = new AbortController();

  outLoading.set(true);
  outError.set("");
```

```

try {
  const res = await fetch(url, { signal: controller.signal });
  if (!res.ok) throw new Error(`HTTP ${res.status}`);
  const json = await res.json();
  outData.set(json);
} catch (e) {
  // ignore abort errors as "expected"
  if (e && e.name === "AbortError") return;
  outError.set(String(e && e.message ? e.message : e));
  outData.set(null);
} finally {
  outLoading.set(false);
}
};

```

3. Create a simple state machine op
4. Build an op that fetches and parses CSV data

## 7.19.2 Loading Semantics (Patch-Friendly)

If an op blocks the patch from being “ready” until something loads, use the patch loading tracking mechanism shown earlier (`op.patch.loading.start()` / `finished()`), and keep those calls paired even on error paths.

## 7.20 Performance Tips for Custom Ops

- **Avoid allocations in per-frame triggers:** reuse arrays/objects when possible.
- **Minimize DOM work:** avoid creating elements repeatedly; cache references.
- **Don't spam logs:** logging inside every-frame triggers will kill performance.
- **Prefer simple math:** it's easy to do too much in JS when the GPU could do it (shader).

## 7.21 Featured Videos

## 7.22 Exercises

1. Create a custom op that formats a number with a prefix and suffix
  2. Build an array shuffler op
-

# 8 Audio & Sound in Cables.gl

## 8.1 Introduction

Cables.gl has powerful audio capabilities, enabling you to create audio-reactive visuals, music visualizations, and interactive sound experiences.

## 8.2 Audio Sources

### 8.2.1 AudioFile

Load and play audio files:

```
AudioFile -> AudioAnalyzer -> Visual ops
```

**Supported Formats:** - MP3 - WAV - OGG

**Key Parameters:** - URL - Path to audio file - Loop - Repeat playback - Volume - Playback volume - Playback Rate - Speed control

### 8.2.2 Microphone

Capture live audio input:

```
Microphone -> AudioAnalyzer -> Visual ops
```

**Note:** Requires user permission in browser.

### 8.2.3 AudioBuffer

Load audio into memory for precise control.

### 8.2.4 WebAudio Oscillator

Generate synthetic sounds:

```
Oscillator -> Audio output
```

**Types:** - Sine - Square - Sawtooth - Triangle

## 8.3 Audio Analysis

### 8.3.1 AudioAnalyzer

The core op for audio-reactive visuals:

```
 AudioSource -> AudioAnalyzer  
 |  
 Outputs: FFT, Volume, Bass, Mid, High
```

**Key Outputs:** - FFT Array - Frequency spectrum data - Volume - Overall loudness - Bass - Low frequency level - Mid - Middle frequency level - High - High frequency level

### 8.3.2 FFT (Fast Fourier Transform)

Breaks audio into frequency bands:

```
 AudioAnalyzer -> FFTArray -> ArrayIterator  
 |  
 Visualize each band
```

**FFT Size Options:** - 32, 64, 128, 256, 512, 1024, 2048, 4096 - Larger = more detail, but slower

### 8.3.3 Smoothing

Apply smoothing to prevent jittery visuals:

```
AudioValue -> Smooth -> Visual parameter
```

## 8.4 Common Audio-Reactive Patterns

### 8.4.1 Volume-Based Scaling

```
AudioAnalyzer (volume) -> Scale input of shape
```

### 8.4.2 Frequency Band Visualization

```
MainLoop  
|  
BasicMaterial  
|  
AudioAnalyzer -> FFTArray  
|  
ArrayIterator  
|  
Transform (X position from index)  
|  
Transform (Y scale from FFT value)  
|  
Rectangle
```

### 8.4.3 Color from Audio

```
AudioAnalyzer (bass) -> Hue input of HSBtoRGB  
HSBtoRGB -> BasicMaterial (color input)
```

### 8.4.4 Beat Detection

```
AudioAnalyzer (volume) -> Threshold -> Trigger  
|
```

(triggers on beat)

## 8.5 Audio Effects

### 8.5.1 Gain

Control volume:

```
 AudioSource -> Gain -> Output
```

### 8.5.2 Filter

Shape the frequency content:

```
 AudioSource -> Filter -> Output
```

**Filter Types:**  
- Lowpass - Removes high frequencies  
- Highpass - Removes low frequencies  
- Bandpass - Keeps only middle frequencies  
- Notch - Removes specific frequency

### 8.5.3 Delay

Add echo effect:

```
 AudioSource -> Delay -> Output
```

### 8.5.4 Reverb

Add space/ambience:

```
 AudioSource -> Reverb -> Output
```

## 8.5.5 Compressor

Even out dynamics:

```
 AudioSource -> Compressor -> Output
```

## 8.6 Building a Visualizer

### 8.6.1 Step 1: Set Up Audio

```
 AudioFile (your music)
 |
 AudioAnalyzer
```

### 8.6.2 Step 2: Create Base Render

```
 MainLoop
 |
 Camera (for 3D) or BasicMaterial (for 2D)
```

### 8.6.3 Step 3: Add Audio-Reactive Elements

Example: Pulsing Circle

```
 MainLoop -> BasicMaterial
           |
           AudioAnalyzer (volume)
           |
           Smooth (for smoother animation)
```

```
   |
   Math (multiply by desired scale)
   |
   Circle (size input)
```

### 8.6.4 Step 4: Add Frequency Visualization

```
 AudioAnalyzer -> FFTArray
   |
   ArrayIterator (iterate through frequencies)
   |
   Index -> Calculate X position
   |
   FFT Value -> Calculate height/color
   |
   Rectangle (bar for each frequency)
```

## 8.7 Synchronizing to Music

### 8.7.1 BPM and Beat Sync

```
 AudioFile
   |
   BPMSync (set your song's BPM)
   |
   Beat triggers for animations
```

### 8.7.2 Timeline with Audio

1. Load audio file
2. Add to timeline
3. Use timeline markers for sync points
4. Keyframe animations to match audio

## 8.8 Advanced Audio Techniques (Make It Feel “Musical”)

Audio-reactive visuals often fail in the same way: they’re *too jittery* and *too literal*. The goal is usually:

- stable motion with **musical** response (not “random noise” response)
- clear separation between **slow energy** (overall level) and **fast transients** (kicks/snare hits)
- mappings that feel good: log frequency, clamped ranges, smoothing that doesn’t lag

### 8.8.1 Technique: Energy vs Transient (Two-Signal Approach)

Treat audio as two complementary control signals:

- **Energy**: smoothed volume/bass/mid/high (drives slow changes: camera drift, fog density, palette)
- **Transients**: thresholded + debounced triggers (drives discrete events: flashes, spawns, scene cuts)

Typical building blocks:

```
AudioAnalyzer (volume/bass/mid/high)
  +-> Smooth (slow) -> Energy signal
      +-> Threshold -> (optional Delay/Interval gating) -
  > Transient trigger
```

### 8.8.2 Technique: Log Frequency Mapping (Better Spectra)

FFT bins are linear in frequency, but our hearing is closer to logarithmic. If your spectrum visualization looks “all action on the left”, try mapping indices in a non-linear way:

- compress the low bins less (give bass more space)
- compress high bins more (reduce over-detail)

Conceptually:

```
Index -> Normalize (0..1) -> Pow (curve) -> Sample FFT
```

### 8.8.3 Technique: Peak Hold (Readable Visuals)

Human-friendly meters often have a “peak hold” that decays slowly. You can build this by:

- capturing the max value over a short window
- then decaying it over time

Conceptually:

```
AudioValue -> Max (with previous peak) -> Decay over time -
  > Peak output
```

### 8.8.4 Technique: Band-Specific Control (Bass Drives Scale, High Drives Detail)

Instead of driving everything from overall volume:

- **bass** -> big scale/position changes
- **mid** -> color shifts or mid-size motion
- **high** -> small jitter/detail/particles

This makes visuals feel much more “mixed”.

### 8.8.5 Technique: Audio -> Shader (The “Pro” Move)

Shading is where audio-reactive projects often become cinematic.

High-level pattern:

```
AudioAnalyzer (energy) -> Smooth -> Shader uniform (e.g., amount)  
FFTArray -> (reduce / select bands) -> Shader uniform(s)  
Time -> Shader uniform (time)
```

Then, in the shader, use audio as **a modulation source**, not as the final value.  
(Example: warp UVs slightly, not wildly.)

## 8.9 Advanced Patch Recipes

### 8.9.1 Recipe: Stable Beat Trigger (Avoid Double-Triggers)

The simplest fix for “machine-gun” beats is gating:

```
AudioAnalyzer (volume or bass)  
|  
Threshold (set just above noise floor)  
|  
(Gate / minimum time between triggers)  
|  
Trigger (spawn / flash / step timeline)
```

### 8.9.2 Recipe: Audio-Reactive Post-Processing

Drive a texture effect strength from music:

```
MainLoop -> Camera -> RenderToTexture -> TextureEffect -> Output  
^  
AudioAnalyzer (volume) -> Smooth -> Map -> effect strength
```

### 8.9.3 Recipe: Audio-Reactive 3D Equalizer (Optimized)

If you build an equalizer with many bars:

- keep geometry simple
- reduce FFT size to what you need
- avoid doing heavy work per bar per frame

Conceptually:

```
AudioAnalyzer -> FFTArray  
|  
ArrayIterator (N bands)  
|  
Transform (X from index, Y scale from FFT)  
|  
Cube (bar)
```

### 8.9.4 Recipe: Audio-Driven Palette

Map energy to hue/saturation to get coherent color shifts:

```
AudioAnalyzer (mid) -> Smooth -> Map -> Hue  
AudioAnalyzer (bass) -> Smooth -> Map -> Saturation  
HSBtoRGB -> BasicMaterial (color)
```

## 8.10 Practical Examples

### 8.10.1 Example 1: Bass-Reactive Background

```
MainLoop  
|  
AudioFile -> AudioAnalyzer (bass)  
|  
Smooth (0.9)  
|  
Map (0-1 to desired range)  
|  
HSBtoRGB (bass controls saturation) -> BasicMaterial (color input)
```

```
|  
BasicMaterial  
|  
FullscreenRectangle
```

### 8.10.2 Example 2: Circular Spectrum

```
MainLoop  
|  
BasicMaterial  
|  
AudioAnalyzer -> FFTArray  
|  
ArrayIterator  
|  
Transform (rotate based on index)  
|  
Transform (translate by FFT value)  
|  
Circle (small)
```

### 8.10.3 Example 3: Waveform Display

```
MainLoop  
|  
BasicMaterial  
|  
AudioAnalyzer -> WaveformArray  
|  
PointCloud or LineStrip
```

### 8.10.4 Example 4: 3D Audio Visualization

```
MainLoop  
|
```

```
Camera -> OrbitControls  
|  
AudioAnalyzer -> FFTArray  
|  
ArrayIterator (creates ring)  
|  
Transform (position in circle)  
|  
Transform (scale Y by FFT)  
|  
Cube
```

## 8.11 Performance Considerations

1. **FFT Size** - Use smallest size that gives needed detail
2. **Smoothing** - Higher smoothing = less CPU for animations
3. **Update Rate** - Don't need 60fps for all audio analysis
4. **Visualizer Complexity** - Balance detail with performance

## 8.12 Browser Audio Policies

Modern browsers require user interaction before playing audio:

1. Add a "Start" button
2. Start audio on button click
3. Or use `AudioContext.resume()` on first interaction

```
// In custom op or patch  
document.addEventListener('click', () => {  
    if (audioContext.state === 'suspended') {  
        audioContext.resume();  
    }  
}, { once: true });
```

## 8.13 Featured Videos

## 8.14 Exercises

1. Create a simple volume meter with animated bars
2. Build a circular frequency spectrum visualizer
3. Make a 3D landscape that morphs to music
4. Create a beat-triggered strobe effect

# 9 Animation & Timeline in Cables.gl

## 9.1 Introduction

Cables.gl provides multiple ways to create animations, from simple time-based movements to complex keyframed sequences using the timeline.

## 9.2 Types of Animation

### 9.2.1 1. Procedural Animation

Using math and time to create continuous motion.

### 9.2.2 2. Keyframe Animation

Defining specific values at specific times.

### 9.2.3 3. Physics-Based Animation

Simulating natural motion with springs, gravity, etc.

### 9.2.4 4. Data-Driven Animation

Animating based on input data or user interaction.

## 9.3 Procedural Animation

### 9.3.1 The Time Op

The foundation of procedural animation:

Time -> Outputs current time in seconds

**Uses:** - Input for trigonometric functions - Driving continuous rotation - Creating loops and cycles

## 9.3.2 Basic Movement Patterns

### Linear Movement:

```
Time -> Modulo (loop duration) -> Position
```

### Oscillation (Sine Wave):

```
Time -> Sin -> Scale/Position
```

### Bounce:

```
Time -> Sin -> Abs -> Position
```

### Circular Motion:

```
Time -> Cos -> X position
```

```
Time -> Sin -> Y position
```

## 9.3.3 Easing Functions

Transform linear time into smooth curves:

### Ease In (slow start):

```
t * t // Quadratic  
t * t * t // Cubic
```

### Ease Out (slow end):

```
1 - (1 - t) * (1 - t)
```

### Ease In-Out (smooth both):

```
t < 0.5 ? 2 * t * t : 1 - pow(-2 * t + 2, 2) / 2
```

## 9.3.4 The Smooth Op

Smoothly interpolate towards target values:

```
TargetValue -> Smooth -> AnimatedValue
```

**Parameter:** - Smoothing - Higher = slower, smoother transitions

## 9.3.5 Spring Animation

Create bouncy, natural motion:

```
TargetValue -> Spring -> AnimatedValue
```

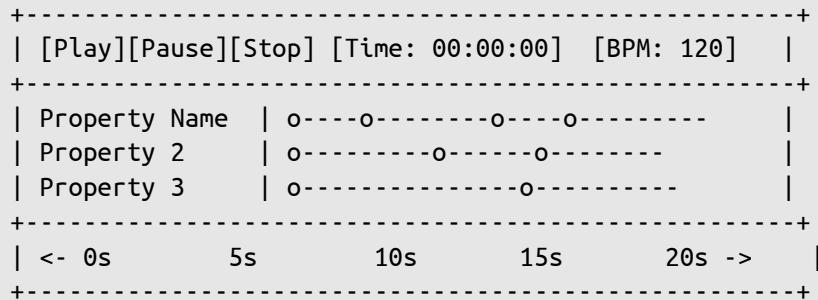
**Parameters:** - Stiffness - How quickly it moves - Damping - How quickly it settles

## 9.4 Timeline Animation

### 9.4.1 Opening the Timeline

1. Click the timeline icon in the toolbar
2. Or press T to toggle timeline visibility

### 9.4.2 Timeline Interface



### 9.4.3 Adding Keyframes

1. Select the op with the property to animate
2. Move the timeline playhead to the desired time
3. Set the value
4. Click the keyframe button (or right-click the property)

### 9.4.4 Keyframe Types

- **Linear** - Straight line between keyframes
- **Step** - Instant change at keyframe
- **Ease In** - Slow start
- **Ease Out** - Slow end
- **Ease In-Out** - Smooth start and end
- **Bezier** - Custom curve with handles

### 9.4.5 Editing Keyframes

- **Move:** Drag keyframe left/right (time) or up/down (value)
- **Delete:** Select and press Delete
- **Copy/Paste:** Ctrl+C, Ctrl+V
- **Multi-select:** Shift+click or drag box

### 9.4.6 Timeline Tracks

Organize animations into tracks:

- **Property tracks** - Individual values

- **Trigger tracks** - Fire events at specific times
- **Audio tracks** - Sync with music

## 9.5 Non-Linear Animation Clips (New Animation System - November 2025)

The new animation system in Cables.gl introduces powerful non-linear animation capabilities through **animation clips**. Clips are reusable, addable, and mixable animation sequences that can be layered and blended to create complex motion.

### 9.5.1 What Are Animation Clips?

Animation clips are self-contained animation sequences that can be:

- **Reusable** - Create once, apply to multiple parameters
- **Addable** - Layer multiple clips together (additive blending)
- **Mixable** - Blend between clips with different weights
- **Non-linear** - Don't require strict sequential playback

### 9.5.2 Creating Animation Clips

#### Step 1: Enable Clip Mode

1. Add an **Anim** operator to your patch
2. Connect it to the parameter you want to animate
3. Open the Anim operator's properties
4. Enable the **Clip** option
5. Assign a **Clip Name** (e.g., "bounce", "fadeIn", "rotate360")

Parameter -> Anim (Clip enabled, Name: "myClip") -> Animated Value

#### Step 2: Define Keyframes

1. With the Anim operator selected, open the Timeline
2. Set keyframes for your animation sequence
3. Adjust easing curves and timing
4. The animation is now stored as a named clip

### Step 3: Apply Clips to Other Parameters

Once created, clips can be applied to any other Anim operator:

1. Add another Anim operator
2. In the Timeline, right-click on a keyframe
3. Select “Apply Clip” and choose your clip name
4. The clip’s animation will be applied at that keyframe

### 9.5.3 Clip Properties and Options

#### Looping Modes

Clips support different looping behaviors:

- **None** - Play once and stop
- **Repeat** - Loop from start to end
- **Mirror** - Play forward, then backward
- **Offset** - Continue from end value

#### Interpolation Methods

- **Linear** - Straight interpolation
- **Ease In/Out** - Smooth acceleration/deceleration
- **Bezier** - Custom curve control
- **Step** - Instant value changes

### 9.5.4 Additive Animation (Layering Clips)

Multiple clips can be **added together** to create combined effects:

```
Base Value
|
Anim Clip 1 ("bounce") -> Add
|
Anim Clip 2 ("rotate") -> Add
|
Anim Clip 3 ("scale") -> Add
```

Final Animated Value

**Use Cases:** - Base idle animation + triggered bounce effect - Procedural motion + keyframed structure - Multiple independent motion layers

#### Example: Character Animation

```
Idle Clip (continuous breathing)
|
Walk Clip (additive, triggered on movement)
|
Jump Clip (additive, triggered on jump)
|
Final Position
```

### 9.5.5 Mixable Animation (Blending Clips)

Clips can be **mixed** with different weights to blend between animations:

```
Clip A ("walk") -> Mix (weight: 0.7)
Clip B ("run")   -> Mix (weight: 0.3)
|
Blended Animation
```

**Blending Modes:** - **Linear Blend** - Simple weighted average - **Smooth Blend** - Eased transition between clips - **Additive Blend** - Add clips together with weights

#### Example: Walk-to-Run Transition

```
Walk Clip -> Mix (weight: 1.0 - runProgress)
Run Clip  -> Mix (weight: runProgress)
|
Smooth transition from walk to run
```

## 9.5.6 Clip Management

### Organizing Clips

Clips are stored within your project and can be:

- **Renamed** - Right-click clip in timeline -> Rename
- **Duplicated** - Copy clip to create variations
- **Deleted** - Remove unused clips
- **Exported/Imported** - Share clips between projects

### Clip Library

Access all clips in your project:

1. Open Timeline
2. Click "Clips" tab
3. View all available clips
4. Drag clips onto timeline tracks

## 9.5.7 Advanced Clip Techniques

### Clip Offsets and Time Remapping

Apply clips at different time offsets:

```
Clip "bounce" (duration: 2s)
  |
  Apply at t=0s: Full clip
  Apply at t=5s: Clip starts here
  Apply at t=10s: Clip with 0.5x speed (time remap)
```

### Clip Masking

Use clips to mask or modulate other animations:

```
Base Animation -> Multiply
Clip "mask" (0 to 1) -> Multiply
  |
Masked Animation (only active where mask = 1)
```

### Conditional Clip Playback

Control clip playback based on conditions:

```
Condition -> If
  +-> True: Play Clip A
  +-> False: Play Clip B
```

## 9.6 JavaScript Custom Op Integration with Animation System

The new animation system integrates seamlessly with JavaScript custom operators, allowing programmatic control and extension of animation capabilities.

### 9.6.1 Accessing Animation Data from Custom Ops

#### Reading Animation Values

```
// Get current animation value from an Anim op
const animOp = op.patch.findOpByName("MyAnimOp");
if (animOp) {
  const currentValue = animOp.outValue.get();
  // Use the animated value
}
```

#### Monitoring Animation State

```
const inTrigger = op.inTrigger("Render");
const outAnimValue = op.outNumber("Animation Value");
const outIsPlaying = op.outBool("Is Playing");

let animOp = null;

// Find the Anim op (call once on init)
```

```

op.onInit = function() {
    animOp = op.patch.findOpByName("MyAnimOp");
};

inTrigger.onTriggered = function() {
    if (animOp) {
        // Get current animated value
        outAnimValue.set(animOp.outValue.get());

        // Check if timeline is playing
        const timeline = op.patch.timeline;
        if (timeline) {
            outIsPlaying.set(timeline.isPlaying());
        }
    }
};

```

## 9.6.2 Controlling Timeline from Custom Ops

### Playback Control

```

const inPlay = op.inTriggerButton("Play");
const inPause = op.inTriggerButton("Pause");
const inStop = op.inTriggerButton("Stop");
const inSeek = op.inFloat("Seek Time", 0);
const inSeekTrigger = op.inTrigger("Seek");

inPlay.onTriggered = function() {
    const timeline = op.patch.timeline;
    if (timeline) timeline.play();
};

inPause.onTriggered = function() {
    const timeline = op.patch.timeline;
    if (timeline) timeline.pause();
};

inStop.onTriggered = function() {

```

```

    const timeline = op.patch.timeline;
    if (timeline) timeline.stop();
};

inSeekTrigger.onTriggered = function() {
    const timeline = op.patch.timeline;
    if (timeline) {
        timeline.seek(inSeek.get());
    }
};

```

### Timeline Time and Progress

```

const inTrigger = op.inTrigger("Render");
const outTime = op.outNumber("Current Time");
const outProgress = op.outNumber("Progress (0-1)");
const outDuration = op.outNumber("Total Duration");

inTrigger.onTriggered = function() {
    const timeline = op.patch.timeline;
    if (timeline) {
        const currentTime = timeline.getTime();
        const duration = timeline.getDuration();

        outTime.set(currentTime);
        outDuration.set(duration);
        outProgress.set(duration > 0 ? currentTime / duration : 0);
    }
};

```

## 9.6.3 Creating Animation Clips Programmatically

### Generating Clip Data

```

// Custom op that generates animation clip data
const inDuration = op.inFloat("Duration", 2.0);

```

```

const inAmplitude = op.inFloat("Amplitude", 1.0);
const inFrequency = op.inFloat("Frequency", 1.0);
const inGenerate = op.inTriggerButton("Generate Clip");
const outClipData = op.outObject("Clip Data");

inGenerate.onTriggered = function() {
    const duration = inDuration.get();
    const amplitude = inAmplitude.get();
    const freq = inFrequency.get();
    const sampleRate = 60; // samples per second
    const numSamples = Math.floor(duration * sampleRate);

    const keyframes = [];
    for (let i = 0; i <= numSamples; i++) {
        const t = i / numSamples;
        const time = t * duration;
        // Generate sine wave animation
        const value = Math.sin(time * freq * Math.PI * 2) * amplitude;
        keyframes.push({
            time: time,
            value: value,
            easing: "easeInOut"
        });
    }

    outClipData.set({
        name: "generatedSine",
        duration: duration,
        keyframes: keyframes,
        loop: "repeat"
    });
};

```

## 9.6.4 Manipulating Animation Clips

### Blending Multiple Clips

```

// Custom op that blends multiple animation clips
const inClipA = op.inObject("Clip A Data");
const inClipB = op.inObject("Clip B Data");
const inBlendFactor = op.inFloat("Blend Factor", 0.5); // 0 = A, 1 = B
const inTime = op.inFloat("Time", 0);
const outBlendedValue = op.outNumber("Blended Value");

inTime.onChange = function() {
    const clipA = inClipA.get();
    const clipB = inClipB.get();
    const blend = inBlendFactor.get();
    const t = inTime.get();

    if (!clipA || !clipB) return;

    // Sample both clips at time t
    const valueA = sampleClip(clipA, t);
    const valueB = sampleClip(clipB, t);

    // Blend
    const blended = valueA * (1 - blend) + valueB * blend;
    outBlendedValue.set(blended);
};

function sampleClip(clip, time) {
    const keyframes = clip.keyframes;
    if (!keyframes || keyframes.length === 0) return 0;

    // Clamp time to clip duration
    time = time % clip.duration;

    // Find surrounding keyframes
    for (let i = 0; i < keyframes.length - 1; i++) {
        if (time >= keyframes[i].time && time <= keyframes[i + 1].time) {
            // Interpolate
            const t0 = keyframes[i].time;
            const t1 = keyframes[i + 1].time;
            const v0 = keyframes[i].value;
        }
    }
}

```

```

        const v1 = keyframes[i + 1].value;

        const t = (time - t0) / (t1 - t0);
        return v0 + (v1 - v0) * t;
    }

    return keyframes[keyframes.length - 1].value;
}

```

```

// Custom op that adds multiple clips together
const inClips = op.inArray("Clips Array");
const inTime = op.inFloat("Time", 0);
const outCombinedValue = op.outNumber("Combined Value");

inTime.onChange = function() {
    const clips = inClips.get();
    const t = inTime.get();

    if (!clips || clips.length === 0) {
        outCombinedValue.set(0);
        return;
    }

    let sum = 0;
    for (let i = 0; i < clips.length; i++) {
        const clip = clips[i];
        if (clip && clip.keyframes) {
            sum += sampleClip(clip, t);
        }
    }

    outCombinedValue.set(sum);
};

```

## 9.6.5 Advanced: Custom Easing Functions

```

// Custom op with advanced easing functions
const inValue = op.inFloat("Input (0-1)", 0);
const inEasingType = op.inSwitch("Easing",
    ["linear", "easeInQuad", "easeOutQuad", "easeInOutQuad",
     "easeInCubic", "easeOutCubic", "easeInOutCubic",
     "easeInElastic", "easeOutBounce"],
     "easeInOutQuad");
const outEased = op.outNumber("Eased Value");

inValue.onChange = function() {
    const t = Math.max(0, Math.min(1, inValue.get()));
    const type = inEasingType.get();
    let eased = 0;

    switch(type) {
        case "linear":
            eased = t;
            break;
        case "easeInQuad":
            eased = t * t;
            break;
        case "easeOutQuad":
            eased = 1 - (1 - t) * (1 - t);
            break;
        case "easeInOutQuad":
            eased = t < 0.5
                ? 2 * t * t
                : 1 - Math.pow(-2 * t + 2, 2) / 2;
            break;
        case "easeInCubic":
            eased = t * t * t;
            break;
        case "easeOutCubic":
            eased = 1 - Math.pow(1 - t, 3);
            break;
        case "easeInOutCubic":
            eased = t < 0.5

```

```

        ? 4 * t * t * t
        : 1 - Math.pow(-2 * t + 2, 3) / 2;
    break;
case "easeInElastic":
    const c4 = (2 * Math.PI) / 3;
    eased = t === 0 ? 0 : t === 1 ? 1
        : -Math.pow(2, 10 * t - 10) * Math.sin((t * 10 -
10.75) * c4);
    break;
case "easeOutBounce":
    const n1 = 7.5625;
    const d1 = 2.75;
    if (t < 1 / d1) {
        eased = n1 * t * t;
    } else if (t < 2 / d1) {
        eased = n1 * (t -= 1.5 / d1) * t + 0.75;
    } else if (t < 2.5 / d1) {
        eased = n1 * (t -= 2.25 / d1) * t + 0.9375;
    } else {
        eased = n1 * (t -= 2.625 / d1) * t + 0.984375;
    }
    break;
}
outEased.set(eased);
};

```

## 9.6.6 Real-Time Animation Modification

```

// Custom op that modifies animation in real-time based on input
const inBaseAnim = op.inObject("Base Animation Clip");
const inModifier = op.inFloat("Modifier", 1.0);
const inTime = op.inFloat("Time", 0);
const outModifiedValue = op.outNumber("Modified Value");

inTime.onChange = function() {
    const clip = inBaseAnim.get();
    const mod = inModifier.get();

```

```

    const t = inTime.get();

    if (!clip) return;

    // Sample base animation
    let value = sampleClip(clip, t);

    // Apply modifier (could be scale, offset, etc.)
    value *= mod;

    outModifiedValue.set(value);
};

```

## 9.6.7 Integration Example: Physics-Driven Animation

```

// Custom op that combines physics simulation with animation clips
const inAnimClip = op.inObject("Animation Clip");
const inPhysicsForce = op.inFloat("Physics Force", 0);
const inDamping = op.inFloat("Damping", 0.9);
const inTime = op.inFloat("Time", 0);
const outCombinedValue = op.outNumber("Combined Value");

let velocity = 0;
let position = 0;

inTime.onChange = function() {
    const clip = inAnimClip.get();
    const force = inPhysicsForce.get();
    const damp = inDamping.get();
    const t = inTime.get();

    // Get base animation value
    const animValue = clip ? sampleClip(clip, t) : 0;

    // Apply physics
    velocity += force;
    velocity *= damp;
    position += velocity;
};

```

```
// Combine animation + physics
const combined = animValue + position;
outCombinedValue.set(combined);
};
```

## 9.6.8 Best Practices for Animation + Custom Ops

- Cache Clip Sampling** - If sampling clips every frame, cache results when time hasn't changed
- Batch Operations** - Process multiple clips in one op rather than multiple ops
- Use Native Anim Op When Possible** - Only use custom ops when you need functionality beyond built-in features
- Optimize Keyframe Lookups** - Use binary search for large clip keyframe arrays
- Handle Edge Cases** - Always check for null/undefined clips and handle time out of bounds

## 9.6.9 Example: Complete Animation Controller Op

```
// Comprehensive animation controller custom op
const inPlay = op.inTriggerButton("Play");
const inPause = op.inTriggerButton("Pause");
const inStop = op.inTriggerButton("Stop");
const inSeek = op.inFloat("Seek", 0);
const inSpeed = op.inFloat("Speed", 1.0);
const inLoop = op.inBool("Loop", true);

const outTime = op.outNumber("Current Time");
const outProgress = op.outNumber("Progress");
const outIsPlaying = op.outBool("Is Playing");

let currentTime = 0;
let isPlaying = false;
let lastFrameTime = 0;
```

```
op.onInit = function() {
    lastFrameTime = op.patch.timer.getTime();
};

const inRender = op.inTrigger("Render");
inRender.onTriggered = function() {
    const now = op.patch.timer.getTime();
    const delta = now - lastFrameTime;
    lastFrameTime = now;

    if (isPlaying) {
        currentTime += delta * inSpeed.get();

        const timeline = op.patch.timeline;
        if (timeline) {
            const duration = timeline.getDuration();
            if (currentTime >= duration) {
                if (inLoop.get()) {
                    currentTime = currentTime % duration;
                } else {
                    currentTime = duration;
                    isPlaying = false;
                }
            }
            timeline.seek(currentTime);
        }
    }

    outTime.set(currentTime);
    const timeline = op.patch.timeline;
    if (timeline) {
        const duration = timeline.getDuration();
        outProgress.set(duration > 0 ? currentTime / duration : 0);
    }
    outIsPlaying.set(isPlaying);
};

inPlay.onTriggered = function() {
```

```

isPlaying = true;
const timeline = op.patch.timeline;
if (timeline) timeline.play();
};

inPause.onTriggered = function() {
  isPlaying = false;
  const timeline = op.patch.timeline;
  if (timeline) timeline.pause();
};

inStop.onTriggered = function() {
  isPlaying = false;
  currentTime = 0;
  const timeline = op.patch.timeline;
  if (timeline) {
    timeline.stop();
    timeline.seek(0);
  }
};

inSeek.onChange = function() {
  currentTime = inSeek.get();
  const timeline = op.patch.timeline;
  if (timeline) timeline.seek(currentTime);
};

```

## 9.7 Sequence and Timing Ops

### 9.7.1 Sequence

Chain multiple actions in order:

```

Trigger -> Sequence
  +-> Action 1
  +-> Action 2 (after delay)
  +-> Action 3 (after delay)

```

### 9.7.2 Delay

Pause before triggering:

```
Trigger -> Delay (seconds) -> DelayedAction
```

### 9.7.3 Timer

Count down or up:

```
StartTrigger -> Timer -> TimeValue
```

### 9.7.4 Interval

Trigger repeatedly:

```
Interval (every X seconds) -> RepeatedAction
```

## 9.8 Animation Patterns

### 9.8.1 Staggered Animation

Animate multiple items with offset timing:

```

ArrayIterator
  |
Index -> Delay offset
  |
AnimatedProperty

```

## 9.8.2 Loop with Pause

```
Time -> Modulo (total duration)
-> If < activeTime: animate
-> Else: hold at end value
```

```
case "exit":
    // Set exit animation params
    break;
}
```

## 9.8.3 Ping-Pong (Back and Forth)

```
Time -> Sin -> Map to range -> Property
```

Or with timeline: set keyframes to go forward then backward.

## 9.8.4 One-Shot Animation

```
Trigger -> SetValue (start)
-> Smooth -> AnimatedValue
```

## 9.9 State Machines

Create complex animation logic:

### 9.9.1 Simple States

```
// In custom op
let state = "idle";

function setState(newState) {
    state = newState;
    switch(state) {
        case "idle":
            // Set idle animation params
            break;
        case "active":
            // Set active animation params
            break;
```

## 9.9.2 Transition Between States

Use Smooth or Spring ops to blend between state values.

## 9.10 Interactive Animation

### 9.10.1 Mouse-Based

```
MouseX -> Map to range -> Target value -> Smooth -> Property
```

### 9.10.2 Scroll-Based

```
ScrollPosition -> Map (0 to page height) -> (0 to 1) -> Animation progress
```

### 9.10.3 Click-Triggered

```
MouseClick -> Toggle state -> Smooth -> Animated property
```

## 9.11 Advanced Animation Systems (How to Build "Scenes")

As patches grow, animation becomes less about a single value moving and more about **systems**:

- multiple objects animated together ("shots" / "scenes")
- blending procedural motion with keyframed structure
- sequencing events reliably (no double-triggers, no race conditions)

- keeping things readable and maintainable

### 9.11.1 Layering: Timeline for Structure, Procedural for Life

A reliable pattern is:

- **Timeline**: controls the big structure (when things appear, when the camera moves, when a section starts/ends)
- **Procedural**: adds micro-motion (subtle noise, breathing, idle motion, wobble)

Example idea:

```
Timeline -> Base position
Time -> Sin (small) -> Add
Result -> Transform position
```

### 9.11.2 Shot-Based Timelines (Cinematic Organization)

Instead of one giant timeline track list, treat the timeline as a set of “shots”:

- Shot 1: intro framing
- Shot 2: reveal
- Shot 3: close-up detail
- Shot 4: outro / logo

Each shot has:

- a start time, end time
- a camera pose
- a set of object visibility/alpha states

### 9.11.3 Animation Curves: Clamp Early, Map Late

If you see overshoot or sudden jumps, it’s usually a range mismatch.

Good practice:

- normalize to 0..1 early

- clamp to 0..1 before sensitive operations
- map to target range at the end

Conceptually:

```
t (0..1) -> Clamp -> Ease -> Map (min..max)
```

### 9.11.4 Reusable “Rig” Pattern

For any object you animate often, create a mini rig:

- one Transform for position
- one Transform for rotation
- one Transform for scale
- optional “wobble” layer

This makes it easy to swap animation sources later without rewiring the whole patch.

### 9.11.5 Avoiding Jitter in Interactive Animation

If input is noisy (mouse, audio, sensors):

- map input into a safe range
- apply Smooth/Spring
- optionally add dead zones

```
Input -> Map -> Clamp -> Smooth -> Property
```

### 9.11.6 Choreographing Triggers Reliably

For sequences of actions:

- use Sequence for deterministic ordering
- use Delay for spacing
- use Interval for periodic triggers

The key is to avoid “implicit timing” where the order depends on frame timing.

## 9.12 Advanced Recipes

### 9.12.1 Recipe: Scroll-Driven Scene (Interactive Storytelling)

Use scroll position as a normalized progress value:

```
ScrollPosition -> Map (0..pageHeight -> 0..1) -> Clamp -> progress  
progress -> Ease -> Drive camera/object parameters
```

Then you can tie multiple properties to the same progress signal for a coherent experience.

### 9.12.2 Recipe: Beat-Synced Timeline Sections

Use BPM sync to trigger timeline jumps or section changes:

```
AudioFile -> BPMSync -> Beat trigger  
Beat trigger -> Sequence -> (advance state) ->  
> set target animation values
```

### 9.12.3 Recipe: One-Shot “Punch” Animation (No Keyframes)

Great for UI hits, impacts, kick drums:

```
Trigger -> SetValue (1)  
-> Smooth (fast decay) -> scale/brightness
```

You can combine a fast rise + slower decay by chaining two Smooth ops with different parameters.

### 9.12.4 Recipe: Camera Rig (Orbit + Handheld Micro Motion)

```
Time -> Sin/Cos -> Orbit position  
Random (small) -> Smooth -> micro offset  
Add (orbit + micro) -> Camera position  
LookAt -> Camera aim
```

This produces camera movement that feels “alive” but still controlled.

## 9.13 Practical Examples

### 9.13.1 Example 1: Bouncing Ball

```
MainLoop  
|  
BasicMaterial  
|  
Time -> Sin -> Abs -> Y position  
|  
Transform  
|  
Circle
```

### 9.13.2 Example 2: Rotating Carousel

```
MainLoop  
|  
Camera  
|  
ArrayIterator (items)  
|  
Time + (Index * offset) -> Cos -> X position  
Time + (Index * offset) -> Sin -> Z position  
|  
Transform  
|
```

### 9.13.3 Example 3: Fade In Sequence

```
MainLoop
  |
BasicMaterial
  |
ArrayIterator
  |
Time - (Index * staggerDelay) -> Clamp (0, 1) -
> BasicMaterial (alpha input)
  |
Shape
```

### 9.13.4 Example 4: Timeline-Based Scene

```
Timeline
+-- 0s: Camera position keyframe
+-- 2s: Object appears (alpha 0->1)
+-- 4s: Object rotates
+-- 6s: Color change
+-- 8s: Fade out
```

### 9.13.5 Example 5: Layered Animation Clips (Additive)

Create a character with multiple animation layers:

```
Base Position (0, 0, 0)
  |
Anim Clip "idleBreath" (vertical oscillation) -> Add
  |
Anim Clip "walkCycle" (horizontal movement, triggered) -> Add
```

```
Anim Clip "jump" (vertical boost, triggered) -> Add
  |
Final Position -> Transform
```

**Setup:** 1. Create “idleBreath” clip: 2-second vertical sine wave (amplitude: 0.1) 2. Create “walkCycle” clip: 1-second horizontal movement (0 to 1, repeat) 3. Create “jump” clip: 0.5-second vertical boost (0 to 2, one-shot) 4. Connect all three Anim ops to Add ops in sequence 5. Trigger walkCycle and jump clips via user input

### 9.13.6 Example 6: Blended Animation Clips (Mixable)

Smooth transition between walk and run:

```
Walk Clip -> Anim (weight: 1.0 - runBlend)
Run Clip -> Anim (weight: runBlend)
  |
Mix -> Final Position
```

**Setup:** 1. Create “walk” clip: slow horizontal movement 2. Create “run” clip: fast horizontal movement 3. Use a Smooth op to blend between 0 (walk) and 1 (run) 4. Connect both clips to Mix op with blend factor

### 9.13.7 Example 7: Reusable Clip System

Create a library of reusable animation clips:

**Clip Library:**

- "fadeIn" (alpha 0->1, 1s, easeOut)
- "fadeOut" (alpha 1->0, 1s, easeIn)
- "bounce" (scale 1->1.2->1, 0.5s, easeOut)
- "slideInLeft" (x: -100->0, 1s, easeOut)
- "rotate360" (rotation 0->360, 2s, linear)

**Apply to multiple objects:**  
Object 1: fadeIn at t=0s, bounce at t=2s

```
Object 2: slideInLeft at t=1s, fadeOut at t=5s  
Object 3: rotate360 at t=3s (looping)
```

### 9.13.8 Example 8: JavaScript-Controlled Animation

Custom op that controls animation based on game state:

```
Game State -> Custom Op  
  +-> State = "idle": Play "idle" clip  
  +-> State = "walk": Play "walk" clip  
  +-> State = "run": Play "run" clip  
  +-> State = "jump": Play "jump" clip (one-shot)  
  |  
Selected Clip -> Anim -> Position
```

**Custom Op Code:**

```
const inState = op.inString("State", "idle");  
const inTime = op.inFloat("Time", 0);  
const outClipName = op.outString("Clip Name");  
const outValue = op.outNumber("Animation Value");  
  
let currentClip = null;  
  
inState.onChange = function() {  
  const state = inState.get();  
  switch(state) {  
    case "idle":  
      currentClip = "idle";  
      break;  
    case "walk":  
      currentClip = "walk";  
      break;  
    case "run":  
      currentClip = "run";  
      break;  
  }  
};
```

```
break;  
case "jump":  
  currentClip = "jump";  
  break;  
}  
outClipName.set(currentClip);  
};  
  
inTime.onChange = function() {  
  // Sample the current clip  
  if (currentClip) {  
    const animOp = op.patch.findOpByName("Anim_" + currentClip);  
    if (animOp) {  
      outValue.set(animOp.outValue.get());  
    }  
  }  
};
```

### 9.13.9 Example 9: Physics + Animation Clip Hybrid

Combine procedural physics with keyframed animation:

```
Anim Clip "baseMotion" (keyframed path)  
  |  
  Add  
  Physics Force (gravity, wind) -> Integrate -> Add  
  |  
  Final Position
```

**Custom Op for Physics Integration:**

```
const inAnimValue = op.inFloat("Animation Value", 0);  
const inPhysicsForce = op.inFloat("Physics Force", 0);  
const inDamping = op.inFloat("Damping", 0.95);  
const inRender = op.inTrigger("Render");
```

```

const outCombined = op.outNumber("Combined Value");

let velocity = 0;
let position = 0;

inRender.onTriggered = function() {
    const delta = op.patch.timer.getDelta();
    const anim = inAnimValue.get();
    const force = inPhysicsForce.get();
    const damp = inDamping.get();

    // Update physics
    velocity += force * delta;
    velocity *= damp;
    position += velocity * delta;

    // Combine with animation
    outCombined.set(anim + position);
};


```

### 9.13.10 Example 10: Conditional Clip Playback

Play different clips based on conditions:

```

Condition A -> If (True: Clip A, False: Clip B)
Condition B -> If (True: Clip C, False: Clip D)
|
Mix (blend between conditional results)
|
Final Animation

```

## 9.14 Performance Tips

- 1. Limit active animations** - Don't animate everything
- 2. Use requestAnimationFrame** - Built into cables.gl
- 3. Cache calculations** - Don't recalculate every frame
- 4. Simplify when far** - Reduce animation complexity for distant objects

- 5. Use GPU** - Animate in shaders when possible

## 9.15 Debugging Animation

### 9.15.1 Slow Motion

Time -> Multiply (0.1) -> SlowTime

### 9.15.2 Visualize Values

Add a DrawNumber op to see animated values in real-time.

### 9.15.3 Pause at Problem

Use timeline pause to inspect a specific frame.

## 9.16 Featured Videos

## 9.17 Exercises

### 9.17.1 Basic Animation

- Create a loading animation with staggered dots
- Build an interactive hover animation
- Design a full intro sequence with timeline
- Create a physics-based pendulum

### 9.17.2 Animation Clips

- Create a reusable "bounce" clip and apply it to 5 different objects
- Build a character animation system with 3 additive clips (idle, walk, jump)
- Create a smooth walk-to-run transition using clip blending
- Design a clip library with 5 common animations (fade, slide, scale, rotate, bounce)

### 9.17.3 JavaScript Integration

9. Build a custom op that generates a sine wave animation clip programmatically
10. Create an animation controller op with play/pause/stop/seek functionality
11. Design a custom op that blends two animation clips with a configurable blend factor
12. Build a state machine op that switches between different animation clips based on input

### 9.17.4 Advanced

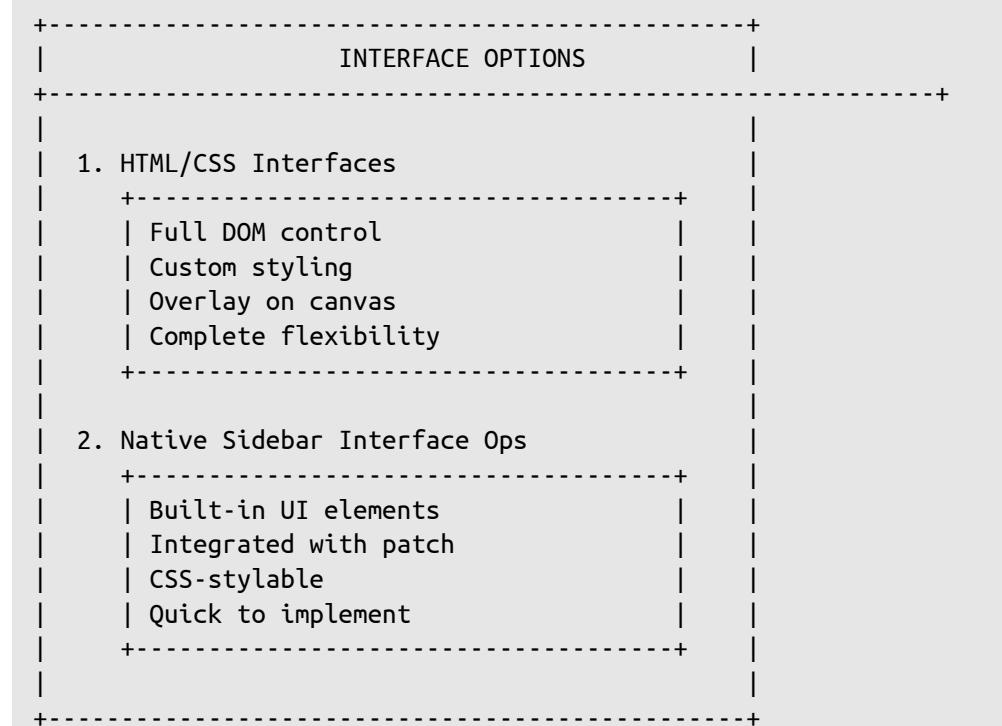
13. Combine procedural animation (Time -> Sin) with a keyframed clip using additive blending
14. Create a custom easing function op and apply it to an animation clip
15. Build a system that plays different animation clips based on user interaction (mouse, keyboard, touch)
16. Design a complex scene with multiple objects, each using a combination of clips and procedural motion

## 10 Interfaces in Cables.gl

### 10.1 Introduction

Cables.gl provides multiple ways to create user interfaces for your patches. You can build interfaces using HTML and CSS for full customization, or use native Cables sidebar interface operators for quick, integrated controls. This chapter covers both approaches in detail.

### 10.2 Interface Approaches Overview



### 10.3 HTML/CSS Interfaces

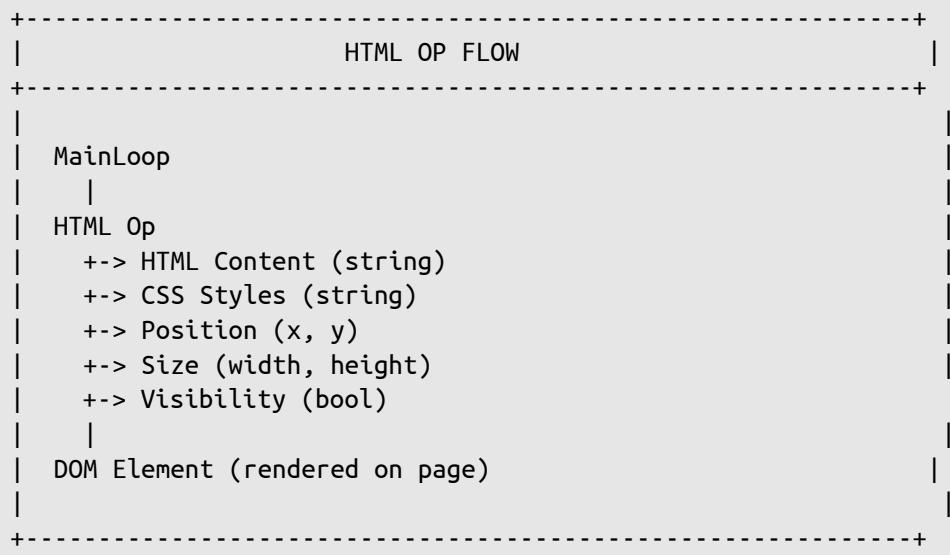
### 10.3.1 Overview

HTML/CSS interfaces give you complete control over the user interface. You can create custom overlays, forms, buttons, and any HTML element positioned over or alongside your canvas.

### 10.3.2 The HTML Op

The HTML op allows you to create and manipulate DOM elements directly within your patch.

#### Basic HTML Op Setup



#### Creating a Simple HTML Interface

##### Step 1: Add HTML Op

1. Add a MainLoop op
2. Add an HTML op
3. Connect MainLoop -> HTML

##### Step 2: Define HTML Content

In the HTML op's "HTML" parameter, enter your HTML:

```
<div id="myInterface">
  <h1>My Interface</h1>
  <button id="myButton">Click Me</button>
  <input type="range" id="mySlider" min="0" max="100" value="50">
  <p id="myText">Value: <span id="valueDisplay">50</span></p>
</div>
```

##### Step 3: Add CSS Styling

In the HTML op's "CSS" parameter:

```
#myInterface {
  position: absolute;
  top: 20px;
  left: 20px;
  background: rgba(30, 30, 30, 0.9);
  padding: 20px;
  border-radius: 8px;
  color: white;
  font-family: Arial, sans-serif;
  z-index: 1000;
}

#myButton {
  background: #4a9eff;
  color: white;
  border: none;
  padding: 10px 20px;
  border-radius: 4px;
  cursor: pointer;
  font-size: 16px;
}

#myButton:hover {
  background: #5aaeff;
```

```

}

#mySlider {
    width: 200px;
    margin: 10px 0;
}

#myText {
    margin-top: 10px;
    font-size: 14px;
}

```

#### Step 4: Position the Interface

Set the HTML op's position parameters: - X: 0 (or desired x position) - Y: 0 (or desired y position) - Width: 300 - Height: 200

### 10.3.3 Connecting HTML to Patch Logic

#### Using JavaScript Custom Op for Interaction

To make HTML elements interactive with your patch, use a JavaScript custom op:

```

// Custom Op: HTML Controller
const inTrigger = op.inTrigger("Render");
const outSliderValue = op.outNumber("Slider Value");
const outButtonClicked = op.outTrigger("Button Clicked");

let sliderValue = 50;
let buttonClicked = false;

// Access DOM elements
op.onInit = function() {
    const slider = document.getElementById("mySlider");
    const button = document.getElementById("myButton");
    const display = document.getElementById("valueDisplay");
}

```

```

if (slider) {
    slider.addEventListener("input", function(e) {
        sliderValue = parseFloat(e.target.value);
        if (display) {
            display.textContent = sliderValue;
        }
        outSliderValue.set(sliderValue);
    });
}

if (button) {
    button.addEventListener("click", function() {
        buttonClicked = true;
        outButtonClicked.trigger();
    });
}

inTrigger.onTriggered = function() {
    outSliderValue.set(sliderValue);
    if (buttonClicked) {
        buttonClicked = false;
    }
};

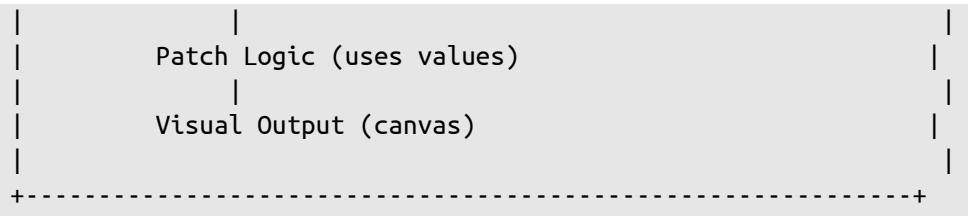
```

#### Complete Example: Interactive Control Panel

```

+-----+
|           INTERACTIVE SETUP          |
+-----+
|
| MainLoop
|   +-> HTML Op (UI elements)
|   +-> Custom Op (JavaScript controller)
|       +-> Reads DOM events
|       +-> Outputs: Slider Value
|       +-> Outputs: Button Trigger
|       +-> Outputs: Text Input
|

```



## HTML Content:

```

<div id="controlPanel">
  <h2>Animation Controls</h2>

  <div class="control-group">
    <label>Speed:</label>
    <input type="range" id="speedSlider" min="0.1" max="5" step="0.1" va
        <span id="speedValue">1.0</span>
  </div>

  <div class="control-group">
    <label>Color:</label>
    <input type="color" id="colorPicker" value="#4a9eff">
  </div>

  <div class="control-group">
    <label>Mode:</label>
    <select id="modeSelect">
      <option value="normal">Normal</option>
      <option value="fast">Fast</option>
      <option value="slow">Slow</option>
    </select>
  </div>

  <button id="resetButton">Reset</button>
  <button id="playButton">Play/Pause</button>
</div>
  
```

## CSS Styling:

```

#controlPanel {
  position: fixed;
  top: 20px;
  right: 20px;
  width: 280px;
  background: linear-gradient(135deg, #1e1e1e 0%, #2d2d2d 100%);
  padding: 24px;
  border-radius: 12px;
  box-shadow: 0 8px 32px rgba(0, 0, 0, 0.4);
  border: 1px solid rgba(255, 255, 255, 0.1);
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  z-index: 1000;
}

#controlPanel h2 {
  margin: 0 0 20px 0;
  color: #ffffff;
  font-size: 20px;
  font-weight: 600;
  border-bottom: 2px solid #4a9eff;
  padding-bottom: 10px;
}

.control-group {
  margin-bottom: 20px;
}

.control-group label {
  display: block;
  color: #b0b0b0;
  font-size: 14px;
  margin-bottom: 8px;
  font-weight: 500;
}

#speedSlider {
  width: 100%;
  height: 6px;
}
  
```

```

border-radius: 3px;
background: #3a3a3a;
outline: none;
-webkit-appearance: none;
}

#speedSlider::-webkit-slider-thumb {
    -webkit-appearance: none;
    appearance: none;
    width: 18px;
    height: 18px;
    border-radius: 50%;
    background: #4a9eff;
    cursor: pointer;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.3);
}

#speedSlider::-moz-range-thumb {
    width: 18px;
    height: 18px;
    border-radius: 50%;
    background: #4a9eff;
    cursor: pointer;
    border: none;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.3);
}

#speedValue {
    color: #4a9eff;
    font-weight: 600;
    margin-left: 10px;
}

#colorPicker {
    width: 100%;
    height: 40px;
    border: 2px solid #3a3a3a;
    border-radius: 6px;
    cursor: pointer;
}

background: transparent;
}

#modeSelect {
    width: 100%;
    padding: 10px;
    background: #3a3a3a;
    color: #ffffff;
    border: 2px solid #3a3a3a;
    border-radius: 6px;
    font-size: 14px;
    cursor: pointer;
}

#modeSelect:hover {
    border-color: #4a9eff;
}

#modeSelect:focus {
    outline: none;
    border-color: #4a9eff;
}

button {
    width: 100%;
    padding: 12px;
    margin-top: 10px;
    background: #4a9eff;
    color: white;
    border: none;
    border-radius: 6px;
    font-size: 14px;
    font-weight: 600;
    cursor: pointer;
    transition: all 0.2s ease;
}

button:hover {
    background: #5aaeff;
}

```

```
    transform: translateY(-1px);
    box-shadow: 0 4px 12px rgba(74, 158, 255, 0.3);
}

button:active {
    transform: translateY(0);
}
```

### 10.3.4 Advanced HTML Interface Patterns

#### Pattern 1: Responsive Overlay

```
#myInterface {
    position: fixed;
    top: 0;
    left: 0;
    width: 100vw;
    height: 100vh;
    background: rgba(0, 0, 0, 0.8);
    display: flex;
    align-items: center;
    justify-content: center;
    z-index: 10000;
}

#myInterface .content {
    background: #2d2d2d;
    padding: 40px;
    border-radius: 12px;
    max-width: 500px;
    width: 90%;
}
```

#### Pattern 2: Sidebar Panel

```
#sidebar {
```

```
position: fixed;
top: 0;
right: 0;
width: 300px;
height: 100vh;
background: #1e1e1e;
box-shadow: -4px 0 16px rgba(0, 0, 0, 0.3);
padding: 20px;
overflow-y: auto;
z-index: 1000;
transform: translateX(0);
transition: transform 0.3s ease;
}

#sidebar.hidden {
    transform: translateX(100%);
}
```

#### Pattern 3: HUD (Heads-Up Display)

```
#hud {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    pointer-events: none;
    z-index: 100;
}

#hud .info {
    position: absolute;
    top: 20px;
    left: 20px;
    color: white;
    font-family: monospace;
    font-size: 14px;
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.8);
```

```
}
```

```
#hud .crosshair {  
    position: absolute;  
    top: 50%;  
    left: 50%;  
    transform: translate(-50%, -50%);  
    width: 20px;  
    height: 20px;  
    border: 2px solid rgba(255, 255, 255, 0.5);  
    border-radius: 50%;  
}
```

## 10.4 Native Sidebar Interface Ops

### 10.4.1 Overview

Cables.gl provides native interface operators that create UI elements directly in the sidebar. These are faster to set up and integrate seamlessly with the patch system.

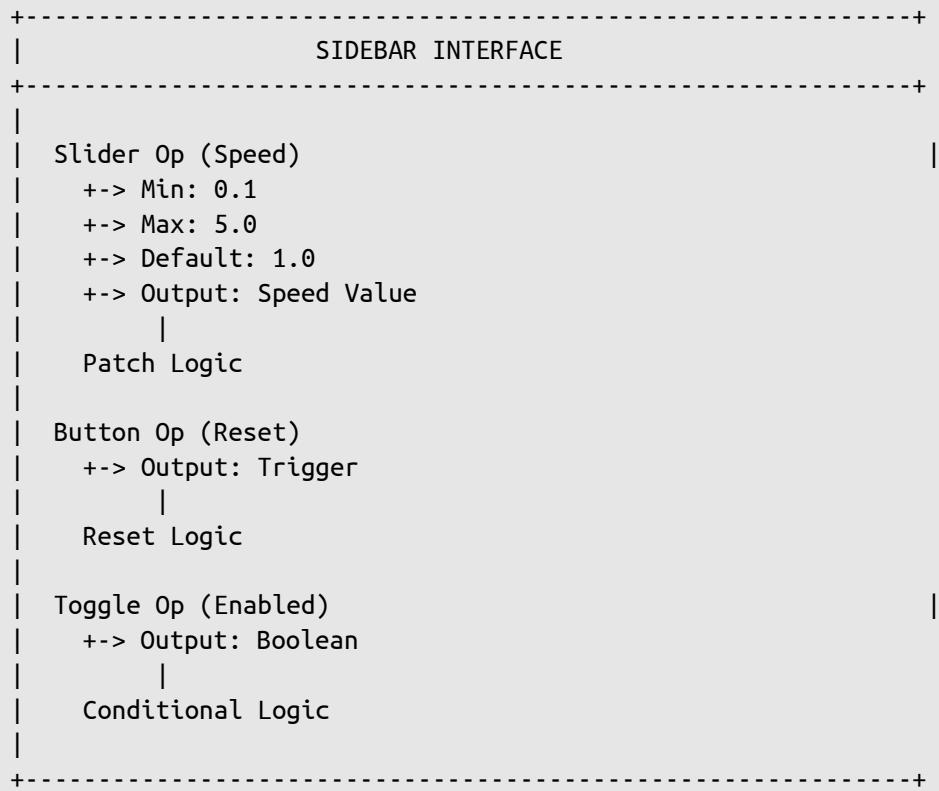
### 10.4.2 Available Interface Ops

#### NATIVE INTERFACE OPS

- Slider      - Numeric input with range
- Button      - Clickable trigger
- Toggle      - Boolean on/off switch
- Text Input   - String input field
- Color Picker - Color selection
- Dropdown     - Selection from options
- Number Input - Direct numeric input
- Text Display  - Display text/values

### 10.4.3 Basic Interface Op Setup

#### Example: Simple Control Panel



#### Step-by-Step: Creating a Sidebar Interface

##### Step 1: Add Interface Ops

1. Add a Slider op for speed control
2. Add a Button op for actions
3. Add a Toggle op for enable/disable
4. Add a ColorPicker op for color selection

##### Step 2: Configure Each Op

**Slider Op:** - Name: "Speed" - Min: 0.1 - Max: 5.0 - Default: 1.0 - Step: 0.1

**Button Op:** - Name: "Reset" - Label: "Reset Animation"

**Toggle Op:** - Name: "Enabled" - Default: true

**ColorPicker Op:** - Name: "Base Color" - Default: #4a9eff

### Step 3: Connect to Patch

```
Speed Slider -> Multiply -> Animation Speed  
Reset Button -> SetValue -> Reset Position  
Enabled Toggle -> If -> Conditional Execution  
ColorPicker -> SetColor -> Material Color
```

## 10.4.4 Styling Native Sidebar with CSS

This is a powerful technique that allows you to customize the appearance of native sidebar interface ops using CSS.

### Understanding the Sidebar Structure

The sidebar interface ops render in a specific DOM structure that you can target with CSS:

```
+-----+  
|       SIDEBAR DOM STRUCTURE |  
+-----+  
  
<div class="cables-sidebar">  
  <div class="cables-sidebar-content">  
    <div class="cables-op-slider" data-op-name="Speed">  
      <label>Speed</label>  
      <input type="range" ...>  
      <span class="value">1.0</span>  
    </div>  
    <div class="cables-op-button" data-op-name="Reset">  
      <button>Reset</button>  
    </div>
```

```
|       ...  
|   </div>  
|   </div>  
|  
+-----+
```

### Method 1: Global CSS Injection

Use an HTML op to inject CSS that styles the entire sidebar:

#### HTML Op Setup:

```
<style id="sidebar-styles">  
/* Sidebar styling will go here */  
</style>
```

#### CSS Content:

```
/* Target the entire sidebar */  
.cables-sidebar {  
  background: linear-gradient(180deg, #1a1a1a 0%, #2d2d2d 100%);  
  border-left: 2px solid #4a9eff;  
}  
  
/* Style all interface ops */  
.cables-sidebar-content > div {  
  background: rgba(255, 255, 255, 0.05);  
  border-radius: 8px;  
  padding: 16px;  
  margin-bottom: 12px;  
  border: 1px solid rgba(255, 255, 255, 0.1);  
  transition: all 0.2s ease;  
}  
  
.cables-sidebar-content > div:hover {  
  background: rgba(255, 255, 255, 0.08);
```

```

border-color: #4a9eff;
}

/* Style slider ops specifically */
.cables-op-slider {
    /* Custom slider container */
}

.cables-op-slider label {
    color: #b0b0b0;
    font-size: 14px;
    font-weight: 600;
    margin-bottom: 8px;
    display: block;
    text-transform: uppercase;
    letter-spacing: 0.5px;
}

.cables-op-slider input[type="range"] {
    width: 100%;
    height: 6px;
    border-radius: 3px;
    background: #3a3a3a;
    outline: none;
    -webkit-appearance: none;
    margin: 10px 0;
}

.cables-op-slider input[type="range"]::-webkit-slider-thumb {
    -webkit-appearance: none;
    appearance: none;
    width: 20px;
    height: 20px;
    border-radius: 50%;
    background: #4a9eff;
    cursor: pointer;
    box-shadow: 0 2px 8px rgba(74, 158, 255, 0.4);
    transition: all 0.2s ease;
}

.cables-op-slider input[type="range"]::-webkit-slider-thumb:hover {
    background: #5aaeff;
    transform: scale(1.1);
    box-shadow: 0 4px 12px rgba(74, 158, 255, 0.6);
}

.cables-op-slider input[type="range"]::-moz-range-thumb {
    width: 20px;
    height: 20px;
    border-radius: 50%;
    background: #4a9eff;
    cursor: pointer;
    border: none;
    box-shadow: 0 2px 8px rgba(74, 158, 255, 0.4);
}

.cables-op-slider .value {
    color: #4a9eff;
    font-weight: 600;
    font-size: 16px;
    float: right;
    margin-top: -24px;
}

/* Style button ops */
.cables-op-button button {
    width: 100%;
    padding: 12px 24px;
    background: linear-gradient(135deg, #4a9eff 0%, #3a8eef 100%);
    color: white;
    border: none;
    border-radius: 6px;
    font-size: 14px;
    font-weight: 600;
    cursor: pointer;
    transition: all 0.2s ease;
    text-transform: uppercase;
}

```

```

letter-spacing: 1px;
box-shadow: 0 4px 12px rgba(74, 158, 255, 0.3);
}

.cables-op-button button:hover {
  background: linear-gradient(135deg, #5aaeff 0%, #4a9eff 100%);
  transform: translateY(-2px);
  box-shadow: 0 6px 16px rgba(74, 158, 255, 0.4);
}

.cables-op-button button:active {
  transform: translateY(0);
  box-shadow: 0 2px 8px rgba(74, 158, 255, 0.3);
}

/* Style toggle ops */
.cables-op-toggle {
  display: flex;
  align-items: center;
  justify-content: space-between;
}

.cables-op-toggle label {
  color: #b0b0b0;
  font-size: 14px;
  font-weight: 500;
}

.cables-op-toggle input[type="checkbox"] {
  width: 50px;
  height: 26px;
  -webkit-appearance: none;
  appearance: none;
  background: #3a3a3a;
  border-radius: 13px;
  position: relative;
  cursor: pointer;
  transition: background 0.3s ease;
  border: 2px solid #2a2a2a;
}

}

.cables-op-toggle input[type="checkbox"]:checked {
  background: #4a9eff;
  border-color: #4a9eff;
}

.cables-op-toggle input[type="checkbox"]::before {
  content: '';
  position: absolute;
  width: 20px;
  height: 20px;
  border-radius: 50%;
  background: white;
  top: 1px;
  left: 1px;
  transition: transform 0.3s ease;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.3);
}

.cables-op-toggle input[type="checkbox"]:checked::before {
  transform: translateX(24px);
}

/* Style color picker ops */
.cables-op-colorpicker {
  display: flex;
  align-items: center;
  gap: 12px;
}

.cables-op-colorpicker label {
  color: #b0b0b0;
  font-size: 14px;
  font-weight: 500;
  flex: 1;
}

.cables-op-colorpicker input[type="color"] {

```

```

width: 60px;
height: 40px;
border: 2px solid #3a3a3a;
border-radius: 6px;
cursor: pointer;
background: transparent;
transition: border-color 0.2s ease;
}

.cables-op-colorpicker input[type="color"]:hover {
    border-color: #4a9eff;
}

/* Style text input ops */
.cables-op-textinput input[type="text"] {
    width: 100%;
    padding: 10px 12px;
    background: #3a3a3a;
    color: #ffffff;
    border: 2px solid #3a3a3a;
    border-radius: 6px;
    font-size: 14px;
    transition: all 0.2s ease;
}

.cables-op-textinput input[type="text"]:focus {
    outline: none;
    border-color: #4a9eff;
    background: #404040;
    box-shadow: 0 0 0 3px rgba(74, 158, 255, 0.1);
}

/* Style dropdown ops */
.cables-op-dropdown select {
    width: 100%;
    padding: 10px 12px;
    background: #3a3a3a;
    color: #ffffff;
    border: 2px solid #3a3a3a;
}

```

```

border-radius: 6px;
font-size: 14px;
cursor: pointer;
transition: all 0.2s ease;
}

.cables-op-dropdown select:hover {
    border-color: #4a9eff;
}

.cables-op-dropdown select:focus {
    outline: none;
    border-color: #4a9eff;
    box-shadow: 0 0 0 3px rgba(74, 158, 255, 0.1);
}

```

## Method 2: Targeted Op Styling

Style specific ops by their data attributes:

```

/* Style a specific slider by op name */
.cables-op-slider[data-op-name="Speed"] {
    background: rgba(74, 158, 255, 0.1);
    border: 2px solid #4a9eff;
}

.cables-op-slider[data-op-name="Speed"] label {
    color: #4a9eff;
    font-weight: 600;
}

/* Style a specific button */
.cables-op-button[data-op-name="Reset"] button {
    background: linear-gradient(135deg, #ff4a4a 0%, #ef3a3a 100%);
}

.cables-op-button[data-op-name="Reset"] button:hover {

```

```
background: linear-gradient(135deg, #ff5a5a 0%, #ff4a4a 100%);  
}
```

### Method 3: Dynamic CSS with JavaScript Custom Op

Create a custom op that injects CSS based on patch state:

```
// Custom Op: Dynamic Sidebar Styling  
const inTheme = op.inSwitch("Theme", ["dark", "light", "neon"], "dark");  
const inAccentColor = op.inString("Accent Color", "#4a9eff");  
  
let currentTheme = "dark";  
let currentAccent = "#4a9eff";  
  
function updateStyles() {  
    const theme = inTheme.get();  
    const accent = inAccentColor.get();  
  
    if (theme === currentTheme && accent === currentAccent) return;  
  
    currentTheme = theme;  
    currentAccent = accent;  
  
    let styleElement = document.getElementById("dynamic-sidebar-styles");  
    if (!styleElement) {  
        styleElement = document.createElement("style");  
        styleElement.id = "dynamic-sidebar-styles";  
        document.head.appendChild(styleElement);  
    }  
  
    let css = "";  
  
    if (theme === "dark") {  
        css = `  
            .cables-sidebar {  
                background: linear-
```

```
gradient(180deg, #1a1a1a 0%, #2d2d2d 100%);  
            }  
            .cables-sidebar-content > div {  
                background: rgba(255, 255, 255, 0.05);  
                border-color: rgba(255, 255, 255, 0.1);  
            }  
            `;  
    } else if (theme === "light") {  
        css = `  
            .cables-sidebar {  
                background: linear-  
gradient(180deg, #f5f5f5 0%, #e0e0e0 100%);  
            }  
            .cables-sidebar-content > div {  
                background: rgba(0, 0, 0, 0.05);  
                border-color: rgba(0, 0, 0, 0.1);  
            }  
            .cables-op-slider label,  
            .cables-op-button label {  
                color: #333;  
            }  
            `;  
    } else if (theme === "neon") {  
        css = `  
            .cables-sidebar {  
                background: #0a0a0a;  
                border-left: 2px solid ${accent};  
                box-shadow: -4px 0 20px ${accent}40;  
            }  
            .cables-sidebar-content > div {  
                background: rgba(0, 0, 0, 0.5);  
                border: 1px solid ${accent}40;  
                box-shadow: 0 0 10px ${accent}20;  
            }  
            `;  
    }  
  
    // Apply accent color
```

```

css += `
    .cables-op-slider input[type="range"]::-webkit-slider-thumb {
        background: ${accent};
        box-shadow: 0 2px 8px ${accent}60;
    }
    .cables-op-button button {
        background: linear-gradient(135deg, ${accent} 0%, ${adjustBrightness(accent, -20)} 100%);
    }
    .cables-op-toggle input[type="checkbox"]:checked {
        background: ${accent};
    }
`;

styleElement.textContent = css;
}

function adjustBrightness(color, percent) {
    // Simple brightness adjustment (simplified)
    const num = parseInt(color.replace("#", ""), 16);
    const r = Math.max(0, Math.min(255, (num >> 16) + percent));
    const g = Math.max(0, Math.min(255, ((num >> 8) & 0x00FF) + percent));
    const b = Math.max(0, Math.min(255, (num & 0x0000FF) + percent));
    return "#" + ((r << 16) | (g << 8) | b).toString(16).padStart(6, "0");
}

inTheme.onChange = updateStyles;
inAccentColor.onChange = updateStyles;

op.onInit = function() {
    updateStyles();
};

```

## 10.4.5 Complete Styling Example: Professional Control Panel

Here's a complete example that styles all interface ops with a cohesive, professional design:

### HTML Op (CSS Injection):

```

<style id="professional-sidebar-styles">
/* Professional Sidebar Styling */

/* Sidebar Container */
.cables-sidebar {
    background: linear-gradient(180deg,
        #1a1a1a 0%,
        #1e1e1e 50%,
        #2d2d2d 100%);
    border-left: 3px solid #4a9eff;
    box-shadow: -4px 0 24px rgba(0, 0, 0, 0.5);
    font-family: 'Inter', 'Segoe UI', system-ui, sans-serif;
}

/* Sidebar Header (if exists) */
.cables-sidebar-header {
    padding: 20px;
    border-bottom: 2px solid rgba(74, 158, 255, 0.2);
    background: rgba(74, 158, 255, 0.05);
}

.cables-sidebar-header h2 {
    margin: 0;
    color: #ffffff;
    font-size: 18px;
    font-weight: 600;
    text-transform: uppercase;
    letter-spacing: 1px;
}

/* Content Container */

```

```

.cables-sidebar-content {
    padding: 16px;
}

/* All Interface Op Containers */
.cables-sidebar-content > div {
    background: rgba(255, 255, 255, 0.03);
    border: 1px solid rgba(255, 255, 255, 0.08);
    border-radius: 10px;
    padding: 18px;
    margin-bottom: 16px;
    transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
    position: relative;
    overflow: hidden;
}

.cables-sidebar-content > div::before {
    content: '';
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 2px;
    background: linear-gradient(90deg,
        transparent 0%,
        #4a9eff 50%,
        transparent 100%);
    opacity: 0;
    transition: opacity 0.3s ease;
}

.cables-sidebar-content > div:hover {
    background: rgba(255, 255, 255, 0.06);
    border-color: rgba(74, 158, 255, 0.3);
    transform: translateX(4px);
    box-shadow: 0 4px 16px rgba(0, 0, 0, 0.3);
}

.cables-sidebar-content > div:hover::before {
    opacity: 1;
}

/* Slider Styling */
.cables-op-slider label {
    display: block;
    color: #b0b0b0;
    font-size: 12px;
    font-weight: 600;
    margin-bottom: 10px;
    text-transform: uppercase;
    letter-spacing: 0.5px;
}

.cables-op-slider input[type="range"] {
    width: 100%;
    height: 8px;
    border-radius: 4px;
    background: linear-gradient(90deg,
        #2a2a2a 0%,
        #3a3a3a 100%);
    outline: none;
    -webkit-appearance: none;
    margin: 12px 0;
    position: relative;
}

.cables-op-slider input[type="range"]::-webkit-slider-thumb {
    -webkit-appearance: none;
    appearance: none;
    width: 24px;
    height: 24px;
    border-radius: 50%;
    background: linear-gradient(135deg, #4a9eff 0%, #3a8eef 100%);
    cursor: pointer;
    box-shadow:
        0 2px 8px rgba(74, 158, 255, 0.4),
        0 0 0 4px rgba(74, 158, 255, 0.1),
        inset 0 1px 0 rgba(255, 255, 255, 0.2);
}

```

```

transition: all 0.2s ease;
border: 2px solid rgba(255, 255, 255, 0.1);
}

.cables-op-slider           input[type="range"]::-webkit-slider-
thumb:hover {
  background: linear-gradient(135deg, #5aaeff 0%, #4a9eff 100%);
  transform: scale(1.15);
  box-shadow:
    0 4px 12px rgba(74, 158, 255, 0.6),
    0 0 6px rgba(74, 158, 255, 0.15),
    inset 0 1px 0 rgba(255, 255, 255, 0.3);
}

.cables-op-slider           input[type="range"]::-webkit-slider-
thumb:active {
  transform: scale(1.05);
}

.cables-op-slider input[type="range"]::-moz-range-thumb {
  width: 24px;
  height: 24px;
  border-radius: 50%;
  background: linear-gradient(135deg, #4a9eff 0%, #3a8eef 100%);
  cursor: pointer;
  border: 2px solid rgba(255, 255, 255, 0.1);
  box-shadow:
    0 2px 8px rgba(74, 158, 255, 0.4),
    0 0 0 4px rgba(74, 158, 255, 0.1);
}

.cables-op-slider .value {
  color: #4a9eff;
  font-weight: 700;
  font-size: 18px;
  float: right;
  margin-top: -32px;
  font-variant-numeric: tabular-nums;
  text-shadow: 0 0 8px rgba(74, 158, 255, 0.5);
}

}

/* Button Styling */
.cables-op-button button {
  width: 100%;
  padding: 14px 24px;
  background: linear-gradient(135deg, #4a9eff 0%, #3a8eef 100%);
  color: white;
  border: none;
  border-radius: 8px;
  font-size: 14px;
  font-weight: 600;
  cursor: pointer;
  transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
  text-transform: uppercase;
  letter-spacing: 1.2px;
  box-shadow:
    0 4px 12px rgba(74, 158, 255, 0.3),
    inset 0 1px 0 rgba(255, 255, 255, 0.2);
  position: relative;
  overflow: hidden;
}

.cables-op-button button::before {
  content: '';
  position: absolute;
  top: 50%;
  left: 50%;
  width: 0;
  height: 0;
  border-radius: 50%;
  background: rgba(255, 255, 255, 0.3);
  transform: translate(-50%, -50%);
  transition: width 0.6s, height 0.6s;
}

.cables-op-button button:hover {
  background: linear-gradient(135deg, #5aaeff 0%, #4a9eff 100%);
  transform: translateY(-2px);
}

```

```

box-shadow:
  0 6px 20px rgba(74, 158, 255, 0.4),
  inset 0 1px 0 rgba(255, 255, 255, 0.3);
}

.cables-op-button button:hover::before {
  width: 300px;
  height: 300px;
}

.cables-op-button button:active {
  transform: translateY(0);
  box-shadow:
    0 2px 8px rgba(74, 158, 255, 0.3),
    inset 0 1px 0 rgba(255, 255, 255, 0.1);
}

/* Toggle Styling */
.cables-op-toggle {
  display: flex;
  align-items: center;
  justify-content: space-between;
}

.cables-op-toggle label {
  color: #b0b0b0;
  font-size: 14px;
  font-weight: 500;
  flex: 1;
}

.cables-op-toggle input[type="checkbox"] {
  width: 56px;
  height: 30px;
  -webkit-appearance: none;
  appearance: none;
  background: #2a2a2a;
  border-radius: 15px;
  position: relative;
}

.cables-op-button button {
  cursor: pointer;
  transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
  border: 2px solid #1a1a1a;
  box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.3);
}

.cables-op-toggle input[type="checkbox"]:checked {
  background: linear-gradient(135deg, #4a9eff 0%, #3a8eef 100%);
  border-color: #4a9eff;
  box-shadow:
    inset 0 2px 4px rgba(0, 0, 0, 0.2),
    0 0 12px rgba(74, 158, 255, 0.4);
}

.cables-op-toggle input[type="checkbox"]::before {
  content: '';
  position: absolute;
  width: 24px;
  height: 24px;
  border-radius: 50%;
  background: linear-gradient(135deg, #ffffff 0%, #f0f0f0 100%);
  top: 1px;
  left: 1px;
  transition: transform 0.3s cubic-bezier(0.4, 0, 0.2, 1);
  box-shadow:
    0 2px 6px rgba(0, 0, 0, 0.3),
    inset 0 1px 0 rgba(255, 255, 255, 0.5);
}

.cables-op-toggle input[type="checkbox"]::checked::before {
  transform: translateX(26px);
}

/* Color Picker Styling */
.cables-op-colorpicker {
  display: flex;
  align-items: center;
  gap: 16px;
}

```

```

.cables-op-colorpicker label {
  color: #b0b0b0;
  font-size: 14px;
  font-weight: 500;
  flex: 1;
}

.cables-op-colorpicker input[type="color"] {
  width: 70px;
  height: 50px;
  border: 3px solid #3a3a3a;
  border-radius: 8px;
  cursor: pointer;
  background: transparent;
  transition: all 0.3s ease;
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.3);
}

.cables-op-colorpicker input[type="color"]:hover {
  border-color: #4a9eff;
  transform: scale(1.05);
  box-shadow:
    0 4px 12px rgba(0, 0, 0, 0.4),
    0 0 0 4px rgba(74, 158, 255, 0.1);
}

/* Text Input Styling */
.cables-op-textinput label {
  display: block;
  color: #b0b0b0;
  font-size: 12px;
  font-weight: 600;
  margin-bottom: 8px;
  text-transform: uppercase;
  letter-spacing: 0.5px;
}

.cables-op-textinput input[type="text"] {
  width: 100%;
  padding: 12px 16px;
  background: #2a2a2a;
  color: #ffffff;
  border: 2px solid #3a3a3a;
  border-radius: 8px;
  font-size: 14px;
  transition: all 0.3s ease;
  box-sizing: border-box;
}

.cables-op-textinput input[type="text"]:focus {
  outline: none;
  border-color: #4a9eff;
  background: #333333;
  box-shadow:
    0 0 0 4px rgba(74, 158, 255, 0.1),
    inset 0 2px 4px rgba(0, 0, 0, 0.2);
}

/* Dropdown Styling */
.cables-op-dropdown label {
  display: block;
  color: #b0b0b0;
  font-size: 12px;
  font-weight: 600;
  margin-bottom: 8px;
  text-transform: uppercase;
  letter-spacing: 0.5px;
}

.cables-op-dropdown select {
  width: 100%;
  padding: 12px 16px;
  background: #2a2a2a;
  color: #ffffff;
  border: 2px solid #3a3a3a;
  border-radius: 8px;
  font-size: 14px;
}

```

```
background: #2a2a2a;
color: #ffffff;
border: 2px solid #3a3a3a;
border-radius: 6px;
font-size: 14px;
text-align: center;
transition: all 0.3s ease;
}

.cables-op-numberinput input[type="number"]:focus {
outline: none;
border-color: #4a9eff;
background: #333333;
box-shadow: 0 0 0 3px rgba(74, 158, 255, 0.1);
}

/* Text Display Styling */
.cables-op-textdisplay {
padding: 12px;
background: rgba(74, 158, 255, 0.1);
border: 1px solid rgba(74, 158, 255, 0.3);
border-radius: 6px;
color: #4a9eff;
font-family: 'Courier New', monospace;
font-size: 14px;
text-align: center;
font-weight: 600;
}

/* Responsive adjustments */
@media (max-width: 768px) {
.cables-sidebar {
width: 100% !important;
height: auto !important;
position: relative !important;
}
}

</style>
```

## 10.4.6 Advanced CSS Techniques

### Technique 1: Animated Transitions

```
.cables-sidebar-content > div {  
    animation: slideIn 0.3s ease-out;  
}  
  
@keyframes slideIn {  
    from {  
        opacity: 0;  
        transform: translateX(-20px);  
    }  
    to {  
        opacity: 1;  
        transform: translateX(0);  
    }  
}  
  
/* Stagger animation delays */  
.cables-sidebar-content > div:nth-child(1) { animation-delay: 0.05s; }  
.cables-sidebar-content > div:nth-child(2) { animation-delay: 0.10s; }  
.cables-sidebar-content > div:nth-child(3) { animation-delay: 0.15s; }  
.cables-sidebar-content > div:nth-child(4) { animation-delay: 0.20s; }
```

### Technique 2: Custom Scrollbar

```
.cables-sidebar-content::-webkit-scrollbar {  
    width: 8px;  
}  
  
.cables-sidebar-content::-webkit-scrollbar-track {  
    background: #1a1a1a;  
    border-radius: 4px;
```

```
}  
  
.cables-sidebar-content::-webkit-scrollbar-thumb {  
    background: #4a9eff;  
    border-radius: 4px;  
    border: 2px solid #1a1a1a;  
}  
  
.cables-sidebar-content::-webkit-scrollbar-thumb:hover {  
    background: #5aaeff;
```

### Technique 3: Glassmorphism Effect

```
.cables-sidebar {  
    background: rgba(30, 30, 30, 0.7);  
    backdrop-filter: blur(20px);  
    -webkit-backdrop-filter: blur(20px);  
    border-left: 1px solid rgba(255, 255, 255, 0.1);  
}  
  
.cables-sidebar-content > div {  
    background: rgba(255, 255, 255, 0.05);  
    backdrop-filter: blur(10px);  
    -webkit-backdrop-filter: blur(10px);  
    border: 1px solid rgba(255, 255, 255, 0.1);  
}
```

## 10.5 Combining HTML and Native Interfaces

You can combine both approaches for maximum flexibility:



```

| Native Sidebar Ops
|   +-> Quick controls (sliders, buttons)
|   +-> Styled with CSS
|
| HTML Overlay
|   +-> Complex UI elements
|   +-> Custom layouts
|   +-> Interactive components
|
| JavaScript Custom Op
|   +-> Bridges both systems
|   +-> Syncs values
|   +-> Handles interactions
|
+-----+

```

## 10.6 Best Practices

### 10.6.1 1. Performance

- **Minimize DOM Manipulation:** Cache element references
- **Use CSS Transforms:** For animations instead of position changes
- **Debounce Inputs:** For sliders and text inputs that trigger heavy computations

### 10.6.2 2. Accessibility

- **Labels:** Always provide clear labels for controls
- **Keyboard Navigation:** Ensure keyboard accessibility
- **Color Contrast:** Maintain sufficient contrast ratios
- **Focus States:** Provide visible focus indicators

### 10.6.3 3. Responsive Design

```

/* Mobile-first approach */
.cables-sidebar {
  width: 100%;
  height: auto;
}

```

```

  position: relative;
}

@media (min-width: 768px) {
  .cables-sidebar {
    width: 320px;
    height: 100vh;
    position: fixed;
  }
}

```

### 10.6.4 4. Organization

- **Group Related Controls:** Use visual grouping
- **Clear Hierarchy:** Use size, color, and spacing
- **Consistent Spacing:** Maintain uniform margins and padding

## 10.7 Practical Examples

### 10.7.1 Example 1: Animation Control Panel

Create a comprehensive control panel for animation parameters:

```

Speed Slider -> Animation Speed
Color Picker -> Material Color
Toggle (Loop) -> Loop Animation
Button (Reset) -> Reset Animation
Text Display -> Current Frame

```

### 10.7.2 Example 2: Game UI Overlay

HTML overlay for game-like interface:

```

<div id="gameUI">
  <div class="hud-top">
    <div class="score">Score: <span id="score">0</span></div>
    <div class="health">Health: <span id="health">100</span></div>
  </div>
  <div class="hud-bottom">
    <button id="pauseBtn">Pause</button>
    <button id="menuBtn">Menu</button>
  </div>
</div>

```

### 10.7.3 Example 3: Data Visualization Dashboard

Combine native ops with HTML for a data dashboard:

- Native sliders for filtering
- HTML charts and graphs
- Real-time data display

## 10.8 Debugging Interface Issues

### 10.8.1 Common Issues

#### 1. CSS Not Applying

- Check selector specificity
- Verify CSS is injected after sidebar renders
- Use !important sparingly

#### 2. Elements Not Visible

- Check z-index values
- Verify position properties
- Check for overflow: hidden

#### 3. Events Not Firing

- Ensure JavaScript runs after DOM is ready
- Check event listener attachment
- Verify element selectors

### 10.8.2 Debugging Tools

```

// Log sidebar structure
console.log(document.querySelector('.cables-sidebar'));

// Check computed styles
const element = document.querySelector('.cables-op-slider');
console.log(window.getComputedStyle(element));

// Monitor style changes
const observer = new MutationObserver((mutations) => {
  console.log('DOM changed:', mutations);
});
observer.observe(document.querySelector('.cables-sidebar'), {
  childList: true,
  subtree: true,
  attributes: true
});

```

## 10.9 Exercises

- Basic HTML Interface:** Create a simple HTML overlay with a button and slider that control patch parameters
- Styled Sidebar:** Style native sidebar ops with a cohesive color scheme and modern design
- Responsive Panel:** Create a sidebar that adapts to different screen sizes
- Interactive Dashboard:** Build a complete control panel combining HTML and native ops
- Theme Switcher:** Create a custom op that dynamically changes sidebar styling based on user selection
- Advanced Styling:** Implement glassmorphism or other modern design trends in your sidebar

# 11 Export & Deployment in Cables.gl

## 11.1 Introduction

Once you've created your cables.gl project, you'll want to share it with the world. This chapter covers all the ways to export and deploy your creations.

## 11.2 Export Options

### 11.2.1 1. Public Patch Link

The simplest way to share - just make your patch public and share the URL.

**Pros:** - Instant sharing - Always up-to-date - No hosting needed

**Cons:** - Requires internet - Cables.gl branding - Limited customization

### 11.2.2 2. Embedded iframe

Embed your patch in any website:

```
<iframe  
  src="https://cables.gl/view/YOUR_PATCH_ID"  
  width="800"  
  height="600"  
  frameborder="0"  
  allowfullscreen>  
</iframe>
```

### 11.2.3 3. Standalone Export

Download your patch as a standalone web application.

**Includes:** - HTML file - JavaScript bundle - Assets (textures, models, audio) - No cables.gl dependency

### 11.2.4 4. npm Package Export

Export as an npm package for integration with other JavaScript projects.

## 11.3 Standalone Export Process

### 11.3.1 Step 1: Prepare Your Patch

1. Test thoroughly in the editor
2. Optimize assets (compress images, reduce model complexity)
3. Remove unused ops and connections
4. Set default camera/view position

### 11.3.2 Step 2: Export

1. Click the export/download button in the editor
2. Choose "Standalone" export
3. Configure options:
  - Include minified code
  - Include source maps (for debugging)
  - Asset optimization level

### 11.3.3 Step 3: Download

You'll receive a ZIP file containing:

```
exported-patch/  
++ index.html          # Main HTML file  
++ js/  
|  +- cables.min.js   # Cables runtime  
|  +- ops.js           # Your patch's operators  
|  +- patch.js         # Patch configuration  
++ assets/  
|  +- textures/        # Image files  
|  +- audio/            # Sound files  
|  +- models/           # 3D models
```

```
+-- css/  
  +- style.css      # Optional styles
```

### 11.3.4 Step 4: Test Locally

```
# Using Python  
python -m http.server 8000  
  
# Using Node.js  
npx serve .  
  
# Using PHP  
php -S localhost:8000
```

Then open <http://localhost:8000> in your browser.

## 11.4 Customizing the Export

### 11.4.1 Custom HTML Template

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width, initial-  
scale=1">  
    <title>My Cables Project</title>  
    <style>  
      body { margin: 0; overflow: hidden; }  
      #cables-container { width: 100vw; height: 100vh; }  
    </style>  
  </head>  
  <body>  
    <div id="cables-container"></div>  
  
    <script src="js/cables.min.js"></script>  
    <script src="js/ops.js"></script>
```

```
<script>  
  CABLES.patch = new CABLES.Patch({  
    patchFile: 'js/patch.js',  
    prefixAssetPath: 'assets/',  
    glCanvasId: 'cables-container',  
    onFinishedLoading: function() {  
      console.log('Patch loaded!');  
    }  
  });  
</script>  
</body>  
</html>
```

### 11.4.2 Configuration Options

```
new CABLES.Patch({  
  patchFile: 'js/patch.js',  
  prefixAssetPath: 'assets/',  
  glCanvasId: 'myCanvas',  
  glCanvasResizeToWindow: true,  
  onFinishedLoading: callback,  
  onError: errorCallback,  
  variables: {  
    // Pass custom variables to the patch  
    customColor: '#ff0000',  
    userName: 'Guest'  
  }  
});
```

## 11.5 Communicating with Your Patch

### 11.5.1 Setting Variables from JavaScript

```
// Get the patch instance  
const patch = CABLES.patch;
```

```
// Set a variable  
patch.setVariable('myValue', 42);  
patch.setVariable('myColor', [1, 0, 0, 1]);
```

## 11.5.2 Getting Values from the Patch

```
// Get a variable  
const value = patch.getVariable('myValue');  
  
// Listen for variable changes  
patch.on('variableChanged', function(name, value) {  
  console.log(name, 'changed to', value);  
});
```

## 11.5.3 Triggering Events

```
// Trigger an op  
patch.getOpById('YOUR_OP_ID').trigger();  
  
// Or use variables as triggers  
patch.setVariable('doSomething', true);
```

# 11.6 Advanced Embedding & Integration

When cables.gl becomes part of a larger website/app, you want the embed to be **robust**:

- correct sizing and device pixel ratio handling
- pause/resume behavior when the tab is hidden
- a clean integration API (events in, telemetry out)
- predictable asset paths across dev/staging/prod

## 11.6.1 Responsive Canvas: Beyond Width/Height

If you embed into dynamic layouts (resizable panels, CSS grid, etc.), treat `resize` as a first-class event:

- call your `resize` function on load
- call it on `resize`
- call it when layout changes (route changes, UI toggles, etc.)

## 11.6.2 Pausing When Not Visible

For performance and battery life, consider pausing expensive animation when the page is hidden:

```
document.addEventListener("visibilitychange", () => {  
  if (!window.CABLES || !CABLES.patch) return;  
  // Depending on your patch/runtime, you may gate updates via a variable:  
  CABLES.patch.setVariable("isVisible", !document.hidden);  
});
```

Then in your patch, use `isVisible` to reduce workload (lower particle count, skip effects, etc.).

## 11.6.3 postMessage Integration (iframe Control)

If you embed via iframe, `postMessage` is the clean way to send commands and data.

**Parent page -> iframe:**

```
const iframe = document.getElementById("cablesFrame");  
iframe.contentWindow.postMessage(  
  { type: "CABLES_SET", name: "myValue", value: 0.75 },  
  "*"  
);
```

**Inside the exported patch wrapper page:**

```

window.addEventListener("message", (event) => {
  const msg = event.data;
  if (!msg || !window.CABLES || !CABLES.patch) return;

  if (msg.type === "CABLES_SET") {
    CABLES.patch.setVariable(msg.name, msg.value);
  }
});

```

## 11.6.4 Environment-Specific Configuration (dev / test / prod)

Keep environment differences in **configuration**, not in the patch logic:

- dev: verbose logging, source maps, local asset path
- test/staging: production-like hosting + debug overlays
- prod: minified, caching enabled, stable URLs

Common patterns:

- query string flags: ?debug=1
- separate config.json loaded at runtime
- environment variables handled by the site that embeds the patch

## 11.6.5 Asset Path Gotchas

Most “works locally but not in prod” issues come down to:

- wrong prefixAssetPath
- case-sensitive paths on Linux hosts
- missing assets in the exported zip upload

If you deploy under a sub-path (e.g., <https://site.com/myproject/>), ensure all paths are relative or correctly prefixed.

# 11.7 Hosting Options

## 11.7.1 Static Hosting

Your exported patch is static files - host anywhere:

- **GitHub Pages** - Free, great for projects
- **Netlify** - Free tier, easy deployment
- **Vercel** - Free tier, automatic deploys
- **Amazon S3** - Scalable, pay-per-use
- **Any web server** - Apache, Nginx, etc.

## 11.7.2 GitHub Pages Deployment

```

# Create a gh-pages branch
git checkout -b gh-pages

# Add your exported files
git add .
git commit -m "Deploy cables patch"

# Push to GitHub
git push origin gh-pages

```

Enable GitHub Pages in repository settings.

## 11.7.3 Netlify Deployment

1. Connect your GitHub repository
2. Set build command: (none needed for static)
3. Set publish directory: / or your export folder
4. Deploy!

# 11.8 Embedding in Existing Websites

## 11.8.1 As a Background

```

<style>
  #cables-bg {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;

```

```

        height: 100%;
        z-index: -1;
    }

```

```

</style>
<canvas id="cables-bg"></canvas>
<script>
    CABLES.patch = new CABLES.Patch({
        patchFile: 'patch.js',
        glCanvasId: 'cables-bg'
    });
</script>

```

## 11.8.2 As a Hero Section

```

<section class="hero">
    <div id="cables-hero"></div>
    <div class="hero-content">
        <h1>Welcome</h1>
        <p>Your content here</p>
    </div>
</section>

```

## 11.8.3 Responsive Embedding

```

function resizeCables() {
    const container = document.getElementById('cables-container');
    container.style.width = window.innerWidth + 'px';
    container.style.height = window.innerHeight + 'px';

    // Notify cables of resize
    if (CABLES.patch) {
        CABLES.patch.cgl.setSize(window.innerWidth, window.innerHeight);
    }
}

window.addEventListener('resize', resizeCables);

```

```
resizeCables();
```

## 11.9 Performance Optimization

### 11.9.1 Before Export

- Remove unused ops** - Clean up your patch
- Optimize textures** - Use appropriate sizes
- Reduce polygon count** - Simplify 3D models
- Minimize audio files** - Compress audio

### 11.9.2 Asset Optimization

**Images:** - Use WebP format when possible - Use power-of-2 dimensions - Compress with tools like TinyPNG

**3D Models:** - Use glTF/GLB format - Remove unnecessary detail - Use Draco compression

**Audio:** - Use MP3 or OGG - Compress appropriately - Consider streaming for long files

### 11.9.3 Loading Optimization

```

// Show loading progress
CABLES.patch = new CABLES.Patch({
    patchFile: 'patch.js',
    onLoadingProgress: function(percent) {
        document.getElementById('loader').style.width = percent + '%';
    },
    onFinishedLoading: function() {
        document.getElementById('loader').style.display = 'none';
    }
});

```

## 11.10 Deployment Checklist (The Stuff That Breaks at the Worst Time)

Before you publish, run through this list:

- **Loading:** Do you show a loader/progress bar for heavy patches?
- **Autoplay policies:** If you use audio/video/webcam, do you require a user click?
- **Mobile sanity:** Does it run on a mid-tier phone without overheating?
- **Resize:** Does it handle orientation changes and dynamic layout resizing?
- **Asset paths:** Are all assets included and paths correct on a case-sensitive host?
- **Cache behavior:** Are you accidentally serving old JS after updates?
- **Console:** Is the browser console clean (no noisy logs, no repeated warnings)?

### 11.10.1 Cache Busting and Versioning

Static hosts cache aggressively. If you deploy a new version and still see the old one:

- add a version/hash to filenames (e.g. ops.v123.js)
- or configure cache headers (short cache for HTML, long cache for hashed assets)

### 11.10.2 MIME Types (Especially for Wasm / Binary Assets)

Some servers mis-serve file types. If a resource fails to load, check response headers:

- .wasm should be served as application/wasm
- .json as application/json
- textures as correct image mime types

### 11.10.3 CORS (Cross-Origin Assets)

If you load assets from another domain:

- ensure that server sends correct CORS headers

- prefer hosting assets alongside the patch when possible (simpler)

### 11.10.4 Content Security Policy (CSP)

If your patch is embedded into a site with strict CSP, you may need to allow:

- fetching assets from required domains
- media playback sources

When possible, avoid “unsafe-inline” and instead rely on your host app’s approved patterns.

## 11.11 CI/CD Ideas (Optional, But Great for Teams)

If you repeatedly export and deploy:

- treat the export zip as a build artifact
- deploy to staging on every change
- promote to prod when approved

Even a simple workflow that publishes static files to GitHub Pages can save time and reduce mistakes.

## 11.12 Offline/PWA

Make your patch work offline as a Progressive Web App:

### 11.12.1 manifest.json

```
{  
  "name": "My Cables App",  
  "short_name": "CablesApp",  
  "start_url": "/",  
  "display": "standalone",  
  "background_color": "#000000",  
  "theme_color": "#000000",  
  "icons": [  
    {
```

```

        "src": "icon-192.png",
        "sizes": "192x192",
        "type": "image/png"
    },
    {
        "src": "icon-512.png",
        "sizes": "512x512",
        "type": "image/png"
    }
]
}

```

## 11.12.2 Service Worker

```

// sw.js
const CACHE_NAME = 'cables-app-v1';
const urlsToCache = [
    '/',
    '/index.html',
    '/js/cables.min.js',
    '/js/ops.js',
    '/js/patch.js',
    // Add your assets
];

self.addEventListener('install', event => {
    event.waitUntil(
        caches.open(CACHE_NAME)
            .then(cache => cache.addAll(urlsToCache))
    );
});

self.addEventListener('fetch', event => {
    event.respondWith(
        caches.match(event.request)
            .then(response => response || fetch(event.request))
    );
});

```

```

        );
    });
}

```

## 11.13 Electron Desktop Applications

For a truly native desktop experience, you can package your cables.gl export as an Electron application. Electron allows you to create cross-platform desktop apps using web technologies, perfect for distributing your cables.gl creations as standalone applications.

### 11.13.1 Why Electron?

**Advantages:** - Native desktop experience (menus, system tray, notifications)  
- Full file system access - Better performance control - No browser UI chrome  
- Can work offline completely - Access to native OS APIs - Professional distribution via installers

**Considerations:** - Larger app size (~100-200MB) - Requires code signing for distribution - More complex build process - Platform-specific considerations

### 11.13.2 Getting Started with Electron

#### Project Structure

After exporting your cables.gl patch, set up an Electron project:

```

electron-app/
    +- package.json
    +- main.js          # Main Electron process
    +- preload.js       # Preload script (optional)
    +- renderer/
        |   +- index.html    # Your exported cables HTML
        |   +- js/
        |       |   +- cables.min.js
        |       |   +- ops.js
        |       |   +- patch.js
        |   +- assets/       # Your exported assets

```

```

+-- assets/
|   +-+ icon.ico          # Windows icon
|   +-+ icon.icns         # macOS icon
|   +-+ icon.png          # Linux icon
+-- build/
    +-+ mac/              # Build configuration
    +-+ win/
    +-+ linux/

```

## Initial Setup

### package.json:

```

{
  "name": "my-cables-app",
  "version": "1.0.0",
  "description": "My Cables.gl Desktop App",
  "main": "main.js",
  "scripts": {
    "start": "electron .",
    "build": "electron-builder",
    "build:mac": "electron-builder --mac",
    "build:win": "electron-builder --win",
    "build:linux": "electron-builder --linux"
  },
  "build": {
    "appId": "com.yourcompany.cablesapp",
    "productName": "My Cables App",
    "directories": {
      "output": "dist"
    },
    "files": [
      "main.js",
      "preload.js",
      "renderer/**/*"
    ],
    "mac": {

```

```

      "icon": "assets/icon.icns",
      "category": "public.app-category.graphics-design"
    },
    "win": {
      "icon": "assets/icon.ico",
      "target": ["nsis", "portable"]
    },
    "linux": {
      "icon": "assets/icon.png",
      "target": ["AppImage", "deb"]
    }
  },
  "devDependencies": {
    "electron": "^28.0.0",
    "electron-builder": "^24.9.1"
  }
}

```

### Install dependencies:

```
npm install --save-dev electron electron-builder
```

### 11.13.3 Main Process (main.js)

The main process controls the application lifecycle and creates windows:

```

const { app, BrowserWindow, Menu, ipcMain, dialog, shell } = require('electron');
const path = require('path');
const fs = require('fs').promises;

// Keep a global reference of the window object
let mainWindow;
let splashWindow;

// Determine if we're in development

```

```

const isDev = process.env.NODE_ENV === 'development' || !app.isPackaged;

function createSplashWindow() {
  splashWindow = new BrowserWindow({
    width: 400,
    height: 300,
    frame: false,
    transparent: true,
    alwaysOnTop: true,
    resizable: false,
    webPreferences: {
      nodeIntegration: false,
      contextIsolation: true
    }
  });
}

// Load splash screen HTML
splashWindow.loadFile('splash.html');

// Center the window
splashWindow.center();

return splashWindow;
}

function createMainWindow() {
  // Create the browser window
  mainWindow = new BrowserWindow({
    width: 1280,
    height: 720,
    minWidth: 800,
    minHeight: 600,
    show: false, // Don't show until ready
    frame: true,
    titleBarStyle: process.platform === 'darwin' ? 'hiddenInset' : 'default',
    backgroundColor: '#000000',
    icon: getIconPath(),
    webPreferences: {
      nodeIntegration: false, // Security: don't expose Node.js
      contextIsolation: true,
      preload: path.join(__dirname, 'preload.js'), // Preload script
      webSecurity: !isDev, // Disable in dev for easier debugging
      enableRemoteModule: false
    }
  });

  // Load your exported cables.gl patch
  if (isDev) {
    mainWindow.loadFile('renderer/index.html');
    // Open DevTools in development
    mainWindow.webContents.openDevTools();
  } else {
    mainWindow.loadFile(path.join(__dirname, 'renderer/index.html'));
  }

  // Show window when ready to prevent visual flash
  mainWindow.once('ready-to-show', () => {
    if (splashWindow) {
      splashWindow.close();
      splashWindow = null;
    }
    mainWindow.show();
  });

  // Focus the window
  if (isDev) {
    mainWindow.focus();
  }
};

// Handle window closed
mainWindow.on('closed', () => {
  mainWindow = null;
});

// Handle external links
mainWindow.webContents.setWindowOpenHandler(({ url }) => {
  shell.openExternal(url);
  return { action: 'deny' };
}

```

```

});
```

// Prevent navigation to external URLs

```

mainWindow.webContents.on('will-navigate', (event, navigationUrl) => {
  const parsedUrl = new URL(navigationUrl);

  if (parsedUrl.origin !== 'file://') {
    event.preventDefault();
    shell.openExternal(navigationUrl);
  }
});
```

```

return mainWindow;
}
```

```

function getIconPath() {
  if (process.platform === 'win32') {
    return path.join(__dirname, 'assets/icon.ico');
  } else if (process.platform === 'darwin') {
    return path.join(__dirname, 'assets/icon.icns');
  } else {
    return path.join(__dirname, 'assets/icon.png');
  }
}
```

```

function createMenu() {
  const template = [
    {
      label: 'File',
      submenu: [
        {
          label: 'Load Settings',
          accelerator: 'CmdOrCtrl+O',
          click: async () => {
            const result = await dialog.showOpenDialog(mainWindow, {
              properties: ['openFile'],
              filters: [
                { name: 'JSON Files', extensions: ['json'] },
                { name: 'All Files', extensions: ['*'] }
              ]
            });
          }
        }
      ]
    }
  ];
}
```

```

      ]
    });

    if (!result.canceled && result.filePaths.length > 0) {
      mainWindow.webContents.send('load-settings', result.filePaths[0]);
    }
  },
  {
    label: 'Save Settings',
    accelerator: 'CmdOrCtrl+S',
    click: async () => {
      const result = await dialog.showSaveDialog(mainWindow, {
        filters: [
          { name: 'JSON Files', extensions: ['json'] },
          { name: 'All Files', extensions: ['*'] }
        ],
        defaultPath: 'settings.json'
      });
      if (!result.canceled) {
        mainWindow.webContents.send('save-settings', result.filePath);
      }
    }
  },
  { type: 'separator' },
  {
    label: 'Exit',
    accelerator: process.platform === 'darwin' ? 'Cmd+Q' : 'Ctrl+Q',
    click: () => {
      app.quit();
    }
  }
],
{
  label: 'Edit',
}
```

```

submenu: [
  { role: 'undo', label: 'Undo' },
  { role: 'redo', label: 'Redo' },
  { type: 'separator' },
  { role: 'cut', label: 'Cut' },
  { role: 'copy', label: 'Copy' },
  { role: 'paste', label: 'Paste' },
  { role: 'selectAll', label: 'Select All' }
]
},
{
  label: 'View',
  submenu: [
    { role: 'reload', label: 'Reload' },
    { role: 'forceReload', label: 'Force Reload' },
    { role: 'toggleDevTools', label: 'Toggle Developer Tools' },
    { type: 'separator' },
    { role: 'resetZoom', label: 'Actual Size' },
    { role: 'zoomIn', label: 'Zoom In' },
    { role: 'zoomOut', label: 'Zoom Out' },
    { type: 'separator' },
    { role: 'togglefullscreen', label: 'Toggle Fullscreen' }
  ]
},
{
  label: 'Window',
  submenu: [
    { role: 'minimize', label: 'Minimize' },
    { role: 'close', label: 'Close' }
  ]
},
{
  label: 'Help',
  submenu: [
    {
      label: 'About',
      click: () => {
        dialog.showMessageBox(mainWindow, {
          type: 'info',

```

```

          title: 'About',
          message: 'My Cables App',
          detail: 'Version 1.0.0\nBuilt with cables.gl and Electron'
        });
      }
    ]
  ];
}

// macOS specific menu adjustments
if (process.platform === 'darwin') {
  template.unshift({
    label: app.getName(),
    submenu: [
      { role: 'about', label: 'About ' + app.getName() },
      { type: 'separator' },
      { role: 'services', label: 'Services' },
      { type: 'separator' },
      { role: 'hide', label: 'Hide ' + app.getName() },
      { role: 'hideOthers', label: 'Hide Others' },
      { role: 'unhide', label: 'Show All' },
      { type: 'separator' },
      { role: 'quit', label: 'Quit ' + app.getName() }
    ]
  });
}

// Window menu
template[4].submenu = [
  { role: 'close', label: 'Close' },
  { role: 'minimize', label: 'Minimize' },
  { role: 'zoom', label: 'Zoom' },
  { type: 'separator' },
  { role: 'front', label: 'Bring All to Front' }
];
}

const menu = Menu.buildFromTemplate(template);
Menu.setApplicationMenu(menu);

```

```

}

// IPC Handlers for inter-process communication
function setupIpcHandlers() {
  // Handle file reading
  ipcMain.handle('read-file', async (event, filePath) => {
    try {
      const data = await fs.readFile(filePath, 'utf-8');
      return { success: true, data: JSON.parse(data) };
    } catch (error) {
      return { success: false, error: error.message };
    }
  });
}

// Handle file writing
ipcMain.handle('write-file', async (event, filePath, data) => {
  try {
    await fs.writeFile(filePath, JSON.stringify(data, null, 2), 'utf-8');
    return { success: true };
  } catch (error) {
    return { success: false, error: error.message };
  }
});

// Get app version
ipcMain.handle('get-app-version', () => {
  return app.getVersion();
});

// Get user data path
ipcMain.handle('get-user-data-path', () => {
  return app.getPath('userData');
});

// Window control
ipcMain.on('window-minimize', () => {
  if (mainWindow) mainWindow.minimize();
});

ipcMain.on('window-maximize', () => {
  if (mainWindow) {
    if (mainWindow.isMaximized()) {
      mainWindow.unmaximize();
    } else {
      mainWindow.maximize();
    }
  }
});

ipcMain.on('window-close', () => {
  if (mainWindow) mainWindow.close();
});

// App event handlers
app.whenReady().then(() => {
  // Create splash screen
  createSplashWindow();

  // Create main window after a short delay (simulate loading)
  setTimeout(() => {
    createMainWindow();
    createMenu();
    setupIpcHandlers();
  }, 1500);

  app.on('activate', () => {
    // On macOS, re-create window when dock icon is clicked
    if (BrowserWindow.getAllWindows().length === 0) {
      createMainWindow();
    }
  });
});

app.on('window-all-closed', () => {
  // On macOS, keep app running even when all windows are closed
  if (process.platform !== 'darwin') {

```

```

    app.quit();
}

// Security: Prevent new window creation
app.on('web-contents-created', (event, contents) => {
  contents.on('new-window', (event, navigationUrl) => {
    event.preventDefault();
    shell.openExternal(navigationUrl);
  });
});

```

#### 11.13.4 Preload Script (preload.js)

The preload script safely exposes Node.js APIs to the renderer process:

```

const { contextBridge, ipcRenderer } = require('electron');

// Expose protected methods that allow the renderer process
// to use ipcRenderer without exposing the entire object
contextBridge.exposeInMainWorld('electronAPI', {
  // File operations
  readFile: (filePath) => ipcRenderer.invoke('read-file', filePath),
  writeFile: (filePath, data) => ipcRenderer.invoke('write-file', filePath, data),

  // App info
  getAppVersion: () => ipcRenderer.invoke('get-app-version'),
  getUserDataPath: () => ipcRenderer.invoke('get-user-data-path'),

  // Window control
  minimizeWindow: () => ipcRenderer.send('window-minimize'),
  maximizeWindow: () => ipcRenderer.send('window-maximize'),
  closeWindow: () => ipcRenderer.send('window-close'),

  // Listen for messages from main process
  onMessage: (event, message) => {
    if (message === 'close') {
      app.quit();
    }
  }
});

```

```

onLoadSettings: (callback) => {
  ipcRenderer.on('load-settings', (event, filePath) => callback(filePath));
},
onSaveSettings: (callback) => {
  ipcRenderer.on('save-settings', (event, filePath) => callback(filePath));
},
// Remove listeners
removeAllListeners: (channel) => {
  ipcRenderer.removeAllListeners(channel);
}
});

```

#### 11.13.5 Advanced Window Configuration

##### Window Options Deep Dive

```

const mainWindow = new BrowserWindow({
  // Size and position
  width: 1280,
  height: 720,
  minWidth: 800,
  minHeight: 600,
  maxWidth: 3840,
  maxHeight: 2160,
  x: undefined, // Center if undefined
  y: undefined,
  center: true, // Center on screen

  // Appearance
  frame: true, // Show window frame
  titleBarStyle: 'default', // 'default', 'hidden', 'hiddenInset', 'customButton'
  transparent: false, // Transparent window (performance impact)
  backgroundColor: '#000000', // Background color before content loads
  opacity: 1.0, // Window opacity (0.0 to 1.0)
  vibrancy: 'ultra-dark', // macOS only: 'appearance-based', 'light', 'dark', etc.
  visualEffectState: 'active', // macOS only: 'active', 'inactive', 'followsSystem'
});

```

```

// Behavior
show: false, // Don't show until ready
alwaysOnTop: false, // Keep window on top
fullscreen: false, // Start in fullscreen
fullscreenable: true, // Allow fullscreen
simpleFullscreen: false, // macOS simple fullscreen
skipTaskbar: false, // Don't show in taskbar
kiosk: false, // Kiosk mode (fullscreen, no exit)
closable: true, // Allow closing
minimizable: true, // Allow minimizing
maximizable: true, // Allow maximizing
resizable: true, // Allow resizing
movable: true, // Allow moving
focusable: true, // Can receive focus

// Window state
autoHideMenuBar: false, // Auto-hide menu bar
useContentSize: false, // Use content size instead of window size
title: 'My Cables App', // Window title

// Icon
icon: getIconPath(), // Window icon

// Web preferences
webPreferences: {
  nodeIntegration: false,
  contextIsolation: true,
  preload: path.join(__dirname, 'preload.js'),
  webSecurity: true,
  allowRunningInsecureContent: false,
  experimentalFeatures: false,
  enableBlinkFeatures: '',
  disableBlinkFeatures: '',
  sandbox: false, // Enable sandbox for extra security
  enableRemoteModule: false,
  backgroundThrottling: true, // Throttle when backgrounded
  offscreen: false, // Use offscreen rendering
  webviewTag: false // Disable webview tag
}

```

```

  }
});


```

## Window State Persistence

Save and restore window position and size:

```

const Store = require('electron-store');

const store = new Store({
  name: 'window-state',
  defaults: {
    width: 1280,
    height: 720,
    x: undefined,
    y: undefined,
    isMaximized: false
  }
});

function createMainWindow() {
  const windowState = store.get('windowState', {});

  const mainWindow = new BrowserWindow({
    width: windowState.width || 1280,
    height: windowState.height || 720,
    x: windowState.x,
    y: windowState.y,
    // ... other options
  });

  // Restore maximized state
  if (windowState.isMaximized) {
    mainWindow.maximize();
  }

  // Save window state on move/resize
}


```

```

const saveWindowState = () => {
  const bounds = mainWindow.getBounds();
  store.set('windowState', {
    width: bounds.width,
    height: bounds.height,
    x: bounds.x,
    y: bounds.y,
    isMaximized: mainWindow.isMaximized()
  });
};

mainWindow.on('moved', saveWindowState);
mainWindow.on('resized', saveWindowState);
mainWindow.on('maximize', () => {
  store.set('windowState.isMaximized', true);
});
mainWindow.on('unmaximize', () => {
  store.set('windowState.isMaximized', false);
});

return mainWindow;
}

```

Install electron-store:

```
npm install electron-store
```

## 11.13.6 Inter-Window Communication

Electron supports multiple windows with various communication patterns:

### Method 1: IPC (Inter-Process Communication)

**Main Process -> Renderer Process:**

```

// In main.js
mainWindow.webContents.send('message-from-main', {
  type: 'update',
  data: { value: 42 }
});

// In renderer (index.html or your cables patch)
window.electronAPI.onMessage((data) => {
  console.log('Received:', data);
});

```

### Renderer Process -> Main Process:

```

// In preload.js
contextBridge.exposeInMainWorld('electronAPI', {
  sendToMain: (channel, data) => {
    ipcRenderer.send(channel, data);
  },
  onMessage: (callback) => {
    ipcRenderer.on('message-from-main', (event, data) => callback(data));
  }
});

// In renderer
window.electronAPI.sendToMain('message-from-renderer', {
  action: 'save',
  data: { settings: {...} }
});

```

### Method 2: Multiple Windows Communication

```
// In main.js
let windows = [];
```

```

function createWindow(id) {
  const window = new BrowserWindow({
    // ... window options
    webPreferences: {
      // ... web preferences
    }
  });

  window.id = id;
  windows.push(window);

  window.on('closed', () => {
    windows = windows.filter(w => w.id !== id);
  });

  return window;
}

// Broadcast to all windows
function broadcastToAllWindows(channel, data) {
  windows.forEach(window => {
    if (window && !window.isDestroyed()) {
      window.webContents.send(channel, data);
    }
  });
}

// Send to specific window
function sendToWindow(windowId, channel, data) {
  const window = windows.find(w => w.id === windowId);
  if (window && !window.isDestroyed()) {
    window.webContents.send(channel, data);
  }
}

// Example: Sync settings across windows
ipcMain.on('update-settings', (event, settings) => {
  // Save settings
  store.set('settings', settings);
}

```

```

    // Broadcast to all windows
    broadcastToAllWindows('settings-updated', settings);
  });
}

```

### Method 3: Shared Data via Main Process

```

// In main.js
let sharedData = {
  settings: {},
  state: {}
};

// Get shared data
ipcMain.handle('get-shared-data', (event, key) => {
  return sharedData[key];
});

// Set shared data
ipcMain.handle('set-shared-data', (event, key, value) => {
  sharedData[key] = value;
  // Notify all windows
  broadcastToAllWindows('shared-data-changed', { key, value });
  return true;
});

```

### Method 4: Window-to-Window via Main Process

```

// Window A sends message to Window B
ipcMain.on('send-to-window', (event, targetWindowId, channel, data) => {
  sendToWindow(targetWindowId, channel, data);
});

// In preload.js
contextBridge.exposeInMainWorld('electronAPI', {
  sendToWindow: (targetWindowId, channel, data) => {
    ipcRenderer.send('send-to-window', targetWindowId, channel, data);
  }
});

```

```

},
onWindowMessage: (callback) => {
  ipcRenderer.on('window-message', (event, data) => callback(data));
}
});

```

## 11.13.7 Splash Screen Implementation

A professional splash screen improves perceived performance:

**splash.html:**

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<style>
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  width: 400px;
  height: 300px;
  background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, sans-serif;
  color: white;
  overflow: hidden;
}

```

```

.logo {
  width: 80px;
  height: 80px;
  margin-bottom: 20px;
  animation: pulse 2s ease-in-out infinite;
}

@keyframes pulse {
  0%, 100% { transform: scale(1); opacity: 1; }
  50% { transform: scale(1.1); opacity: 0.8; }
}

.app-name {
  font-size: 24px;
  font-weight: 600;
  margin-bottom: 10px;
}

.version {
  font-size: 12px;
  opacity: 0.8;
  margin-bottom: 30px;
}

.loader {
  width: 200px;
  height: 4px;
  background: rgba(255, 255, 255, 0.2);
  border-radius: 2px;
  overflow: hidden;
  position: relative;
}

.loader-bar {
  height: 100%;
  background: white;
  width: 0;
  animation: loading 2s ease-in-out infinite;
  border-radius: 2px;
}

```

```

}

@keyframes loading {
  0% { width: 0%; }
  50% { width: 70%; }
  100% { width: 100%; }
}

.status {
  margin-top: 20px;
  font-size: 12px;
  opacity: 0.7;
}
</style>
</head>
<body>
  <div class="logo">
    <!-- Your logo SVG or image -->
    <svg viewBox="0 0 100 100" fill="white">
      <circle cx="50" cy="50" r="40" stroke="white" stroke-width="2" fill="none"/>
      <path d="M30 50 L45 65 L70 35" stroke="white" stroke-width="3" fill="none"/>
    </svg>
  </div>
  <div class="app-name">My Cables App</div>
  <div class="version">Version 1.0.0</div>
  <div class="loader">
    <div class="loader-bar"></div>
  </div>
  <div class="status" id="status">Loading...</div>

<script>
  // Update status from main process
  const { ipcRenderer } = require('electron');

  ipcRenderer.on('splash-status', (event, message) => {
    document.getElementById('status').textContent = message;
  });

```

```

    ipcRenderer.on('splash-progress', (event, percent) => {
      document.querySelector('.loader-bar').style.width = percent + '%';
    });
  </script>
</body>
</html>

```

### Enhanced main.js with splash screen:

```

function createSplashWindow() {
  splashWindow = new BrowserWindow({
    width: 400,
    height: 300,
    frame: false,
    transparent: true,
    alwaysOnTop: true,
    resizable: false,
    webPreferences: {
      nodeIntegration: true, // Needed for splash screen
      contextIsolation: false
    }
  });

  splashWindow.loadFile('splash.html');
  splashWindow.center();

  // Update splash screen status
  const updateSplashStatus = (message) => {
    if (splashWindow && !splashWindow.isDestroyed()) {
      splashWindow.webContents.send('splash-status', message);
    }
  };

  const updateSplashProgress = (percent) => {
    if (splashWindow && !splashWindow.isDestroyed()) {
      splashWindow.webContents.send('splash-progress', percent);
    }
  };
}

```

```

    }

    // Simulate loading progress
    updateSplashStatus('Initializing...');

    updateSplashProgress(10);

    setTimeout(() => {
        updateSplashStatus('Loading assets...');

        updateSplashProgress(40);

    }, 300);

    setTimeout(() => {
        updateSplashStatus('Preparing renderer...');

        updateSplashProgress(70);

    }, 800);

    setTimeout(() => {
        updateSplashStatus('Almost ready...');

        updateSplashProgress(90);

    }, 1200);

    return { splashWindow, updateSplashStatus, updateSplashProgress };
}

// In app.whenReady()
app.whenReady().then(() => {
    const { splashWindow: splash, updateSplashStatus } = createSplashWindow()

    updateSplashStatus('Creating main window...');

    setTimeout(() => {
        createMainWindow();
        createMenu();
        setupIpcHandlers();

        // Close splash when main window is ready
        mainWindow.once('ready-to-show', () => {
            setTimeout(() => {

```

```

                if (splash && !splash.isDestroyed()) {
                    splash.close();
                }
                mainWindow.show();
            }, 500); // Small delay for smooth transition
        });
    }, 1500);
});

```

## 11.13.8 JSON File Operations

Saving and loading JSON data is essential for app settings, user preferences, and state persistence:

### Method 1: Using IPC Handlers (Recommended)

In `main.js`:

```

const fs = require('fs').promises;
const path = require('path');

// Get user data directory
const getUserDataPath = () => {
    return app.getPath('userData');
};

// Ensure directory exists
async function ensureDirectory(dirPath) {
    try {
        await fs.mkdir(dirPath, { recursive: true });
    } catch (error) {
        console.error('Error creating directory:', error);
    }
}

// IPC Handlers for JSON operations
ipcMain.handle('save-json', async (event, filename, data) => {

```

```

try {
  const userDataPath = getUserDataPath();
  const filePath = path.join(userDataPath, filename);

  await ensureDirectory(path.dirname(filePath));
  await fs.writeFile(filePath, JSON.stringify(data, null, 2), 'utf-8');

  return { success: true, path: filePath };
} catch (error) {
  console.error('Error saving JSON:', error);
  return { success: false, error: error.message };
}
});

ipcMain.handle('load-json', async (event, filename) => {
try {
  const userDataPath = getUserDataPath();
  const filePath = path.join(userDataPath, filename);

  const data = await fs.readFile(filePath, 'utf-8');
  return { success: true, data: JSON.parse(data) };
} catch (error) {
  if (error.code === 'ENOENT') {
    // File doesn't exist, return default
    return { success: true, data: null };
  }
  console.error('Error loading JSON:', error);
  return { success: false, error: error.message };
}
};

ipcMain.handle('delete-json', async (event, filename) => {
try {
  const userDataPath = getUserDataPath();
  const filePath = path.join(userDataPath, filename);

  await fs.unlink(filePath);
  return { success: true };
}

```

```

} catch (error) {
  if (error.code === 'ENOENT') {
    return { success: true }; // Already deleted
  }
  console.error('Error deleting JSON:', error);
  return { success: false, error: error.message };
}

ipcMain.handle('list-json-files', async (event, directory = '') => {
try {
  const userDataPath = getUserDataPath();
  const dirPath = path.join(userDataPath, directory);

  const files = await fs.readdir(dirPath);
  const jsonFiles = files.filter(file => file.endsWith('.json'));

  return { success: true, files: jsonFiles };
} catch (error) {
  console.error('Error listing JSON files:', error);
  return { success: false, error: error.message };
}
});

```

### In preload.js:

```

contextBridge.exposeInMainWorld('electronAPI', {
  // JSON file operations
  saveJSON: async (filename, data) => {
    return await ipcRenderer.invoke('save-json', filename, data);
  },

  loadJSON: async (filename) => {
    return await ipcRenderer.invoke('load-json', filename);
  },

  deleteJSON: async (filename) => {
    return await ipcRenderer.invoke('delete-json', filename);
  }
});

```

```

},
listJSONFiles: async (directory = '') => {
  return await ipcRenderer.invoke('list-json-files', directory);
}
);

```

### In your renderer (cables patch or HTML):

```

// Save settings
async function saveSettings(settings) {
  const result = await window.electronAPI.saveJSON('settings.json', settings);
  if (result.success) {
    console.log('Settings saved to:', result.path);
  } else {
    console.error('Failed to save settings:', result.error);
  }
}

// Load settings
async function loadSettings() {
  const result = await window.electronAPI.loadJSON('settings.json');
  if (result.success) {
    if (result.data) {
      console.log('Settings loaded:', result.data);
      return result.data;
    } else {
      // Return default settings
      return getDefaultSettings();
    }
  } else {
    console.error('Failed to load settings:', result.error);
    return getDefaultSettings();
  }
}

// Example usage with cables.gl patch
async function initializeApp() {

```

```

// Load saved settings
const settings = await loadSettings();

// Apply settings to cables patch
if (window.CABLES && window.CABLES.patch) {
  Object.keys(settings).forEach(key => {
    window.CABLES.patch.setVariable(key, settings[key]);
  });
}

// Listen for settings changes and auto-save
if (window.CABLES && window.CABLES.patch) {
  window.CABLES.patch.on('variableChanged', async (name, value) => {
    const currentSettings = await loadSettings();
    currentSettings[name] = value;
    await saveSettings(currentSettings);
  });
}

// Save cables patch state
async function savePatchState() {
  if (!window.CABLES || !window.CABLES.patch) return;

  const state = {
    timestamp: new Date().toISOString(),
    variables: {},
    camera: {
      position: window.CABLES.patch.cgl?.camera?.position || null,
      rotation: window.CABLES.patch.cgl?.camera?.rotation || null
    }
  };

  // Save all variables
  // (You'll need to track variable names or get them from your patch)
  const variableNames = ['color', 'speed', 'intensity']; // Your variable names
  variableNames.forEach(name => {
    state.variables[name] = window.CABLES.patch.getVariable(name);
  });
}
```

```

    await window.electronAPI.saveJSON('patch-state.json', state);
}

// Load patch state
async function loadPatchState() {
    const result = await window.electronAPI.loadJSON('patch-
state.json');
    if (result.success && result.data) {
        const state = result.data;

        // Restore variables
        Object.keys(state.variables).forEach(name => {
            window.CABLES.patch.setVariable(name, state.variables[name]);
        });

        // Restore camera if available
        if (state.camera && window.CABLES.patch.cgl?.camera) {
            // Camera restoration depends on your cables setup
        }
    }
}

```

## Method 2: Using electron-store (Simpler)

```
npm install electron-store
```

```

// In main.js
const Store = require('electron-store');

const store = new Store({
    name: 'app-settings',
    defaults: {
        theme: 'dark',
        windowState: {
            width: 1280,

```

```

        height: 720
    },
    cablesSettings: {
        color: [1, 0, 0, 1],
        speed: 1.0
    }
});

// Expose store to renderer
ipcMain.handle('store-get', (event, key) => {
    return store.get(key);
});

ipcMain.handle('store-set', (event, key, value) => {
    store.set(key, value);
    return true;
});

ipcMain.handle('store-delete', (event, key) => {
    store.delete(key);
    return true;
});

ipcMain.handle('store-clear', () => {
    store.clear();
    return true;
});

ipcMain.handle('store-all', () => {
    return store.store;
});

```

```

// In preload.js
contextBridge.exposeInMainWorld('electronAPI', {
    store: {
        get: (key) => ipcRenderer.invoke('store-get', key),

```

```

    set: (key, value) => ipcRenderer.invoke('store-set', key, value),
    delete: (key) => ipcRenderer.invoke('store-delete', key),
    clear: () => ipcRenderer.invoke('store-clear'),
    all: () => ipcRenderer.invoke('store-all')
  }
);

```

```

// In renderer
// Get setting
const theme = await window.electronAPI.store.get('theme');

// Set setting
await window.electronAPI.store.set('cablesSettings.color', [0, 1, 0, 1]);

// Get all settings
const allSettings = await window.electronAPI.store.all();

```

### 11.13.9 Code Signing for Distribution

Code signing is essential for smooth app distribution on macOS and Windows. Unsigned apps trigger security warnings and may be blocked.

#### macOS Code Signing

**Requirements:** - Apple Developer Account (\$99/year) - Valid code signing certificate - Notarization (required for macOS 10.15+)

#### package.json configuration:

```

{
  "build": {
    "appId": "com.yourcompany.cablesapp",
    "mac": {
      "icon": "assets/icon.icns",
      "category": "public.app-category.graphics-design",
    }
  }
};

```

```

  "target": [
    {
      "target": "dmg",
      "arch": ["x64", "arm64"]
    },
    {
      "target": "zip",
      "arch": ["x64", "arm64"]
    }
  ],
  "hardenedRuntime": true,
  "gatekeeperAssess": false,
  "entitlements": "build/mac/entitlements.mac.plist",
  "entitlementsInherit": "build/mac/entitlements.mac.plist"
},
"afterSign": "scripts/notarize.js",
"notarize": {
  "teamId": "YOUR_TEAM_ID"
}
}
]
}

```

#### entitlements.mac.plist:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>com.apple.security.cs.allow-jit</key>
  <true/>
  <key>com.apple.security.cs.allow-unsigned-executable-memory</key>
  <true/>
  <key>com.apple.security.cs.allow-dyld-environment-variables</key>
  <true/>
  <key>com.apple.security.cs.disable-library-validation</key>

```

```
<true/>  
</dict>  
</plist>
```

### scripts/notarize.js:

```
const { notarize } = require('@electron/notarize');

exports.default = async function notarizing(context) {
  const { electronPlatformName, appOutDir } = context;

  if (electronPlatformName !== 'darwin') {
    return;
  }

  const appName = context.packager.appInfo.productFilename;

  return await notarize({
    appBundleId: 'com.yourcompany.cablesapp',
    appPath: `${appOutDir}/${appName}.app`,
    appleId: process.env.APPLE_ID,
    appleIdPassword: process.env.APPLE_ID_PASSWORD,
    teamId: process.env.APPLE_TEAM_ID
  });
};
```

### Environment variables (.env or export):

```
export APPLE_ID="your@email.com"
export APPLE_ID_PASSWORD="app-specific-password"
export APPLE_TEAM_ID="YOUR_TEAM_ID"
```

### Build command:

```
npm run build:mac
```

### Windows Code Signing

**Requirements:** - Code signing certificate (purchased from certificate authority)  
- Or use self-signed certificate for testing (not recommended for distribution)

### package.json configuration:

```
{
  "build": {
    "win": {
      "icon": "assets/icon.ico",
      "target": [
        {
          "target": "nsis",
          "arch": ["x64", "ia32"]
        },
        {
          "target": "portable",
          "arch": ["x64"]
        }
      ],
      "signingHashAlgorithms": ["sha256"],
      "sign": "build/win/sign.js",
      "certificateFile": "path/to/certificate.pfx",
      "certificatePassword": "${env.CERTIFICATE_PASSWORD}"
    }
  }
}
```

### build/win/sign.js:

```

const path = require('path');

exports.default = async function(configuration) {
  const { path: filePath } = configuration;

  // Only sign on Windows
  if (process.platform !== 'win32') {
    return;
  }

  // Use electron-builder's built-in signing
  // Or use signtool directly
  const { execSync } = require('child_process');

  const certPath = process.env.CERTIFICATE_PATH;
  const certPassword = process.env.CERTIFICATE_PASSWORD;

  if (!certPath || !certPassword) {
    console.warn('Certificate not configured, skipping signing');
    return;
  }

  try {
    execSync(
      `signtool sign /f "${certPath}" /p "${certPassword}" /t http://timestamp.com/auth/namesign
      { stdio: 'inherit' }
    );
  } catch (error) {
    console.error('Signing failed:', error);
    throw error;
  }
};

```

#### Alternative: Using electron-builder's built-in signing:

```
{
  "build": {
```

```

    "win": {
      "certificateFile": "path/to/certificate.pfx",
      "certificatePassword": "${env.CERTIFICATE_PASSWORD}",
      "signingHashAlgorithms": ["sha256"],
      "signDlIs": true
    }
  }
}
```

#### Build command:

```
npm run build:win
```

#### App Registration and Metadata

#### package.json - Complete build configuration:

```
{
  "name": "my-cables-app",
  "version": "1.0.0",
  "description": "My amazing Cables.gl application",
  "author": {
    "name": "Your Name",
    "email": "your@email.com"
  },
  "license": "MIT",
  "main": "main.js",
  "build": {
    "appId": "com.yourcompany.cablesapp",
    "productName": "My Cables App",
    "copyright": "Copyright © 2024 Your Company",
    "directories": {
      "output": "dist",
      "buildResources": "build"
    },
    "files": [

```

```

"main.js",
"preload.js",
"renderer/**/*",
"!renderer/**/*.map"
],
"extraResources": [
{
  "from": "assets/",
  "to": "assets/",
  "filter": ["**/*"]
}
],
"mac": {
  "icon": "assets/icon.icns",
  "category": "public.app-category.graphics-design",
  "minimumSystemVersion": "10.13",
  "darkModeSupport": true,
  "target": [
    {
      "target": "dmg",
      "arch": ["x64", "arm64"]
    }
  ],
  "hardenedRuntime": true,
  "entitlements": "build/mac/entitlements.mac.plist",
  "entitlementsInherit": "build/mac/entitlements.mac.plist"
},
"win": {
  "icon": "assets/icon.ico",
  "target": [
    {
      "target": "nsis",
      "arch": ["x64"]
    }
  ],
  "publisherName": "Your Company Name",
  "verifyUpdateCodeSignature": false
},
"linux": {
  "icon": "assets/icon.png",
  "target": [
    {
      "target": "AppImage",
      "arch": ["x64"]
    },
    {
      "target": "deb",
      "arch": ["x64"]
    }
  ],
  "category": "Graphics"
},
"nsis": {
  "oneClick": false,
  "allowToChangeInstallationDirectory": true,
  "createDesktopShortcut": true,
  "createStartMenuShortcut": true,
  "shortcutName": "My Cables App"
},
"dmg": {
  "title": "${productName} ${version}",
  "icon": "assets/icon.icns",
  "background": "build/mac/dmg-background.png",
  "contents": [
    {
      "x": 410,
      "y": 150,
      "type": "link",
      "path": "/Applications"
    },
    {
      "x": 130,
      "y": 150,
      "type": "file"
    }
  ],
  "window": {
    "width": 540,

```

```
        "height": 380
    }
}
}
```

## 11.13.10 Building and Distributing

### Development Build

```
# Start in development mode
npm start
```

### Production Build

```
# Build for current platform
npm run build

# Build for specific platforms
npm run build:mac
npm run build:win
npm run build:linux

# Build for all platforms (requires platform-specific tools)
npm run build:all
```

### Distribution Checklist

**Before Building:** - [ ] Update version in package.json - [ ] Test app thoroughly  
- [ ] Optimize assets - [ ] Prepare code signing certificates - [ ] Set up environment variables - [ ] Test on target platforms

**After Building:** - [ ] Test installer on clean system - [ ] Verify code signing -  
[ ] Test auto-updater (if implemented) - [ ] Check file associations - [ ] Verify menu items work - [ ] Test file operations - [ ] Check window state persistence

## 11.13.11 Advanced Electron Features

### Auto-Updater

```
npm install electron-updater
```

```
// In main.js
const { autoUpdater } = require('electron-updater');

autoUpdater.checkForUpdatesAndNotify();

autoUpdater.on('update-available', () => {
  dialog.showMessageBox(mainWindow, {
    type: 'info',
    title: 'Update Available',
    message: 'A new version is available. It will be downloaded in the background',
    buttons: ['OK']
  });
});

autoUpdater.on('update-downloaded', () => {
  dialog.showMessageBox(mainWindow, {
    type: 'info',
    title: 'Update Ready',
    message: 'Update downloaded. The application will restart to apply the update',
    buttons: ['Restart Now', 'Later']
}).then(result => {
  if (result.response === 0) {
    autoUpdater.quitAndInstall();
  }
});
});
```

### System Tray

```

const { Tray, Menu } = require('electron');
const path = require('path');

let tray = null;

function createTray() {
  const iconPath = path.join(__dirname, 'assets', 'tray-icon.png');
  tray = new Tray(iconPath);

  const contextMenu = Menu.buildFromTemplate([
    {
      label: 'Show App',
      click: () => {
        mainWindow.show();
      }
    },
    {
      label: 'Quit',
      click: () => {
        app.quit();
      }
    }
  ]);

  tray.setToolTip('My Cables App');
  tray.setContextMenu(contextMenu);

  tray.on('click', () => {
    mainWindow.isVisible() ? mainWindow.hide() : mainWindow.show();
  });
}

```

## Native Notifications

```

const { Notification } = require('electron');

function showNotification(title, body) {

```

```

if (Notification.isSupported()) {
  new Notification({
    title: title,
    body: body,
    icon: getIconPath()
  }).show();
}

```

## 11.13.12 Performance Optimization for Electron

- Disable Node Integration in Renderer** - Use contextBridge instead
- Enable Context Isolation** - Better security and performance
- Use Hardware Acceleration** - Enabled by default
- Optimize Asset Loading** - Lazy load when possible
- Throttle Background Processes** - Use backgroundThrottling: true
- Monitor Memory Usage** - Use DevTools memory profiler

## 11.13.13 Security Best Practices

- Never use nodeIntegration: true** - Use preload scripts instead
- Always use contextIsolation: true** - Isolates your code
- Validate all IPC messages** - Don't trust renderer input
- Use Content Security Policy** - Restrict resource loading
- Keep Electron updated** - Security patches are important
- Sanitize file paths** - Prevent directory traversal attacks

## 11.13.14 Troubleshooting Electron Issues

**App won't start:** - Check main.js for syntax errors - Verify all dependencies are installed - Check console for error messages

**Window is blank:** - Verify file paths are correct - Check DevTools for errors - Ensure renderer files are included in build

**Code signing fails:** - Verify certificate is valid - Check environment variables are set - Ensure certificate password is correct

**App is slow:** - Check for memory leaks - Optimize asset loading - Use performance profiling tools

## 11.14 Troubleshooting

### 11.14.1 Common Issues

**"Assets not loading"** - Check file paths are correct - Ensure CORS headers are set for cross-origin assets - Verify assets are included in export

**"Blank screen"** - Check browser console for errors - Verify all JavaScript files loaded - Test on a local server (not file://)

**"Poor performance"** - Reduce canvas resolution - Lower texture sizes - Simplify shaders - Check for memory leaks

**"Works locally but not on server"** - Check file paths (case-sensitive on Linux) - Verify all files uploaded - Check server MIME types

6. **Electron Exercise:** Implement JSON save/load functionality to persist your patch settings
7. **Electron Exercise:** Set up code signing for macOS or Windows (requires developer account/certificate)
8. **Electron Exercise:** Create a multi-window Electron app with inter-window communication
9. **Electron Exercise:** Implement window state persistence (save/restore window position and size)
10. **Electron Exercise:** Add a system tray icon with context menu for your Electron app

## 11.15 Featured Videos



The thumbnail features a black background with a colorful, swirling logo in the center. The word "OPEN-SOURCE" is at the top left, and "OFFLINE" is at the bottom left. A small portrait of a man is on the right side.

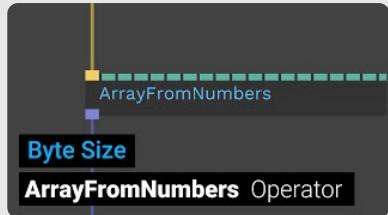
<https://youtu.be/hVxrxXhH7vQ>  
**Cables.gl Standalone (Offline) Build: Create Without Limits!**  
by Decode GL

## 11.16 Exercises

1. Export a simple patch and host it on GitHub Pages
2. Embed a cables patch as a website background
3. Create a loading screen for your patch
4. Set up communication between your patch and external JavaScript
5. **Electron Exercise:** Package your cables.gl export as an Electron app with a custom splash screen

# 12 Video Tutorials

## 12.1 Getting Started & Overviews



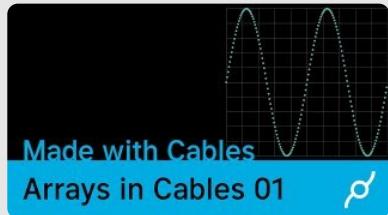
<https://youtu.be/iXKo7mU422M>

**Array from Numbers Operator tutorial - byte size**  
by cables\_gl



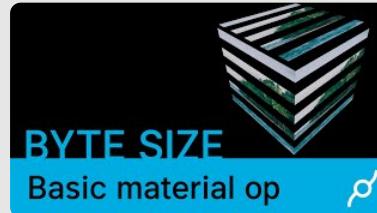
<https://youtu.be/koLSrHFyIUY>

**Arrays in cables - tutorial 03**  
by cables\_gl



<https://youtu.be/FRFFfvVgWFcs>

**Arrays in cables tutorial 01**  
by cables\_gl



<https://youtu.be/F-CUDHq40Pc>  
**Basic material op tutorial - Byte size**  
by cables\_gl



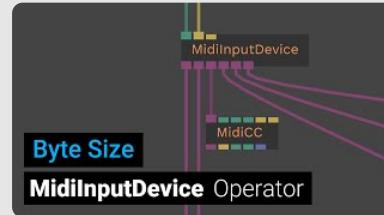
<https://youtu.be/EPFNHYah9F4>  
**cables gl introduction**  
by cables\_gl



<https://youtu.be/MODVmJ6MYQE>  
**Creating your own cables.gl operators - custom and user ops tutorial**  
by cables\_gl



<https://youtu.be/EzV5CRAMyTA>  
**Depth texture op tutorial - Byte size**  
by cables\_gl



<https://youtu.be/XvVBnPakE28>  
**Midi Input Device - intro to MIDI in cables - Byte Size**  
by cables\_gl



<https://youtu.be/knGnukutZeM>  
**Lights and Shadows Operators - getting started - Video Tutorial**  
by cables\_gl



<https://youtu.be/Ds4fPcxyBvM>  
**Noise Texture Operator for generating color palettes for various design techniques - Video Tutorial**  
by cables\_gl



<https://youtu.be/7xlElfMWgw>  
**MeshInstancer tutorial 01**  
by cables\_gl



<https://youtu.be/P6esDOFHM6w>  
**Particle system in cables tutorial 01**  
by cables\_gl



Made with Cables  
Particle system 02

<https://youtu.be/Nre7LH0OVw4>  
**Particle system in cables tutorial 02**  
by cables\_gl



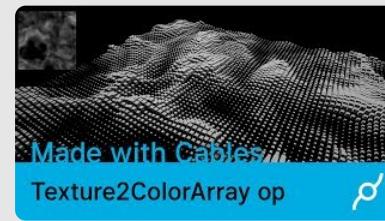
Made with Cables  
Post processing part 1

<https://youtu.be/x2jKZgmFVq4>  
**Post processing tutorial for beginners**  
by cables\_gl



BYTE SIZE  
Text Texture op

<https://youtu.be/z1QF9dE67-w>  
**Text Texture op tutorial - Byte size**  
by cables\_gl



Made with Cables  
Texture2ColorArray op

<https://youtu.be/mQN8VtVOltQ>  
**Texture2ColorArray op tutorial**  
by cables\_gl



<https://youtu.be/wzpKR7vbCXg>  
**Timeline - Part 1: Overview**  
by cables\_gl



<https://youtu.be/SaKWF6Rnsyl>  
**Transform Vertex Operator tutorial (GPU vs CPU based animation) - byte size**  
by cables\_gl

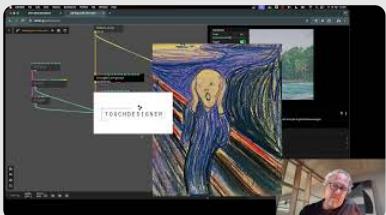


<https://youtu.be/B9GyRzov5Bg>

**tutorial demo effect / render2textures world position target tricks**  
by cables\_gl



<https://youtu.be/T0djoWQkBew>  
**Cables.GL: Introduction**  
by Creative Tech Talks

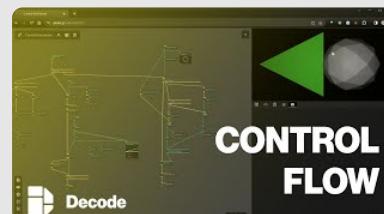


<https://youtu.be/sbML3B3Vu4g>  
**Cables.GL: Tutorial**  
by Creative Tech Talks



<https://youtu.be/kgXpXsLtv1M>

**Assets (6/13) - Intro to Cables.gl**  
by Decode GL



<https://youtu.be/vzWrCGfU7uw>  
**Control Flow (3/13) - Intro to Cables.gl**  
by Decode GL



<https://youtu.be/2YFB4MuN8y8>  
**Data Types (2/13) - Intro to Cables.gl**  
by Decode GL



<https://youtu.be/Z4gReZ34SHU>  
**Interactions (5/13) - Intro to Cables.gl**  
by Decode GL



<https://youtu.be/qEno30S8CBc>  
**Glitch Art Tutorial using Cables.gl**  
by Jaalibandar



<https://youtu.be/VsS4gaJ7pMw>  
**Introduction to Cables.gl (1/13)**  
by Decode GL



<https://youtu.be/goO3PhuenBI>  
**First Steps in Cables.gl - Tutorial**  
by The Interactive & Immersive HQ



<https://youtu.be/RhbId-kUWig>  
**Texture Effects (8/13) - Intro to Cables.gl**  
by Decode GL

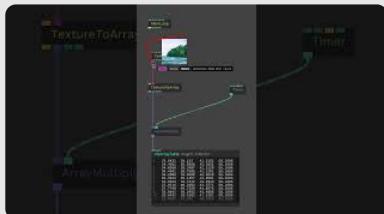


<https://youtu.be/lj6REnNZU0s>  
**converter ops**  
by cables\_gl

## 12.2 Core Concepts & Workflow



<https://youtu.be/M1A8S98UOuI>  
**how to reroute cables #gui #uxdesign #motiondesign**  
by cables\_gl



<https://youtu.be/ZCKrhswQiyc>  
**you can cut cables with the [Y] key #animation #motiondesign #design #web #3danimation**  
by cables\_gl



<https://youtu.be/xawlfxKpxRQ>  
**you can replace cables that easy #animation #motiondesign #design #web**  
by cables\_gl



[https://youtu.be/GQc6JF\\_jy6M](https://youtu.be/GQc6JF_jy6M)  
**Debug View in Cables.gl | Setting up multiple views in your patch**  
by Jaalibandar



<https://youtu.be/uzqplBUGMWg>  
**01 Jam Sessions : Generative Fluid Graphic in Cables.gl**  
by FahmiMursyid



<https://youtu.be/wERboDg6zOI>  
**Impactful Transitions under 10 minutes using cables.gl | Genuary 04: Intersections**  
by Jaalibandar



[https://youtu.be/\\_CltN9uQhoU](https://youtu.be/_CltN9uQhoU)  
**Procedurally generated plants in Cables.gl #genuary**  
by Jaalibandar



<https://youtu.be/5Jc3woVozNc>  
**Cables.gl | Generative Poster 05**  
by Karthik Dondeti



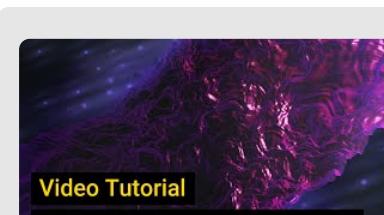
<https://youtu.be/DsSPcNSLyAw>  
**Cables.gl | Generative Poster 06**  
by Karthik Dondeti



<https://youtu.be/gRV0DqpSd-4>  
**Cables.gl | Generative Poster 09**  
by Karthik Dondeti



<https://youtu.be/3tZQt5Eiicw>  
**February 2022 Release Chat - cables.gl updated - PBR, Geometry from Textures, Teams, EXR support**  
by cables\_gl



**Video Tutorial**  
**Vertex Displacement & Normals**  
<https://youtu.be/a56wk9Xm9dY>  
**Using Vertex Displacement with Normal maps in cables.gl**  
by cables\_gl



<https://youtu.be/NjG85QbbI0w>  
**Vertex displacement op - byte size**  
by cables\_gl



[https://youtu.be/lOMplXy\\_JV0](https://youtu.be/lOMplXy_JV0)  
**Visualize any YouTube playlist in 3D with n8n.io & cables.gl (part 1)**  
by Decode GL



<https://youtu.be/AZrWNl3MwHQ>  
**Scrolling Terrain with UFO in 10 minutes using cables.gl**  
by Jaalibandar



<https://youtu.be/sbqE83ZHiTU>  
**Scrolling Terrain with UFO in 10 minutes using cables.gl**  
by Jaalibandar

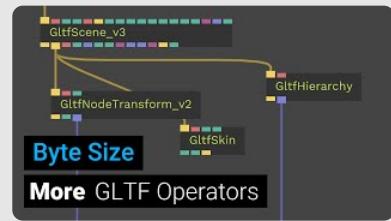


<https://youtu.be/3KSS1nrv6t0>  
**cables.gl web demo - realtime visualizer soundcloud globe | Exyl - Ping! Moai**  
by stobelights

## 12.3 3D / 3D Meshes



<https://youtu.be/iqIXSb-kAws>  
**Importing GLTF 3D Scenes with Camera positions and animating them in cables.gl**  
by cables\_gl



[https://youtu.be/l\\_eD5nml\\_5A](https://youtu.be/l_eD5nml_5A)

**More GLTF operators - animated rig support, position data, separate animation timing - Byte Size**

by cables\_gl



<https://youtu.be/uwoj7R52yU8>

**PBR Material & PBR Environment Light Op - Byte Size - Physically Based Rendering in Cables**

by cables\_gl



<https://youtu.be/DW9U5tv1GHM>

**Varying Mesh Instances with color, animation and textures - Video Tutorial**

by cables\_gl



<https://youtu.be/Yf84KQc9jzU>

**Copy Texture operator deep dive - basics and use cases**

by cables\_gl



<https://youtu.be/PrkdNENo8wQ>

**Vertex Textures - Point Clouds and Mesh Instancing from Textures - Introduction**

by cables\_gl



<https://youtu.be/cc5Vlmvlq6A>

**Pixel displace op - byte size**

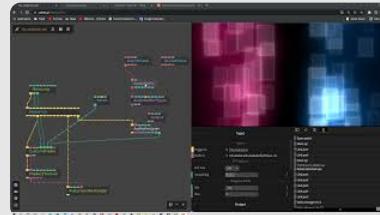
by cables\_gl

## 12.4 Textures / Post-Processing



<https://youtu.be/rtDA2S9SPQ4>

**Exploring Matcap Creator by bagoof - a new tool made with cables**  
by cables\_gl



<https://youtu.be/nil-HkZgNZ8>

**Programmation d'un shadertoy avec Cables.gl Partie 8.**  
by Meletou1

## 12.5 Shaders / Shadertoy / GLSL



<https://youtu.be/Zfhn8xSM0SE>  
**Coding with cables - custom shader op**  
by cables\_gl



[https://youtu.be/j\\_ins4RW0c8](https://youtu.be/j_ins4RW0c8)  
**Shadertoy to cables - part 01**  
by cables\_gl

## 12.6 Audio / Music / MIDI



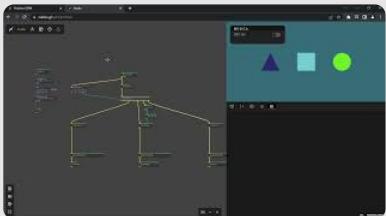
<https://youtu.be/SFXvtm-vkvE>  
**Introduction to Generative Music and Audio Reactive Systems with Cables.gl**  
by Jaalibandar



<https://youtu.be/h20ZH-xD8Ts>  
**Microphone Input & Audio Reactivity in Cables.gl - Tutorial**  
by The Interactive & Immersive HQ



<https://youtu.be/uYk7-9dZ8Ys>  
**MidiFighter cables.gl Vjing**  
by Alberto Barrios L. (nahui-ocelotl.com)



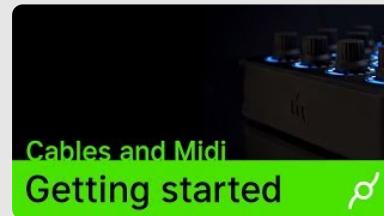
<https://youtu.be/KZbhVClahv4>  
**Páginas WEB Interactivas con cables.gl | 13 Audio**  
by Alberto Barrios L. (nahui-ocelotl.com)



<https://youtu.be/3m-2F2T1f6w>  
**Audio analyzer op - audio reactive**  
by cables\_gl



<https://youtu.be/68iSILnuLnA>  
**BiQuadFilter op- audio reactive tut**  
by cables\_gl



[https://youtu.be/eDlaFD\\_d5lc](https://youtu.be/eDlaFD_d5lc)  
**Connecting Midi controllers to Cables**  
by cables\_gl

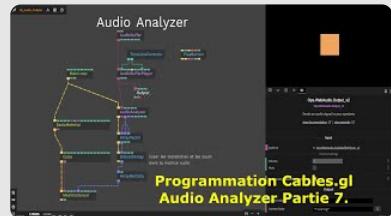


<https://youtu.be/wKQN2BZPtyU>  
**Exploring Spatial Audio in Cables.gl**  
by cables\_gl



<https://youtu.be/3owzslzvkdQ>

**Let's make some noise! Building a drum machine with Cables.gl.**  
by Kirell Benzi



<https://youtu.be/KtREXHa9tS8>

**Programmation Cables.gl Audio Analyzer Partie 7.**  
by Meletou1



[https://youtu.be/TyElawM\\_ll0](https://youtu.be/TyElawM_ll0)  
**Syncing Cables.gl with Bitwig Studio**  
by Stefan Sauer



<https://youtu.be/TlDHrXS06-A>  
**[animatic] Better! // bitwig studio, cables.gl**  
by voz-h-ko



<https://youtu.be/S-KyCySVucM>  
**[HD] Sidereal Collapse // cables.gl, Bitwig Studio**  
by voz-h-ko

## 12.7 Physics



[https://youtu.be/hlmNF\\_42raY](https://youtu.be/hlmNF_42raY)  
**AmmoRaycast Operator - creating a simple 3D menu UI - Tutorial**  
by cables\_gl



[https://youtu.be/TAhAqgY\\_EEs](https://youtu.be/TAhAqgY_EEs)

**AmmoWorld and AmmoBody Operators - physics simulations in cables.gl - Video Tutorial**  
by cables\_gl

## 12.8 Export / Deployment / Embedding

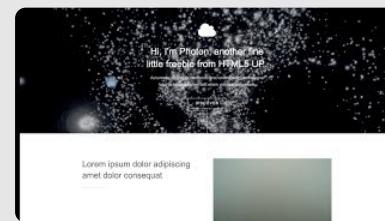


[https://youtu.be/DX0sISkR\\_Hg](https://youtu.be/DX0sISkR_Hg)

**Páginas WEB Interactivas con cables.gl | 20 Exportación**  
by Alberto Barrios L. (nahui-ocelotl.com)



<https://youtu.be/J8yJtcd1Jeg>  
**CABLES Command Line export**  
by cables\_gl



[https://youtu.be/YUAYs\\_NcwTA](https://youtu.be/YUAYs_NcwTA)

**embed a cables patch into a html website**  
by cables\_gl



<https://youtu.be/B4M9FddXk1I>

**Exporting your Project - .zip Export - Byte Size**  
by cables\_gl



<https://youtu.be/L5BGMs7vKuI>

**Exporting your Project - Netlify export - Byte Size**  
by cables\_gl



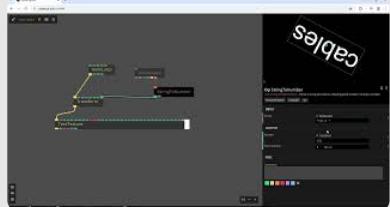
<https://youtu.be/hVxrxXhH7vQ>

**Cables.gl Standalone (Offline) Build: Create Without Limits!**  
by Decode GL



<https://youtu.be/4YsuGFAEvEE>  
**Phidget Encoder in cables.gl**  
by wirmachenbunt

## 12.9 Hardware / External Tools



<https://youtu.be/vebGfUp9vJ4>

**Getting cables.gl to talk to hardware, using Chataigne!**  
by Rob Duarte

## 12.10 Talks / Meetups / Release Notes



[https://youtu.be/FvC3Ec\\_38Jo](https://youtu.be/FvC3Ec_38Jo)  
**Inércia 2023 | Seminar: Cables.gl as a demo making tool by anticore feat. liquebe**  
by Associação Inércia



<https://youtu.be/xLBL06O1kXg>  
**cables.gl october meetup**  
by cables\_gl



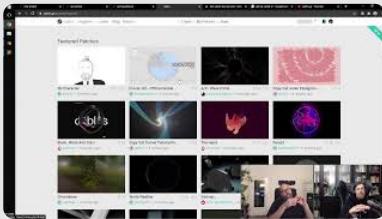
<https://youtu.be/xRbg1Az0k8k>

**November Update - cables.gl monthly meetup**  
by cables\_gl



<https://youtu.be/C2FjpdRWPxw>

**Updated Physically Based Rendering Operators - discussion with the developer AMajesticSeaFlapFlap**  
by cables\_gl



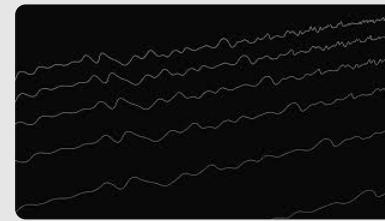
<https://youtu.be/v4rYqHuT-0E>

**Seminar: Making demos with cables.gl (speaker: pandur)**  
by psenough



<https://youtu.be/oLPsJd0e4Gc>

**antonymph - vylet pony (avoset remix; cables.gl visualiser)**  
by avoset



<https://youtu.be/CfPJZMAsxcTU>

**Lines / Live experience with Cables.gl**  
by BoatBoat\_Station



[https://youtu.be/Zr\\_7wRBmRmA](https://youtu.be/Zr_7wRBmRmA)

**Building a VJ patch mixer with cables.gl**  
by cables\_gl

## 12.11 Showcases / Demos / Visualizers



<https://youtu.be/84pXsmJghdM>  
**demomaking with cables**  
by cables\_gl



<https://youtu.be/M8ls131LSzE>  
**hydra - demo by mfx**  
by cables\_gl



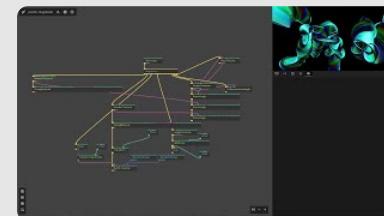
<https://youtu.be/R9-D4SxBd90>  
**Ninja de Gaia - Inércia 2023 - creating a demo with cables.gl**  
by cables\_gl



<https://youtu.be/auvD8oSxMew>  
**cables.gl - Drifting Apart (FXHash Project)**  
by Creative Exploration /w Purz



[https://youtu.be/9vZzrXX\\_2jM](https://youtu.be/9vZzrXX_2jM)  
**cables.gl - purzOS - Low Poly Lavalamp (FXHash Project)**  
by Creative Exploration /w Purz

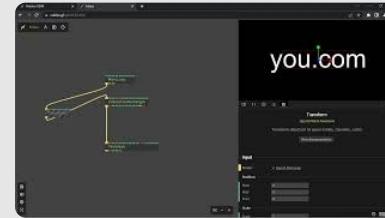


<https://youtu.be/a0IJ8DF-v8o>  
**cables.gl - purzOS - Ring Worlds (Screensaver)**  
by Creative Exploration /w Purz



<https://youtu.be/EO3UdeBQ9m0>

**EroLogo - Visual Demo Length 12:37 made with Cables.gl**  
by faktisProductions



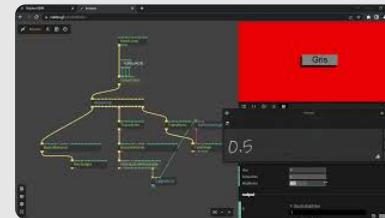
[https://youtu.be/oBoH\\_7uHv-E](https://youtu.be/oBoH_7uHv-E)

**Páginas WEB Interactivas con cables.gl | 02 Enlace**  
by Alberto Barrios L. (nahui-ocelotl.com)



<https://youtu.be/xba3e91Fum4>

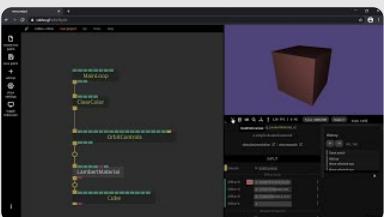
**Design Designs Design - "Smorp" (A Cables.gl demo for Evoke 2022)**  
by Jan-Jozef Tuigstra



<https://youtu.be/-9QrZSoAPpQ>

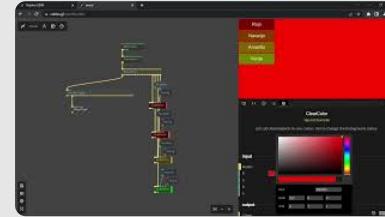
**Páginas WEB Interactivas con cables.gl | 07 Botones**  
by Alberto Barrios L. (nahui-ocelotl.com)

## 12.12 Unsorted (Still cables.gl-related)



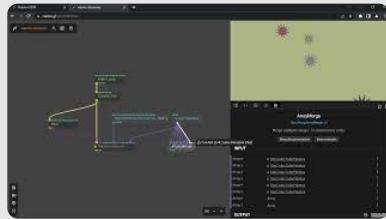
<https://youtu.be/1FqBKJ1RXdY>

**Entornos virtuales WEB con programación visual en cables.gl Parte 1**  
by Alberto Barrios L. (nahui-ocelotl.com)



<https://youtu.be/MTeI06T-kGw>

**Páginas WEB Interactivas con cables.gl | 08 Menu**  
by Alberto Barrios L. (nahui-ocelotl.com)



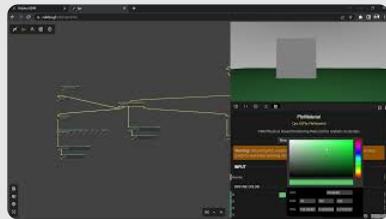
<https://youtu.be/iFDD4tm7-Uw>

**Páginas WEB Interactivas con cables.gl | 15 Valores Aleatorios**  
by Alberto Barrios L. (nahui-ocelotl.com)



<https://youtu.be/7BiDxNc7D7g>

**Create awesome Visuals using OpenDAW and cables.gl!**  
by BeatMax\_Prediction



<https://youtu.be/a2H8vk3Ko1M>

**Páginas WEB Interactivas con cables.gl | 18 FPS**  
by Alberto Barrios L. (nahui-ocelotl.com)



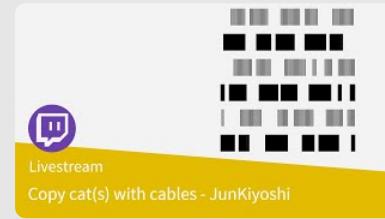
<https://youtu.be/omIK1YOtB70>

**Copy cat with cables- live stream - Inconvergent**  
by cables\_gl



<https://youtu.be/cVpC9IS6kI0>

**Substitution Pattern / Testing / CABLES.GL /**  
by Antiguo Autómata Mexicano



<https://youtu.be/tu49qg8BpBU>

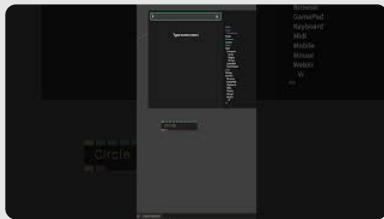
**copy cat(s) with cables live stream - Junkiyoshi**  
by cables\_gl



Tyler hobbs - Untitled Cityscape



[https://youtu.be/Gr3iVMUs\\_hA](https://youtu.be/Gr3iVMUs_hA)  
**Copycat with Cables - Tyler hobbs - Untitled**  
by cables\_gl



<https://youtu.be/hZQZsh5UHSE>  
**did you know, you can add multiple ops one go**  
by cables\_gl



Repeat op tutorial 01



<https://youtu.be/jiOLZaMUH78>  
**Repeat op tut 01 - Byte size**  
by cables\_gl



<https://youtu.be/00Rvb749wrc>  
**Smooth Operator - Byte Size**  
by cables\_gl



<https://youtu.be/8Lfr8iLLbMA>  
**Infinite Looping Motion Graphic in 10 minutes using cables.gl**  
by Jaalibandar

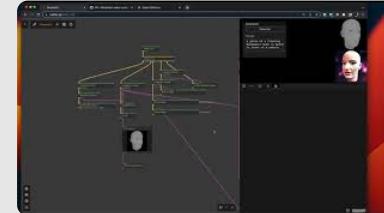


<https://youtu.be/WGoM1AmfW7g>  
**Getting data from an API with cables.gl - data-driven gradient from geo-located weather - part 1**  
by Kirell Benzi



<https://youtu.be/G1HKysL8iVw>

**Présentation du logiciel Cables.gl par les étudiants en UI/UX design**  
by L'École de design Nantes Atlantique



<https://youtu.be/n4UPiZhbcRU>

**StableDiffusion and ControlNet in Cables.gl via the WebUI**  
by Nighth Allen



<https://youtu.be/4Op74ulzH5c>

**Retour sur le programme Cables.gl**  
by Meletou1



<https://youtu.be/lImv9ZJshUE>

**cables.gl and ollama API**  
by Tobias Hartmann



<https://youtu.be/tdbTTxDu7Qk>

**Cables.gl**  
by Nathan Sonzogni



<https://youtu.be/vOVKpppw1ds>

**Class 30: Learning how to mint a cables.gl patch on fx hash w/ Somaticbits**  
by VERTICAL

# 13 Ops.Anim

## 13.1 Ops.Anim

### 13.1.1 AnimNumber



**Full Name:** Ops.Anim.AnimNumber

**Description:** Always animates to the current value

#### > Input Ports:

- **Exe** (Trigger)
- **Value** (Number)
- **Duration** (Number)
- **Easing Index** (Number: Integer)

#### < Output Ports:

- **Next** (Trigger)
- **Result** (Number)
- **Finished** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Anim.AnimNumber>

### 13.1.2 Bang



**Full Name:** Ops.Anim.Bang

**Description:** Trigger a simple bang animation going from 1 to 0

#### > Input Ports:

- **Update** (Trigger)
- **Bang** (Trigger)
- **Duration** (Number)
- **Invert** (Number: Boolean)

**< Output Ports:**

- **Trigger Out** (Trigger)
- **Value** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Anim.Bang>

### 13.1.3 BoolAnim



**Full Name:** Ops.Anim.BoolAnim

**Description:** Animate between two numbers based on a boolean value

**> Input Ports:**

- **Exe** (Trigger)
- **Bool** (Number: Boolean)
- **Easing Index** (Number: Integer)
- **Duration** (Number)
- **Direction Index** (Number: Integer)
- **Value False** (Number)
- **Value True** (Number)

**< Output Ports:**

- **Trigger** (Trigger)
- **Value** (Number)
- **Finished** (booleanNumber)
- **Finished Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Anim.BoolAnim>

### 13.1.4 Crossfade



**Full Name:** Ops.Anim.Crossfade

**Description:** Crossfade between 2 values

**> Input Ports:**

- **Crossfade** (Number)
- **Out Min** (Number)
- **Out Max** (Number)
- **Easing Index** (Number: Integer)

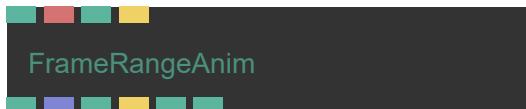
**< Output Ports:**

- **A** (Number)
- **B** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Anim.Crossfade>

### 13.1.5 FrameRangeAnim\_v2



**Full Name:** Ops.Anim.FrameRangeAnim\_v2

**Description:** Parses string containing ranges of frames and play as coherent animation

**> Input Ports:**

- **Time** (Number)
- **Frames** (String)
- **frame range** (ex. "0-10")
- **Loop** (Number: Boolean)
- **Rewind** (Trigger)

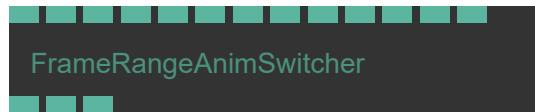
**< Output Ports:**

- **Result Time** (Number)
- **Expanded Frames** (Array)
- **Finished** (booleanNumber)
- **Finished Trigger** (Trigger)
- **Anim Length** (Number)
- **Progress** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Anim.FrameRangeAnim\\_v2](https://cables.gl/op/Ops.Anim.FrameRangeAnim_v2)

### 13.1.6 FrameRangeAnimSwitcher



**Full Name:** Ops.Anim.FrameRangeAnimSwitcher

**Description:** Switch between multiple anim ranges of a keyframed 3d scene

#### > Input Ports:

- **Index** (Number: Integer)
- **Duration** (Number)
- **Easing Index** (Number: Integer)
- **Value 0** (Number)
- **Value 1** (Number)
- **Value 2** (Number)
- **Value 3** (Number)
- **Value 4** (Number)
- **Value 5** (Number)
- **Value 6** (Number)
- **Value 7** (Number)
- **Value 8** (Number)
- **Value 9** (Number)

#### < Output Ports:

- **Time 1** (Number)
- **Time Fade** (Number)
- **Time 2** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Anim.FrameRangeAnimSwitcher>

### 13.1.7 InOutInAnim



**Full Name:** Ops.Anim.InOutInAnim

**Description:** Animates after a trigger from 1 to 0 to 1

#### > Input Ports:

- **Update** (Trigger)
- **Duration In** (Number)
- **Easing In Index** (Number: Integer)
- **Value In** (Number)
- **Hold Duration** (Number)
- **Duration Out** (Number)
- **Easing Out Index** (Number: Integer)
- **Value Out** (Number)
- **Start** (Trigger)

#### < Output Ports:

- **Next** (Trigger)
- **Result** (Number)
- **Started** (Trigger)
- **Middle** (Trigger)
- **Finished** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Anim.InOutInAnim>

### 13.1.8 LFO\_v3



**Full Name:** Ops.Anim.LFO\_v3

**Description:** Low-frequency oscillation for animations

**> Input Ports:**

- **Time** (Number)
- **Frequency** (Number)
- **Type Index** (Number: Integer)
- **Phase** (Number)
- **Range Min** (Number)
- **Range Max** (Number)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Anim.LFO\\_v3](https://cables.gl/op/Ops.Anim.LFO_v3)

## 13.1.9 RandomAnim\_v2



**Full Name:** Ops.Anim.RandomAnim\_v2

**Description:** Animates between random values defined by a min and max value

**> Input Ports:**

- **Exe** (Trigger)
- **Min** (Number)
- **Max** (Number)
- **Duration** (Number)
- **Pause Between** (Number)
- **Easing Index** (Number: Integer)

**< Output Ports:**

- **Next** (Trigger)
- **Result** (Number)
- **Looped** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Anim.RandomAnim\\_v2](https://cables.gl/op/Ops.Anim.RandomAnim_v2)

## 13.1.10 SimpleAnim



**Full Name:** Ops.Anim.SimpleAnim

**Description:** Simple animation between two values

**> Input Ports:**

- **Exe** (Trigger)
- **Reset** (Trigger)
- **Rewind** (Trigger)
- **Start** (Number)
- **End** (Number)
- **Duration** (Number)
- **Loop** (Number: Boolean)
- **Wait For Reset** (Number: Boolean)
- **Easing Index** (Number: Integer)

**< Output Ports:**

- **Next** (Trigger)
- **Result** (Number)
- **Finished** (Number)
- **Finished Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Anim.SimpleAnim>

## 13.1.11 SineAnim



**Full Name:** Ops.Anim.SineAnim

**Description:** Animation in the form of a sine/cosine curve (sinus/cos)

#### > Input Ports:

- **Exe** (Trigger)
- **Mode Index** (Number: Integer)
- **Phase** (Number)
- **Frequency** (Number)
- **Amplitude** (Number)

#### < Output Ports:

- **Trigger Out** (Trigger)
- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Anim.SineAnim>

## 13.1.12 Smooth



**Full Name:** Ops.Anim.Smooth

**Description:** Smooths out jumps in values (AverageInterpolation)

#### > Input Ports:

- **Update** (Trigger)
- **Value** (Number)
- **Dec Factor** (Number)

#### < Output Ports:

- **Next** (Trigger)
- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Anim.Smooth>

## 13.1.13 Snap



**Full Name:** Ops.Anim.Snap

**Description:** Snap at certain points (e.g. while scrolling)

#### > Input Ports:

- **Delta** (Number)
- **Snap At Values** (Array)
- **Snap Distance** (Number)
- **Snap Distance Release** (Number)
- **Slowdown** (Number)
- **Block Input After Snap** (Number)
- **Reset** (Trigger)
- **Min** (Number)
- **Max** (Number)
- **Value Mul** (Number)
- **Enabled** (Number: Boolean)

#### < Output Ports:

- **Result** (Number)
- **Distance** (Number)
- **Snapped** (Number)
- **Was Snapped** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Anim.Snap>

## 13.1.14 Spring



**Full Name:** Ops.Anim.Spring

**Description:** Spring simulation based on input target value.

#### > Input Ports:

- **Exe** (Trigger)
- **Value** (Number)

- **Damping** (Number)
- **Stiffness** (Number)

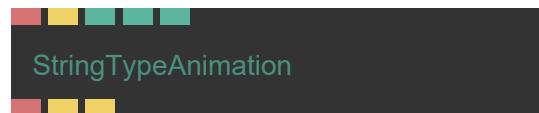
**< Output Ports:**

- **Trigger** (Trigger)
- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Anim.Spring>

### 13.1.15 StringTypeAnimation\_v2



**Full Name:** Ops.Anim.StringTypeAnimation\_v2

**Description:** Animates a text/string, like it is being typed out by a person

**> Input Ports:**

- **Text** (String)
- **Restart** (Trigger)
- **Speed** (Number)
- **Speed Variation** (Number)
- **Show Cursor** (Number: Boolean)

**< Output Ports:**

- **Result** (String)
- **Changed** (Trigger)
- **Finished** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Anim.StringTypeAnimation\\_v2](https://cables.gl/op/Ops.Anim.StringTypeAnimation_v2)

### 13.1.16 TimeDelta



**Full Name:** Ops.Anim.TimeDelta

**Description:** Measure the time difference between two triggers

**> Input Ports:**

- **Exe** (Trigger)
- **Smooth** (Number: Boolean)
- **Seconds** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)
- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Anim.TimeDelta>

### 13.1.17 Timer\_v2



**Full Name:** Ops.Anim.Timer\_v2

**Description:** A timer that can be started, paused and reset by triggering

**> Input Ports:**

- **Speed** (Number)
- **Play** (Number: Boolean)
- **Reset** (Trigger)
- **Sync To Timeline** (Number: Boolean)

**< Output Ports:**

- **Time** (Number)

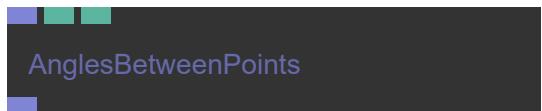
**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Anim.Timer\\_v2](https://cables.gl/op/Ops.Anim.Timer_v2)

# 14 Ops.Array

## 14.1 Ops.Array

### 14.1.1 AnglesBetweenPoints



**Full Name:** Ops.Array.AnglesBetweenPoints

**Description:** Outputs the angle between points in 3D space (degree)

#### > Input Ports:

- Points (Array)
- Theta (Number)
- Phi (Number)

#### < Output Ports:

- Rotations (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.AnglesBetweenPoints>

### 14.1.2 AnimArray\_v2



**Full Name:** Ops.Array.AnimArray\_v2

**Description:** Animate values in an array to another array

#### > Input Ports:

- Update (Trigger)
- Next Array (Array)
- Duration (Number)
- Easing Index (Number: Integer)

#### < Output Ports:

- Next (Trigger)
- Matrix (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.AnimArray\\_v2](https://cables.gl/op/Ops.Array.AnimArray_v2)

### 14.1.3 Array1toX\_v2



**Full Name:** Ops.Array.Array1toX\_v2

**Description:** convert an array1 to array2,3,4 by choosing content for new axis

#### > Input Ports:

- Array1x (Array)

#### < Output Ports:

- Array3x (Array)
- Total Points (Number)
- Array Length (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.Array1toX\\_v2](https://cables.gl/op/Ops.Array.Array1toX_v2)

### 14.1.4 Array2To3



**Full Name:** Ops.Array.Array2To3

**Description:** Inserts zeroes every third item

#### > Input Ports:

- Array2x (Array)

#### < Output Ports:

- Array3x (Array)
- Total Points (Number)
- Array Length (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Array2To3>

## 14.1.5 Array3



**Full Name:** Ops.Array.Array3

**Description:** Create an array of num triplets set to default values xyz

**> Input Ports:**

- **Num Triplets** (Number: Integer)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)

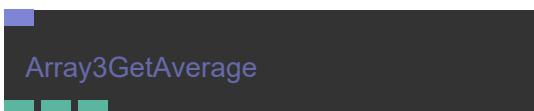
**< Output Ports:**

- **Array** (Array)
- **Total Points** (Number)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Array3>

## 14.1.6 Array3GetAverage



**Full Name:** Ops.Array.Array3GetAverage

**Description:** Average x,y,z values of an array3x

**> Input Ports:**

- **Array** (Array)

**< Output Ports:**

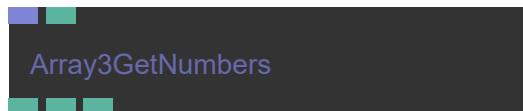
- **Average X** (Number)
- **Average Y** (Number)

- **Average Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Array3GetAverage>

## 14.1.7 Array3GetNumbers



**Full Name:** Ops.Array.Array3GetNumbers

**Description:** Get 3 values XYZ from an array

**> Input Ports:**

- **Array** (Array)
- **Index** (Number: Integer)

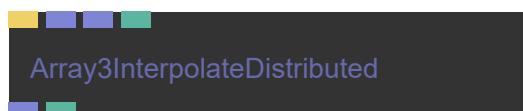
**< Output Ports:**

- **X** (Number)
- **Y** (Number)
- **Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Array3GetNumbers>

## 14.1.8 Array3InterpolateDistributed



**Full Name:** Ops.Array.Array3InterpolateDistributed

**Description:** Interpolate between two arrays

**> Input Ports:**

- **Update** (Trigger)
- **Array 1** (Array)
- **Array 2** (Array)
- **Progress** (Number)

**< Output Ports:**

- **Result** (Array)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Array3InterpolateDistributed>

## 14.1.9 Array3Iterator



**Full Name:** Ops.Array.Array3Iterator

**Description:** Iterate over an array in steps of three and outputs three values

**> Input Ports:**

- **Execute** (Trigger)
- **Array** (Array)
- **Step** (Number)

**< Output Ports:**

- **Trigger** (Trigger)
- **Triggers for every iteration step** (triplet in the array)
- **Index** (Number)
- **Value 1** (Number)
- **First value of the current triplet** (e.g. x)
- **Value 2** (Number)
- **Second value of the current triplet** (e.g. y)
- **Value 3** (Number)
- **Third value of the current triplet** (e.g. z)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Array3Iterator>

## 14.1.10 Array3Multiply



**Full Name:** Ops.Array.Array3Multiply

**Description:** Multiply every XYZ member of array3x

**> Input Ports:**

- **Array3x** (Array)
- **Mul X** (Number)
- **Mul Y** (Number)
- **Mul Z** (Number)

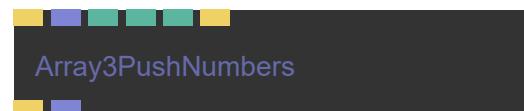
**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Array3Multiply>

## 14.1.11 Array3PushNumbers\_v2



**Full Name:** Ops.Array.Array3PushNumbers\_v2

**Description:** Push three numbers to the end of an array (was ArrayPushValue3x)

**> Input Ports:**

- **Execute** (Trigger)
- **Array** (Array)
- **Value 1** (Number)
- **Value 2** (Number)
- **Value 3** (Number)
- **Reset** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **Result Array** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.Array3PushNumbers\\_v2](https://cables.gl/op/Ops.Array.Array3PushNumbers_v2)

## 14.1.12 Array3RandomSelection



**Full Name:** Ops.Array.Array3RandomSelection

**Description:** Extract definable amount of random xyz points from an array

**> Input Ports:**

- **Array** (Array)
- **Elements** (Number: Integer)
- **Seed** (Number)

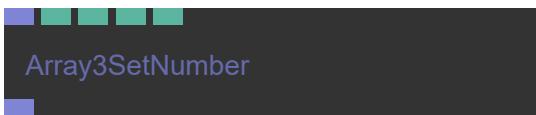
**< Output Ports:**

- **Result** (Array)
- **Total Points** (Number)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Array3RandomSelection>

## 14.1.13 Array3SetNumber



**Full Name:** Ops.Array.Array3SetNumber

**Description:** Set three numbers at index in an array

**> Input Ports:**

- **Array** (Array)
- **Index** (Number: Integer)
- **Value X** (Number)
- **Value Y** (Number)
- **Value Z** (Number)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Array3SetNumber>

## 14.1.14 Array3Sum



**Full Name:** Ops.Array.Array3Sum

**Description:** Add number to every XYZ member of array3x

**> Input Ports:**

- **Array3x** (Array)
- **Add X** (Number)
- **Add Y** (Number)
- **Add Z** (Number)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Array3Sum>

## 14.1.15 Array3To2



**Full Name:** Ops.Array.Array3To2

**Description:** Remove every 3rd item of an array - changes array length

**> Input Ports:**

- **Array3x** (Array)

**< Output Ports:**

- **Array2x** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Array3To2>

## 14.1.16 Array3To4



**Full Name:** Ops.Array.Array3To4

**Description:** Convert an array3 to an array4 by filling it up with 1

**> Input Ports:**

- **Array3x** (Array)

**< Output Ports:**

- **Array4x** (Array)
- **Total Points** (Number)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Array3To4>

## 14.1.17 Array3VectorLength



**Full Name:** Ops.Array.Array3VectorLength

**Description:** Return the length of a vector from an array 3

**> Input Ports:**

- **Array In** (Array)

**< Output Ports:**

- **Array Out** (Array)
- **Array Lengths** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Array3VectorLength>

## 14.1.18 Array4



**Full Name:** Ops.Array.Array4

**Description:** Create an array of num quadruples set to default values xyz

**> Input Ports:**

- **Num Quadruplets** (Number: Integer)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **W** (Number)

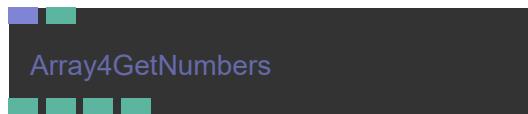
**< Output Ports:**

- **Array** (Array)
- **Total Quadruplets** (Number)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Array4>

## 14.1.19 Array4GetNumbers



**Full Name:** Ops.Array.Array4GetNumbers

**Description:** Get 4 values from an array

**> Input Ports:**

- **Array** (Array)
- **Index** (Number: Integer)

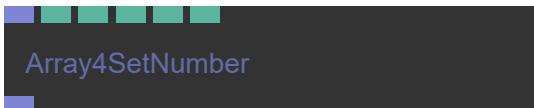
**< Output Ports:**

- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **W** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Array4GetNumbers>

## 14.1.20 Array4SetNumber



**Full Name:** Ops.Array.Array4SetNumber

**Description:** Set four numbers at index in an array

**> Input Ports:**

- **Array** (Array)
- **Index** (Number: Integer)
- **Value X** (Number)
- **Value Y** (Number)
- **Value Z** (Number)
- **Value W** (Number)

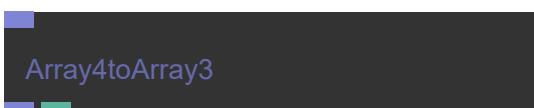
**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Array4SetNumber>

## 14.1.21 Array4toArray3



**Full Name:** Ops.Array.Array4toArray3

**Description:** Convert an array4 to array3 by dropping every 4th number

**> Input Ports:**

- **Array** (Array)

**< Output Ports:**

- **Result** (Array)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Array4toArray3>

## 14.1.22 Array\_v3



**Full Name:** Ops.Array.Array\_v3

**Description:** Can generate 3 kinds of arrays: Number - 1,2,3,4 - Normalized - (ContinuousNumberArray)

**> Input Ports:**

- **Array Length** (Number: Integer)
- **Mode Select Index** (Number: Integer)
- **Default Value** (Number)
- **Reverse** (Number: Boolean)

**< Output Ports:**

- **Array** (Array)
- **Array Length Out** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.Array\\_v3](https://cables.gl/op/Ops.Array.Array_v3)

## 14.1.23 ArrayAbs



**Full Name:** Ops.Array.ArrayAbs

**Description:** Converts array contents to absolute values - converts all negative numbers to positive numbers

**> Input Ports:**

- **In** (Array)

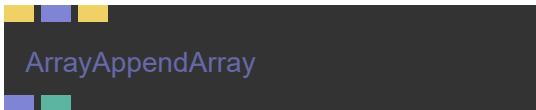
**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayAbs>

## 14.1.24 ArrayAppendArray



**Full Name:** Ops.Array.ArrayAppendArray

**Description:** Append an array to an existing array

**> Input Ports:**

- **Join** (Trigger)
- **Array** (Array)
- **Reset** (Trigger)

**< Output Ports:**

- **Result** (Array)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayAppendArray>

## 14.1.25 ArrayBuffer



**Full Name:** Ops.Array.ArrayBuffer

**Description:** Store values in an array / fifo array buffer

**> Input Ports:**

- **Exec** (Trigger)
- **Value** (Number)
- **Max Length** (Number: Integer)
- **Reset** (Trigger)

**< Output Ports:**

- **Trigger Out** (Trigger)

- **Result** (Array)

- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayBuffer>

## 14.1.26 ArrayBuffer3



**Full Name:** Ops.Array.ArrayBuffer3

**Description:** Circular buffer for xyz values

**> Input Ports:**

- **Exec** (Trigger)
- **Max Num Elements** (Number)
- **Value X** (Number)
- **Value Y** (Number)
- **Value Z** (Number)
- **Reset** (Trigger)

**< Output Ports:**

- **Trigger Out** (Trigger)
- **Result** (Array)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayBuffer3>

## 14.1.27 ArrayCeil



**Full Name:** Ops.Array.ArrayCeil

**Description:** Round numbers up

**> Input Ports:**

- **In** (Array)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayCeil>

## 14.1.28 ArrayChunk



**Full Name:** Ops.Array.ArrayChunk

**Description:** Extracts x elements from an array

**> Input Ports:**

- **Input Array** (Array)
- **Begin Index** (Number: Integer)
- **Chunk Size** (Number: Integer)
- **Circular** (Number: Boolean)

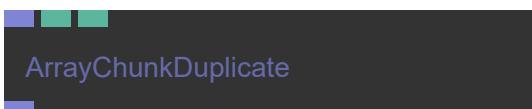
**< Output Ports:**

- **Output Array** (Array)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayChunk>

## 14.1.29 ArrayChunkDuplicate



**Full Name:** Ops.Array.ArrayChunkDuplicate

**Description:** Repeat chunks of an array multiple times

**> Input Ports:**

- **Array** (Array)
- **Chunk Size** (Number: Integer)

- **Repeats** (Number: Integer)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayChunkDuplicate>

## 14.1.30 ArrayClamp



**Full Name:** Ops.Array.ArrayClamp

**Description:** Clamp the values of an array to a min and max value

**> Input Ports:**

- **Array In** (Array)
- **Min** (Number)
- **Max** (Number)

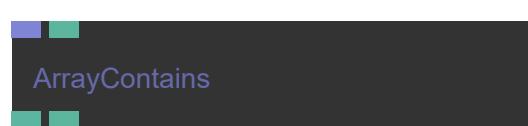
**< Output Ports:**

- **Array Out** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayClamp>

## 14.1.31 ArrayContains\_v2



**Full Name:** Ops.Array.ArrayContains\_v2

**Description:** Check if an array contains a number (find,search,indexOf)

**> Input Ports:**

- **Array** (Array)
- **SearchValue** (Number)

**< Output Ports:**

- **Found** (booleanNumber)
- **Index** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.ArrayContains\\_v2](https://cables.gl/op/Ops.Array.ArrayContains_v2)

## 14.1.32 ArrayDivide



**Full Name:** Ops.Array.ArrayDivide

**Description:** Divide all values in an array by one number

**> Input Ports:**

- **Array In** (Array)
- **Value** (Number)

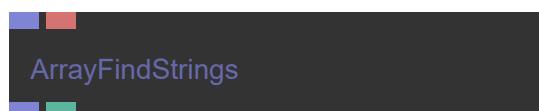
**< Output Ports:**

- **Array Out** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayDivide>

## 14.1.33 ArrayFindStrings



**Full Name:** Ops.Array.ArrayFindStrings

**Description:** Return all the indexes of a string in an array

**> Input Ports:**

- **Array** (Array)
- **SearchValue** (String)

**< Output Ports:**

- **Index** (Array)
- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayFindStrings>

## 14.1.34 ArrayFloor



**Full Name:** Ops.Array.ArrayFloor

**Description:** Round numbers down

**> Input Ports:**

- **In** (Array)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayFloor>

## 14.1.35 ArrayFract



**Full Name:** Ops.Array.ArrayFract

**Description:** Return the fractional remainder of all values in an array

**> Input Ports:**

- **In** (Array)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayFract>

## 14.1.36 ArrayFromNumbers



**Full Name:** Ops.Array.ArrayFromNumbers

**Description:** Simple way to create small arrays of numbers

### > Input Ports:

- **Update** (Trigger)
- **Limit** (Number: Integer)
- **Slider** (Number: Boolean)
- **Index 0** (Number)
- **Index 1** (Number)
- **Index 2** (Number)
- **Index 3** (Number)
- **Index 4** (Number)
- **Index 5** (Number)
- **Index 6** (Number)
- **Index 7** (Number)
- **Index 8** (Number)
- **Index 9** (Number)
- **Index 10** (Number)
- **Index 11** (Number)
- **Index 12** (Number)
- **Index 13** (Number)
- **Index 14** (Number)
- **Index 15** (Number)
- **Index 16** (Number)
- **Index 17** (Number)
- **Index 18** (Number)
- **Index 19** (Number)
- **Index 20** (Number)
- **Index 21** (Number)
- **Index 22** (Number)
- **Index 23** (Number)
- **Index 24** (Number)
- **Index 25** (Number)
- **Index 26** (Number)
- **Index 27** (Number)

- **Index 28** (Number)
- **Index 29** (Number)

### < Output Ports:

- **Next** (Trigger)
- **Array** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayFromNumbers>

## 14.1.37 ArrayGetArray



**Full Name:** Ops.Array.ArrayGetArray

**Description:** Get an array from an array of arrays

### > Input Ports:

- **Array Of Arrays** (Array)
- **Index** (Number: Integer)

### < Output Ports:

- **Result Array** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayGetArray>

## 14.1.38 ArrayGetNumber



**Full Name:** Ops.Array.ArrayGetNumber

**Description:** Return a value from an array

### > Input Ports:

- **Array** (Array)
- **Index** (Number: Integer)
- **Value Invalid Index** (Number)

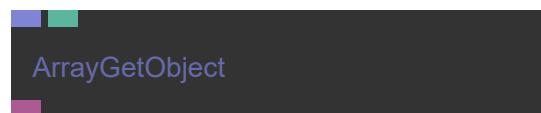
**< Output Ports:**

- **Value** (Number)
- **Valid Index** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayGetNumber>

### 14.1.39 ArrayGetObject



**Full Name:** Ops.Array.ArrayGetObject

**Description:** Get an object from an array

**> Input Ports:**

- **Array** (Array)
- **Index** (Number: Integer)

**< Output Ports:**

- **Value** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayGetObject>

### 14.1.40 ArrayGetString\_v2



**Full Name:** Ops.Array.ArrayGetString\_v2

**Description:** Get a string from an array at [index]

**> Input Ports:**

- **Array** (Array)
- **Index** (Number: Integer)

**< Output Ports:**

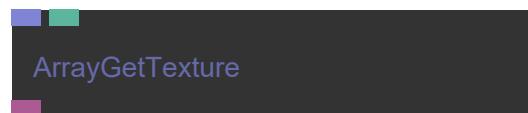
- **Result** (String)

- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.ArrayGetString\\_v2](https://cables.gl/op/Ops.Array.ArrayGetString_v2)

### 14.1.41 ArrayGetTexture



**Full Name:** Ops.Array.ArrayGetTexture

**Description:** Get texture from array at index

**> Input Ports:**

- **Array** (Array)
- **Index** (Number: Integer)

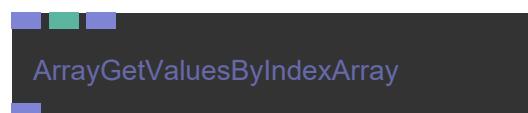
**< Output Ports:**

- **Value** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayGetTexture>

### 14.1.42 ArrayGetValuesByIndexArray



**Full Name:** Ops.Array.ArrayGetValuesByIndexArray

**Description:** Pick values from input array at given indices and stride

**> Input Ports:**

- **Array** (Array)
- **Array Stride Index** (Number: Integer)
- **Indices** (Array)

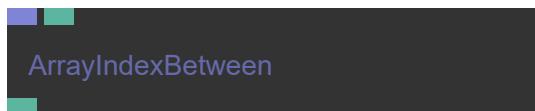
**< Output Ports:**

- **Results** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayGetValuesByIndexArray>

## 14.1.43 ArrayIndexBetween



**Full Name:** Ops.Array.ArrayIndexBetween

**Description:** Output index where value is greater than number and smaller than next number

**> Input Ports:**

- **Array** (Array)
- **Value** (Number)

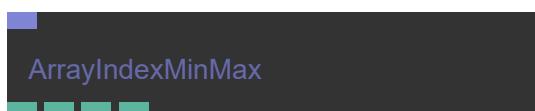
**< Output Ports:**

- **Index** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayIndexBetween>

## 14.1.44 ArrayIndexMinMax



**Full Name:** Ops.Array.ArrayIndexMinMax

**Description:** Find lowest/highest numbers in an array

**> Input Ports:**

- **Array** (Array)

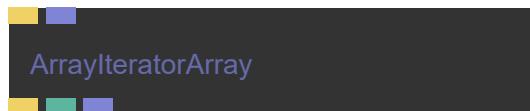
**< Output Ports:**

- **Max** (Number)
- **Index Max** (Number)
- **Min** (Number)
- **Index Min** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayIndexMinMax>

## 14.1.45 ArrayIteratorArray



**Full Name:** Ops.Array.ArrayIteratorArray

**Description:** Iterate over an array of arrays

**> Input Ports:**

- **Exe** (Trigger)
- **Array** (Array)

**< Output Ports:**

- **Trigger** (Trigger)
- **Index** (Number)
- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayIteratorArray>

## 14.1.46 ArrayIteratorNumbers



**Full Name:** Ops.Array.ArrayIteratorNumbers

**Description:** Loop over every element of an array

**> Input Ports:**

- **Exe** (Trigger)
- **Array** (Array)

**< Output Ports:**

- **Trigger** (Trigger)
- **Index** (Number)
- **Value** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayIteratorNumbers>

## 14.1.47 ArrayIteratorObjects

ArrayIteratorObjects

**Full Name:** Ops.Array.ArrayIteratorObjects

**Description:** Iterate over an array of objects

**> Input Ports:**

- **Exe** (Trigger)
- **Array** (Array)

**< Output Ports:**

- **Trigger** (Trigger)
- **Finished** (Trigger)
- **Index** (Number)
- **Value** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayIteratorObjects>

## 14.1.48 ArrayIteratorStrings

ArrayIteratorStrings

**Full Name:** Ops.Array.ArrayIteratorStrings

**Description:** Loop over every element of an array

**> Input Ports:**

- **Exe** (Trigger)
- **Array** (Array)

**< Output Ports:**

- **Trigger** (Trigger)
- **Index** (Number)
- **Value** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayIteratorStrings>

## 14.1.49 ArrayIteratorTextures

ArrayIteratorTextures

**Full Name:** Ops.Array.ArrayIteratorTextures

**Description:** Iterate over an array of objects

**> Input Ports:**

- **Exe** (Trigger)
- **Array** (Array)

**< Output Ports:**

- **Trigger** (Trigger)
- **Finished** (Trigger)
- **Index** (Number)
- **Value** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayIteratorTextures>

## 14.1.50 ArrayLength\_v2

ArrayLength

**Full Name:** Ops.Array.ArrayLength\_v2

**Description:** Number of items in an array

**> Input Ports:**

- **Array** (Array)

**< Output Ports:**

- **Length** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.ArrayLength\\_v2](https://cables.gl/op/Ops.Array.ArrayLength_v2)

## 14.1.51 ArrayLogic



**Full Name:** Ops.Array.ArrayLogic

**Description:** Performs logical comparison operations on a single array of numbers

**> Input Ports:**

- **Array 0** (Array)
- **Comparison Mode Index** (Number: Integer)
- **Number For Comparison** (Number)
- **Value If True** (Number)
- **Value If False** (Number)

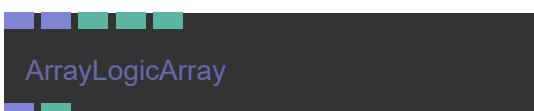
**< Output Ports:**

- **Array Result** (Array)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayLogic>

## 14.1.52 ArrayLogicArray



**Full Name:** Ops.Array.ArrayLogicArray

**Description:** Performs logical comparison operations on two arrays

**> Input Ports:**

- **Array 0** (Array)
- **Array 1** (Array)
- **Value If True** (Number)
- **Value If False** (Number)
- **Comparison Mode Index** (Number: Integer)

**< Output Ports:**

- **Array Result** (Array)

- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayLogicArray>

## 14.1.53 ArrayLogicBetween\_v2



**Full Name:** Ops.Array.ArrayLogicBetween\_v2

**Description:** If value of array is between min and max then the value is 1 else 0

**> Input Ports:**

- **Array** (Array)
- **Min** (Number)
- **Max** (Number)
- **Pass Value When True** (Number: Boolean)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.ArrayLogicBetween\\_v2](https://cables.gl/op/Ops.Array.ArrayLogicBetween_v2)

## 14.1.54 ArrayLookup



**Full Name:** Ops.Array.ArrayLookup

**Description:** Create an array that is filled with values looked up by index from another array

**> Input Ports:**

- **Indices** (Array)
- **Values** (Array)
- **Stride** (Number: Integer)

#### < Output Ports:

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayLookup>

## 14.1.55 ArrayMath



**Full Name:** Ops.Array.ArrayMath

**Description:** Pick from multiple mathematical modes which can all be applied to a single array

#### > Input Ports:

- **Array 0** (Array)
- **Number For Math** (Number)
- **Math Function Index** (Number: Integer)

#### < Output Ports:

- **Array Result** (Array)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayMath>

## 14.1.56 ArrayMathArray



**Full Name:** Ops.Array.ArrayMathArray

**Description:** Perform a math operations on two arrays

#### > Input Ports:

- **Array 0** (Array)
- **Array 1** (Array)
- **Math Function Index** (Number: Integer)

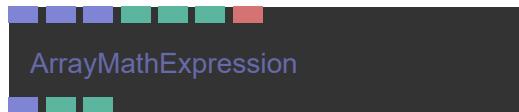
#### < Output Ports:

- **Array Result** (Array)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayMathArray>

## 14.1.57 ArrayMathExpression



**Full Name:** Ops.Array.ArrayMathExpression

**Description:** Calculate a user-defined mathematical expression

#### > Input Ports:

- **A** (Array)
- **B** (Array)
- **C** (Array)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Expression** (String)

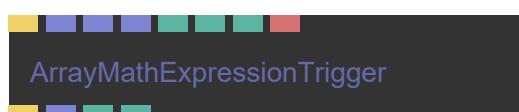
#### < Output Ports:

- **Result Array** (Array)
- **Array Length** (Number)
- **Expression Valid** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayMathExpression>

## 14.1.58 ArrayMathExpressionTrigger



**Full Name:** Ops.Array.ArrayMathExpressionTrigger

**Full Name:** Ops.Array.ArrayMathExpressionTrigger

**Description:** Calculate a user-defined mathematical expression

**> Input Ports:**

- **Update** (Trigger)
- **A** (Array)
- **B** (Array)
- **C** (Array)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Expression** (String)

**< Output Ports:**

- **Next** (Trigger)
- **Result Array** (Array)
- **Array Length** (Number)
- **Expression Valid** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayMathExpressionTrigger>

## 14.1.59 ArrayMax



**Full Name:** Ops.Array.ArrayMax

**Description:** Apply a max operation to all values in an array

**> Input Ports:**

- **Array In** (Array)
- **Value** (Number)

**< Output Ports:**

- **Array Out** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayMax>

## 14.1.60 ArrayMerge\_v3



**Full Name:** Ops.Array.ArrayMerge\_v3

**Description:** Merge multiple arrays - in consecutive order

**> Input Ports:**

- **Array 0** (Array)
- **Array 1** (Array)
- **Array 2** (Array)
- **Array 3** (Array)
- **Array 4** (Array)
- **Array 5** (Array)
- **Array 6** (Array)
- **Array 7** (Array)

**< Output Ports:**

- **Result** (Array)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.ArrayMerge\\_v3](https://cables.gl/op/Ops.Array.ArrayMerge_v3)

## 14.1.61 ArrayMergeTrigger



**Full Name:** Ops.Array.ArrayMergeTrigger

**Description:** Merge / concatenate arrays by trigger

**> Input Ports:**

- **Merge** (Trigger)
- **Array 0** (Array)
- **Array 1** (Array)
- **Array 2** (Array)
- **Array 3** (Array)

- **Array 4** (Array)
- **Array 5** (Array)
- **Array 6** (Array)
- **Array 7** (Array)
- **Array 8** (Array)
- **Array 9** (Array)
- **Array 10** (Array)
- **Array 11** (Array)

**< Output Ports:**

- **Next** (Trigger)
- **Result** (Array)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayMergeTrigger>

## 14.1.62 ArrayMin



**Full Name:** Ops.Array.ArrayMin

**Description:** Apply a min operation to all values in an array

**> Input Ports:**

- **Array In** (Array)
- **Value** (Number)

**< Output Ports:**

- **Array Out** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayMin>

## 14.1.63 ArrayModulo



**Full Name:** Ops.Array.ArrayModulo

**Description:** Apply a modulo operation to all values in an array

**> Input Ports:**

- **Array In** (Array)
- **Value** (Number)

**< Output Ports:**

- **Array Out** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayModulo>

## 14.1.64 ArrayMultiply



**Full Name:** Ops.Array.ArrayMultiply

**Description:** Multiply every number in an array

**> Input Ports:**

- **In** (Array)
- **Value** (Number)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayMultiply>

## 14.1.65 ArrayNumberRamp\_v2



**Full Name:** Ops.Array.ArrayNumberRamp\_v2

**Description:** Create an array that contains X numbers between start and end values

> **Input Ports:**

- **Start Value** (Number)
- **End Value** (Number)
- **Entries** (Number: Integer)

< **Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.ArrayNumberRamp\\_v2](https://cables.gl/op/Ops.Array.ArrayNumberRamp_v2)

## 14.1.66 ArrayOfArrays



**Full Name:** Ops.Array.ArrayOfArrays

**Description:** Create an array filled with other arrays

> **Input Ports:**

- **Update** (Trigger)
- **Array 0** (Array)
- **Array 1** (Array)
- **Array 2** (Array)
- **Array 3** (Array)
- **Array 4** (Array)
- **Array 5** (Array)
- **Array 6** (Array)
- **Array 7** (Array)
- **Array 8** (Array)
- **Array 9** (Array)

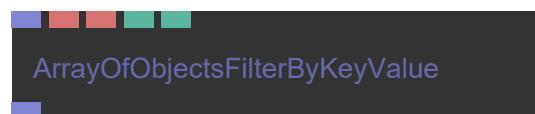
< **Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayOfArrays>

## 14.1.67 ArrayOfObjectsFilterByKeyValue\_v3



ArrayOfObjectsFilterByKeyValue

**Full Name:** Ops.Array.ArrayOfObjectsFilterByKeyValue\_v3

**Description:** Filter key-value pairs in objects in an array of objects

> **Input Ports:**

- **Array** (Array)
- **Filter Key** (String)
- **Filter Value** (String)
- **Invert Filter** (Number: Boolean)
- **invert result** (discard all objects that have key-value pair)

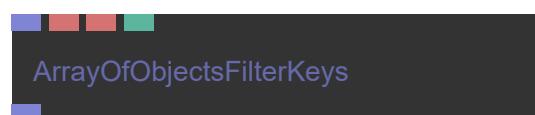
< **Output Ports:**

- **ArrayOut** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.ArrayOfObjectsFilterByKeyValue\\_v3](https://cables.gl/op/Ops.Array.ArrayOfObjectsFilterByKeyValue_v3)

## 14.1.68 ArrayOfObjectsFilterKeys



ArrayOfObjectsFilterKeys

**Full Name:** Ops.Array.ArrayOfObjectsFilterKeys

**Description:** Remove key-value pairs from objects in an array of objects

> **Input Ports:**

- **Array** (Array)
- **Keys** (String)
- **Separator** (String)
- **Invert Filter** (Number: Boolean)

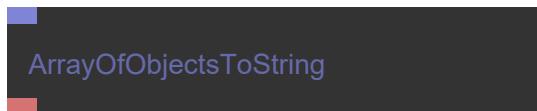
< **Output Ports:**

- **Array Out** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayOfObjectsFilterKeys>

## 14.1.69 ArrayOfObjectsToString



**Full Name:** Ops.Array.ArrayOfObjectsToString

**Description:** Convert an array of objects into readable string format

**> Input Ports:**

- **Array In** (Array)

**< Output Ports:**

- **String** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayOfObjectsToString>

## 14.1.70 ArrayPack



**Full Name:** Ops.Array.ArrayPack

**Description:** Pack multiple arrays into a new array

**> Input Ports:**

- **Array 0** (Array)
- **Array 1** (Array)
- **Array 2** (Array)
- **Array 3** (Array)
- **Array 4** (Array)
- **Array 5** (Array)
- **Array 6** (Array)
- **Array 7** (Array)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayPack>

## 14.1.71 ArrayPack2



**Full Name:** Ops.Array.ArrayPack2

**Description:** Pack two individual arrays into a new array

**> Input Ports:**

- **Trigger In** (Trigger)
- **Array 1** (Array)
- **Array 2** (Array)

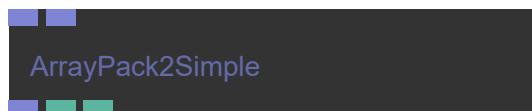
**< Output Ports:**

- **Trigger Out** (Trigger)
- **Array Out** (Array)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayPack2>

## 14.1.72 ArrayPack2Simple



**Full Name:** Ops.Array.ArrayPack2Simple

**Description:** Pack 2 individual arrays into an array2 - without needing a trigger

**> Input Ports:**

- **Array 1** (Array)
- **Array 2** (Array)

**< Output Ports:**

- **Array Out** (Array)
- **Num Points** (Number)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayPack2Simple>

### 14.1.73 ArrayPack3



**Full Name:** Ops.Array.ArrayPack3

**Description:** Pack 3 individual arrays into a xyz array

#### > Input Ports:

- **Trigger In** (Trigger)
- **Array 1** (Array)
- **Array 2** (Array)
- **Array 3** (Array)

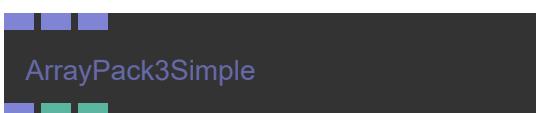
#### < Output Ports:

- **Trigger Out** (Trigger)
- **Array Out** (Array)
- **Num Points** (Number)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayPack3>

### 14.1.74 ArrayPack3Simple



**Full Name:** Ops.Array.ArrayPack3Simple

**Description:** Pack 3 individual arrays into an array3 - without needing a trigger

#### > Input Ports:

- **Array 1** (Array)
- **Array 2** (Array)
- **Array 3** (Array)

#### < Output Ports:

- **Array Out** (Array)
- **Num Points** (Number)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayPack3Simple>

### 14.1.75 ArrayPack4



**Full Name:** Ops.Array.ArrayPack4

**Description:** Pack 4 arrays into one array

#### > Input Ports:

- **Trigger In** (Trigger)
- **Array 1** (Array)
- **Array 2** (Array)
- **Array 3** (Array)
- **Array 4** (Array)

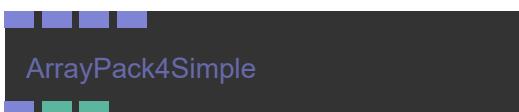
#### < Output Ports:

- **Trigger Out** (Trigger)
- **Array Out** (Array)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayPack4>

### 14.1.76 ArrayPack4Simple



**Full Name:** Ops.Array.ArrayPack4Simple

**Description:** Pack 3 individual arrays into an array3 - without needing a trigger

**> Input Ports:**

- **Array 1** (Array)
- **Array 2** (Array)
- **Array 3** (Array)
- **Array 4** (Array)

**< Output Ports:**

- **Array Out** (Array)
- **Num Points** (Number)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayPack4Simple>

## 14.1.77 ArrayPow



**Full Name:** Ops.Array.ArrayPow

**Description:** Values below 0 are not accepted. 1 = Array in is unaltered

**> Input Ports:**

- **Array In** (Array)
- **Pow Factor** (Number)

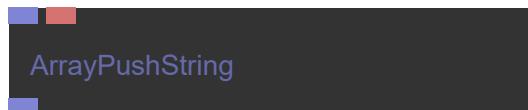
**< Output Ports:**

- **Array Out** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayPow>

## 14.1.78 ArrayPushString



**Full Name:** Ops.Array.ArrayPushString

**Description:** Push/Append a string to the end of an array

**> Input Ports:**

- **Array** (Array)
- **String** (String)

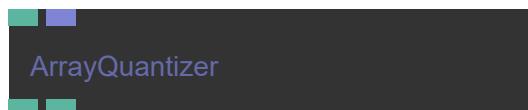
**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayPushString>

## 14.1.79 ArrayQuantizer



**Full Name:** Ops.Array.ArrayQuantizer

**Description:** Quantize input to nearest number in array

**> Input Ports:**

- **Value** (Number)
- **Constraints Array Input** (Array)

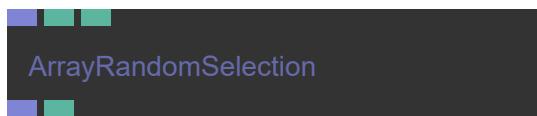
**< Output Ports:**

- **Quantized Value** (Number)
- **Quantization Error** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayQuantizer>

## 14.1.80 ArrayRandomSelection



**Full Name:** Ops.Array.ArrayRandomSelection

**Description:** Extract a definable amount of values from an array

**> Input Ports:**

- **Array** (Array)
- **Elements** (Number: Integer)
- **Seed** (Number)

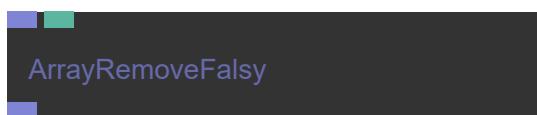
**< Output Ports:**

- **Result** (Array)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayRandomSelection>

## 14.1.81 ArrayRemoveFalsy



**Full Name:** Ops.Array.ArrayRemoveFalsy

**Description:** Remove falsy items from an array

**> Input Ports:**

- **Array** (Array)
- **Remove Falsy** (Number: Boolean)

**< Output Ports:**

- **Result Array** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayRemoveFalsy>

## 14.1.82 ArrayReverse



**Full Name:** Ops.Array.ArrayReverse

**Description:** Reverse an array

**> Input Ports:**

- **Active** (Number: Boolean)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayReverse>

## 14.1.83 ArrayRound



**Full Name:** Ops.Array.ArrayRound

**Description:** Round numbers up

**> Input Ports:**

- **In** (Array)
- **Method Index** (Number: Integer)
- **Decimal Places** (Number: Integer)

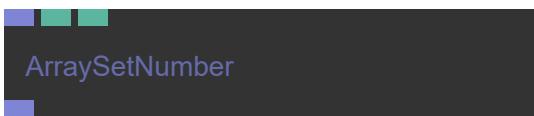
**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayRound>

## 14.1.84 ArraySetNumber\_v3



**Full Name:** Ops.Array.ArraySetNumber\_v3

**Description:** Set a number at index in an array

**> Input Ports:**

- **Array** (Array)
- **Index** (Number: Integer)
- **Number** (Number)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.ArraySetNumber\\_v3](https://cables.gl/op/Ops.Array.ArraySetNumber_v3)

## 14.1.85 ArraySetString



**Full Name:** Ops.Array.ArraySetString

**Description:** Set a string at index in an array

**> Input Ports:**

- **Array** (Array)
- **Index** (Number: Integer)
- **Value** (String)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArraySetString>

## 14.1.86 ArraySin



**Full Name:** Ops.Array.ArraySin

**Description:** Perform a sin or cos operation on the contents of an array

**> Input Ports:**

- **Array In** (Array)
- **Math Function Index** (Number: Integer)
- **Phase** (Number)
- **Frequency** (Number)
- **Amplitude** (Number)

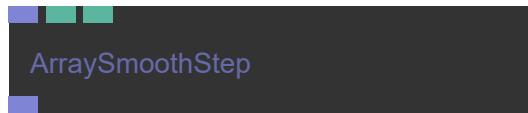
**< Output Ports:**

- **Array Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArraySin>

## 14.1.87 ArraySmoothStep



**Full Name:** Ops.Array.ArraySmoothStep

**Description:** The fancy way of saying it is Perform Hermite interpolation between two values

**> Input Ports:**

- **Array In** (Array)
- **Min** (Number)
- **Max** (Number)

**< Output Ports:**

- **Array Out** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArraySmoothStep>

## 14.1.88 ArraySqrt



**Full Name:** Ops.Array.ArraySqrt

**Description:** Return the square root of all values in the array

**> Input Ports:**

- In (Array)

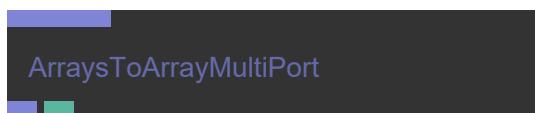
**< Output Ports:**

- Result (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArraySqrt>

## 14.1.89 ArraysToArrayMultiPort



**Full Name:** Ops.Array.ArraysToArrayMultiPort

**Description:** Create an array from multiple string

**> Input Ports:**

- Arrays\_0 (Array)
- Arrays\_1 (Array)
- Add Port (Array)

**< Output Ports:**

- Result (Array)
- Num Values (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArraysToArrayMultiPort>

## 14.1.90 ArraySubtract



**Full Name:** Ops.Array.ArraySubtract

**Description:** Subtract one number from all values in an array

**> Input Ports:**

- Array In (Array)
- Value (Number)

**< Output Ports:**

- Array Out (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArraySubtract>

## 14.1.91 ArraySum



**Full Name:** Ops.Array.ArraySum

**Description:** Add one number to all values in an array

**> Input Ports:**

- In (Array)
- Value (Number)

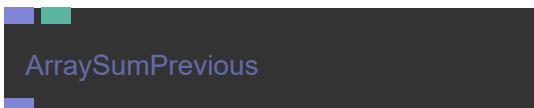
**< Output Ports:**

- Result (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArraySum>

## 14.1.92 ArraySumPrevious



**Full Name:** Ops.Array.ArraySumPrevious

**Description:** Sum up every number in an array with the sum of the previous

**> Input Ports:**

- **Array** (Array)
- **Padding** (Number)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArraySumPrevious>

## 14.1.93 ArraySumUp



**Full Name:** Ops.Array.ArraySumUp

**Description:** Sum of every number in an array

**> Input Ports:**

- **Array** (Array)

**< Output Ports:**

- **Sum** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArraySumUp>

## 14.1.94 ArraySwizzle



ArraySwizzle

**Full Name:** Ops.Array.ArraySwizzle

**Description:** Manage/re-order components of an array (stride)

**> Input Ports:**

- **Array** (Array)

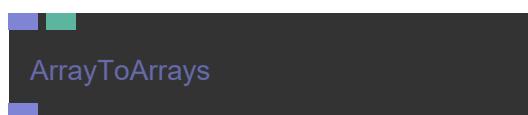
**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArraySwizzle>

## 14.1.95 ArrayToArrays



**Full Name:** Ops.Array.ArrayToArrays

**Description:** Split an array up into an array of arrays

**> Input Ports:**

- **Array** (Array)
- **Stride** (Number: Integer)

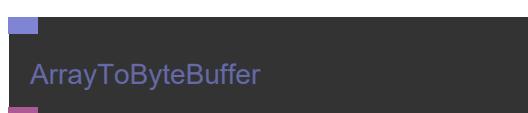
**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayToArrays>

## 14.1.96 ArrayToByteBuffer



**Full Name:** Ops.Array.ArrayToByteBuffer

**Description:** Convert an array to a byte buffer (Uint8ClampedArray)

**> Input Ports:**

- **Array** (Array)

#### < Output Ports:

- **Buffer** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayToByteBuffer>

### 14.1.97 ArrayToString\_v3



**Full Name:** Ops.Array.ArrayToString\_v3

**Description:** Join array values to a string (concat)

#### > Input Ports:

- **Array** (Array)
- **Seperator** (String)
- **New Line** (Number: Boolean)

#### < Output Ports:

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.ArrayToString\\_v3](https://cables.gl/op/Ops.Array.ArrayToString_v3)

### 14.1.98 ArrayTrigger



**Full Name:** Ops.Array.ArrayTrigger

**Description:** Trigger an array

#### > Input Ports:

- **Exec** (Trigger)
- **Array** (Array)

#### < Output Ports:

- **Trigger Out** (Trigger)

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayTrigger>

### 14.1.99 ArrayUnique



**Full Name:** Ops.Array.ArrayUnique

**Description:** Filter an array for duplicate items and returns all unique items in a new array

#### > Input Ports:

- **Array** (Array)
- **Format Index** (Number: Integer)
- **Format** (String)

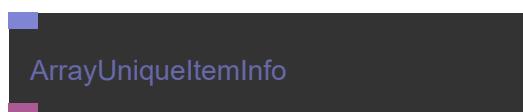
#### < Output Ports:

- **ArrayOut** (Array)
- **Array Length Out** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayUnique>

### 14.1.100 ArrayUniqueItemInfo



**Full Name:** Ops.Array.ArrayUniqueItemInfo

**Description:** Return information about the count of “duplicates” in an array, as an object

#### > Input Ports:

- **Array** (Array)

#### < Output Ports:

- **ObjectOut** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayUniqueItemInfo>

### 14.1.101 ArrayUnpack2



**Full Name:** Ops.Array.ArrayUnpack2

**Description:** Unpack an xy array into separate arrays

**> Input Ports:**

- **Array In Xyz** (Array)

**< Output Ports:**

- **Array 1 Out** (Array)
- **Array 2 Out** (Array)
- **Array Lengths** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayUnpack2>

### 14.1.102 ArrayUnpack3



**Full Name:** Ops.Array.ArrayUnpack3

**Description:** Split an xyz array into 3 individual arrays

**> Input Ports:**

- **Array In Xyz** (Array)

**< Output Ports:**

- **Array 1 Out** (Array)
- **Array 2 Out** (Array)
- **Array 3 Out** (Array)
- **Array Lengths** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayUnpack3>

### 14.1.103 ArrayUnpack4



**Full Name:** Ops.Array.ArrayUnpack4

**Description:** Split an xyzw array into 4 individual arrays

**> Input Ports:**

- **Array In Xyzw** (Array)

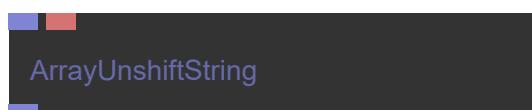
**< Output Ports:**

- **Array 1 Out** (Array)
- **Array 2 Out** (Array)
- **Array 3 Out** (Array)
- **Array 4 Out** (Array)
- **Array Lengths** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayUnpack4>

### 14.1.104 ArrayUnshiftString



**Full Name:** Ops.Array.ArrayUnshiftString

**Description:** Insert/add/unshift a string to the beginning of an array

**> Input Ports:**

- **Array** (Array)
- **String** (String)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ArrayUnshiftString>

## 14.1.105 AverageArray



**Full Name:** Ops.Array.AverageArray

**Description:** Smooth/average values in an array

**> Input Ports:**

- **Array** (Array)
- **Iterations** (Number: Integer)
- **Mode Index** (Number: Integer)

**< Output Ports:**

- **Smoothed Array** (Array)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.AverageArray>

## 14.1.106 BoolStateArray



**Full Name:** Ops.Array.BoolStateArray

**Description:** Array filled with 0, only one can be 1

**> Input Ports:**

- **Array Length** (Number)
- **Active Index** (Number)
- **Inactive Value** (Number)
- **Active Value** (Number)

**< Output Ports:**

- **State Array** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.BoolStateArray>

## 14.1.107 CopyArray



**Full Name:** Ops.Array.CopyArray

**Description:** Copy an array with a trigger, reset to use a default array

**> Input Ports:**

- **Exec** (Trigger)
- **Array** (Array)
- **Reset** (Trigger)
- **Default** (Array)

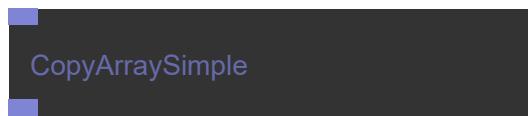
**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.CopyArray>

## 14.1.108 CopyArraySimple



**Full Name:** Ops.Array.CopyArraySimple

**Description:** Create a copy of an array

**> Input Ports:**

- Visit *Ops.Array.CopyArraySimple documentation* for input port details

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.CopyArraySimple>

## 14.1.109 CropArray



**Full Name:** Ops.Array.CropArray

**Description:** The array to crop

**> Input Ports:**

- **Source Array** (Array)
- **Start Index** (Number: Integer)
- **New Length** (Number: Integer)

**< Output Ports:**

- **Cropped Array** (Array)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.CropArray>

## 14.1.110 CutArray



**Full Name:** Ops.Array.CutArray

**Description:** Remove elements from an array from the beginning and/or the end

**> Input Ports:**

- **Source Array** (Array)
- **Remove From Start** (Number: Integer)
- **Remove From End** (Number: Integer)

**< Output Ports:**

- **Cut Array** (Array)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.CutArray>

## 14.1.111 EaseArray



**Full Name:** Ops.Array.EaseArray

**Description:** Apply easing curve to numbers in an array

**> Input Ports:**

- **Array** (Array)
- **Min** (Number)
- **Max** (Number)
- **Easing Index** (Number: Integer)

**< Output Ports:**

- **Result Array** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.EaseArray>

## 14.1.112 EmptyArray



**Full Name:** Ops.Array.EmptyArray

**Description:** Visit documentation for details

**> Input Ports:**

- Visit *Ops.Array.EmptyArray* documentation for input port details

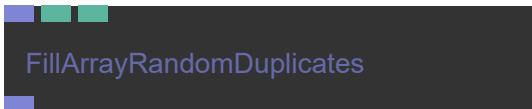
**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.EmptyArray>

## 14.1.113 FillArrayRandomDuplicates\_v2



**Full Name:** Ops.Array.FillArrayRandomDuplicates\_v2

**Description:** Fill an array with random duplicates

**> Input Ports:**

- **Array** (Array)
- **Num Elements** (Number: Integer)
- **Random Seed** (Number)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.FillArrayRandomDuplicates\\_v2](https://cables.gl/op/Ops.Array.FillArrayRandomDuplicates_v2)

## 14.1.114 FilterArray



**Full Name:** Ops.Array.FilterArray

**Description:** Compare elements from an array and remove not matching ones

**> Input Ports:**

- **Array** (Array)
- **Stride Index** (Number: Integer)
- **the type of the array** (Array3, Array2, ...)
- **Compare Element Index** (Number: Integer)
- **which element to compare** (see stride)
- **Filter Method Index** (Number: Integer)
- **Compare To** (Number)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.FilterArray>

## 14.1.115 FilterValidArray



**Full Name:** Ops.Array.FilterValidArray

**Description:** Filter valid arrays

**> Input Ports:**

- **Array** (Array)
- **Invalid When Length Is 0** (Number: Boolean)

**< Output Ports:**

- **Last Valid Array** (Array)
- **Is Valid** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.FilterValidArray>

## 14.1.116 FlattenArray



**Full Name:** Ops.Array.FlattenArray

**Description:** Create a new array with all sub-array elements concatenated into it

**> Input Ports:**

- **Array** (Array)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.FlattenArray>

## 14.1.117 FreezeArray



**Full Name:** Ops.Array.FreezeArray

**Description:** Capture the current input and copy it to the output, even after a reload

**> Input Ports:**

- **Number** (Array)
- **Button** (Trigger)

**< Output Ports:**

- **Frozen Array** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.FreezeArray>

## 14.1.118 GateArray\_v2



**Full Name:** Ops.Array.GateArray\_v2

**Description:** Only allow an array through if pass through is true

**> Input Ports:**

- **Array In** (Array)
- **Pass Through** (Number: Boolean)

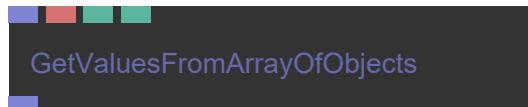
**< Output Ports:**

- **Array Out** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.GateArray\\_v2](https://cables.gl/op/Ops.Array.GateArray_v2)

## 14.1.119 GetValuesFromArrayOfObjects



**Full Name:** Ops.Array.GetValuesFromArrayOfObjects

**Description:** Get an array of values by key of objects in an array

**> Input Ports:**

- **Array** (Array)
- **Key** (String)
- **Numbers Only** (Number: Boolean)

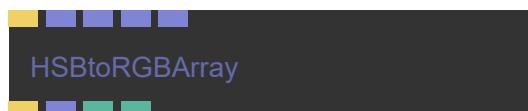
**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.GetValuesFromArrayOfObjects>

## 14.1.120 HSBTorGBAarray



**Full Name:** Ops.Array.HSBToRGBArray

**Description:** Generate an RGBA array from up to 4 arrays (HSBA)

**> Input Ports:**

- **Trigger Input** (Trigger)
- **In Hue Array** (Array)
- **In Saturation Array** (Array)
- **In Brightness Array** (Array)
- **In Alpha Array** (Array)

**< Output Ports:**

- **Trigger Output** (Trigger)
- **Result Array** (Array)
- **Array Length** (Number)
- **RGBA Tuple Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.HSBtoRGBArray>

### 14.1.121 InfoArray



**Full Name:** Ops.Array.InfoArray

**Description:** Min, Max and Average value from an array

**> Input Ports:**

- **Array** (Array)

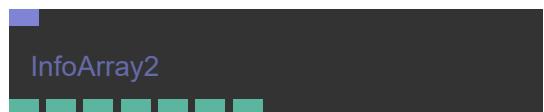
**< Output Ports:**

- **Min** (Number)
- **Max** (Number)
- **Average** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.InfoArray>

### 14.1.122 InfoArray2



**Full Name:** Ops.Array.InfoArray2

**Description:** Min, Max and Average values of an array2

**> Input Ports:**

- **Array** (Array)

**< Output Ports:**

- **Num Items** (Number)
- **Min X** (Number)
- **Max X** (Number)
- **Average X** (Number)
- **Min Y** (Number)
- **Max Y** (Number)

- **Average Y** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.InfoArray2>

### 14.1.123 InfoArray3



**Full Name:** Ops.Array.InfoArray3

**Description:** Min, Max and Average values of an array3

**> Input Ports:**

- **Array** (Array)

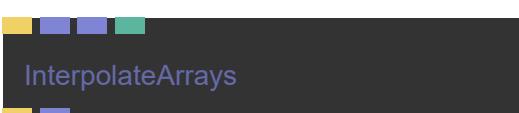
**< Output Ports:**

- **Num Items** (Number)
- **Min X** (Number)
- **Min Y** (Number)
- **Min Z** (Number)
- **Max X** (Number)
- **Max Y** (Number)
- **Max Z** (Number)
- **Average X** (Number)
- **Average Y** (Number)
- **Average Z** (Number)
- **Center X** (Number)
- **Center Y** (Number)
- **Center Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.InfoArray3>

### 14.1.124 InterpolateArrays



**Full Name:** Ops.Array.InterpolateArrays

**Description:** Interpolate between two arrays (lerp) - linear interpolation

**> Input Ports:**

- **Exe** (Trigger)
- **Array 1** (Array)
- **Array 2** (Array)
- **Perc** (Number)

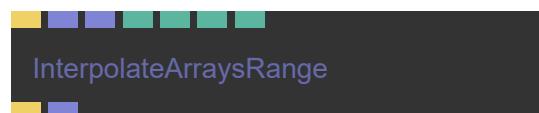
**< Output Ports:**

- **Next** (Trigger)
- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.InterpolateArrays>

## 14.1.125 InterpolateArraysRange



**Full Name:** Ops.Array.InterpolateArraysRange

**Description:** Interpolate between two arrays, only a few numbers at the same time

**> Input Ports:**

- **Exe** (Trigger)
- **Array 1** (Array)
- **Array 2** (Array)
- **Pos** (Number)
- **Width** (Number)
- **Easing Index** (Number: Integer)
- **Reverse** (Number: Boolean)

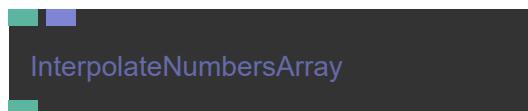
**< Output Ports:**

- **Next** (Trigger)
- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.InterpolateArraysRange>

## 14.1.126 InterpolateNumbersArray



**Full Name:** Ops.Array.InterpolateNumbersArray

**Description:** Interpolate between all values of an array

**> Input Ports:**

- **Index Position** (Number)
- **Array** (Array)

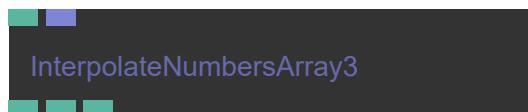
**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.InterpolateNumbersArray>

## 14.1.127 InterpolateNumbersArray3



**Full Name:** Ops.Array.InterpolateNumbersArray3

**Description:** Get interpolated values between the indices of an array3x

**> Input Ports:**

- **Index Position** (Number)
- **Array** (Array)

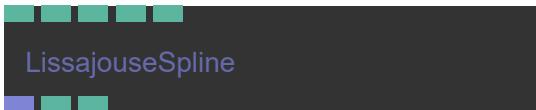
**< Output Ports:**

- **X** (Number)
- **Y** (Number)
- **Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.InterpolateNumbersArray3>

## 14.1.128 LissajouseSpline



**Full Name:** Ops.Array.LissajouseSpline

**Description:** Generate spline using lissajous formulas

**> Input Ports:**

- **Formula Index** (Number: Integer)
- **A** (Number: Integer)
- **B** (Number: Integer)
- **C** (Number: Integer)
- **D** (Number: Integer)

**< Output Ports:**

- **Result** (Array)
- **Total Points** (Number)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.LissajouseSpline>

## 14.1.129 LoopArray3



**Full Name:** Ops.Array.LoopArray3

**Description:** Make the 1st and last point of an array the same, good for closing splines and shapes

**> Input Ports:**

- **Array In** (Array)

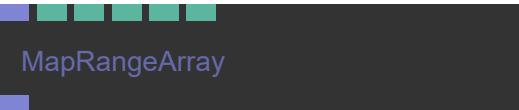
**< Output Ports:**

- **Array Out** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.LoopArray3>

## 14.1.130 MapRangeArray



**Full Name:** Ops.Array.MapRangeArray

**Description:** Map values in an array from one range into another.

**> Input Ports:**

- **Array** (Array)
- **Old Min** (Number)
- **Old Max** (Number)
- **New Min** (Number)
- **New Max** (Number)
- **Easing Index** (Number: Integer)

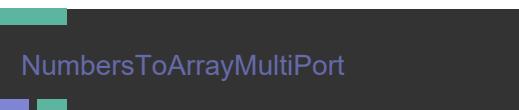
**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.MapRangeArray>

## 14.1.131 NumbersToArrayMultiPort\_v2



**Full Name:** Ops.Array.NumbersToArrayMultiPort\_v2

**Description:** Create an array from multiple number inputs

**> Input Ports:**

- **Numbers\_0** (Number)
- **Add Port** (Number)

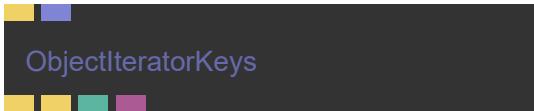
**< Output Ports:**

- **Result** (Array)
- **Num Values** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.NumbersToArrayMultiPort\\_v2](https://cables.gl/op/Ops.Array.NumbersToArrayMultiPort_v2)

## 14.1.132 ObjectIteratorKeys



**Full Name:** Ops.Array.ObjectIteratorKeys

**Description:** Iterate over an array of objects

**> Input Ports:**

- **Exe** (Trigger)
- **Array** (Array)

**< Output Ports:**

- **Trigger** (Trigger)
- **Finished** (Trigger)
- **Index** (Number)
- **Value** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ObjectIteratorKeys>

## 14.1.133 PaletteLibrary



**Full Name:** Ops.Array.PaletteLibrary

**Description:** Contains a collection of color palettes in groups of 5 in an array

**> Input Ports:**

- Visit *Ops.Array.PaletteLibrary documentation for input port details*

**< Output Ports:**

- **Palette Array Out** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.PaletteLibrary>

## 14.1.134 PerlinArray



**Full Name:** Ops.Array.PerlinArray

**Description:** Create an array filled with Perlin noise values

**> Input Ports:**

- **Array In X** (Array)
- **Array Time** (Array)
- **Time In Y** (Number)
- **Seed 0-1** (Number)
- **Frequency** (Number)

**< Output Ports:**

- **Array Out** (Array)
- **Array Length Out** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.PerlinArray>

## 14.1.135 Phyllotaxis



**Full Name:** Ops.Array.Phyllotaxis

**Description:** Coordinate generation like arrangement of leaves in some plants

**> Input Ports:**

- **Render** (Trigger)
- **Num** (Number: Integer)
- **Scale** (Number)
- **Param** (Number)

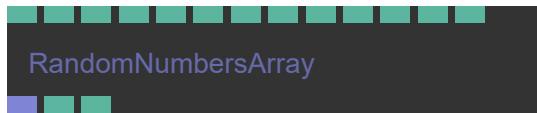
**< Output Ports:**

- **Coordinates** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.Phyllotaxis>

### 14.1.136 RandomNumbersArray\_v4



**Full Name:** Ops.Array.RandomNumbersArray\_v4

**Description:** Create a random array of 1 to 4 dimensions

#### > Input Ports:

- **Num Values** (Number: Integer)
- **Mode Index** (Number: Integer)
- **Random Seed** (Number)
- **Integer** (Number: Boolean)
- **Min A** (Number)
- **Max A** (Number)
- **Min B** (Number)
- **Max B** (Number)
- **Min C** (Number)
- **Max C** (Number)
- **Min D** (Number)
- **Max D** (Number)

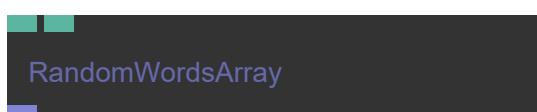
#### < Output Ports:

- **Array Out** (Array)
- **Chunks Amount** (Number)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.RandomNumbersArray\\_v4](https://cables.gl/op/Ops.Array.RandomNumbersArray_v4)

### 14.1.137 RandomWordsArray



RandomWordsArray

**Full Name:** Ops.Array.RandomWordsArray

**Description:** Generate an array filled with random english words

#### > Input Ports:

- **Random Seed** (Number)
- **Content Index** (Number: Integer)

#### < Output Ports:

- **Words** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.RandomWordsArray>

### 14.1.138 ReduceArray3\_v3



**Full Name:** Ops.Array.ReduceArray3\_v3

**Description:** Remove points from an array, e.g. xth points, random, duplicates

#### > Input Ports:

- **Array** (Array)
- **Remove Index** (Number: Integer)
- **Every Xth Item** (Number: Integer)
- **Threshold** (Number)
- **Seed** (Number)

#### < Output Ports:

- **Result Array** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.ReduceArray3\\_v3](https://cables.gl/op/Ops.Array.ReduceArray3_v3)

### 14.1.139 ReverseArray3



ReverseArray3

**Full Name:** Ops.Array.ReverseArray3

**Description:** Reverse an array with value triplets [x, y, z, ...]

**> Input Ports:**

- **Array (Array)**
- **The Array you want to reverse** (containing triplets)

**< Output Ports:**

- **Reversed Array (Array)**

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.ReverseArray3>

## 14.1.140 RingBuffer



**Full Name:** Ops.Array.RingBuffer

**Description:** Array of fixed size, index is automatically incremented and restarts after reaching the end

**> Input Ports:**

- **Value (Number)**
- **Write (Trigger)**
- **Length (Number: Integer)**
- **Reset Index (Trigger)**

**< Output Ports:**

- **Result (Array)**
- **Index (Number)**

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.RingBuffer>

## 14.1.141 RotateArray



**Full Name:** Ops.Array.RotateArray

**Description:** Shift array contents based upon rotate amount

**> Input Ports:**

- **Array In (Array)**
- **Rotate Amount (Number: Integer)**

**< Output Ports:**

- **ArrayOut (Array)**

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.RotateArray>

## 14.1.142 RouteArray



**Full Name:** Ops.Array.RouteArray

**Description:** Route an array to an output port

**> Input Ports:**

- **Index (Number: Integer)**
- **Array In (Array)**
- **Default Array (Array)**

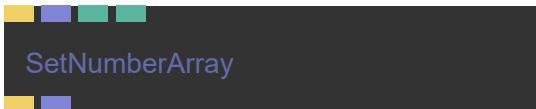
**< Output Ports:**

- **Index 0 Array (Array)**
- **Index 1 Array (Array)**
- **Index 2 Array (Array)**
- **Index 3 Array (Array)**
- **Index 4 Array (Array)**
- **Index 5 Array (Array)**
- **Index 6 Array (Array)**
- **Index 7 Array (Array)**
- **Index 8 Array (Array)**
- **Index 9 Array (Array)**

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.RouteArray>

## 14.1.143 SetNumberArray



**Full Name:** Ops.Array.SetNumberArray

**Description:** Change the number of an array at an index

**> Input Ports:**

- **Exe** (Trigger)
- **Array** (Array)
- **Index** (Number: Integer)
- **Value** (Number)

**< Output Ports:**

- **Next** (Trigger)
- **Values** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.SetNumberArray>

**Docs:** <https://cables.gl/op/Ops.Array.SetNumbersArray3>

## 14.1.145 ShuffleArray3\_v3



**Full Name:** Ops.Array.ShuffleArray3\_v3

**Description:** Shuffle/Randomize the order of an array of triplets

**> Input Ports:**

- **Array3** (Array)
- **Seed** (Number)

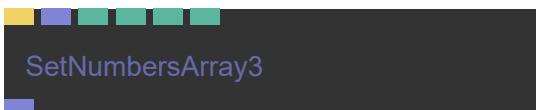
**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.ShuffleArray3\\_v3](https://cables.gl/op/Ops.Array.ShuffleArray3_v3)

## 14.1.144 SetNumbersArray3



**Full Name:** Ops.Array.SetNumbersArray3

**Description:** Set three values at position index in an array

**> Input Ports:**

- **Exe** (Trigger)
- **Array** (Array)
- **Index** (Number: Integer)
- **Value 1** (Number)
- **Value 2** (Number)
- **Value 3** (Number)

**< Output Ports:**

- **Values** (Array)

**Example Patch:** Open in Editor

## 14.1.146 ShuffleArray\_v3



**Full Name:** Ops.Array.ShuffleArray\_v3

**Description:** Randomize the order of elements inside an array

**> Input Ports:**

- **Array3** (Array)
- **Seed** (Number)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.ShuffleArray\\_v3](https://cables.gl/op/Ops.Array.ShuffleArray_v3)

## 14.1.147 SimplexArray



**Full Name:** Ops.Array.SimplexArray

**Description:** Create an array filled with Simplex noise values (Range: -1, 1)

**> Input Ports:**

- **Array In X** (Array)
- **Array Time** (Array)
- **Time In Y** (Number)
- **Seed 0-1** (Number)
- **Frequency** (Number)

**< Output Ports:**

- **Array Out** (Array)
- **Array Length Out** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.SimplexArray>

## 14.1.148 SmoothArray



**Full Name:** Ops.Array.SmoothArray

**Description:** Smooth out changes in values of an array

**> Input Ports:**

- **Execute** (Trigger)
- **Array In** (Array)
- **Inc Factor** (Number)
- **Dec Factor** (Number)

**< Output Ports:**

- **Next** (Trigger)
- **Array Out** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.SmoothArray>

## 14.1.149 SortArray



**Full Name:** Ops.Array.SortArray

**Description:** Sort an array of numbers with one of two modes - ascending or descending

**> Input Ports:**

- **Array To Sort** (Array)

**< Output Ports:**

- **Sorted Array** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.SortArray>

## 14.1.150 SortArray3



**Full Name:** Ops.Array.SortArray3

**Description:** Sort an array with the lowest values of the selected component.

**> Input Ports:**

- **Array** (Array)
- **What Index** (Number: Integer)

**< Output Ports:**

- **Result** (Array)
- **The sorted array** (new array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.SortArray3>

## 14.1.151 SortArrayOfObjects



**Full Name:** Ops.Array.SortArrayOfObjects

**Description:** Sort an array of objects by the values of a key

**> Input Ports:**

- **Array** (Array)
- **Key** (String)
- **Reverse** (Number: Boolean)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.SortArrayOfObjects>

## 14.1.152 SortArrayWithIndices\_v2



**Full Name:** Ops.Array.SortArrayWithIndices\_v2

**Description:** Sort an array of numbers and also get sorted indices

**> Input Ports:**

- **Array To Sort** (Array)
- **Sorting Mode Index** (Number: Integer)

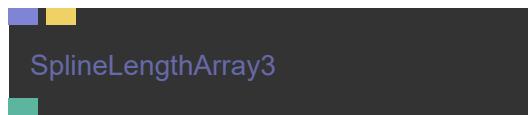
**< Output Ports:**

- **Sorted Array** (Array)
- **Sorted Indices** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.SortArrayWithIndices\\_v2](https://cables.gl/op/Ops.Array.SortArrayWithIndices_v2)

## 14.1.153 SplineLengthArray3



**Full Name:** Ops.Array.SplineLengthArray3

**Description:** Return a number with the total distance between the points/items in an array3

**> Input Ports:**

- **Array3x** (Array)
- **Calculate** (Trigger)

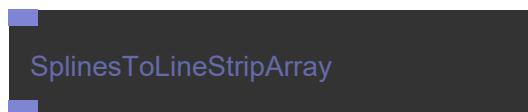
**< Output Ports:**

- **Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.SplineLengthArray3>

## 14.1.154 SplinesToLineStripArray



**Full Name:** Ops.Array.SplinesToLineStripArray

**Description:** Convert an array of splines to one “line striped” array

**> Input Ports:**

- **Array** (Array)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.SplinesToLineStripArray>

## 14.1.155 StringToArray\_v2



**Full Name:** Ops.Array.StringToArray\_v2

**Description:** Parse a string into an array (create, split string, stringToArray)

**> Input Ports:**

- **Text** (String)
- **Separator** (String)
- **Numbers** (Number: Boolean)
- **Trim** (Number: Boolean)
- **Split Lines** (Number: Boolean)

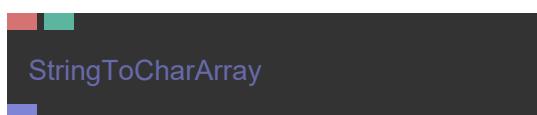
**< Output Ports:**

- **Array** (Array)
- **Parsed** (Trigger)
- **Length** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.StringToArray\\_v2](https://cables.gl/op/Ops.Array.StringToArray_v2)

## 14.1.156 StringToCharArray



**Full Name:** Ops.Array.StringToCharArray

**Description:** Turn a string into an array of single characters or ASCII numbers

**> Input Ports:**

- **String** (String)
- **Convert To Numbers** (Number: Boolean)

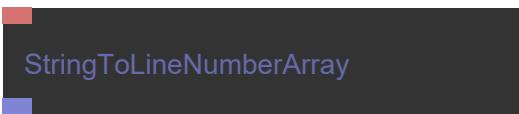
**< Output Ports:**

- **Array Out** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.StringToCharArray>

## 14.1.157 StringToLineNumberArray



**Full Name:** Ops.Array.StringToLineNumberArray

**Description:** Output an array containing a line number for every character

**> Input Ports:**

- **String** (String)

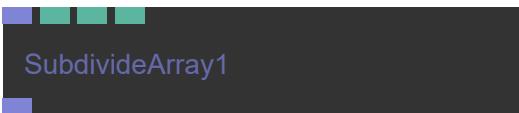
**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.StringToLineNumberArray>

## 14.1.158 SubdivideArray1



**Full Name:** Ops.Array.SubdivideArray1

**Description:** For subdividing splines, smoothing lines using cubic bezier interpolation

**> Input Ports:**

- **Points** (Array)
- **Num Subdivs** (Number: Integer)
- **Smooth** (Number: Boolean)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.SubdivideArray1>

## 14.1.159 SwitchArray



SwitchArray

**Full Name:** Ops.Array.SwitchArray

**Description:** Switch between multiple arrays

### > Input Ports:

- **Index** (Number: Integer)
- **Array 0** (Array)
- **Array 1** (Array)
- **Array 2** (Array)
- **Array 3** (Array)
- **Array 4** (Array)
- **Array 5** (Array)
- **Array 6** (Array)
- **Array 7** (Array)
- **Array 8** (Array)
- **Array 9** (Array)

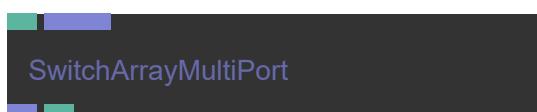
### < Output Ports:

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.SwitchArray>

## 14.1.160 SwitchArrayMultiPort\_v2



SwitchArrayMultiPort

**Full Name:** Ops.Array.SwitchArrayMultiPort\_v2

**Description:** Switch between multiple input arrays

### > Input Ports:

- **Index** (Number: Integer)
- **Arrays\_0** (Array)
- **Add Port** (Array)

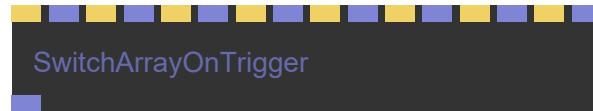
### < Output Ports:

- **Number** (Array)
- **Num Values** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.SwitchArrayMultiPort\\_v2](https://cables.gl/op/Ops.Array.SwitchArrayMultiPort_v2)

## 14.1.161 SwitchArrayOnTrigger



SwitchArrayOnTrigger

**Full Name:** Ops.Array.SwitchArrayOnTrigger

**Description:** Switch between multiple arrays on trigger

### > Input Ports:

- **Trigger 1** (Trigger)
- **Array 1** (Array)
- **Trigger 2** (Trigger)
- **Array 2** (Array)
- **Trigger 3** (Trigger)
- **Array 3** (Array)
- **Trigger 4** (Trigger)
- **Array 4** (Array)
- **Trigger 5** (Trigger)
- **Array 5** (Array)
- **Trigger 6** (Trigger)
- **Array 6** (Array)
- **Trigger 7** (Trigger)
- **Array 7** (Array)
- **Trigger 8** (Trigger)
- **Array 8** (Array)

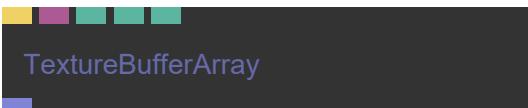
### < Output Ports:

- **Out Array** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.SwitchArrayOnTrigger>

## 14.1.162 TextureBufferArray



TextureBufferArray

**Full Name:** Ops.Array.TextureBufferArray

**Description:** Store various textures in an array, starts at the beginning again when end reached

### > Input Ports:

- **Write** (Trigger)
- **Texture** (Object:Texture)
- **Num** (Number: Integer)
- **The size of the ring buffer** (how many textures it can hold)
- **Order** (Number: Boolean)
- **Clear** (Number: Boolean)

### < Output Ports:

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.TextureBufferArray>

## 14.1.163 WeaveArrays



WeaveArrays

**Full Name:** Ops.Array.WeaveArrays

**Description:** Weave two arrays together (combine, join, merge)

### > Input Ports:

- **Array 1** (Array)
- **Array 2** (Array)
- **Chunk Size** (Number)

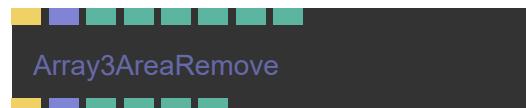
### < Output Ports:

- **Combined Array** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.WeaveArrays>

## 15 Ops.Array.PointArray



Array3AreaRemove

**Full Name:** Ops.Array.PointArray.Array3AreaRemove

**Description:** Remove points from an array3 with different shapes

### > Input Ports:

- **In Trigger** (Trigger)
- **In Array** (Array)
- **Mode Index** (Number: Integer)
- **Size** (Number)
- **Invert** (Number: Boolean)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)

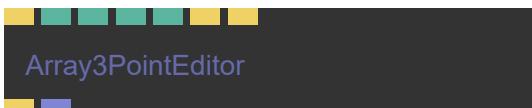
### < Output Ports:

- **Out Trigger** (Trigger)
- **Out Array** (Array)
- **Array Length** (Number)
- **Out X** (Number)
- **Out Y** (Number)
- **Out Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.PointArray.Array3AreaRemove>

## 15.1.2 Array3PointEditor



**Full Name:** Ops.Array.PointArray.Array3PointEditor

**Description:** Visually edit positions in an array of point coordinates

**> Input Ports:**

- **Execute** (Trigger)
- **Total Points** (Number: Integer)
- **Edit** (Number: Boolean)
- **Index** (Number: Integer)
- **Copy From Index** (Number: Integer)
- **Copy Coordinates** (Trigger)
- **Reset** (Trigger)

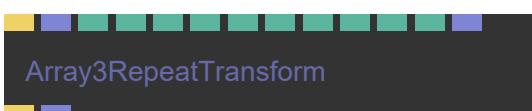
**< Output Ports:**

- **Next** (Trigger)
- **Coordinates** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.PointArray.Array3PointEditor>

## 15.1.3 Array3RepeatTransform



**Full Name:** Ops.Array.PointArray.Array3RepeatTransform

**Description:** Repeat an array by transforming it x times

**> Input Ports:**

- **Trigger** (Trigger)
- **Array** (Array)
- **Times** (Number: Integer)
- **Translate X** (Number)
- **Translate Y** (Number)
- **Translate Z** (Number)

- **Scale X** (Number)
- **Scale Y** (Number)
- **Scale Z** (Number)
- **Rotation X** (Number)
- **Rotation Y** (Number)
- **Rotation Z** (Number)
- **Position Array** (Array)

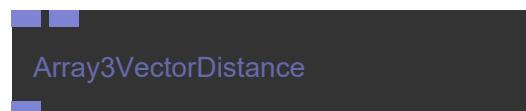
**< Output Ports:**

- **Next** (Trigger)
- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.PointArray.Array3RepeatTransform>

## 15.1.4 Array3VectorDistance



**Full Name:** Ops.Array.PointArray.Array3VectorDistance

**Description:** Return the distance between 2 points from an array

**> Input Ports:**

- **Array In 1** (Array)
- **Array In 2** (Array)

**< Output Ports:**

- **Array Out** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.PointArray.Array3VectorDistance>

## 15.1.5 ArraySpray



**Full Name:** Ops.Array.PointArray.ArraySpray

**Description:** Particle spray simulation

**> Input Ports:**

- **Exe** (Trigger)
- **Time** (Number)
- **Num** (Number)
- **Size X** (Number)
- **Size Y** (Number)
- **Size Z** (Number)
- **Movement X** (Number)
- **Movement Y** (Number)
- **Movement Z** (Number)
- **Center X** (Number: Boolean)
- **Center Y** (Number: Boolean)
- **Center Z** (Number: Boolean)
- **Reset** (Trigger)
- **Lifetime** (Number)
- **Lifetime Minimum** (Number)

**< Output Ports:**

- **Trigger Out** (Trigger)
- **Positions** (Array)
- **Lifetime** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.PointArray.ArraySpray>

## 15.1.6 CircularPoints\_v2



**Full Name:** Ops.Array.PointArray.CircularPoints\_v2

**Description:** Create arrays for circular shapes, helix, circle, etc.

**> Input Ports:**

- **Radius** (Number)
- **Round Segments** (Number)
- **Rounds** (Number)

• **Radius Add Round** (Number)

• **Radius Add Point** (Number)

• **Offset** (Number)

• **Point Offset XY** (Number)

• **Point Offset Z** (Number)

• **Offset Rotation** (Number)

• **Loop** (Number: Boolean)

**< Output Ports:**

- **Points** (Array)

- **Rotation** (Array)

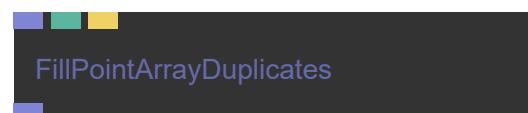
- **Total Points** (Number)

- **Array Lengths** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.CircularPoints\\_v2](https://cables.gl/op/Ops.Array.CircularPoints_v2)

## 15.1.7 FillPointArrayDuplicates



**Full Name:** Ops.Array.PointArray.FillPointArrayDuplicates

**Description:** Fill an XYZ array with existing duplicate points until it reaches the length

**> Input Ports:**

- **Array** (Array)
- **Num Elements** (Number: Integer)
- **Calculate** (Trigger)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.FillPointArrayDuplicates>

## 15.1.8 PointsCube



**Full Name:** Ops.Array.PointArray.PointsCube

**Description:** Generate a 3d point field with controllable amount of xyz points  
(was PointsField3d)

### > Input Ports:

- **Num X** (Number: Integer)
- **Num Y** (Number: Integer)
- **Num Z** (Number: Integer)
- **Mul** (Number)
- **Center** (Number: Boolean)

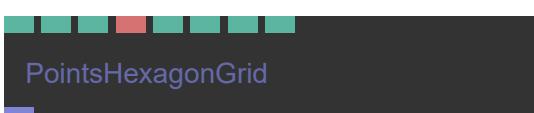
### < Output Ports:

- **Array Out** (Array)
- **Total Points** (Number)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.PointArray.PointsCube>

## 15.1.9 PointsHexagonGrid



**Full Name:** Ops.Array.PointArray.PointsHexagonGrid

**Description:** Generate coordinates for a hexagon grid, outputs array3x

### > Input Ports:

- **Rows** (Number: Integer)
- **Columns** (Number: Integer)
- **Hex Facing Index** (Number: Integer)
- **Hex Facing** (String)
- **Flip Corners** (Number: Boolean)
- **Tile X Offset** (Number)

- **Tile Y Offset** (Number)

- **Multiplier** (Number)

### < Output Ports:

- **Array Out** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.PointArray.PointsHexagonGrid>

## 15.1.10 PointsPlane\_v2



**Full Name:** Ops.Array.PointArray.PointsPlane\_v2

**Description:** Generate coordinates for a rectangular field / grid of points

### > Input Ports:

- **Rows** (Number: Integer)
- **Columns** (Number: Integer)
- **Width** (Number)
- **Height** (Number)
- **Row Offset** (Number)
- **Center** (Number: Boolean)

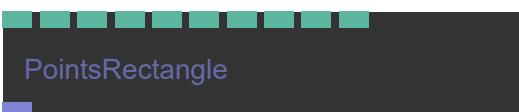
### < Output Ports:

- **Result** (Array)
- **Total Points** (Number)
- **Array Length** (Number)
- **Row Numbers** (Array)
- **Column Numbers** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.PointArray.PointsPlane\\_v2](https://cables.gl/op/Ops.Array.PointArray.PointsPlane_v2)

## 15.1.11 PointsRectangle\_v2



**Full Name:** Ops.Array.PointArray.PointsRectangle\_v2

**Description:** Generate an array of XYZ coordinates of an rectangle

**> Input Ports:**

- **Line Strip** (Number: Boolean)
- **Segments** (Number: Integer)
- **Width** (Number)
- **Height** (Number)
- **Border Radius** (Number)
- **Loop** (Number: Boolean)
- **Top Left** (Number: Boolean)
- **Top Right** (Number: Boolean)
- **Bottom Left** (Number: Boolean)
- **Bottom Right** (Number: Boolean)

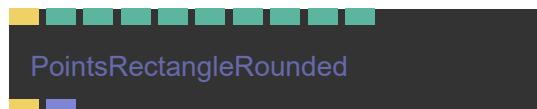
**< Output Ports:**

- **Points** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.PointArray.PointsRectangle\\_v2](https://cables.gl/op/Ops.Array.PointArray.PointsRectangle_v2)

## 15.1.12 PointsRectangleRounded\_v2



PointsRectangleRounded

**Full Name:** Ops.Array.PointArray.PointsRectangleRounded\_v2

**Description:** Generate an array of points of a rectangle with rounded corners

**> Input Ports:**

- **Render** (Trigger)
- **Segments** (Number: Integer)
- **Width** (Number)
- **Height** (Number)
- **Border Radius** (Number)
- **Top Left** (Number: Boolean)
- **Top Right** (Number: Boolean)
- **Bottom Left** (Number: Boolean)
- **Bottom Right** (Number: Boolean)
- **Draw** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)
- **Points** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.PointsRectangleRounded\\_v2](https://cables.gl/op/Ops.Array.PointsRectangleRounded_v2)

## 15.1.13 PointsSphereRandom



PointsSphereRandom

**Full Name:** Ops.Array.PointArray.PointsSphereRandom

**Description:** Generate a point field mapped to the surface of a sphere

**> Input Ports:**

- **Amount Of Points** (Number: Integer)
- **Sphere Size** (Number)
- **Random Seed** (Number)
- **Random Distance From Sphere** (Number)
- **Distribution Index** (Number: Integer)

**< Output Ports:**

- **Array Out** (Array)
- **Total Points** (Number)
- **Array Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.PointsSphereRandom>

## 15.1.14 RedistributeSplinePoints



RedistributeSplinePoints

**Full Name:** Ops.Array.PointArray.RedistributeSplinePoints

**Description:** Recalculate a spline / change number of points of a spline

**> Input Ports:**

- **Array3x** (Array)
- **Num Points** (Number: Integer)
- **Calculate** (Trigger)
- **Normalized** (Number: Boolean)

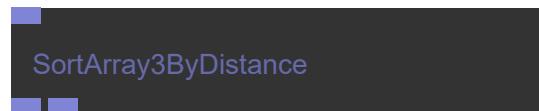
**< Output Ports:**

- **Result** (Array)
- **Spline Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.PointArray.RedistributeSplinePoints>

## 15.1.15 SortArray3ByDistance



**Full Name:** Ops.Array.PointArray.SortArray3ByDistance

**Description:** Sort an array3, by the distance of each point to the previous point

**> Input Ports:**

- **Array** (Array)

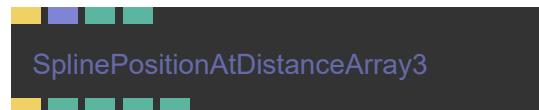
**< Output Ports:**

- **Result** (Array)
- **Result Index** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.PointArray.SortArray3ByDistance>

## 15.1.16 SplinePositionAtDistanceArray3



**Full Name:** Ops.Array.PointArray.SplinePositionAtDistanceArray3

**Description:** Get position in array3/spline at distance from start

**> Input Ports:**

- **Calculate** (Trigger)
- **Array3x** (Array)
- **Distance** (Number)
- **Normalized** (Number: Boolean)

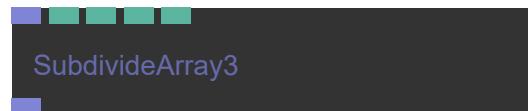
**< Output Ports:**

- **Next** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Spline Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.PointArray.SplinePositionAtDistanceArray3>

## 15.1.17 SubdivideArray3\_v2



**Full Name:** Ops.Array.PointArray.SubdivideArray3\_v2

**Description:** For subdividing splines, smoothing lines using cubic bezier interpolation

**> Input Ports:**

- **Points** (Array)
- **Num Subdivs** (Number: Integer)
- **Smooth** (Number: Boolean)
- **Loop** (Number: Boolean)

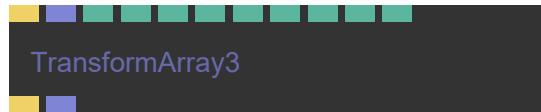
**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Array.PointArray.SubdivideArray3\\_v2](https://cables.gl/op/Ops.Array.PointArray.SubdivideArray3_v2)

## 15.1.18 TransformArray3



**Full Name:** Ops.Array.PointArray.TransformArray3

**Description:** Transform (translate, rotate, scale) positions in an array3x

### > Input Ports:

- **Transform** (Trigger)
- **Array** (Array)
- **Translate X** (Number)
- **Translate Y** (Number)
- **Translate Z** (Number)
- **Scale X** (Number)
- **Scale Y** (Number)
- **Scale Z** (Number)
- **Rotation X** (Number)
- **Rotation Y** (Number)
- **Rotation Z** (Number)

### < Output Ports:

- **Next** (Trigger)
- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Array.PointArray.TransformArray3>

# 16 Ops.Audio

## 16.1 Ops.Audio

### 16.1.1 BpmTap



**Full Name:** Ops.Audio.BpmTap

**Description:** Let's you tap in a beat, useful to synchronise visuals to music (VJ, sync, sound)

### > Input Ports:

- **Exe** (Trigger)
- **Tap** (Trigger)
- **Sync** (Trigger)
- **NudgeLeft** (Trigger)
- **NudgeRight** (Trigger)
- **Active** (Number: Boolean)

### < Output Ports:

- **Beat** (Trigger)
- **Bpm** (Number)
- **The resulting BPM** (beats per minute)
- **States** (Array)
- **Beat Index** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Audio.BpmTap>

### 16.1.2 MidiJson



**Full Name:** Ops.Audio.MidiJson

**Description:** read MIDI information at time x

**> Input Ports:**

- **MidiJson** (Object)
- **Time** (Number)

**< Output Ports:**

- **Beat** (Number)
- **Track Names** (Array)
- **Names** (Array)
- **Progress** (Array)
- **Velocity** (Array)
- **Num Tracks** (Number)
- **BPM** (Number)
- **Data** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Audio.MidiJson>

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Audio.MidiJsonNote\\_v2](https://cables.gl/op/Ops.Audio.MidiJsonNote_v2)

## 16.1.3 MidiJsonNote\_v2



**Full Name:** Ops.Audio.MidiJsonNote\_v2

**Description:** Filter MidiJson for notes

**> Input Ports:**

- **Data** (Object)
- **Note** (String)
- **Channel** (Number: String)
- **Beat Start** (Number: Integer)
- **Beat End** (Number: Integer)

**< Output Ports:**

- **Count** (Number)
- **Progress** (Number)
- **Time Since Last** (Number)
- **Trigger** (Trigger)
- **Reseted** (Trigger)

# 17 Ops.Boolean

## 17.1 Ops.Boolean

### 17.1.1 And



**Full Name:** Ops.Boolean.And

**Description:** Outputs true if both input values are true (boolean)

#### > Input Ports:

- **Bool 1** (Number: Boolean)
- **Bool 2** (Number: Boolean)

#### < Output Ports:

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Boolean.And>

### 17.1.2 AndMultiPort\_v2



**Full Name:** Ops.Boolean.AndMultiPort\_v2

**Description:** Outputs true if all input values are true (boolean)

#### > Input Ports:

- **Booleans\_0** (Number: Boolean)
- **Add Port** (Number: Boolean)

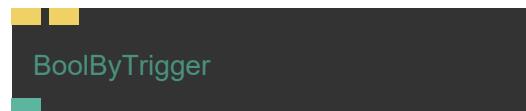
#### < Output Ports:

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Boolean.AndMultiPort\\_v2](https://cables.gl/op/Ops.Boolean.AndMultiPort_v2)

### 17.1.3 BoolByTrigger



**Full Name:** Ops.Boolean.BoolByTrigger

**Description:** Trigger true or false values

#### > Input Ports:

- **True** (Trigger)
- **False** (Trigger)

#### < Output Ports:

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Boolean.BoolByTrigger>

### 17.1.4 Boolean



**Full Name:** Ops.Boolean.Boolean

**Description:** Stores a boolean value

#### > Input Ports:

- **Value** (Number: Boolean)

#### < Output Ports:

- **Result** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Boolean.Boolean>

## 17.1.5 BoolToColor



**Full Name:** Ops.Boolean.BoolToColor

**Description:** Convert boolean to RGB color

### > Input Ports:

- **Boolean** (Number: Boolean)
- **R True** (Number)
- **G True** (Number)
- **B True** (Number)
- **A True** (Number)
- **R False** (Number)
- **G False** (Number)
- **B False** (Number)
- **A False** (Number)

### < Output Ports:

- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Boolean.BoolToColor>

## 17.1.6 BoolToNumber\_v2



**Full Name:** Ops.Boolean.BoolToNumber\_v2

**Description:** Switches two number values using a boolean

### > Input Ports:

- **Use Value 1** (Number: Boolean)
- **Value 0** (Number)

- **Value 1** (Number)

### < Output Ports:

- **Out Value** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Boolean.BoolToNumber\\_v2](https://cables.gl/op/Ops.Boolean.BoolToNumber_v2)

## 17.1.7 BoolToString



**Full Name:** Ops.Boolean.BoolToString

**Description:** convert boolean to string

### > Input Ports:

- **Boolean** (Number: Boolean)
- **False** (String)
- **True** (String)

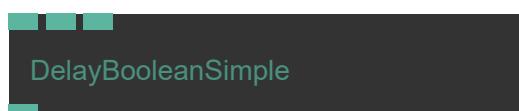
### < Output Ports:

- **String** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Boolean.BoolToString>

## 17.1.8 DelayBooleanSimple



**Full Name:** Ops.Boolean.DelayBooleanSimple

**Description:** Delay the input/output of a boolean by x seconds

### > Input Ports:

- **Value** (Number)
- **Delay True** (Number)
- **Delay False** (Number)

#### < Output Ports:

- **Out Value** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Boolean.DelayBooleanSimple>

## 17.1.9 IfFalseThen



**Full Name:** Ops.Boolean.IfFalseThen

**Description:** Triggers if input value is false

#### > Input Ports:

- **Exe** (Trigger)
- **Boolean** (Number: Boolean)

#### < Output Ports:

- **Exe** (Trigger)
- **Boolean** (Number: Boolean)
- **Then** (Trigger)
- **Else** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Boolean.IfFalseThen>

## 17.1.10 IfTrueThen\_v2



**Full Name:** Ops.Boolean.IfTrueThen\_v2

**Description:** Switch, trigger one or the other trigger port based on the input value

#### > Input Ports:

- **Exe** (Trigger)
- **Boolean** (Number: Boolean)

#### < Output Ports:

- **Then** (Trigger)
- **Else** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Boolean.IfTrueThen\\_v2](https://cables.gl/op/Ops.Boolean.IfTrueThen_v2)

## 17.1.11 IsOne



**Full Name:** Ops.Boolean.IsOne

**Description:** Returns true if input value is 1

#### > Input Ports:

- **Value** (Number)

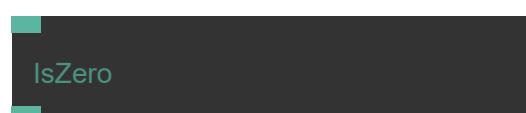
#### < Output Ports:

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Boolean.IsOne>

## 17.1.12 IsZero



**Full Name:** Ops.Boolean.IsZero

**Description:** Returns true if input value is 0

#### > Input Ports:

- **Value** (Number)

#### < Output Ports:

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Boolean.IsZero>

## 17.1.13 MonoFlop



**Full Name:** Ops.Boolean.MonoFlop

**Description:** Sets output to 1 when triggered, turns back to 0 automatically after x seconds

**> Input Ports:**

- **Trigger** (Trigger)
- **Duration** (Number)
- **Value True** (Number)
- **Value False** (Number)
- **Reset** (Trigger)

**< Output Ports:**

- **Activated** (Trigger)
- **Ended** (Trigger)
- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Boolean.MonoFlop>

## 17.1.14 Not



**Full Name:** Ops.Boolean.Not

**Description:** result is false if input is true and vice versa (negate/toggle/switch/!=)

**> Input Ports:**

- **Boolean** (Number: Boolean)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Boolean.Not>

## 17.1.15 Or



**Full Name:** Ops.Boolean.Or

**Description:** Returns true if one or more of the input booleans are true

**> Input Ports:**

- **Bool 1** (Number: Boolean)
- **Bool 2** (Number: Boolean)
- **Bool 3** (Number: Boolean)
- **Bool 4** (Number: Boolean)
- **Bool 5** (Number: Boolean)
- **Bool 6** (Number: Boolean)
- **Bool 7** (Number: Boolean)
- **Bool 8** (Number: Boolean)
- **Bool 9** (Number: Boolean)
- **Bool 10** (Number: Boolean)

**< Output Ports:**

- **Result** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Boolean.Or>

## 17.1.16 OrNumber\_v2



**Full Name:** Ops.Boolean.OrNumber\_v2

**Description:** Output another number if input number is zero

**> Input Ports:**

- **Number** (Number)
- **Number 2** (Number)
- **Number 3** (Number)
- **Number 4** (Number)

- **Number 5** (Number)
- **Number 6** (Number)
- **Number 7** (Number)
- **Number 8** (Number)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Boolean.OrNumber\\_v2](https://cables.gl/op/Ops.Boolean.OrNumber_v2)

### 17.1.17 ParseBoolean\_v2



**Full Name:** Ops.Boolean.ParseBoolean\_v2

**Description:** parse boolean from string/number

**> Input Ports:**

- **String** (String)

**< Output Ports:**

- **Result** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Boolean.ParseBoolean\\_v2](https://cables.gl/op/Ops.Boolean.ParseBoolean_v2)

### 17.1.18 RouteBoolean



**Full Name:** Ops.Boolean.RouteBoolean

**Description:** Route a boolean to an output port

**> Input Ports:**

- **Index** (Number: Integer)
- **Boolean In** (Number: Boolean)
- **Default Boolean** (Number: Boolean)

**< Output Ports:**

- **Index 0 Boolean** (booleanNumber)
- **Index 1 Boolean** (booleanNumber)
- **Index 2 Boolean** (booleanNumber)
- **Index 3 Boolean** (booleanNumber)
- **Index 4 Boolean** (booleanNumber)
- **Index 5 Boolean** (booleanNumber)
- **Index 6 Boolean** (booleanNumber)
- **Index 7 Boolean** (booleanNumber)
- **Index 8 Boolean** (booleanNumber)
- **Index 9 Boolean** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Boolean.RouteBoolean>

### 17.1.19 ToggleBool\_v2



**Full Name:** Ops.Boolean.ToggleBool\_v2

**Description:** Toggle a boolean value by triggering

**> Input Ports:**

- **Trigger** (Trigger)
- **Reset** (Trigger)
- **Default** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)
- **Result** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Boolean.ToggleBool\\_v2](https://cables.gl/op/Ops.Boolean.ToggleBool_v2)

## 17.1.20 TriggerChangedFalse



**Full Name:** Ops.Boolean.TriggerChangedFalse

**Description:** Triggers next only after value has changed to false

**> Input Ports:**

- **Value** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Boolean.TriggerChangedFalse>

**Full Name:** Ops.Boolean.TriggerOnChangeBoolean\_v2

**Description:** Triggers when boolean value has changed

**> Input Ports:**

- **Value** (Number: Boolean)

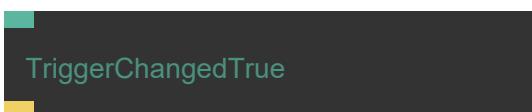
**< Output Ports:**

- **True** (Trigger)
- **False** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Boolean.TriggerOnChangeBoolean\\_v2](https://cables.gl/op/Ops.Boolean.TriggerOnChangeBoolean_v2)

## 17.1.21 TriggerChangedTrue



**Full Name:** Ops.Boolean.TriggerChangedTrue

**Description:** Triggers next only after value has changed to true

**> Input Ports:**

- **Value** (Number: Boolean)

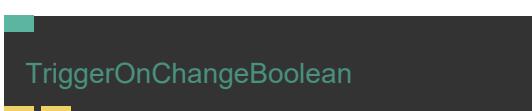
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Boolean.TriggerChangedTrue>

## 17.1.22 TriggerOnChangeBoolean\_v2



# 18 Ops.Cables

## 18.1 Ops.Cables

### 18.1.1 AssetPathURL

AssetPathURL

**Full Name:** Ops.Cables.AssetPathURL

**Description:** outputs the path to the assets

#### > Input Ports:

- **Filename** (String)

#### < Output Ports:

- **Path** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Cables.AssetPathURL>

### 18.1.2 CablesInfo

CablesInfo

**Full Name:** Ops.Cables.CablesInfo

**Description:** Output the cables URL of the current editor environment

#### > Input Ports:

- Visit *Ops.Cables.CablesInfo documentation for input port details*

#### < Output Ports:

- **URL** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Cables.CablesInfo>

### 18.1.3 CallBack\_v2

CallBack

**Full Name:** Ops.Cables.CallBack\_v2

**Description:** Useful when a cables patch is embedded into a website. All parameters (Value 1, Value 2, Value 3) will be send as a parameter array. So e.g. if Callback Name is foo cables would call: CABLES.patch.config.foo([Value 1, Value 2, Value 3])

#### > Input Ports:

- **Exe** (Trigger)
- **Callback Name** (String)
- **Parameter 1** (String)
- **Parameter 2** (String)
- **Parameter 3** (String)
- **Public** (7): LANDINGPORTAFOLIO
- **LOGICX BED** (PUBLIC): wirmachenbunt - Published Sep 30, 2021 at 12:25

#### < Output Ports:

- Visit *Ops.Cables.CallBack\_v2 documentation for output port details*

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Cables.CallBack\\_v2](https://cables.gl/op/Ops.Cables.CallBack_v2)

### 18.1.4 FPS\_v2

FPS

**Full Name:** Ops.Cables.FPS\_v2

**Description:** output current frames per second

#### > Input Ports:

- Visit *Ops.Cables.FPS\_v2 documentation for input port details*

#### < Output Ports:

- **FPS** (Number)
- **MS** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Cables.FPS\\_v2](https://cables.gl/op/Ops.Cables.FPS_v2)

## 18.1.5 Function\_v2



**Full Name:** Ops.Cables.Function\_v2

**Description:** trigger from external function when embedded into a website

**> Input Ports:**

- **Function Name** (String)
- **Trigger** (Trigger)
- **Default Parameter 1** (String)
- **Default Parameter 2** (String)
- **Default Parameter 3** (String)

**< Output Ports:**

- **Next** (Trigger)
- **Parameter 1** (String)
- **Parameter 2** (String)
- **Parameter 3** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Cables.Function\\_v2](https://cables.gl/op/Ops.Cables.Function_v2)

## 18.1.6 GetOpName



**Full Name:** Ops.Cables.GetOpName

**Description:** Get op name by id

**> Input Ports:**

- **OpId** (String)

**< Output Ports:**

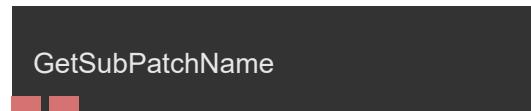
- **Found** (booleanNumber)

- **Name** (String)
- **Shortname** (String)
- **Version** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Cables.GetOpName>

## 18.1.7 GetSubPatchName



**Full Name:** Ops.Cables.GetSubPatchName

**Description:** Outputs the current subpatch op name

**> Input Ports:**

- Visit *Ops.Cables.GetSubPatchName documentation for input port details*

**< Output Ports:**

- **Name** (String)
- **ShortName** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Cables.GetSubPatchName>

## 18.1.8 LoadingJob



**Full Name:** Ops.Cables.LoadingJob

**Description:** Create a loading job while input is true

**> Input Ports:**

- **Loading Active** (Number: Boolean)

**< Output Ports:**

- Visit *Ops.Cables.LoadingJob documentation for output port details*

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Cables.LoadingJob>

### 18.1.9 LoadingStatus\_v2



**Full Name:** Ops.Cables.LoadingStatus\_v2

**Description:** trigger events / get information about asset-loading status

**> Input Ports:**

- **Exe** (Trigger)
- **Play Timeline** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)
- **Finished Initial Loading** (booleanNumber)
- **Loading** (booleanNumber)
- **Progress** (Number)
- **Jobs** (Array)
- **Trigger Loading Finished** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Cables.LoadingStatus\\_v2](https://cables.gl/op/Ops.Cables.LoadingStatus_v2)

### 18.1.10 PatchInfo\_v2



**Full Name:** Ops.Cables.PatchInfo\_v2

**Description:** read patch config when embedding on another page

**> Input Ports:**

- Visit *Ops.Cables.PatchInfo\_v2 documentation for input port details*

**< Output Ports:**

- **Config** (Object)

- **Name** (String)
- **Patch Id** (String)
- **Namespace** (String)
- **Last Saved** (Number)
- **Last Exported** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Cables.PatchInfo\\_v2](https://cables.gl/op/Ops.Cables.PatchInfo_v2)

### 18.1.11 UIMode



**Full Name:** Ops.Cables.UIMode

**Description:** Outputs true if patch is executed in the cables editor (UI)

**> Input Ports:**

- Visit *Ops.Cables.UIMode documentation for input port details*

**< Output Ports:**

- **UI** (booleanNumber)
- **Overlay Mode** (booleanNumber)
- **Remote Viewer** (booleanNumber)
- **Is Standalone** (booleanNumber)
- **Canvas Mode** (Number)
- **Patch Field Visible** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Cables.UIMode>

### 18.1.12 UploadAsset



**Full Name:** Ops.Cables.UploadAsset

**Description:** Upload a file into the cables patch assets using a base64 string

**> Input Ports:**

- **Filename** (String)
- **Base64 String** (String)
- **Upload** (Trigger)

**< Output Ports:**

- **Result** (String)
- **Error** (booleanNumber)
- **Finished** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Cables.UploadAsset>

### 18.1.13 UploadScreenshot



**Full Name:** Ops.Cables.UploadScreenshot

**Description:** Upload an image as screenshot in cables

**> Input Ports:**

- **Trigger** (Trigger)
- **DataUrl** (String)

**< Output Ports:**

- **Result** (Number)

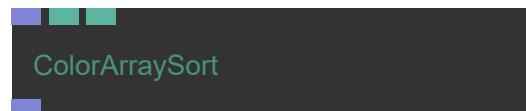
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Cables.UploadScreenshot>

## 19 Ops.Color

### 19.1 Ops.Color

#### 19.1.1 ColorArraySort



**Full Name:** Ops.Color.ColorArraySort

**Description:** Sort an array of colors by saturation/lightness etc.

**> Input Ports:**

- **Colors** (Array)

**< Output Ports:**

- **New Colors** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Color.ColorArraySort>

#### 19.1.2 ColorPalettes



**Full Name:** Ops.Color.ColorPalettes

**Description:** Contains a collection of nice color palettes output to texture or array via index

**> Input Ports:**

- **Index** (Number: Integer)
- **Smooth** (Number: Boolean)

**< Output Ports:**

- **Texture** (Object)
- **Color Array** (Array)

- The color array containing 5 colors (15 values in total, 3 values per color
  - r, g and b)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Color.ColorPalettes>

### 19.1.3 ColorValue



**Full Name:** Ops.Color.ColorValue

**Description:** Use a color value on multiple places

**> Input Ports:**

- R (Number)
- G (Number)
- B (Number)
- A (Number)

**< Output Ports:**

- Outr (Number)
- Outg (Number)
- Outb (Number)
- Outa (Number)
- Hex (Number)
- Array (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Color.ColorValue>

### 19.1.4 EyeDropper



**Full Name:** Ops.Color.EyeDropper

**Description:** Native color picker

**> Input Ports:**

- Open (Trigger)

**< Output Ports:**

- Hex (String)
- R (Number)
- G (Number)
- B (Number)
- Supported (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Color.EyeDropper>

### 19.1.5 Gradient



**Full Name:** Ops.Color.Gradient

**Description:** gradient editor,outputs an objects with gradient information

**> Input Ports:**

- Gradient (Number)
- Randomize Colors (Trigger)

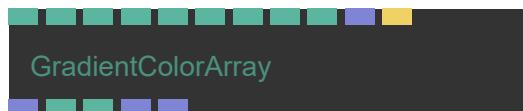
**< Output Ports:**

- Gradient Object (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Color.Gradient>

### 19.1.6 GradientColorArray



**Full Name:** Ops.Color.GradientColorArray

**Description:** texture containing a colour gradient that can be altered with an editor

#### > Input Ports:

- **Gradient** (Number)
- **Direction Index** (Number: Integer)
- **Smoothstep** (Number: Boolean)
- **Step** (Number: Boolean)
- **Flip** (Number: Boolean)
- **SRGB** (Number: Boolean)
- **Oklab** (Number: Boolean)
- **Size** (Number: Integer)
- **Dither** (Number)
- **Gradient Array** (Array)
- **Randomize Colors** (Trigger)

#### < Output Ports:

- **Color Array** (Array)
- **Width** (Number)
- **Height** (Number)
- **Colors** (Array)
- **Colors Pos** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Color.GradientColorArray>

## 19.1.7 HexToRGB\_v2



**Full Name:** Ops.Color.HexToRGB\_v2

**Description:** Converts a hex color like #ff0000 to number values

#### > Input Ports:

- **Hex** (String)
- **Bytes** (Number: Boolean)

#### < Output Ports:

- **R** (Number)
- **G** (Number)
- **B** (Number)
- **RGB Array** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Color.HexToRGB\\_v2](https://cables.gl/op/Ops.Color.HexToRGB_v2)

## 19.1.8 HSLtoRGB



**Full Name:** Ops.Color.HSLtoRGB

**Description:** Convert HSL to RGB

#### > Input Ports:

- **Hue** (Number)
- **Saturation** (Number)
- **Lightness** (Number)

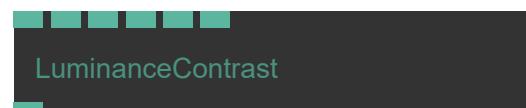
#### < Output Ports:

- **R** (Number)
- **G** (Number)
- **B** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Color.HSLtoRGB>

## 19.1.9 LuminanceContrast



**Full Name:** Ops.Color.LuminanceContrast

**Description:** Calculate the luminance contrast between two colors

#### > Input Ports:

- **R 1** (Number)
- **G 1** (Number)
- **B 1** (Number)
- **R 2** (Number)
- **G 2** (Number)

- **B** (Number)

**< Output Ports:**

- **Contrast** (Number)

**Example Patch:** Open in Editor

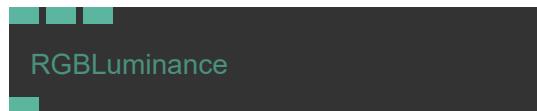
**Docs:** <https://cables.gl/op/Ops.Color.LuminanceContrast>

- **C** (Number)
- **M** (Number)
- **Y** (Number)
- **K** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Color.RGBToCMYK>

## 19.1.10 RGBLuminance



**Full Name:** Ops.Color.RGBLuminance

**Description:** Calculate the luminance of a RGB color

**> Input Ports:**

- **R** (Number)
- **G** (Number)
- **B** (Number)

**< Output Ports:**

- **Luminance** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Color.RGBLuminance>

## 19.1.11 RGBToCMYK



**Full Name:** Ops.Color.RGBToCMYK

**Description:** Output the CMYK value of a RGB color

**> Input Ports:**

- **R** (Number)
- **G** (Number)
- **B** (Number)

**< Output Ports:**

## 19.1.12 RgbToHex



**Full Name:** Ops.Color.RgbToHex

**Description:** convert RGB float values to HEX color String

**> Input Ports:**

- **R** (Number)
- **G** (Number)
- **B** (Number)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Color.RgbToHex>

## 19.1.13 RGBtoHSB



**Full Name:** Ops.Color.RGBtoHSB

**Description:** convert RGB color to HSB Hue, Saturation, Brightness

**> Input Ports:**

- **R** (Number)
- **G** (Number)
- **B** (Number)

#### < Output Ports:

- **Hue** (Number)
- **Saturation** (Number)
- **Brightness** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Color.RGBtoHSB>

### 19.1.14 RGBtoHSL



**Full Name:** Ops.Color.RGBtoHSL

**Description:** Convert RGB color to HSL values

#### > Input Ports:

- **R** (Number)
- **G** (Number)
- **B** (Number)

#### < Output Ports:

- **Hue** (Number)
- **Saturation** (Number)
- **Lightness** (Number)

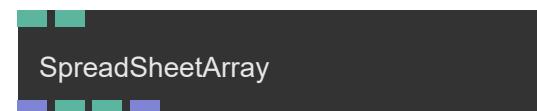
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Color.RGBtoHSL>

## 20 Ops.Data

### 20.1 Ops.Data

#### 20.1.1 SpreadSheetArray



**Full Name:** Ops.Data.SpreadSheetArray

**Description:** Enter data in a spreadsheet table

#### > Input Ports:

- Visit *Ops.Data.SpreadSheetArray documentation for input port details*

#### < Output Ports:

- **Array** (Array)
- **Width** (Number)
- **Height** (Number)
- **Column Names** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.SpreadSheetArray>

# 21 Ops.Data.Compose.Array

## 21.1 Ops.Data.Compose.Array

### 21.1.1 CompArray



**Full Name:** Ops.Data.Compose.Array.CompArray

**Description:** Compose an Array

#### > Input Ports:

- **Update** (Trigger)
- **Active** (Number: Boolean)
- **Clear** (Number: Boolean)
- **Reset** (Trigger)

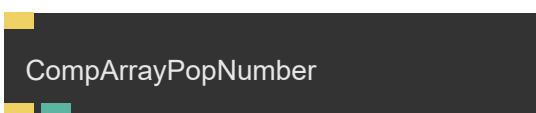
#### < Output Ports:

- **Next** (Trigger)
- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Array.CompArray>

### 21.1.2 CompArrayPopNumber



**Full Name:** Ops.Data.Compose.Array.CompArrayPopNumber

**Description:** pop/remove the last number from an array

#### > Input Ports:

- **Update** (Trigger)

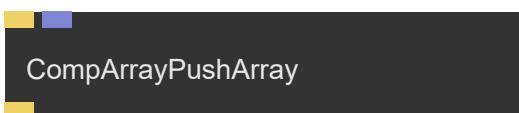
#### < Output Ports:

- **Next** (Trigger)
- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Array.CompArrayPopNumber>

### 21.1.3 CompArrayPushArray



**Full Name:** Ops.Data.Compose.Array.CompArrayPushArray

**Description:** push/append an array to an array

#### > Input Ports:

- **Update** (Trigger)
- **Array** (Array)

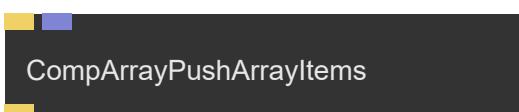
#### < Output Ports:

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Array.CompArrayPushArray>

### 21.1.4 CompArrayPushArrayItems



**Full Name:** Ops.Data.Compose.Array.CompArrayPushArrayItems

**Description:** push/append an array to an array

#### > Input Ports:

- **Update** (Trigger)
- **Array** (Array)

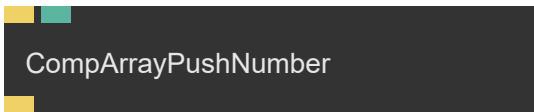
#### < Output Ports:

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Array.CompArrayPushArray>  
Items

## 21.1.5 CompArrayPushNumber



**Full Name:** Ops.Data.Compose.Array.CompArrayPushNumber

**Description:** push/append a number to an array

**> Input Ports:**

- **Update** (Trigger)
- **Number** (Number)

**< Output Ports:**

- **Next** (Trigger)

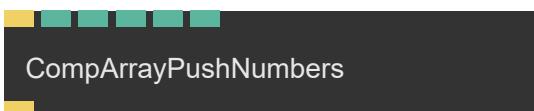
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Array.CompArrayPushNumber>

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Array.CompArrayPushNumbers>

## 21.1.6 CompArrayPushNumbers



**Full Name:** Ops.Data.Compose.Array.CompArrayPushNumbers

**Description:** push/append multiple numbers to an array

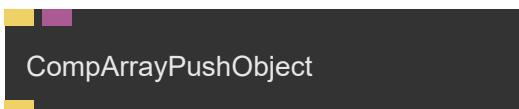
**> Input Ports:**

- **Update** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **W** (Number)

**< Output Ports:**

- **Next** (Trigger)

## 21.1.7 CompArrayPushObject



**Full Name:** Ops.Data.Compose.Array.CompArrayPushObject

**Description:** push/append an object to an array

**> Input Ports:**

- **Update** (Trigger)
- **Object** (Object)

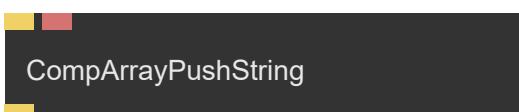
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Array.CompArrayPushObject>

## 21.1.8 CompArrayPushString



**Full Name:** Ops.Data.Compose.Array.CompArrayPushString

**Description:** push/append a string to an array

**> Input Ports:**

- **Update** (Trigger)
- **String** (String)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Array.CompArrayPushString>

## 21.1.9 CompArrayPushTexture

CompArrayPushTexture

**Full Name:** Ops.Data.Compose.Array.CompArrayPushTexture

**Description:** push/append a texture to an array

**> Input Ports:**

- **Update** (Trigger)
- **Object** (Object)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Array.CompArrayPushTexture>

## 21.1.10 CompArraySetNumber

CompArraySetNumber

**Full Name:** Ops.Data.Compose.Array.CompArraySetNumber

**Description:** set a number to an array at index

**> Input Ports:**

- **Update** (Trigger)
- **Index** (Number)
- **Number** (Number)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Array.CompArraySetNumber>

## 21.1.11 CompArraySetObject

CompArraySetObject

**Full Name:** Ops.Data.Compose.Array.CompArraySetObject

**Description:** push/append a number to an array

**> Input Ports:**

- **Update** (Trigger)
- **Index** (Number)
- **Object** (Object)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Array.CompArraySetObject>

## 21.1.12 CompArrayShiftNumber

CompArrayShiftNumber

**Full Name:** Ops.Data.Compose.Array.CompArrayShiftNumber

**Description:** shift/remove the first number from an array

**> Input Ports:**

- **Update** (Trigger)

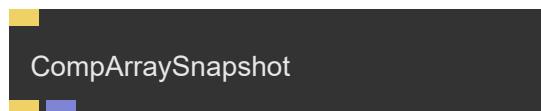
**< Output Ports:**

- **Next** (Trigger)
- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Array.CompArrayShiftNumber>

## 21.1.13 CompArraySnapshot



**Full Name:** Ops.Data.Compose.Array.CompArraySnapshot

**Description:** get a copy of the current state of an array

**> Input Ports:**

- **Update** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Array.CompArraySnapshot>

## 22 Ops.Data.Compose.Object

### 22.1 Ops.Data.Compose.Object

#### 22.1.1 CompObject



**Full Name:** Ops.Data.Compose.Object.CompObject

**Description:** Compose an Object

**> Input Ports:**

- **Update** (Trigger)
- **Clear** (Number: Boolean)
- **Reset** (Trigger)

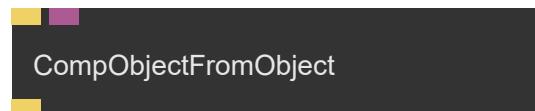
**< Output Ports:**

- **Next** (Trigger)
- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Object.CompObject>

#### 22.1.2 CompObjectFromObject



**Full Name:** Ops.Data.Compose.Object.CompObjectFromObject

**Description:** Set key/values to the current ObjectCompose from an existing object

**> Input Ports:**

- **Update** (Trigger)
- **Object** (Object)

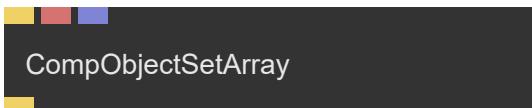
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Object.CompObjectFromObj>

### 22.1.3 CompObjectSetArray



**Full Name:** Ops.Data.Compose.Object.CompObjectSetArray

**Description:** set array as object property

**> Input Ports:**

- **Update** (Trigger)
- **Key** (String)
- **Array** (Array)

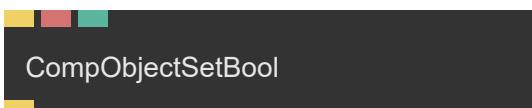
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Object.CompObjectSetArray>

### 22.1.4 CompObjectSetBool



**Full Name:** Ops.Data.Compose.Object.CompObjectSetBool

**Description:** set a boolean as object property

**> Input Ports:**

- **Update** (Trigger)
- **Key** (String)
- **Boolean** (Number: Boolean)

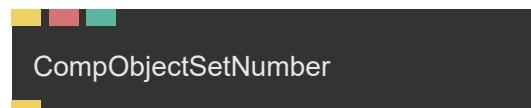
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Object.CompObjectSetBool>

### 22.1.5 CompObjectSetNumber



**Full Name:** Ops.Data.Compose.Object.CompObjectSetNumber

**Description:** set number as object property

**> Input Ports:**

- **Update** (Trigger)
- **Key** (String)
- **Number** (Number)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Object.CompObjectSetNumber>

### 22.1.6 CompObjectSetObject



**Full Name:** Ops.Data.Compose.Object.CompObjectSetObject

**Description:** set object as object property

**> Input Ports:**

- **Update** (Trigger)
- **Key** (String)
- **Object** (Object)

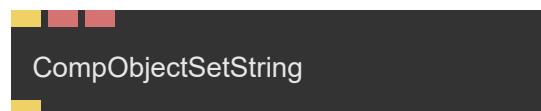
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Object.CompObjectSetObject>

## 22.1.7 CompObjectSetString



**Full Name:** Ops.Data.Compose.Object.CompObjectSetString

**Description:** set string as object property

**> Input Ports:**

- **Update** (Trigger)
- **Key** (String)
- **String** (String)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.Object.CompObjectSetString>

# 23 Ops.Data.Compose.String

## 23.1 Ops.Data.Compose.String

### 23.1.1 CompString



**Full Name:** Ops.Data.Compose.String.CompString

**Description:** Compose a string

**> Input Ports:**

- **Update** (Trigger)
- **Clear** (Number: Boolean)
- **Reset** (Trigger)

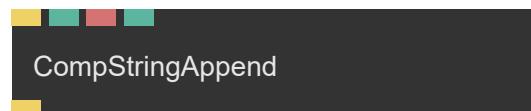
**< Output Ports:**

- **Next** (Trigger)
- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.String.CompString>

### 23.1.2 CompStringAppend



**Full Name:** Ops.Data.Compose.String.CompStringAppend

**Description:** Append a string to a string

**> Input Ports:**

- **Update** (Trigger)
- **String** (String)
- **Add Break** (Number: Boolean)

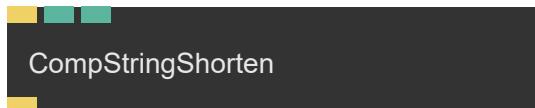
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.String.CompStringAppend>

### 23.1.3 CompStringShorten



**Full Name:** Ops.Data.Compose.String.CompStringShorten

**Description:** Remove characters from the beginning or end of a string

**> Input Ports:**

- **Update** (Trigger)
- **Direction Index** (Number: Integer)
- **Num Chars** (Number: Integer)

**< Output Ports:**

- **Next** (Trigger)

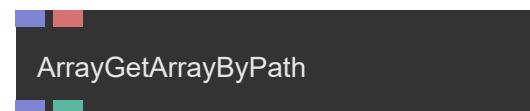
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.Compose.String.CompStringShorten>

## 24 Ops.Data.JsonPath

### 24.1 Ops.Data.JsonPath

#### 24.1.1 ArrayGetArrayByPath



**Full Name:** Ops.Data.JsonPath.ArrayGetArrayByPath

**Description:** returns the array at the position defined by a path

**> Input Ports:**

- **Array** (Array)
- **Path** (String)
- **path to array** (i.e. data.numbers)

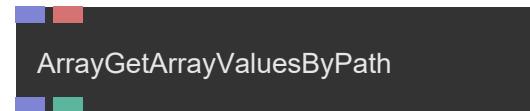
**< Output Ports:**

- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.JsonPath.ArrayGetArrayByPath>

#### 24.1.2 ArrayGetArrayValuesByPath



**Full Name:** Ops.Data.JsonPath.ArrayGetArrayValuesByPath

**Description:** Outputs all the values of the properties of an array of objects given a path

**> Input Ports:**

- **Array** (Array)
- **Path** (String)
- **path to first array field** (i.e. "data.0.firstName")

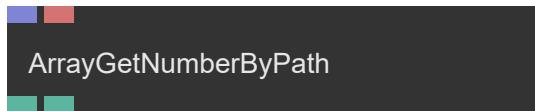
**< Output Ports:**

- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.JsonPath.ArrayGetArrayValuesByPath>

### 24.1.3 ArrayGetNumberByPath



**Full Name:** Ops.Data.JsonPath.ArrayGetNumberByPath

**Description:** finds a number at a position in an array defined by path

➢ **Input Ports:**

- **Array** (Array)
- **Path** (String)
- **the past** (i.e. person.age)

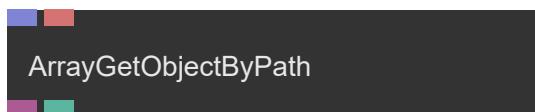
◀ **Output Ports:**

- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.JsonPath.ArrayGetNumberByPath>

### 24.1.4 ArrayGetObjectByPath



**Full Name:** Ops.Data.JsonPath.ArrayGetObjectByPath

**Description:** Returns the object at the position defined by a path

➢ **Input Ports:**

- **Array** (Array)
- **Path** (String)

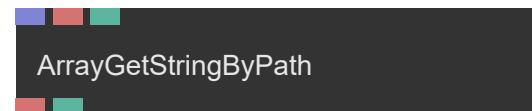
◀ **Output Ports:**

- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.JsonPath.ArrayGetObjectByPath>

### 24.1.5 ArrayGetStringByPath\_v2



**Full Name:** Ops.Data.JsonPath.ArrayGetStringByPath\_v2

**Description:** Finds a string at a position in an array defined by path

➢ **Input Ports:**

- **Array** (Array)
- **Path** (String)
- **the path** (i.e. data.names)
- **Return Path If Missing** (Number: Boolean)

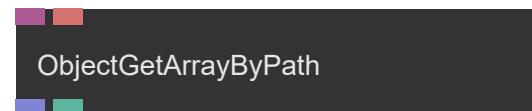
◀ **Output Ports:**

- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Data.JsonPath.ArrayGetStringByPath\\_v2](https://cables.gl/op/Ops.Data.JsonPath.ArrayGetStringByPath_v2)

### 24.1.6 ObjectGetArrayByPath



**Full Name:** Ops.Data.JsonPath.ObjectGetArrayByPath

**Description:** returns the array at the position defined by a path

➢ **Input Ports:**

- **Object** (Object)
- **Path** (String)
- **path to array** (i.e. data.numbers)

◀ **Output Ports:**

- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.JsonPath.ObjectGetArrayByPath>

## 24.1.7 ObjectGetArrayValuesByPath

### ObjectGetArrayValuesByPath

**Full Name:** Ops.Data.JsonPath.ObjectGetArrayValuesByPath

**Description:** Outputs all the values of the properties of an array of objects given a path

**> Input Ports:**

- **Object** (Object)
- **Path** (String)
- **path to first array field** (i.e. "data.0.firstName")

**< Output Ports:**

- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.JsonPath.ObjectGetArrayValuesByPath>

## 24.1.8 ObjectGetNumberByPath

### ObjectGetNumberByPath

**Full Name:** Ops.Data.JsonPath.ObjectGetNumberByPath

**Description:** finds a number at a position in an object defined by path

**> Input Ports:**

- **Object** (Object)
- **Path** (String)
- **the past** (i.e. person.age)

**< Output Ports:**

- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.JsonPath.ObjectGetNumberByPath>

## 24.1.9 ObjectGetObjectByPath

### ObjectGetObjectByPath

**Full Name:** Ops.Data.JsonPath.ObjectGetObjectByPath

**Description:** Returns the object at the position defined by a path

**> Input Ports:**

- **Object** (Object)
- **Path** (String)

**< Output Ports:**

- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.JsonPath.ObjectGetObjectByPath>

## 24.1.10 ObjectGetStringByPath\_v2

### ObjectGetStringByPath

**Full Name:** Ops.Data.JsonPath.ObjectGetStringByPath\_v2

**Description:** Finds a string at a position in an object defined by path

**> Input Ports:**

- **Object** (Object)
- **Path** (String)
- **the path** (i.e. data.names)
- **Output Path If Missing** (Number: Boolean)

**< Output Ports:**

- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Data.JsonPath.ObjectGetStringByPath\\_v2](https://cables.gl/op/Ops.Data.JsonPath.ObjectGetStringByPath_v2)

# 25 Ops.Data.StackValues

## 25.1 Ops.Data.StackValues

### 25.1.1 StackGetArray



**Full Name:** Ops.Data.StackValues.StackGetArray

**Description:** read a value from the stack to use it later in the trigger branch

**> Input Ports:**

- **Trigger** (Trigger)
- **Name** (String)

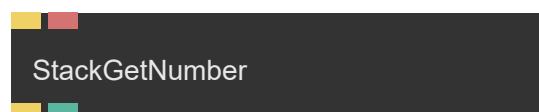
**< Output Ports:**

- **Next** (Trigger)
- **Array** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.StackValues.StackGetArray>

### 25.1.2 StackGetNumber



**Full Name:** Ops.Data.StackValues.StackGetNumber

**Description:** read a value from the stack to use it later in the trigger branch

**> Input Ports:**

- **Exec** (Trigger)
- **Name** (String)

**< Output Ports:**

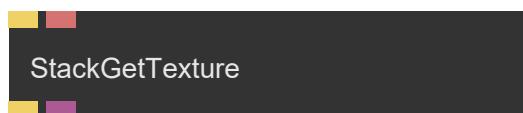
- **Next** (Trigger)

- **Value** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.StackValues.StackGetNumber>

### 25.1.3 StackGetTexture



**Full Name:** Ops.Data.StackValues.StackGetTexture

**Description:** read a value from the stack to use it later in the trigger branch

**> Input Ports:**

- **Exec** (Trigger)
- **Name** (String)

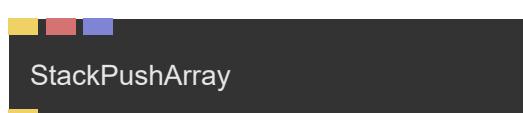
**< Output Ports:**

- **Next** (Trigger)
- **Texture** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.StackValues.StackGetTexture>

### 25.1.4 StackPushArray



**Full Name:** Ops.Data.StackValues.StackPushArray

**Description:** push a value on to the stack to use it later in the trigger branch

**> Input Ports:**

- **Trigger** (Trigger)
- **Name** (String)

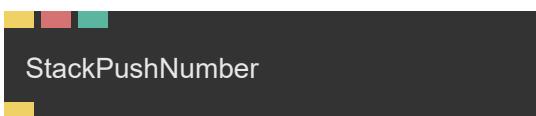
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.StackValues.StackPushArray>

## 25.1.5 StackPushNumber



**Full Name:** Ops.Data.StackValues.StackPushNumber

**Description:** push a value on to the stack to use it later in the trigger branch

### > Input Ports:

- **Exec** (Trigger)
- **Name** (String)
- **Value** (Number)

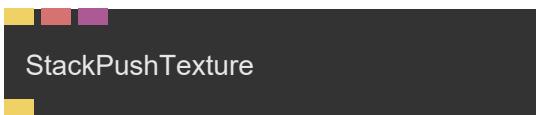
### < Output Ports:

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.StackValues.StackPushNumber>

## 25.1.6 StackPushTexture



**Full Name:** Ops.Data.StackValues.StackPushTexture

**Description:** push a value on to the stack to use it later in the trigger branch

### > Input Ports:

- **Trigger** (Trigger)
- **Name** (String)
- **Texture** (Object:Texture)

### < Output Ports:

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Data.StackValues.StackPushTexture>

# 26 Ops.Date

## 26.1 Ops.Date

### 26.1.1 DateAndTime



**Full Name:** Ops.Date.DateAndTime

**Description:** Returns current date and time and timestamp

### > Input Ports:

- **Update Rate** (Number)
- **How often the op should update the output** (in milliseconds)

### < Output Ports:

- **Year** (Number)
- **Month** (Number)
- **Day** (Number)
- **Hours** (Number)
- **Minutes** (Number)
- **Seconds** (Number)
- **Timestamp** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Date.DateAndTime>

### 26.1.2 DateCalc



**Full Name:** Ops.Date.DateCalc

**Description:** Perform date calculations

### > Input Ports:

- **Timestamp** (Number)
- **Difference** (Number: Integer)
- **Type Index** (Number: Integer)
- **Update** (Trigger)
- **Update time value** (not needed if an timestamp input is used)

**< Output Ports:**

- **Date** (Object)
- **Timestamp** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Date.DateCalc>

### 26.1.3 DateDifference



**Full Name:** Ops.Date.DateDifference

**Description:** Calculates the difference between two timestamps

**> Input Ports:**

- **Timestamp 1** (Number)
- **Timestamp 2** (Number)
- **Stop At 0** (Number: Boolean)

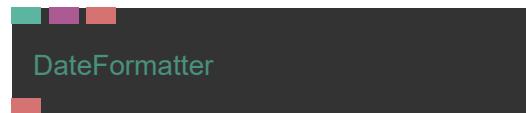
**< Output Ports:**

- **Year** (Number)
- **Month** (Number)
- **Day** (Number)
- **Hours** (Number)
- **Minutes** (Number)
- **Seconds** (Number)
- **Milliseconds** (Number)
- **Diff** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Date.DateDifference>

### 26.1.4 DateFormatter



**Full Name:** Ops.Date.DateFormatter

**Description:** String representation of a date

**> Input Ports:**

- **Timestamp** (Number)
- **Date** (Object)
- **Format** (String)

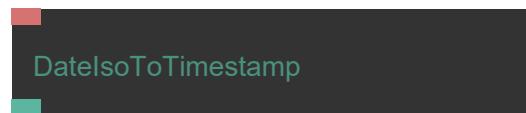
**< Output Ports:**

- **StringDate** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Date.DateFormatter>

### 26.1.5 DatelsoToTimestamp



**Full Name:** Ops.Date.DateIsoToTimestamp

**Description:** parses a date and time in iso format and outputs a millisecond timestamp

**> Input Ports:**

- **Datetime** (String)

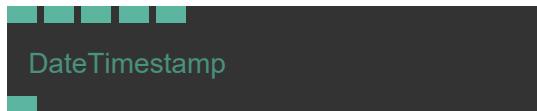
**< Output Ports:**

- **Timestamp** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Date.DatelsoToTimestamp>

## 26.1.6 DateTimestamp



**Full Name:** Ops.Date.DateTimestamp

**Description:** Calculates the timestamp of a date by year / month / day / hour / minute

**> Input Ports:**

- **Year** (Number: Integer)
- **Month** (Number: Integer)
- **Day** (Number: Integer)
- **Hour** (Number: Integer)
- **Minute** (Number: Integer)

**< Output Ports:**

- **Timestamp** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Date.DateTimestamp>

## 26.1.7 Milliseconds



**Full Name:** Ops.Date.Milliseconds

**Description:** Value since the time origin in milliseconds (performance.now())

**> Input Ports:**

- **Update** (Trigger)

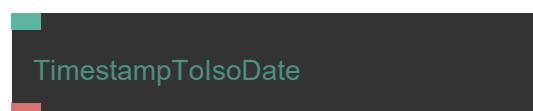
**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Date.Milliseconds>

## 26.1.8 TimestampToIsoDate



**Full Name:** Ops.Date.TimestampToIsoDate

**Description:** convert a timestamp to an ISO date string

**> Input Ports:**

- **Timestamp** (Number)

**< Output Ports:**

- **ISO Date** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Date.TimestampToIsoDate>

# 27 Ops.Debug

## 27.1 Ops.Debug

### 27.1.1 Console



**Full Name:** Ops.Debug.Console

**Description:** Shows console log output on the screen

#### > Input Ports:

- **Visible** (Number: Boolean)
- **Clear** (Trigger)

#### < Output Ports:

- **Element** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Debug.Console>

### 27.1.2 ConsoleLog



**Full Name:** Ops.Debug.ConsoleLog

**Description:** Log incoming values to the console/dev tools

#### > Input Ports:

- **Number** (Number)
- **String** (String)

#### < Output Ports:

- Visit *Ops.Debug.ConsoleLog documentation* for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Debug.ConsoleLog>

### 27.1.3 CrashOp



**Full Name:** Ops.Debug.CrashOp

**Description:** Crash the editor in many ways

#### > Input Ports:

- **Async Crash** (Trigger)
- **Undefined Crash** (Trigger)
- **Throw Exception** (Trigger)
- **Float** (Number)
- **Array Exception** (Trigger)
- **Promise Fail** (Trigger)
- **Shader Error** (Trigger)

#### < Output Ports:

- **Nan** (Number)
- **Infinity** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Debug.CrashOp>

### 27.1.4 GLLogErrors



**Full Name:** Ops.Debug.GLLogErrors

**Description:** execute glGetError after every gl command and log to browser console

#### > Input Ports:

- **Exec** (Trigger)
- **Limit Error Logs Num** (Number: Integer)
- **Stop Trigger After Limit** (Number: Boolean)

- Show Gl History (Trigger)

**< Output Ports:**

- Next (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Debug.GlLogErrors>

## 27.1.5 GlStates



**Full Name:** Ops.Debug.GlStates

**Description:** see current gl states and error message

**> Input Ports:**

- Update (Trigger)

**< Output Ports:**

- Next (Trigger)
- GlGetError (Number)
- Depthtest (Number)
- Stack Depthtest (Number)
- Depth Writing (Number)
- Stack Depth Writing (Number)
- DepthFunc (Number)
- Stack DepthFunc (Number)
- Blend (Number)
- Blend Stack (Number)
- Cull Mode (Number)
- Face Culling (Number)
- Is Shadowpass (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Debug.GlStates>

## 27.1.6 ProfileGL



**Full Name:** Ops.Debug.ProfileGL

**Description:** dump all gl commands of one frame to console

**> Input Ports:**

- Exec (Trigger)
- Debug One Frame (Trigger)

**< Output Ports:**

- Next (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Debug.ProfileGL>

## 27.1.7 StopWatch



**Full Name:** Ops.Debug.StopWatch

**Description:** Measure the time used to render all child nodes in milliseconds

**> Input Ports:**

- Exec (Trigger)

**< Output Ports:**

- Next (Trigger)
- Time Used (Number)
- Times (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Debug.StopWatch>

# 28 Ops.Devices

---

## 28.1 Ops.Devices

### 28.1.1 TouchGesture



**Full Name:** Ops.Devices.TouchGesture

**Description:** detect touch gestures like swipe and pan

#### > Input Ports:

- **Active** (Number: Boolean)
- **Vertical Swipe** (Number: Boolean)
- **Vertical Pan** (Number: Boolean)

#### < Output Ports:

- **Press** (Trigger)
- **Press Up** (Trigger)
- **Pan Left** (Trigger)
- **Pan Right** (Trigger)
- **Swipe Left** (Trigger)
- **Swipe Right** (Trigger)
- **Swipe Up** (Trigger)
- **Swipe Down** (Trigger)
- **Event** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.TouchGesture>

### 28.1.2 TouchScreen



**Full Name:** Ops.Devices.TouchScreen

**Description:** touch screen input: e.g. position of fingers

#### > Input Ports:

- **Disable Scaling** (Number: Boolean)
- **Disable Scroll** (Number: Boolean)
- **HDPI Coordinates** (Number: Boolean)
- **Active** (Number: Boolean)
- **Normalize Coordinates** (Number: Boolean)
- **Flip Y** (Number: Boolean)

#### < Output Ports:

- **Touched** (Number)
- **Fingers** (Number)
- **Finger 1 X** (Number)
- **Finger 1 Y** (Number)
- **Finger 1 Force** (Number)
- **Finger 2 X** (Number)
- **Finger 2 Y** (Number)
- **Finger 2 Force** (Number)
- **Events** (Array)
- **Touch Start** (Trigger)
- **Touch End** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.TouchScreen>

# 29 Ops.Devices.Browser

## 29.1 Ops.Devices.Browser

### 29.1.1 BrowserInfo\_v3



**Full Name:** Ops.Devices.Browser.BrowserInfo\_v3

**Description:** Reports the browser being used

**> Input Ports:**

- Visit *Ops.Devices.Browser.BrowserInfo\_v3 documentation* for input port details

**< Output Ports:**

- **Is Mobile** (booleanNumber)
- **Is Touchscreen** (booleanNumber)
- **Is IE** (booleanNumber)
- **Is Edge** (booleanNumber)
- **Is Chrome** (booleanNumber)
- **Is Firefox** (booleanNumber)
- **Is Opera** (booleanNumber)
- **Is Safari** (booleanNumber)
- **True if the browser is Safari** (iOS & macOS & OS X)
- **Is Windows** (booleanNumber)
- **Is Linux** (booleanNumber)
- **Is Mac** (booleanNumber)
- **Is IOS** (booleanNumber)
- **Is Android** (booleanNumber)
- **Is Electron** (booleanNumber)
- **Operating System** (String)
- **Browser Name** (String)
- **Browser Version** (String)
- **OS Version** (String)
- **Language** (String)

- **User Agent** (String)
- **Platform Object** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Devices.Browser.BrowserInfo\\_v3](https://cables.gl/op/Ops.Devices.Browser.BrowserInfo_v3)

### 29.1.2 ColorScheme



**Full Name:** Ops.Devices.Browser.ColorScheme

**Description:** Get light/dark color scheme preference of the browser

**> Input Ports:**

- Visit *Ops.Devices.Browser.ColorScheme documentation* for input port details

**< Output Ports:**

- **Color Scheme** (String)
- **Dark Mode** (booleanNumber)
- **Light Mode** (booleanNumber)
- **Supported** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Browser.ColorScheme>

### 29.1.3 History



**Full Name:** Ops.Devices.Browser.History

**Description:** Move back or forward in the browser navigation history

**> Input Ports:**

- **Back** (Trigger)
- **Forward** (Trigger)

**< Output Ports:**

- Visit `Ops.Devices.Browser.History` documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Browser.History>

## 29.1.4 JsExpression



**Full Name:** Ops.Devices.Browser.JsExpression

**Description:** evaluate a javascript expression

**> Input Ports:**

- **JS Expression** (String)

**< Output Ports:**

- **Result String** (String)
- **Result Number** (Number)
- **Result Array** (Array)
- **Result Object** (Object)
- **Error** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Browser.JsExpression>

## 29.1.5 JsMemory



**Full Name:** Ops.Devices.Browser.JsMemory

**Description:** browser js memory consumption

**> Input Ports:**

- **Update** (Trigger)

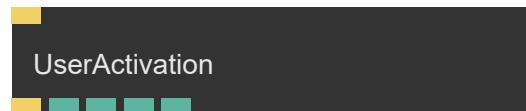
**< Output Ports:**

- **Used Heap Size** (Number)
- **Total Heap Size** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Browser.JsMemory>

## 29.1.6 UserActivation



**Full Name:** Ops.Devices.Browser.UserActivation

**Description:** detect if the user interacted with or activated the page

**> Input Ports:**

- **Update** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **User Is Or Was Active** (booleanNumber)
- **User Has Been Active** (booleanNumber)
- **User Is Active** (booleanNumber)
- **Supported** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Browser.UserActivation>

## 29.1.7 WebShare



**Full Name:** Ops.Devices.Browser.WebShare

**Description:** Opens a sharing dialog to share text and images

**> Input Ports:**

- **Text** (String)
- **URL** (String)
- **Base64 File** (String)
- **Data URL** (String)
- **Filetype** (String)

- **Filename** (String)
- **Share** (Trigger)

◀ **Output Ports:**

- **Status** (String)
- **Supported** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Browser.WebShare>

# 30 Ops.Devices.GamePad

---

## 30.1 Ops.Devices.GamePad

### 30.1.1 GamePad



**Full Name:** Ops.Devices.GamePad.GamePad

**Description:** Outputs the button states of a gamepad

▶ **Input Ports:**

- **GamePad Data** (Object)
- **Analog To Digital** (Number: Boolean)

◀ **Output Ports:**

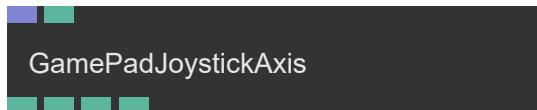
- **ID** (String)
- **Axes** (Array)
- **Pad Left** (booleanNumber)
- **Pad Right** (booleanNumber)
- **Pad Up** (booleanNumber)
- **Pad Down** (booleanNumber)
- **Button 1** (booleanNumber)
- **Button 2** (booleanNumber)
- **Button 3** (booleanNumber)
- **Button 4** (booleanNumber)
- **Left Shoulder** (Number)
- **Left Shoulder Bottom** (Number)
- **Right Shoulder** (Number)
- **Right Shoulder Bottom** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.GamePad.GamePad>

### 30.1.2 GamePadJoystickAxis

Docs: <https://cables.gl/op/Ops.Devices.GamePad.GamePads>



**Full Name:** Ops.Devices.GamePad.GamePadJoystickAxis

**Description:** get axis and angle of a joystick/thumbstick

**> Input Ports:**

- Axis (Array)
- Index (Number: Integer)

**< Output Ports:**

- X (Number)
- Y (Number)
- DeadZone (Number)
- Angle (Number)

**Example Patch:** Open in Editor

Docs: <https://cables.gl/op/Ops.Devices.GamePad.GamePadJoystickAxis>

### 30.1.3 GamePads



**Full Name:** Ops.Devices.GamePad.GamePads

**Description:** list connected gamepads - press a button to connect

**> Input Ports:**

- Exe (Trigger)

**< Output Ports:**

- Num Gamepads (Number)
- Pad 0 (Object)
- Pad 1 (Object)
- Pad 2 (Object)
- Pad 3 (Object)

**Example Patch:** Open in Editor

# 31 Ops.Devices.Keyboard

## 31.1 Ops.Devices.Keyboard

### 31.1.1 CursorKeys



**Full Name:** Ops.Devices.Keyboard.CursorKeys

**Description:** get the state of your keyboards arrow keys

#### > Input Ports:

- **Canvas Only** (Number: Boolean)
- **Cursor Keys** (Number: Boolean)
- **WASD** (Number: Boolean)
- **Active** (Number: Boolean)

#### < Output Ports:

- **Degree** (Number)
- **Up** (booleanNumber)
- **Up Pressed** (Trigger)
- **Down** (booleanNumber)
- **Down Pressed** (Trigger)
- **Left** (booleanNumber)
- **Left Pressed** (Trigger)
- **Right** (booleanNumber)
- **Right Pressed** (Trigger)
- **Any Button Pressed** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Keyboard.CursorKeys>

### 31.1.2 KeyPress\_v2



**Full Name:** Ops.Devices.Keyboard.KeyPress\_v2

**Description:** Triggers when a key is pressed

#### > Input Ports:

- **Area Index** (Number: Integer)
- **Prevent Default** (Number: Boolean)
- **Enabled** (Number: Boolean)

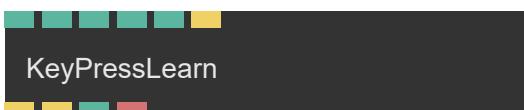
#### < Output Ports:

- **On Press** (Trigger)
- **Key Code** (Number)
- **Key** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Devices.Keyboard.KeyPress\\_v2](https://cables.gl/op/Ops.Devices.Keyboard.KeyPress_v2)

### 31.1.3 KeyPressLearn



**Full Name:** Ops.Devices.Keyboard.KeyPressLearn

**Description:** Triggers when certain key is pressed or released

#### > Input Ports:

- **Key Code** (Number: Integer)
- **Canvas Only** (Number: Boolean)
- **Mod Key Index** (Number: Integer)
- **Enabled** (Number: Boolean)
- **Prevent Default** (Number: Boolean)
- **Learn** (Trigger)

#### < Output Ports:

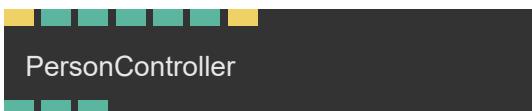
- **On Press** (Trigger)
- **On Release** (Trigger)

- **Pressed** (booleanNumber)
- **Key** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Keyboard.KeyPressLearn>

### 31.1.4 PersonController



**Full Name:** Ops.Devices.Keyboard.PersonController

**Description:** simple controller example op for game characters

#### > Input Ports:

- **Exe** (Trigger)
- **Speed** (Number)
- **North** (Number: Boolean)
- **East** (Number: Boolean)
- **South** (Number: Boolean)
- **West** (Number: Boolean)
- **Reset** (Trigger)

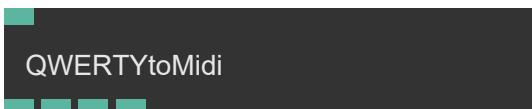
#### < Output Ports:

- **X** (Number)
- **Y** (Number)
- **Dir** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Keyboard.PersonController>

### 31.1.5 QWERTYtoMidi



**Full Name:** Ops.Devices.Keyboard.QWERTYtoMidi

**Description:** Emulates a MIDI keyboard using your regular keyboard

#### > Input Ports:

- **Canvas Only** (Number: Boolean)

#### < Output Ports:

- **Note Number** (Number)
- **Velocity** (Number)
- **Channel** (Number)
- **Command** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Keyboard.QWERTYtoMidi>

# 32 Ops.Devices.Midi

## 32.1 Ops.Devices.Midi

### 32.1.1 DeviceList



**Full Name:** Ops.Devices.Midi.DeviceList

**Description:** list of midi devices

#### > Input Ports:

- Visit *Ops.Devices.Midi.DeviceList documentation* for input port details

#### < Output Ports:

- **Num Devices** (Number)
- **Midi Support** (booleanNumber)
- **Device Names** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Midi.DeviceList>

### 32.1.2 MidiCC\_v3



**Full Name:** Ops.Devices.Midi.MidiCC\_v3

**Description:** read CC value from Midi controller

#### > Input Ports:

- **MIDI Event In** (Object)
- **MIDI Channel Index** (Number: Integer)
- **CC Index** (Number: Integer)
- **Speed** (Number)
- **Learn** (Trigger)

- **Clear** (Trigger)

#### < Output Ports:

- **CC Value Out** (Number)
- **Event** (Object)
- **Trigger Out** (Trigger)
- **CC Index Out** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Devices.Midi.MidiCC\\_v3](https://cables.gl/op/Ops.Devices.Midi.MidiCC_v3)

### 32.1.3 MidiCCOut\_v2



**Full Name:** Ops.Devices.Midi.MidiCCOut\_v2

**Description:** send MIDI CC data to a midi output

#### > Input Ports:

- **Send** (Trigger)
- **MIDI Channel Index** (Number: Integer)
- **CC Index** (Number: Integer)
- **CC Value** (Number: Integer)
- **Auto Send Value Change** (Number: Boolean)

#### < Output Ports:

- **MIDI Event Out** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Devices.Midi.MidiCCOut\\_v2](https://cables.gl/op/Ops.Devices.Midi.MidiCCOut_v2)

### 32.1.4 MidiChord3



**Full Name:** Ops.Devices.Midi.MidiChord3

**Description:** Map 3 midi notes to values

#### > Input Ports:

- **MIDI Event In** (Object)
- **MIDI Channel Index** (Number: Integer)
- **Note 1 Index** (Number: Integer)
- **Note 2 Index** (Number: Integer)
- **Note 3 Index** (Number: Integer)
- **Normalize Velocity Index** (Number: Integer)
- **Learn** (Trigger)
- **Reset** (Trigger)

#### < Output Ports:

- **MIDI Event Out** (Object)
- **Trigger Out** (Trigger)
- **Note Out 1** (Number)
- **Velocity 1** (Number)
- **Gate 1** (booleanNumber)
- **Note Out 2** (Number)
- **Velocity 2** (Number)
- **Gate 2** (booleanNumber)
- **Note Out 3** (Number)
- **Velocity 3** (Number)
- **Gate 3** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Midi.MidiChord3>

## 32.1.5 MidiClock



**Full Name:** Ops.Devices.Midi.MidiClock

**Description:** sends out midi clock signals as triggers

#### > Input Ports:

- **MIDI Event In** (Object)
- **Timing Index** (Number: Integer)

#### < Output Ports:

- **MIDI Event Out** (Object)

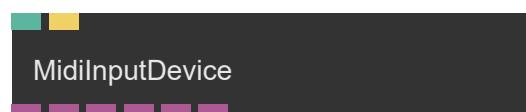
#### > Input Ports:

- **Tick Out** (Trigger)
- **Clock Start** (Trigger)
- **Clock Stop** (Trigger)
- **Clock Continue** (Trigger)
- **BPM** (Number)
- **Tick Duration** (Number)
- **Sub Tick** (Number)
- **current subtick** (value between 0 - 24)
- **outputs a trigger every bar** (dotted: 1.5 bars, triplet: full-note triplet)
- **outputs a trigger every half note** (dotted: trigger every 3/4, triplet: half-note triplet)
- **outputs a trigger every quarter note** (dotted: trigger every 3/8, triplet: quarter-note triplet)
- **outputs a trigger every eighth note** (dotted: trigger every 3/16, triplet: eighth-note triplet)
- **outputs a trigger every sixteenth note** (dotted: trigger every 3/32, triplet: sixteenth-note triplet)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Midi.MidiClock>

## 32.1.6 MidilInputDevice\_v2



**Full Name:** Ops.Devices.Midi.MidilInputDevice\_v2

**Description:** connect to MIDI device output port

#### > Input Ports:

- **Device Index** (Number: Integer)
- **Learn** (Trigger)

#### < Output Ports:

- **Event** (Object)
- **Note** (Object)
- **CC** (Object)
- **NRPN** (Object)
- **Program Change** (Object)
- **Clock** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Devices.Midi.MidiInputDevice\\_v2](https://cables.gl/op/Ops.Devices.Midi.MidiInputDevice_v2)

### 32.1.7 MidiMonitor



**Full Name:** Ops.Devices.Midi.MidiMonitor

**Description:** detailed information about Midi events being sent

**> Input Ports:**

- **Event** (Object)

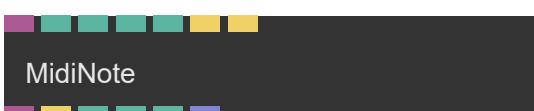
**< Output Ports:**

- **MIDI Event Out** (Object)
- **Trigger Out** (Trigger)
- **Device** (Number)
- **MIDI Channel** (Number)
- **Message Type** (Number)
- **the type of the message** (CC, Note, NRPN, Clock, ...)
- **Note** (Number)
- **Note Velocity** (Number)
- **CC Number** (Number)
- **CC Value** (Number)
- **Pitch Bend Value** (Number)
- **NRPN Number** (Number)
- **NRPN Value** (Number)
- **Program Change Value** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Midi.MidiMonitor>

### 32.1.8 MidiNote



**Full Name:** Ops.Devices.Midi.MidiNote

**Description:** Read a single midi note

**> Input Ports:**

- **MIDI Event In** (Object)
- **MIDI Channel Index** (Number: Integer)
- **Note Index** (Number: Integer)
- **Normalize Velocity Index** (Number: Integer)
- **Toggle Gate** (Number: Boolean)
- **Learn** (Trigger)
- **Clear** (Trigger)

**< Output Ports:**

- **MIDI Event Out** (Object)
- **Trigger Out** (Trigger)
- **Current Note** (Number)
- **Velocity** (Number)
- **Gate** (booleanNumber)
- **Velocity Array** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Midi.MidiNote>

### 32.1.9 MidiNoteFilter



**Full Name:** Ops.Devices.Midi.MidiNoteFilter

**Description:** Only read a range of notes (e.g. C1 to C2)

**> Input Ports:**

- **MIDI Event** (Object)
- **MIDI Channel Index** (Number: Integer)
- **Note Start Index** (Number: Integer)
- **Note End Index** (Number: Integer)
- **Normalize Velocity Index** (Number: Integer)
- **Learn** (Trigger)
- **Reset** (Trigger)

**< Output Ports:**

- **Event** (Object)
- **Trigger Out** (Trigger)
- **Current Note** (Number)
- **Velocity** (Number)
- **Gate** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Midi.MidiNoteFilter>

### 32.1.10 MidiNoteOut



**Full Name:** Ops.Devices.Midi.MidiNoteOut

**Description:** send midi note data to a midi output

**> Input Ports:**

- **MIDI Channel Index** (Number: Integer)
- **Note Index** (Number: Integer)
- **Note Number** (Number: Integer)
- **Velocity** (Number: Integer)
- **Min In Velocity** (Number)
- **Max In Velocity** (Number)
- **Velocity Array In** (Array)

**< Output Ports:**

- **MIDI Event Out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Midi.MidiNoteOut>

### 32.1.11 MidiNRPN



**Full Name:** Ops.Devices.Midi.MidiNRPN

**Description:** read NRPN value from controller

**> Input Ports:**

- **MIDI Event In** (Object)
- **MIDI Channel Index** (Number: Integer)
- **NRPN Index** (Number: Integer)
- **Normalize Index** (Number: Integer)
- **Learn** (Trigger)
- **Clear** (Trigger)

**< Output Ports:**

- **MIDI Event Out** (Object)
- **Trigger Out** (Trigger)
- **NRPN Index Out** (Number)
- **NRPN Value** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Midi.MidiNRPN>

### 32.1.12 MidiNRPNOout



**Full Name:** Ops.Devices.Midi.MidiNRPNOout

**Description:** send midi NRPN data to a midi output

**> Input Ports:**

- **MIDI Channel Index** (Number: Integer)
- **NRPN Index** (Number: Integer)
- **NRPN Value** (Number: Integer)
- **Min In Value** (Number)
- **Max In Value** (Number)

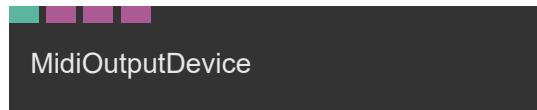
**< Output Ports:**

- **MIDI Event Out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Midi.MidiNRPNOout>

## 32.1.13 MidiOutputDevice



**Full Name:** Ops.Devices.Midi.MidiOutputDevice

**Description:** Connect to MIDI device input port

**> Input Ports:**

- **Device Index** (Number: Integer)
- **Note** (Object)
- **CC** (Object)
- **NRPN** (Object)

**< Output Ports:**

- Visit *Ops.Devices.Midi.MidiOutputDevice* documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Midi.MidiOutputDevice>

**Docs:** <https://cables.gl/op/Ops.Devices.Midi.MidiTranspose>

## 32.1.14 MidiTranspose



**Full Name:** Ops.Devices.Midi.MidiTranspose

**Description:** transpose incoming midi notes

**> Input Ports:**

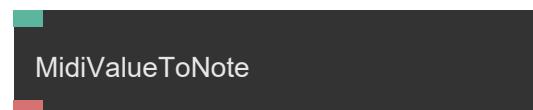
- **MIDI Event In** (Object)
- **MIDI Channel Index** (Number: Integer)
- **Transpose Amount** (Number: Integer)
- **Learn** (Trigger)

**< Output Ports:**

- **MIDI Event Out** (Object)
- **Trigger Out** (Trigger)

**Example Patch:** Open in Editor

## 32.1.15 MidiValueToNote\_v2



**Full Name:** Ops.Devices.Midi.MidiValueToNote\_v2

**Description:** Converts a MIDI value to a note string

**> Input Ports:**

- **Midi Value** (Number)

**< Output Ports:**

- **Note** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Devices.Midi.MidiValueToNote\\_v2](https://cables.gl/op/Ops.Devices.Midi.MidiValueToNote_v2)

# 33 Ops.Devices.Mobile

## 33.1 Ops.Devices.Mobile

### 33.1.1 DeviceVibrate



**Full Name:** Ops.Devices.Mobile.DeviceVibrate

**Description:** vibrating a mobile device

**> Input Ports:**

- **Vibrate** (Trigger)

**< Output Ports:**

- **Supported** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Mobile.DeviceVibrate>

### 33.1.2 GeoLocation



**Full Name:** Ops.Devices.Mobile.GeoLocation

**Description:** tries to get the geo coordinates from the mobile device/browser

**> Input Ports:**

- Visit *Ops.Devices.Mobile.GeoLocation documentation* for input port details

**< Output Ports:**

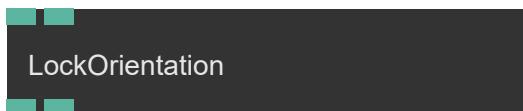
- **Browser Support** (booleanNumber)
- **Latitude** (Number)
- **Longitude** (Number)

- **Data** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Mobile.GeoLocation>

### 33.1.3 LockOrientation



**Full Name:** Ops.Devices.Mobile.LockOrientation

**Description:** locks orientation to landscape or portrait mode

**> Input Ports:**

- **Portrait** (Number: Boolean)
- **Landscape** (Number: Boolean)

**< Output Ports:**

- **Supported** (Number)
- **Locked** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Mobile.LockOrientation>

### 33.1.4 MotionSensor\_v2



**Full Name:** Ops.Devices.Mobile.MotionSensor\_v2

**Description:** get values from the device motion sensor mobile

**> Input Ports:**

- **Mul Orientation** (Number)
- **Request Permissions** (Trigger)

**< Output Ports:**

- **Orientation Alpha** (Number)
- **Orientation Beta** (Number)
- **Orientation Gamma** (Number)

- **Acceleration X** (Number)
- **Acceleration Y** (Number)
- **Acceleration Z** (Number)
- **Acceleration X No Gravity** (Number)
- **Acceleration Y No Gravity** (Number)
- **Acceleration Z No Gravity** (Number)
- **Rotation Rate Alpha** (Number)
- **Rotation Rate Beta** (Number)
- **Rotation Rate Gamma** (Number)
- **Permissions** (String)
- **Object** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Devices.Mobile.MotionSensor\\_v2](https://cables.gl/op/Ops.Devices.Mobile.MotionSensor_v2)

### 33.1.5 Pinch



**Full Name:** Ops.Devices.Mobile.Pinch

**Description:** detect two finger pinch gestures on touchscreens

**> Input Ports:**

- **Enabled** (Number: Boolean)
- **Min Scale** (Number)
- **Max Scale** (Number)
- **Reset Scale** (Trigger)
- **Limit** (Number: Boolean)

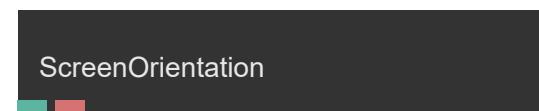
**< Output Ports:**

- **Scale** (Number)
- **Event Details** (Object)
- **Delta** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Mobile.Pinch>

### 33.1.6 ScreenOrientation\_v2



**Full Name:** Ops.Devices.Mobile.ScreenOrientation\_v2

**Description:** get orientation of the physical screen

**> Input Ports:**

- Visit *Ops.Devices.Mobile.ScreenOrientation\_v2 documentation* for input port details

**< Output Ports:**

- **Angle** (Number)
- **Type** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Devices.Mobile.ScreenOrientation\\_v2](https://cables.gl/op/Ops.Devices.Mobile.ScreenOrientation_v2)

### 33.1.7 ShakeGesture



**Full Name:** Ops.Devices.Mobile.ShakeGesture

**Description:** Reads the accelerometer data from a mobile device

**> Input Ports:**

- Visit *Ops.Devices.Mobile.ShakeGesture documentation* for input port details

**< Output Ports:**

- **Acceleration X** (Number)
- **Acceleration Y** (Number)
- **Acceleration Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Mobile.ShakeGesture>

# 34 Ops.Devices.Mouse

---

## 34.1 Ops.Devices.Mouse

### 34.1.1 Mouse\_v4



**Full Name:** Ops.Devices.Mouse.Mouse\_v4

**Description:** Get mouse/touchscreen/pointer coordinates and events

#### > Input Ports:

- **Area Index** (Number: Integer)
- **Flip Y** (Number: Boolean)
- **Right Click Prevent Default** (Number: Boolean)
- **Passive Events** (Number: Boolean)
- **Element** (Object)
- **Active** (Number: Boolean)

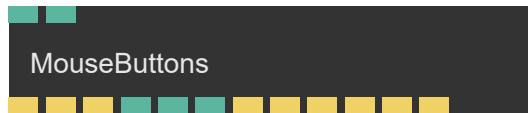
#### < Output Ports:

- **X** (Number)
- **Y** (Number)
- **Click** (Trigger)
- **Click Right** (Trigger)
- **Button Is Down** (booleanNumber)
- **Mouse Is Hovering** (booleanNumber)
- **Movement X** (Number)
- **Movement Y** (Number)
- **Event** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Devices.Mouse.Mouse\\_v4](https://cables.gl/op/Ops.Devices.Mouse.Mouse_v4)

### 34.1.2 MouseButtons



**MouseButtons**

**Full Name:** Ops.Devices.Mouse.MouseButtons

**Description:** Get the state of mouse buttons

#### > Input Ports:

- **Area Index** (Number: Integer)
- **Active** (Number: Boolean)

#### < Output Ports:

- **Click Left** (Trigger)
- **Click Right** (Trigger)
- **Double Click** (Trigger)
- **Button Pressed Left** (Number)
- **Button Pressed Middle** (Number)
- **Button Pressed Right** (Number)
- **Mouse Down Left** (Trigger)
- **Mouse Down Middle** (Trigger)
- **Mouse Down Right** (Trigger)
- **Mouse Up Left** (Trigger)
- **Mouse Up Middle** (Trigger)
- **Mouse Up Right** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Mouse.MouseButtons>

### 34.1.3 MouseDrag



**MouseDrag**

**Full Name:** Ops.Devices.Mouse.MouseDrag

**Description:** get delta of mouse position while dragging

#### > Input Ports:

- **Active** (Number: Boolean)

- **Speed** (Number)
- **Input Type Index** (Number: Integer)
- **Area Index** (Number: Integer)

**< Output Ports:**

- **Delta X** (Number)
- **Delta Y** (Number)
- **Is Dragging** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Mouse.MouseDrag>

### 34.1.5 PointerLock

#### 34.1.4 MouseWheel\_v2



**Full Name:** Ops.Devices.Mouse.MouseWheel\_v2

**Description:** outputs delta values controlled by the mousewheel (scroll, zoom)

**> Input Ports:**

- **Speed** (Number)
- **Prevent Scroll** (Number: Boolean)
- **Flip Direction** (Number: Boolean)
- **Simple Delta** (Number: Boolean)
- **Active** (Number: Boolean)

**< Output Ports:**

- **Delta** (Number)
- **Delta X** (Number)
- **Browser Event Delta** (Number)
- **Wheel Action** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Devices.Mouse.MouseWheel\\_v2](https://cables.gl/op/Ops.Devices.Mouse.MouseWheel_v2)



**Full Name:** Ops.Devices.Mouse.PointerLock

**Description:** locks the pointer to the canvas and hides the cursor

**> Input Ports:**

- **Render** (Trigger)
- **Start** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **Supported** (booleanNumber)
- **Is Locked** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.Mouse.PointerLock>

# 35 Ops.Devices.WebXr.Vr

## 35.1 Ops.Devices.WebXr.Vr

### 35.1.1 Vr



**Full Name:** Ops.Devices.WebXr.Vr.Vr

**Description:** rendering on webxr virtual reality immersive devices

#### > Input Ports:

- **Mainloop** (Trigger)
- **Stop** (Trigger)
- **Show Button** (Number: Boolean)
- **Button Style** (String)
- **Render To Texture** (Number: Boolean)
- **Shader** (Object:Shader)

#### < Output Ports:

- **Next** (Trigger)
- **Render After Eyes** (Trigger)
- **Viewer Pose** (Object)
- **Eye Index** (Number)
- **VR Support** (booleanNumber)
- **Matrix** (Array)
- **DOM Overlay Ele** (Object)
- **In Session** (booleanNumber)
- **Ms Per Eye** (Array)
- **Texture** (Object)
- **Texture Depth** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.WebXr.Vr.Vr>

### 35.1.2 VrController



**Full Name:** Ops.Devices.WebXr.Vr.VrController

**Description:** tracking of vr hand controller

#### > Input Ports:

- **Update** (Trigger)
- **Handedness Index** (Number: Integer)

#### < Output Ports:

- **Next** (Trigger)
- **Axis 1** (Number)
- **Axis 2** (Number)
- **Axis 3** (Number)
- **Axis 4** (Number)
- **Button 1 Pressed** (Number)
- **Button 2 Pressed** (Number)
- **Button 3 Pressed** (Number)
- **Button 4 Pressed** (Number)
- **Button 5 Pressed** (Number)
- **Button 6 Pressed** (Number)
- **Button 7 Pressed** (Number)
- **Button 1 Touched** (Number)
- **Button 2 Touched** (Number)
- **Button 3 Touched** (Number)
- **Button 4 Touched** (Number)
- **Button 5 Touched** (Number)
- **Button 6 Touched** (Number)
- **Button 7 Touched** (Number)
- **Position X** (Number)
- **Position Y** (Number)
- **Position Z** (Number)
- **Gamepad Values** (Object)
- **Transformed Position** (Trigger)
- **Found** (Number)

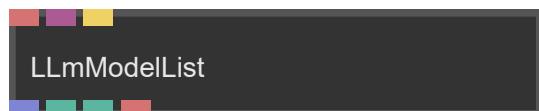
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Devices.WebXr.Vr.VrController>

# 36 Ops.Extension.Ai

## 36.1 Ops.Extension.Ai

### 36.1.1 LLmModelList



**Full Name:** Ops.Extension.Ai.LLmModelList

**Description:** Visit documentation for details

#### > Input Ports:

- **String1** (String)
- **Headers** (Object)
- **Reload** (Trigger)

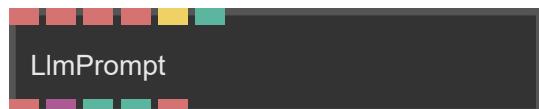
#### < Output Ports:

- **Z2gtag4y7** (Array)
- **Jcju8npa2** (booleanNumber)
- **Ozg9pnd1z** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Ai.LLmModelList>

### 36.1.2 LlmPrompt



**Full Name:** Ops.Extension.Ai.LlmPrompt

**Description:** Visit documentation for details

#### > Input Ports:

- **Prompt** (String)
- **Value** (String)
- **API URL** (String)

- **Authentication** (String)
- **Run** (Trigger)
- **Auto Request** (Number: Boolean)

#### < Output Ports:

- **I4feefw9n** (Object)
- **Klu6r35ga** (booleanNumber)
- **Xs18z73z0** (booleanNumber)
- **Aosval1gx** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Ai.LlmPrompt>

# 37 Ops.Extension.AmmoPhysics

## 37.1 Ops.Extension.AmmoPhysics

### 37.1.1 AmmoBody



**Full Name:** Ops.Extension.AmmoPhysics.AmmoBody

**Description:** Create a physics body/collision shape using any geometry or select a shape

#### > Input Ports:

- **Update** (Trigger)
- **Name** (String)
- **Mass** (Number)
- **Friction** (Number)
- **Rolling Friction** (Number)
- **Restitution** (Number)
- **Shape Index** (Number: Integer)
- **Geometry** (Object: Geometry)
- **Simplify Max Triangles** (Number: Integer)
- **Radius** (Number)
- **Size X** (Number)
- **Size Y** (Number)
- **Size Z** (Number)
- **Positions** (Array)
- **Append Index To Name** (Number: Boolean)
- **Never Deactivate** (Number: Boolean)
- **Ghost Object** (Number: Boolean)
- **Active** (Number: Boolean)
- **Reset** (Trigger)
- **Activate** (Trigger)

#### < Output Ports:

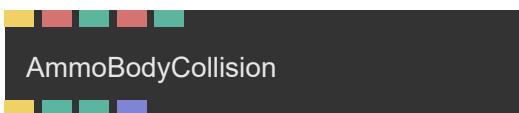
- **Next** (Trigger)

- **Ray Hit** (booleanNumber)
- **Transformed** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.AmmoPhysics.AmmoBody>

### 37.1.2 AmmoBodyCollision



**Full Name:** Ops.Extension.AmmoPhysics.AmmoBodyCollision

**Description:** Check if physics bodies are colliding

#### > Input Ports:

- **Update** (Trigger)
- **Name 1** (String)
- **Match Name 1 Index** (Number: Integer)
- **Name 2** (String)
- **name of physics object** (optional)
- **Match Name 2 Index** (Number: Integer)
- **match name 2** (if set)

#### < Output Ports:

- **Next** (Trigger)
- **Colliding** (Number)
- **collision detected** (Boolean)
- **Num Collisions** (Number)
- **Collisions** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.AmmoPhysics.AmmoBodyCollision>

### 37.1.3 AmmoCharacter



**Full Name:** Ops.Extension.AmmoPhysics.AmmoCharacter

**Description:** Control and move a character in a physics environment

**> Input Ports:**

- **Update** (Trigger)
- **Radius** (Number)
- **View Index** (Number: Integer)
- **Height** (Number)
- **Mass** (Number)
- **Name** (String)
- **Activate** (Trigger)
- **Move X-** (Number: Boolean)
- **Move Y-** (Number: Boolean)
- **Move Z-** (Number: Boolean)
- **Dir X** (Number)
- **X axis rotation value** (from AmmoCharacterFpsCamera for example)
- **Dir Y** (Number)
- **Y axis rotation value** (from AmmoCharacterFpsCamera for example)
- **Dir Z** (Number)
- **Z axis rotation value** (from AmmoCharacterFpsCamera for example)
- **Set Pos X** (Number)
- **Set Pos Y** (Number)
- **Set Pos Z** (Number)
- **Reset** (Trigger)
- **Speed** (Number)
- **Add Velocity Y** (Number)

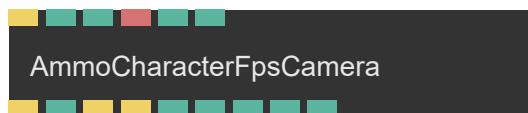
**< Output Ports:**

- **Next** (Trigger)
- **Position X** (Number)
- **Position Y** (Number)
- **Position Z** (Number)
- **Transformed** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.AmmoPhysics.AmmoCharacter>

### 37.1.4 AmmoCharacterFpsCamera



**Full Name:** Ops.Extension.AmmoPhysics.AmmoCharacterFpsCamera

**Description:** First person camera to use with AmmoCharacter

**> Input Ports:**

- **Render** (Trigger)
- **Enable Pointer Lock** (Number: Boolean)
- **Height** (Number)
- **Character Name** (String)
- **Mouse Speed** (Number)
- **Active** (Number: Boolean)

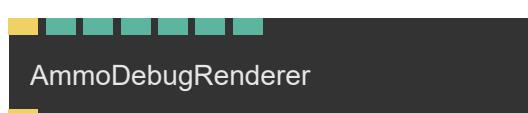
**< Output Ports:**

- **Trigger** (Trigger)
- **IsLocked** (Number)
- **has the mouse cursor been locked** (Boolean)
- **Mouse Left** (Trigger)
- **Mouse Right** (Trigger)
- **Dir X** (Number)
- **Dir Y** (Number)
- **Dir Z** (Number)
- **Rot X** (Number)
- **Rot Y** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.AmmoPhysics.AmmoCharacterFpsCamera>

### 37.1.5 AmmoDebugRenderer



**Full Name:** Ops.Extension.AmmoPhysics.AmmoDebugRenderer

**Description:** Visualize the physical bodies as lines and points

#### > Input Ports:

- **Render** (Trigger)
- **Draw Wireframe** (Number: Boolean)
- **Draw AABB** (Number: Boolean)
- **Draw Contact Points** (Number: Boolean)
- **Draw Constraints** (Number: Boolean)
- **Depth** (Number: Boolean)
- **Active** (Number: Boolean)

#### < Output Ports:

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.AmmoPhysics.AmmoDebugRendrer>

## 37.1.6 AmmoEmitter



**Full Name:** Ops.Extension.AmmoPhysics.AmmoEmitter

**Description:** Emit Ammo physics bodies by triggering

#### > Input Ports:

- **Exec** (Trigger)
- **Limit Bodies** (Number: Integer)
- **Radius** (Number)
- **Mass** (Number)
- **Add Index To Name** (Number: Boolean)
- **Name** (String)
- **Friction** (Number)
- **Rolling Friction** (Number)
- **Restitution** (Number)
- **Dir X** (Number)
- **Dir Y** (Number)
- **Dir Z** (Number)
- **Speed** (Number)
- **Spawn One** (Trigger)
- **Remove All** (Trigger)

- **Activate All** (Trigger)

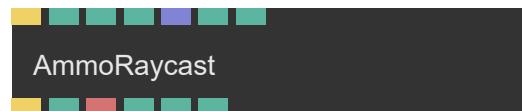
#### < Output Ports:

- **Next** (Trigger)
- **Total Bodies** (Number)
- **Positions** (Array)
- **Rotations Quats** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.AmmoPhysics.AmmoEmitter>

## 37.1.7 AmmoRaycast



**Full Name:** Ops.Extension.AmmoPhysics.AmmoRaycast

**Description:** Cast a ray and detect colliding bodies

#### > Input Ports:

- **Update** (Trigger)
- **Screen X** (Number)
- **Normalize screencoordinates on X Axis** (0-1)
- **Screen Y** (Number)
- **Normalize screencoordinates on Y Axis** (0-1)
- **Ray Points** (Array)
- **Active** (Number: Boolean)
- **Change Cursor** (Number: Boolean)

#### < Output Ports:

- **Next** (Trigger)
- **Has Hit** (booleanNumber)
- **Hit Body Name** (String)
- **Hit X** (Number)
- **Hit Y** (Number)
- **Hit Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.AmmoPhysics.AmmoRaycast>

### 37.1.8 AmmoWorld



AmmoWorld

**Full Name:** Ops.Extension.AmmoPhysics.AmmoWorld

**Description:** Simulate physical world

#### > Input Ports:

- **Update** (Trigger)
- **Simulate** (Number: Boolean)
- **Auto Remove Inactive** (Number: Boolean)
- **Gravity X** (Number)
- **Gravity Y** (Number)
- **Gravity Z** (Number)
- **Activate All** (Trigger)
- **Reset** (Trigger)

#### < Output Ports:

- **Next** (Trigger)
- **Total Bodies** (Number)
- **Debug Points** (Array)
- **Bodies Meta** (Array)
- **Collisions** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.AmmoPhysics.AmmoWorld>

- **Filter Meshes** (String)
- **Mass Kg** (Number)
- **Active** (Number: Boolean)

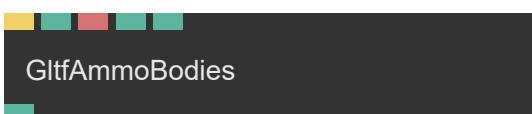
#### < Output Ports:

- **Meshes** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.AmmoPhysics.GltfAmmoBodies>

### 37.1.9 GltfAmmoBodies



GltfAmmoBodies

**Full Name:** Ops.Extension.AmmoPhysics.GltfAmmoBodies

**Description:** Create physics bodies from a GLTF File

#### > Input Ports:

- **Exec** (Trigger)
- **Shape Index** (Number: Integer)

## 38 Ops.Extension.DetectGpu

---

### 38.1 Ops.Extension.DetectGpu

#### 38.1.1 DetectGPU



**Full Name:** Ops.Extension.DetectGpu.DetectGPU

**Description:** Use the detect-gpu library to assess performance of the client running the patch

**> Input Ports:**

- **Run** (Trigger)

**< Output Ports:**

- **Finished** (Trigger)
- **Tier** (Number)
- **Is Mobile** (booleanNumber)
- **GPU Name** (String)
- **FPS** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.DetectGpu.DetectGPU>

## 39 Ops.Extension.ECharts

---

### 39.1 Ops.Extension.ECharts

#### 39.1.1 ECharts



**Full Name:** Ops.Extension.ECharts.ECharts

**Description:** wrapper for echarts-library

**> Input Ports:**

- **Create** (Trigger)
- **Parent DOM Element** (Object)
- **Id** (String)
- **Width** (Number: Integer)
- **Height** (Number: Integer)
- **Chart Object** (Object)
- **Merge Options** (Object)
- **Renderer Index** (Number: Integer)
- **Renderer** (String)
- **Theme Index** (Number: Integer)
- **Theme** (String)
- **Custom Theme Obj** (Object)
- **Init Extra Options** (Object)
- **Style** (Number: String)
- **Visible** (Number: Boolean)

**< Output Ports:**

- **DOM Element** (Object)
- **ECharts Instance** (Object)
- **Chart Updated** (Trigger)
- **Theme Changed** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.ECharts.ECharts>

## 39.1.2 EChartsEvent



**Full Name:** Ops.Extension.ECharts.EChartsEvent

**Description:** capture echart-library-events

**> Input Ports:**

- **ECharts Instance** (Object)
- **Event Name** (String)
- **Query String** (String)
- **Query Object** (Object)
- **Refresh Event Binding** (Trigger)

**< Output Ports:**

- **Out Chart** (Object)
- **Trigger** (Trigger)
- **Event Params** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.ECharts.EChartsEvent>

## 40 Ops.Extension.FxHash

### 40.1 Ops.Extension.FxHash

#### 40.1.1 FxHash



**Full Name:** Ops.Extension.FxHash.FxHash

**Description:** FxHash simulator / generator of seeded random numbers

**> Input Ports:**

- **Hash** (String)
- **Randomize Hash** (Trigger)

**< Output Ports:**

- **Fxhash** (String)
- **Fxrnd 1** (Number)
- **Fxrnd 2** (Number)
- **Fxrnd 3** (Number)
- **Fxrnd 4** (Number)
- **Random Numbers** (Array)
- **Fxhash Environment** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.FxHash.FxHash>

# 41 Ops.Extension.GlParticles

## 41.1 Ops.Extension.GlParticles

### 41.1.1 VelocityBoundaries



**Full Name:** Ops.Extension.GlParticles.VelocityBoundaries

**Description:** Visit documentation for details

#### > Input Ports:

- **Render** (Trigger)
- **Area Index** (Number: Integer)
- **Method Index** (Number: Integer)
- **Invert Area** (Number: Boolean)
- **Strength** (Number)
- **Size** (Number)
- **Falloff** (Number)
- **Bounciness** (Number)
- **Collision Fade** (Number)
- **Dir Randomness** (Number)
- **InForceOutwards** (Number)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Velocity Dir X** (Number)
- **Velocity Dir Y** (Number)
- **Velocity Dir Z** (Number)
- **Size X** (Number)
- **Size Y** (Number)
- **Size Z** (Number)
- **Multiply** (Object:Texture)
- **Age Start** (Number)
- **Age End** (Number)
- **Age Fade** (Number)

#### < Output Ports:

- **Trigger** (Trigger)
- **Velocity** (Object)
- **Collision** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.GlParticles.VelocityBoundaries>

# 42 Ops.Extension.HtmlElementArray

## 42.1 Ops.Extension.HtmlElementArray

### 42.1.1 DivElements



**Full Name:** Ops.Extension.HtmlElementArray.DivElements

**Description:** create an array of div elements

#### > Input Ports:

- **Class** (String)
- **Parent** (Object:Element)
- **Num** (Number: Integer)
- **Active** (Number: Boolean)
- **Text** (Array)
- **Reset Hover** (Trigger)

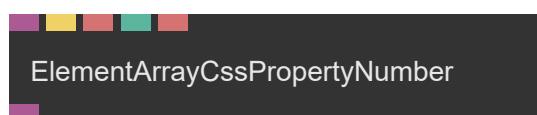
#### < Output Ports:

- **Elements** (Array)
- **Index Clicked** (Number)
- **Element Clicked** (Trigger)
- **Pointer Up** (Trigger)
- **Index Hovered** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.HtmlElementArray.DivElements>

### 42.1.2 ElementArrayCssPropertyName



**Full Name:** Ops.Extension.HtmlElementArray.ElementArrayCssPropertyName

**Description:** Set css style properties of a html element

#### > Input Ports:

- **Element** (Object)
- **Update** (Trigger)
- **Property** (String)
- **Value** (Number)
- **Value Suffix** (String)

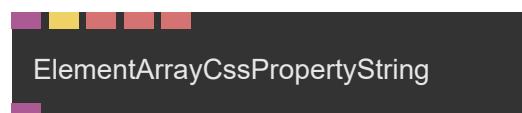
#### < Output Ports:

- **HTML Element** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.HtmlElementArray.ElementArrayCssPropertyName>

### 42.1.3 ElementArrayCssPropertyString



**Full Name:** Ops.Extension.HtmlElementArray.ElementArrayCssPropertyString

**Description:** set css properties

#### > Input Ports:

- **Element** (Object)
- **Update** (Trigger)
- **Property** (String)
- **Value** (String)
- **Value Suffix** (String)

#### < Output Ports:

- **HTML Element** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.HtmlElementArray.ElementArrayCssPropertyString>

## 43 Ops.Extension.HtmlToTexture

### 43.1 Ops.Extension.HtmlToTexture

#### 43.1.1 HtmlToTexture



**Full Name:** Ops.Extension.HtmlToTexture.HtmlToTexture

**Description:** Visit documentation for details

##### > Input Ports:

- **Element** (Object:Element)
- **Update** (Trigger)

##### < Output Ports:

- **Image DataUrl** (String)
- **Progress** (Number)
- **Finished** (Trigger)

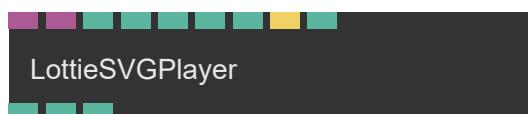
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.HtmlToTexture.HtmlToTexture>

## 44 Ops.Extension.Lottie

### 44.1 Ops.Extension.Lottie

#### 44.1.1 LottieSVGPlayer



**Full Name:** Ops.Extension.Lottie.LottieSVGPlayer

**Description:** Play Bodymovin/Lottie animations as SVG in a HTML element

##### > Input Ports:

- **HTML Element** (Object:Element)
- **JSON Data** (Object)
- **Render Frame** (Number)
- **Loop** (Number: Boolean)
- **Play** (Number: Boolean)
- **Play Backward** (Number: Boolean)
- **Rewind** (Trigger)
- **Active** (Number: Boolean)

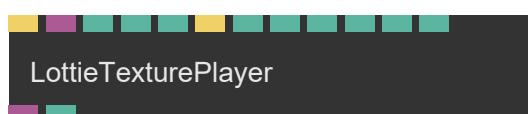
##### < Output Ports:

- **Completed** (booleanNumber)
- **Progress** (Number)
- **Total Frames** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Lottie.LottieSVGPlayer>

#### 44.1.2 LottieTexturePlayer\_v2



**Full Name:** Ops.Extension.Lottie.LottieTexturePlayer\_v2

**Description:** Play a Lottie animation in a texture

**> Input Ports:**

- **Exe** (Trigger)
- **JSON Data** (Object)
- **Play Mode Index** (Number: Integer)
- **Frame** (Number)
- **Play** (Number: Boolean)
- **Rewind** (Trigger)
- **Speed** (Number)
- **Texture Width** (Number: Integer)
- **Texture Height** (Number: Integer)
- **Filter Index** (Number: Integer)
- **Wrap Index** (Number: Integer)
- **Scale Index** (Number: Integer)

**< Output Ports:**

- **Texture** (Object)
- **Total Frames** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Extension.Lottie.LottieTexturePlayer\\_v2](https://cables.gl/op/Ops.Extension.Lottie.LottieTexturePlayer_v2)

## 45 Ops.Extension.LSystem

### 45.1 Ops.Extension.LSystem

#### 45.1.1 Lsystem\_v2



**Full Name:** Ops.Extension.LSystem.Lsystem\_v2

**Description:** Lsystem generator

**> Input Ports:**

- **Trigger** (Trigger)
- **Iterations** (Number: Integer)
- **Step Length** (Number)
- **Step Scale Multiplier** (Number)
- **Default Angle** (Number)
- **Rotation Multiplier** (Number)
- **Random Seed** (Number)
- **Random Strength** (Number)

**< Output Ports:**

- **Out Trigger Geometry** (Trigger)
- **Points Out** (Array)
- **Max Size** (Number)
- **Final Generated String** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Extension.LSystem.Lsystem\\_v2](https://cables.gl/op/Ops.Extension.LSystem.Lsystem_v2)

# 46 Ops.Extension.Mediapipe

## 46.1 Ops.Extension.Mediapipe

### 46.1.1 FaceMesh



**Full Name:** Ops.Extension.Mediapipe.FaceMesh

**Description:** Generate an animated geometry from MpFaceTracking Point Coordinates

**> Input Ports:**

- **Geom** (Object)
- **Points** (Array)

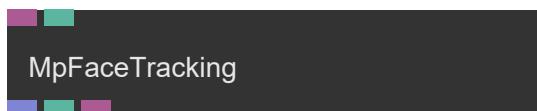
**< Output Ports:**

- **Result Geom** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Mediapipe.FaceMesh>

### 46.1.2 MpFaceTracking



**Full Name:** Ops.Extension.Mediapipe.MpFaceTracking

**Description:** Get face mesh from webcam/video using mediapipe library

**> Input Ports:**

- **Element** (Object)
- **Refine LandMarks** (Number: Boolean)

**< Output Ports:**

- **Points** (Array)

- **Found** (Number)
- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Mediapipe.MpFaceTracking>

### 46.1.3 MpHand



**Full Name:** Ops.Extension.Mediapipe.MpHand

**Description:** Get points and lines for left/right hand from mediapipe

**> Input Ports:**

- **Hands Result** (Object)
- **Hand Index** (Number: Integer)
- **Min Score** (Number)

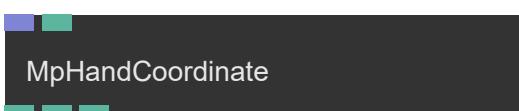
**< Output Ports:**

- **Points** (Array)
- **Lines** (Array)
- **Data** (Object)
- **Found Hand** (Number)
- **Score** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Mediapipe.MpHand>

### 46.1.4 MpHandCoordinate



**Full Name:** Ops.Extension.Mediapipe.MpHandCoordinate

**Description:** Get individual coordinates of fingers or wrist from an array of mediapipe data

**> Input Ports:**

- **Hand Points** (Array)
- **Joint Index** (Number: Integer)

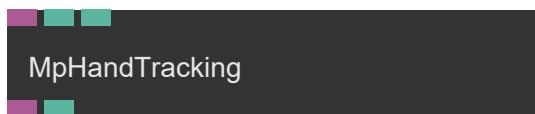
**< Output Ports:**

- **X** (Number)
- **Y** (Number)
- **Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Mediapipe.MpHandCoordinate>

## 46.1.5 MpHandTracking



**Full Name:** Ops.Extension.Mediapipe.MpHandTracking

**Description:** Get hand data from mediapipe library, use with MpHand

**> Input Ports:**

- **Element** (Object:Element)
- **Min Confidence Detect** (Number)
- **Min Confidence Tracking** (Number)

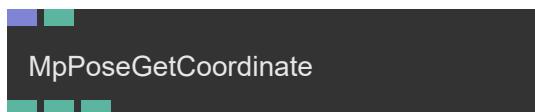
**< Output Ports:**

- **Result** (Object)
- **Found Hands** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Mediapipe.MpHandTracking>

## 46.1.6 MpPoseGetCoordinate



**Full Name:** Ops.Extension.Mediapipe.MpPoseGetCoordinate

**Description:** Get coordinates of specific body parts from mediapipe data

**> Input Ports:**

- **Landmarks** (Array)
- **Landmark Index** (Number: Integer)

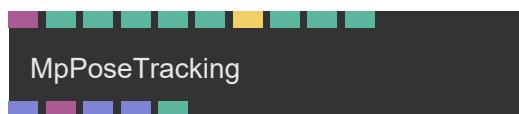
**< Output Ports:**

- **X** (Number)
- **Y** (Number)
- **Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Mediapipe.MpPoseGetCoordinate>

## 46.1.7 MpPoseTracking



**Full Name:** Ops.Extension.Mediapipe.MpPoseTracking

**Description:** Get pose-data (points/landmarks/lines) from webcam using mediapipe library

**> Input Ports:**

- **Element** (Object:Element)
- **Smooth Landmarks** (Number: Boolean)
- **Min Detection Confidence** (Number)
- **Min Tracking Confidence** (Number)
- **Enable Segmentation** (Number: Boolean)
- **Update Texture** (Trigger)
- **Smooth Segmentation** (Number: Boolean)
- **Flip X** (Number: Boolean)
- **Flip Y** (Number: Boolean)

**< Output Ports:**

- **Points** (Array)
- **Segmentation Mask** (Object)
- **Landmarks** (Array)
- **Lines** (Array)
- **Found** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Mediapipe.MpPoseTracking>

# 47 Ops.Extension.OpenType

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.OpenType.OpentypeToSvgPath>

## 47.1 Ops.Extension.OpenType

### 47.1.1 OpentypeFont



**Full Name:** Ops.Extension.OpenType.OpentypeFont

**Description:** Load OTF & TTF fonts via OpenType library

**> Input Ports:**

- **Font File** (String)

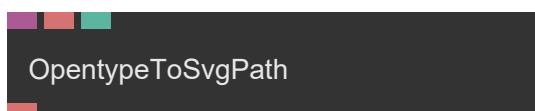
**< Output Ports:**

- **Opentype Font** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.OpenType.OpentypeFont>

### 47.1.2 OpentypeToSvgPath



**Full Name:** Ops.Extension.OpenType.OpentypeToSvgPath

**Description:** get svg path from (OTF) OpentypeFont using the opentype library

**> Input Ports:**

- **Opentype Font** (Object)
- **Text** (String)
- **Letter Spacing** (Number)

**< Output Ports:**

- **Path String** (String)

# 48 Ops.Extension.Osc2Ws

## 48.1 Ops.Extension.Osc2Ws

### 48.1.1 Osc2WsArray



**Full Name:** Ops.Extension.Osc2Ws.Osc2WsArray

**Description:** Outputs an array of data from a user defined OSC address

#### > Input Ports:

- **Message** (Object)
- **Address** (String)
- **Learn** (Trigger)

#### < Output Ports:

- **Result Message** (Object)
- **Array Out** (Array)
- **Array Length** (Number)
- **Received** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Osc2Ws.Osc2WsArray>

### 48.1.2 Osc2WsMessage



**Full Name:** Ops.Extension.Osc2Ws.Osc2WsMessage

**Description:** Shows the current active address of an incoming OSC message

#### > Input Ports:

- **Message** (Object)

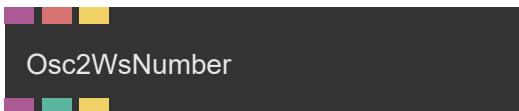
#### < Output Ports:

- **Address** (String)
- **Arguments** (Array)
- **Total Arguments** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Osc2Ws.Osc2WsMessage>

### 48.1.3 Osc2WsNumber



**Full Name:** Ops.Extension.Osc2Ws.Osc2WsNumber

**Description:** Outputs a single number from a user defined OSC address

#### > Input Ports:

- **Message** (Object)
- **Address** (String)
- **Learn** (Trigger)

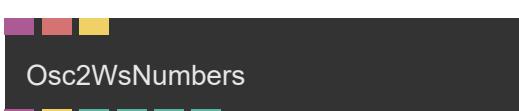
#### < Output Ports:

- **Result Message** (Object)
- **Value** (Number)
- **Received** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Osc2Ws.Osc2WsNumber>

### 48.1.4 Osc2WsNumbers



**Full Name:** Ops.Extension.Osc2Ws.Osc2WsNumbers

**Description:** Outputs up to 4 numbers from a user defined OSC address

#### > Input Ports:

- **Message In** (Object)
- **Osc Address** (String)

- **Learn** (Trigger)

**< Output Ports:**

- **Message Through** (Object)
- **Received** (Trigger)
- **Number 0** (Number)
- **Number 1** (Number)
- **Number 2** (Number)
- **Number 3** (Number)

**Example Patch:** Open in Editor

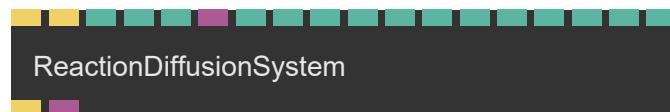
**Docs:** <https://cables.gl/op/Ops.Extension.Osc2Ws.Osc2WsNumbers>

## 49 Ops.Extension.ReactionDiffusion

---

### 49.1 Ops.Extension.ReactionDiffusion

#### 49.1.1 ReactionDiffusionSystem\_v2



**Full Name:** Ops.Extension.ReactionDiffusion.ReactionDiffusionSystem\_v2

**Description:** Cellular automata system as feedback loop texture.

**> Input Ports:**

- **Render** (Trigger)
- **Reset** (Trigger)
- **Use Viewport Size** (Number: Boolean)
- **Width** (Number: Integer)
- **Height** (Number: Integer)
- **Texture In** (Object:Texture)
- **Speed** (Number: Integer)
- **Seed** (Number)
- **Presets Index** (Number: Integer)
- **Feed** (Number)
- **Feed Variation** (Number)
- **Kill** (Number)
- **Kill Variation** (Number)
- **Diffusion Scale** (Number)
- **Diffusion Scale Variation** (Number)
- **Anisotropy** (Number)
- **Noise Scale** (Number)
- **Separate Fields** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)
- **Texture Out** (Object)

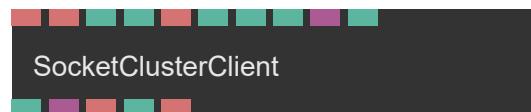
**Example Patch:** Open in Editor

# 50 Ops.Extension.SocketCluster

---

## 50.1 Ops.Extension.SocketCluster

### 50.1.1 SocketClusterClient\_v2



**Full Name:** Ops.Extension.SocketCluster.SocketClusterClient\_v2

**Description:** connect to a socketcluster server and manage the connection

#### ➢ Input Ports:

- **Channel** (String)
- **Server Hostname** (String)
- **Server Port** (Number)
- **Use SSL** (Number: Boolean)
- **enable encryption** (needs to be supported by server)
- **Server Path** (String)
- **Allow Send** (Number: Boolean)
- **Allow Multiple Senders** (Number: Boolean)
- **Additional Serverdata** (Object)
- **additional data send with every message** (can be used for auth-token)
- **Active** (Number: Boolean)

#### ◀ Output Ports:

- **Ready** (booleanNumber)
- **Socket** (Object)
- **Own Client Id** (String)
- **Can Send** (booleanNumber)
- **Error** (String)

**Example Patch:** Open in Editor

Docs: [https://cables.gl/op/Ops.Extension.SocketCluster.SocketClusterClient\\_v2](https://cables.gl/op/Ops.Extension.SocketCluster.SocketClusterClient_v2)

## 50.1.2 SocketClusterReceiveObject

SocketClusterReceiveObject

**Full Name:** Ops.Extension.SocketCluster.SocketClusterReceiveObject

**Description:** Receives object from the socketcluster socket/topic

### > Input Ports:

- **Socket** (Object:Socketcluster)
- **Topic** (String)
- **Receive Own Data** (Number: Boolean)

### < Output Ports:

- **Client Id** (String)
- **Data** (Object)
- **Received** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.SocketCluster.SocketClusterReceiveObject>

## 50.1.3 SocketClusterReceiveTrigger

SocketClusterReceiveTrigger

**Full Name:** Ops.Extension.SocketCluster.SocketClusterReceiveTrigger

**Description:** Receives trigger from the socketcluster socket/topic

### > Input Ports:

- **Socket** (Object:Socketcluster)
- **Topic** (String)
- **Receive Own Data** (Number: Boolean)
- **Use Named Trigger** (Number: Boolean)

### < Output Ports:

- **Client Id** (String)
- **Trigger Name** (String)
- **Received** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.SocketCluster.SocketClusterReceiveTrigger>

## 50.1.4 SocketClusterSendObject

SocketClusterSendObject

**Full Name:** Ops.Extension.SocketCluster.SocketClusterSendObject

**Description:** sends an object via socketcluster/websocket

### > Input Ports:

- **Socket** (Object:Socketcluster)
- **Topic** (String)
- **Data** (Object)
- **Send** (Trigger)

### < Output Ports:

- **Sent Data** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.SocketCluster.SocketClusterSendObject>

## 50.1.5 SocketClusterSendTrigger

SocketClusterSendTrigger

**Full Name:** Ops.Extension.SocketCluster.SocketClusterSendTrigger

**Description:** sends a trigger via socketcluster/websocket

### > Input Ports:

- **Data** (Trigger)
- **Socket** (Object:Socketcluster)
- **Topic** (String)
- **Trigger Name** (String)

- the name of the trigger (created with TriggerSend)

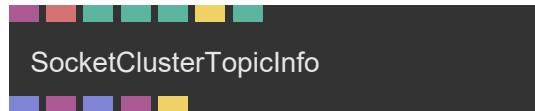
**< Output Ports:**

- Visit `Ops.Extension.SocketCluster.SocketClusterSendTrigger` documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.SocketCluster.SocketClusterSendTrigger>

## 50.1.6 SocketClusterTopicInfo\_v2



**Full Name:** Ops.Extension.SocketCluster.SocketClusterTopicInfo\_v2

**Description:** get info for clients listening on a socketcluster topic

**> Input Ports:**

- **Socket** (Object:Socketcluster)
- **Topic** (String)
- **Timeout Seconds** (Number: Integer)
- **Soft Timeout Seconds** (Number: Integer)
- **Retain Messages** (Number: Integer)
- **Update** (Trigger)
- **Receive My Data** (Number: Boolean)

**< Output Ports:**

- **Active Clients** (Array)
- **Will Time Out** (Object)
- **Timed Out Clients** (Array)
- **Messages** (Object)
- **Updated** (Trigger)

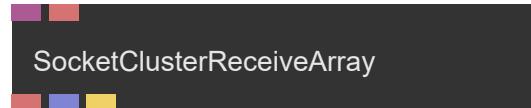
**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Extension.SocketCluster.SocketClusterTopicInfo\\_v2](https://cables.gl/op/Ops.Extension.SocketCluster.SocketClusterTopicInfo_v2)

# 51 Ops.Extension.SocketCluster.Deprecated

## 51.1 Ops.Extension.SocketCluster.Deprecated

### 51.1.1 SocketClusterReceiveArray



**Full Name:** Ops.Extension.SocketCluster.Deprecated.SocketClusterReceiveArray

**Description:** receive an array from the socketcluster topic

**> Input Ports:**

- **Socket** (Object)
- **Topic** (String)

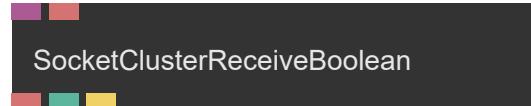
**< Output Ports:**

- **Client Id** (String)
- **Data** (Array)
- **Received** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.SocketCluster.Deprecated.SocketClusterReceiveArray>

### 51.1.2 SocketClusterReceiveBoolean



**Full Name:** Ops.Extension.SocketCluster.Deprecated.SocketClusterReceiveBoolean

**Description:** Receive boolean value from the socketcluster socket/topic

**> Input Ports:**

- **Socket** (Object)
- **Topic** (String)

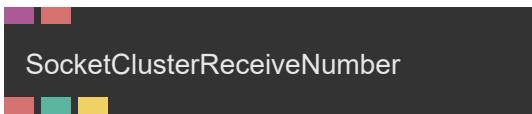
**< Output Ports:**

- **Client Id** (String)
- **Data** (booleanNumber)
- **Received** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.SocketCluster.Deprecated.SocketClusterReceiveBoolean>

### 51.1.3 SocketClusterReceiveNumber



**Full Name:** Ops.Extension.SocketCluster.Deprecated.SocketClusterReceiveNum

**Description:** receive number from the socketcluster socket/topic

**> Input Ports:**

- **Socket** (Object)
- **Topic** (String)

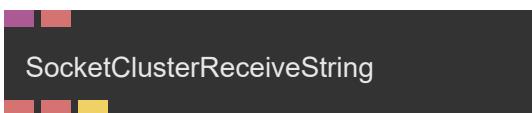
**< Output Ports:**

- **Client Id** (String)
- **Data** (Number)
- **Received** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.SocketCluster.Deprecated.SocketClusterReceiveNumber>

### 51.1.4 SocketClusterReceiveString



**Full Name:** Ops.Extension.SocketCluster.Deprecated.SocketClusterReceiveString

**Description:** receives string from the socketcluster socket/topic

**> Input Ports:**

- **Socket** (Object:Socketcluster)

- **Topic** (String)

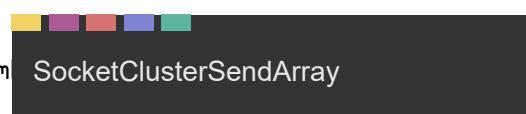
**< Output Ports:**

- **Data** (String)
- **Client Id** (String)
- **Received** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.SocketCluster.Deprecated.SocketClusterReceiveString>

### 51.1.5 SocketClusterSendArray



**Full Name:** Ops.Extension.SocketCluster.Deprecated.SocketClusterSendArray

**Description:** sends an array via socketcluster/websocket

**> Input Ports:**

- **Send** (Trigger)
- **Socket** (Object:Socketcluster)
- **Topic** (String)
- **Data** (Array)
- **Public** (2): MOUSE MOVEMENT SEND

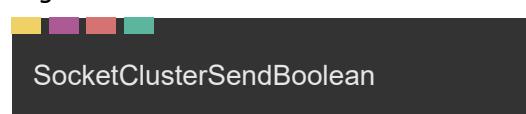
**< Output Ports:**

- **Visit** *Ops.Extension.SocketCluster.Deprecated.SocketClusterSendArray documentation for output port details*

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.SocketCluster.Deprecated.SocketClusterSendArray>

### 51.1.6 SocketClusterSendBoolean



**Full Name:** Ops.Extension.SocketCluster.Deprecated.SocketClusterSendBoolean

**Description:** Sends boolean value via socketcluster/websocket

**> Input Ports:**

- **Send** (Trigger)
- **Socket** (Object)
- **Topic** (String)
- **Data** (Number: Boolean)

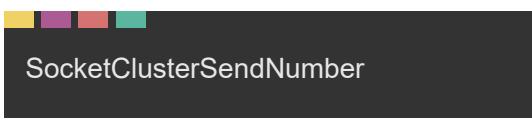
**< Output Ports:**

- Visit *Ops.Extension.SocketCluster.Deprecated.SocketClusterSendBoolean documentation for output port details*

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.SocketCluster.Deprecated.SocketClusterSendBoolean>

## 51.1.7 SocketClusterSendNumber



**Full Name:** Ops.Extension.SocketCluster.Deprecated.SocketClusterSendNumber

**Description:** sends a number via socketcluster/websocket

**> Input Ports:**

- **Send** (Trigger)
- **Socket** (Object)
- **Topic** (String)
- **Data** (Number)

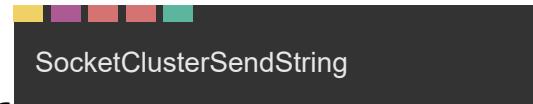
**< Output Ports:**

- Visit *Ops.Extension.SocketCluster.Deprecated.SocketClusterSendNumber documentation for output port details*

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.SocketCluster.Deprecated.SocketClusterSendNumber>

## 51.1.8 SocketClusterSendString



**Full Name:** Ops.Extension.SocketCluster.Deprecated.SocketClusterSendString

**Description:** sends a string via socketcluster/websocket

**> Input Ports:**

- **Send** (Trigger)
- **Socket** (Object:Socketcluster)
- **Topic** (String)
- **Data** (String)

**< Output Ports:**

- Visit *Ops.Extension.SocketCluster.Deprecated.SocketClusterSendString documentation for output port details*

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.SocketCluster.Deprecated.SocketClusterSendString>

## 52 Ops.Extension.Standalone

---

### 52.1 Ops.Extension.Standalone

#### 52.1.1 Ffmpeg



**Full Name:** Ops.Extension.Standalone.Ffmpeg

**Description:** FFmpeg video converter toolbox op

##### > Input Ports:

- **Source Video** (String)
- **Destination File** (String)
- **Set Bitrate** (Number: Boolean)
- **Bitrate** (String)
- **Constant** (Number: Boolean)
- **Set Codec** (Number: Boolean)
- **Codec** (String)
- **Set Size** (Number: Boolean)
- **Size** (String)
- **Crop Time** (Number: Boolean)
- **Start Time** (String)
- **Duration** (String)
- **Process** (Trigger)

##### < Output Ports:

- **Processing** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Ffmpeg>

## 53 Ops.Extension.Standalone.Files

---

### 53.1 Ops.Extension.Standalone.Files

#### 53.1.1 CreateFile



**Full Name:** Ops.Extension.Standalone.Files.CreateFile

**Description:** Create a new empty file on your local harddrive

##### > Input Ports:

- **Default Path** (String)
- **Create File** (Trigger)

##### < Output Ports:

- **Path** (String)
- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Files.CreateFile>

#### 53.1.2 Exist



**Full Name:** Ops.Extension.Standalone.Files.Exist

**Description:** Check if a file exists on the local file system

##### > Input Ports:

- **Path** (String)
- **Execute** (Trigger)

##### < Output Ports:

- **Exists** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Files.Exist>

### 53.1.3 FileUrlToPath



**Full Name:** Ops.Extension.Standalone.Files.FileUrlToPath

**Description:** convert file-url to path

**> Input Ports:**

- **FileUrl** (String)

**< Output Ports:**

- **Path** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Files.FileUrlToPath>

### 53.1.4 Makedir



**Full Name:** Ops.Extension.Standalone.Files.Makedir

**Description:** Create a directory on the local file system

**> Input Ports:**

- **Path** (String)
- **Create** (Trigger)

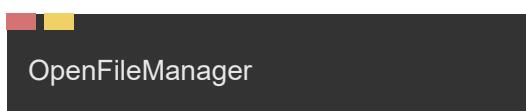
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Files.Makedir>

### 53.1.5 OpenFileManager



**Full Name:** Ops.Extension.Standalone.Files.OpenFileManager

**Description:** Open the native file manager application using that path

**> Input Ports:**

- **Path** (String)
- **Open File Manager** (Trigger)

**< Output Ports:**

- Visit *Ops.Extension.Standalone.Files.OpenFileManager documentation for output port details*

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Files.OpenFileManager>

### 53.1.6 PathToFileUrl



**Full Name:** Ops.Extension.Standalone.Files.PathToFileUrl

**Description:** convert local path to file-url

**> Input Ports:**

- **Path** (String)

**< Output Ports:**

- **FileUrl** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Files.PathToFileUrl>

### 53.1.7 ReadDir



**Full Name:** Ops.Extension.Standalone.Files.ReadDir

**Description:** Read all entries in a directory

**> Input Ports:**

- **Path** (String)
- **Reload** (Trigger)

**< Output Ports:**

- **Entries** (Array)
- **Has Error** (booleanNumber)
- **Error** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Files.ReadDir>

### 53.1.8 ResolvePath



**Full Name:** Ops.Extension.Standalone.Files.ResolvePath

**Description:** Resolves a paths into an absolute path

**> Input Ports:**

- **Path** (String)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Files.ResolvePath>

### 53.1.9 SelectDir



**Full Name:** Ops.Extension.Standalone.Files.SelectDir

**Description:** Choose a directory on your hard drive

**> Input Ports:**

- **Default Path** (String)
- **Select Directory** (Trigger)

**< Output Ports:**

- **Path** (String)
- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Files.SelectDir>

### 53.1.10 SelectFile



**Full Name:** Ops.Extension.Standalone.Files.SelectFile

**Description:** Choose a file on your hard drive

**> Input Ports:**

- **Default Path** (String)
- **Select File** (Trigger)

**< Output Ports:**

- **Path** (String)
- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Files.SelectFile>

## 53.1.11 Stat



**Full Name:** Ops.Extension.Standalone.Files.Stat

**Description:** Get statistics about a file on the local file system

**> Input Ports:**

- **Path** (String)

**< Output Ports:**

- **Stats** (Object)
- **Is Directory** (booleanNumber)
- **Is File** (booleanNumber)
- **Has Error** (booleanNumber)
- **Error** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Files.Stat>

## 53.1.12 SystemDirs



**Full Name:** Ops.Extension.Standalone.Files.SystemDirs

**Description:** Get Default System Directories Paths

**> Input Ports:**

- Visit *Ops.Extension.Standalone.Files.SystemDirs documentation for input port details*

**< Output Ports:**

- **Home** (String)
- **Downloads** (String)
- **Documents** (String)
- **Desktop** (String)
- **Exe** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Files.SystemDirs>

## 53.1.13 Watch



**Full Name:** Ops.Extension.Standalone.Files.Watch

**Description:** Watch a directory, get a trigger when a file changes

**> Input Ports:**

- **Path** (String)
- **Read** (Trigger)

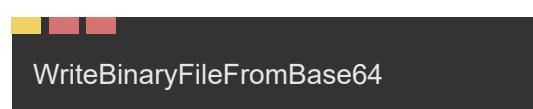
**< Output Ports:**

- **Event Type** (String)
- **Event Filename** (String)
- **Event Happened** (Trigger)
- **Content** (String)
- **Has Error** (booleanNumber)
- **Error** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Files.Watch>

## 53.1.14 WriteBinaryFileFromBase64



**Full Name:** Ops.Extension.Standalone.Files.WriteBinaryFileFromBase64

**Description:** Create a binary file on the local file system from a base64 string

**> Input Ports:**

- **Trigger** (Trigger)
- **Base64** (String)
- **Filename** (String)

**< Output Ports:**

- Visit `Ops.Extension.Standalone.Files.WriteBinaryFileFromBase64` documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Files.WriteBinaryFileFromBase64>

### 53.1.15 WriteTextFile



**Full Name:** Ops.Extension.Standalone.Files.WriteAllTextFile

**Description:** Write a string to a text file on the local file system

**> Input Ports:**

- **Filename** (String)
- **Content** (String)
- **Write** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **Has Error** (booleanNumber)
- **Error** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Files.WriteAllTextFile>

## 54 Ops.Extension.Standalone.Net

### 54.1 Ops.Extension.Standalone.Net

#### 54.1.1 HttpServer



**Full Name:** Ops.Extension.Standalone.Net.HttpServer

**Description:** Create a Web/Http server locally

**> Input Ports:**

- **Hostname** (String)
- **Port** (Number: Integer)

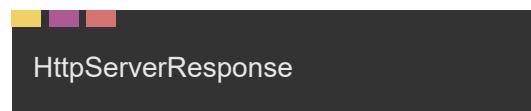
**< Output Ports:**

- **Trigger Request** (Trigger)
- **Response** (Object)
- **Request URL** (String)
- **Request** (Object)
- **Running** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Net.HttpServer>

#### 54.1.2 HttpServerResponse



**Full Name:** Ops.Extension.Standalone.Net.HttpServerResponse

**Description:** Answer http requests by sending string to the browser/client

**> Input Ports:**

- **Trigger** (Trigger)
- **Response** (Object)

- **Body** (String)

#### < Output Ports:

- Visit *Ops.Extension.Standalone.Net.HttpServerResponse* documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Net.HttpServerResponse>

### 54.1.3 IpAddress



IpAddress

**Full Name:** Ops.Extension.Standalone.Net.IpAddress

**Description:** Outputs your local IP Adress

#### > Input Ports:

- Visit *Ops.Extension.Standalone.Net.IpAddress* documentation for input port details

#### < Output Ports:

- **Local IP** (String)
- **Interface** (String)
- **Data** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Net.IpAddress>

### 54.1.4 Osc\_v2



Osc

**Full Name:** Ops.Extension.Standalone.Net.Osc\_v2

**Description:** Visit documentation for details

#### > Input Ports:

- **Port** (Number: Integer)

#### < Output Ports:

- **Message Received** (Trigger)
- **Message** (Object)
- **Connection** (Object)
- **Status** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Extension.Standalone.Net.Osc\\_v2](https://cables.gl/op/Ops.Extension.Standalone.Net.Osc_v2)

### 54.1.5 OscSend



OscSend

**Full Name:** Ops.Extension.Standalone.Net.OscSend

**Description:** send data to a OSC device

#### > Input Ports:

- **Connection** (Object)
- **Net Address** (String)
- **Port** (Number: Integer)
- **OSC Address** (String)
- **Number** (Number)
- **Send** (Trigger)
- **Public** (1): OSC: READ / SEND

#### < Output Ports:

- Visit *Ops.Extension.Standalone.Net.OscSend* documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Net.OscSend>

### 54.1.6 ReadTextFile



ReadTextFile

**Full Name:** Ops.Extension.Standalone.Net.ReadTextFile  
**Description:** Read a text file as string from the local file system

**> Input Ports:**

- **Filename** (String)
- **Read** (Trigger)

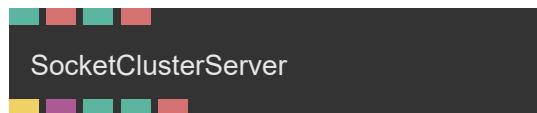
**< Output Ports:**

- **Next** (Trigger)
- **Content** (String)
- **Has Error** (booleanNumber)
- **Error** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Net.ReadTextFile>

## 54.1.7 SocketClusterServer



**Full Name:** Ops.Extension.Standalone.Net.SocketClusterServer

**Description:** start a socketcluster server

**> Input Ports:**

- **Active** (Number: Boolean)
- **Hostname** (String)
- **Port** (Number: Integer)
- **Path** (String)

**< Output Ports:**

- **Receiving** (Trigger)
- **Data** (Object)
- **Listening** (booleanNumber)
- **Clients** (Number)
- **Error** (String)

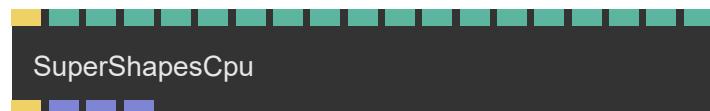
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Standalone.Net.SocketClusterServer>

# 55 Ops.Extension.SuperShapes

## 55.1 Ops.Extension.SuperShapes

### 55.1.1 SuperShapesCpu



**Full Name:** Ops.Extension.SuperShapes.SuperShapesCpu

**Description:** Visit documentation for details

**> Input Ports:**

- **Update** (Trigger)
- **Shape Index** (Number: Integer)
- **Tesselation** (Number: Integer)
- **Param 0** (Number)
- **Param 1** (Number)
- **Param 2** (Number)
- **Param 3** (Number)
- **Param 4** (Number)
- **Param 5** (Number)
- **Param 6** (Number)
- **Param 7** (Number)
- **Param 8** (Number)
- **Param 9** (Number)
- **Param 10** (Number)
- **Param 11** (Number)
- **Param 12** (Number)
- **Param 13** (Number)
- **Param 14** (Number)
- **Param 15** (Number)

**< Output Ports:**

- **Next** (Trigger)
- **Coords** (Array)
- **Faces** (Array)

- **TexCoords** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.SuperShapes.SuperShapesCpu>

## 55.1.2 SuperShapesGpu



**Full Name:** Ops.Extension.SuperShapes.SuperShapesGpu

**Description:** Visit documentation for details

**> Input Ports:**

- **Update** (Trigger)
- **Render** (Number: Boolean)
- **Shape Index** (Number: Integer)
- **Tesselation** (Number: Integer)
- **Param 0** (Number)
- **Param 1** (Number)
- **Param 2** (Number)
- **Param 3** (Number)
- **Param 4** (Number)
- **Param 5** (Number)
- **Param 6** (Number)
- **Param 7** (Number)
- **Param 8** (Number)
- **Param 9** (Number)
- **Param 10** (Number)
- **Param 11** (Number)
- **Param 12** (Number)
- **Param 13** (Number)
- **Param 14** (Number)
- **Param 15** (Number)

**< Output Ports:**

- **Next** (Trigger)

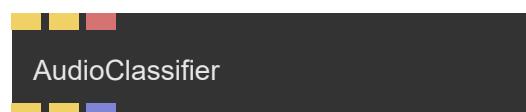
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.SuperShapes.SuperShapesGpu>

# 56 Ops.Extension.TeachableMachines

## 56.1 Ops.Extension.TeachableMachines

### 56.1.1 AudioClassifier



**Full Name:** Ops.Extension.TeachableMachines.AudioClassifier

**Description:** Use the Teachable Machines audio classifier for your microphone. Insert the uploaded model URL.

**> Input Ports:**

- **Trigger In** (Trigger)
- **Initialize** (Trigger)
- **Model URL** (String)

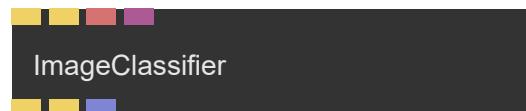
**< Output Ports:**

- **Trigger** (Trigger)
- **Initialized** (Trigger)
- **Classifier** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.TeachableMachines.AudioClassifier>

### 56.1.2 ImageClassifier\_v2



**Full Name:** Ops.Extension.TeachableMachines.ImageClassifier\_v2

**Description:** Use the Teachable Machines image classifier. Insert the uploaded model URL.

**> Input Ports:**

- **Trigger In** (Trigger)
- **Initialize** (Trigger)
- **Model URL** (String)
- **Webcam Element** (Object)

**< Output Ports:**

- **Trigger** (Trigger)
- **Initialized** (Trigger)
- **Classifier** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Extension.TeachableMachines.ImageClassifier\\_v2](https://cables.gl/op/Ops.Extension.TeachableMachines.ImageClassifier_v2)

### 56.1.3 PoseDetection\_v2



**Full Name:** Ops.Extension.TeachableMachines.PoseDetection\_v2

**Description:** Use the Teachable Machines pose detection with your webcam. Insert the uploaded model URL.

**> Input Ports:**

- **Render** (Trigger)
- **Initialize** (Trigger)
- **Model URL** (String)
- **Webcam Element** (Object)
- **Flip Image** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)
- **Initialized** (Trigger)
- **Classifier** (Array)
- **Pose Positions** (Array)
- **Image Flipped** (Number)

**Example Patch:** Open in Editor

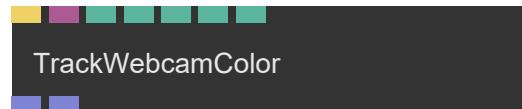
**Docs:** [https://cables.gl/op/Ops.Extension.TeachableMachines.PoseDetection\\_v2](https://cables.gl/op/Ops.Extension.TeachableMachines.PoseDetection_v2)

## 57 Ops.Extension.Trackingjs

---

### 57.1 Ops.Extension.Trackingjs

#### 57.1.1 TrackWebcamColor



**Full Name:** Ops.Extension.Trackingjs.TrackWebcamColor

**Description:** Track a position of a specific color in the current webcam stream

**> Input Ports:**

- **Update** (Trigger)
- **Video Element** (Object)
- **Threshold** (Number)
- **Resize Video** (Number)
- **R** (Number)
- **G** (Number)
- **B** (Number)

**< Output Ports:**

- **Positions** (Array)
- **Sizes** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Trackingjs.TrackWebcamColor>

# 58 Ops.Extension.Voice

## 58.1 Ops.Extension.Voice

### 58.1.1 MeSpeak



**Full Name:** Ops.Extension.Voice.MeSpeak

**Description:** uses mespeak.js to convert text-to-speech

#### > Input Ports:

- **Text** (String)
- **Say** (Trigger)
- **Amplitude** (Number)
- **Pitch** (Number)
- **Voice Index** (Number: Integer)
- **Word Gap** (Number: Integer)
- **Variants Index** (Number: Integer)
- **Line-Break Length** (Number: Integer)
- **Capitals** (Number: Integer)
- **Punctuation** (String)
- **No Stop** (Number: Boolean)
- **UTF16** (Number: Boolean)
- **SSML** (Number: Boolean)
- **Log Console** (Number: Boolean)
- **Pan** (Number)

#### < Output Ports:

- **Audio Out** (Object)
- **Speaking** (booleanNumber)
- **Voice Loaded** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Voice.MeSpeak>

### 58.1.2 Say\_v2



**Full Name:** Ops.Extension.Voice.Say\_v2

**Description:** Text-to-Speech, speaks different languages (speech synthesis)

#### > Input Ports:

- **Update State** (Trigger)
- **Text** (String)
- **Say** (Trigger)
- **Voice** (Number: Select Box)
- **Pitch** (Number)
- **Rate** (Number)
- **Volume** (Number)
- **Say On Text Change** (Number: Boolean)
- **Pause** (Trigger)
- **Resume** (Trigger)
- **Cancel** (Trigger)

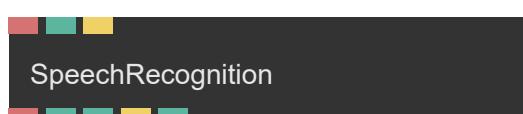
#### < Output Ports:

- **Next** (Trigger)
- **Speaking** (Number)
- **Pending** (Number)
- **Paused** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Extension.Voice.Say\\_v2](https://cables.gl/op/Ops.Extension.Voice.Say_v2)

### 58.1.3 SpeechRecognition



**Full Name:** Ops.Extension.Voice.SpeechRecognition

**Description:** speech to text recognition

#### > Input Ports:

- **Language** (String)
- **Active** (Number: Boolean)
- **Start** (Trigger)

◀ **Output Ports:**

- **Result** (String)
- **Confidence** (Number)
- **Supported** (booleanNumber)
- **New Result** (Trigger)
- **Started** (booleanNumber)

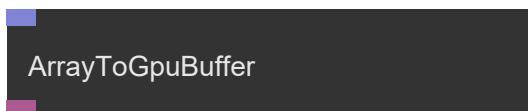
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.Voice.SpeechRecognition>

# 59 Ops.Extension.WebGpu

## 59.1 Ops.Extension.WebGpu

### 59.1.1 ArrayToGpuBuffer



**Full Name:** Ops.Extension.WebGpu.ArrayToGpuBuffer

**Description:** Upload an array to the GPU as a GpuBuffer

▶ **Input Ports:**

- **Arr** (Array)

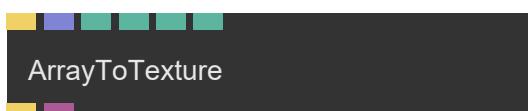
◀ **Output Ports:**

- **GPUBuffer** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.ArrayToGpuBuffer>

### 59.1.2 ArrayToTexture



**Full Name:** Ops.Extension.WebGpu.ArrayToTexture

**Description:** Convert an array of numbers to a webgpu texture

▶ **Input Ports:**

- **Update** (Trigger)
- **Array** (Array)
- **Wrap Index** (Number: Integer)
- **Width** (Number: Integer)
- **Height** (Number: Integer)

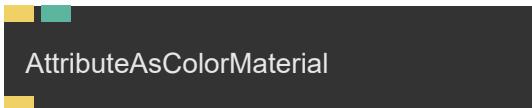
◀ **Output Ports:**

- **Next** (Trigger)
- **Texture** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.ArrayToTexture>

### 59.1.3 AttributeAsColorMaterial



**Full Name:** Ops.Extension.WebGpu.AttributeAsColorMaterial

**Description:** Render mesh attribultes as color

**> Input Ports:**

- **Render** (Trigger)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.AttributeAsColorMaterial>

### 59.1.4 BasicMaterial



**Full Name:** Ops.Extension.WebGpu.BasicMaterial

**Description:** A simple material without shading

**> Input Ports:**

- **Render** (Trigger)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)
- **Colorize Texture** (Number: Boolean)

- **DiffuseRepeatX** (Number)
- **DiffuseRepeatY** (Number)
- **Tex Offset X** (Number)
- **Tex Offset Y** (Number)
- **Texture** (Object:Texture)
- **Mask** (Object:Texture)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.BasicMaterial>

### 59.1.5 ColorTexture



**Full Name:** Ops.Extension.WebGpu.ColorTexture

**Description:** A texture containing only one color

**> Input Ports:**

- **Render** (Trigger)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)

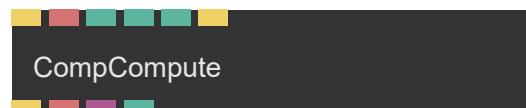
**< Output Ports:**

- **Next** (Trigger)
- **Texture\_out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.ColorTexture>

### 59.1.6 CompCompute



**Full Name:** Ops.Extension.WebGpu.CompCompute

**Description:** Compose a compute shader

**> Input Ports:**

- **Compute** (Trigger)
- **Source** (String)
- **Workgroups 1** (Number: Integer)
- **Workgroups 2** (Number: Integer)
- **Workgroups 3** (Number: Integer)
- **Force Update** (Trigger)

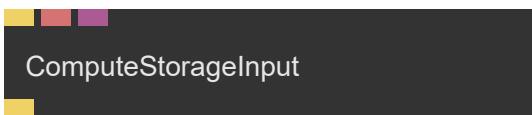
**< Output Ports:**

- **Next** (Trigger)
- **Code** (String)
- **Buffer** (Object)
- **Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.CompCompute>

## 59.1.7 ComputeStorageInput



ComputeStorageInput

**Full Name:** Ops.Extension.WebGpu.ComputeStorageInput

**Description:** Compute shader GPU buffer storage input

**> Input Ports:**

- **Trigger** (Trigger)
- **Name** (String)
- **Buffer** (Object)

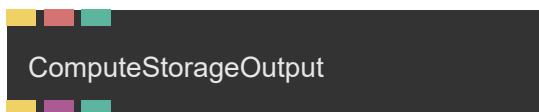
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.ComputeStorageInput>

## 59.1.8 ComputeStorageOutput



ComputeStorageOutput

**Full Name:** Ops.Extension.WebGpu.ComputeStorageOutput

**Description:** Compute shader GPU buffer storage output

**> Input Ports:**

- **Trigger** (Trigger)
- **Name** (String)
- **Length** (Number: Integer)

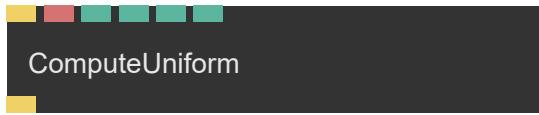
**< Output Ports:**

- **Next** (Trigger)
- **Buffer** (Object)
- **Buffer Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.ComputeStorageOutput>

## 59.1.9 ComputeUniform



ComputeUniform

**Full Name:** Ops.Extension.WebGpu.ComputeUniform

**Description:** Add a uniform input to a compute shader composition

**> Input Ports:**

- **Trigger** (Trigger)
- **Name** (String)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **W** (Number)

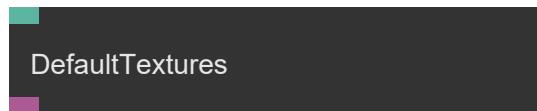
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.ComputeUniform>

## 59.1.10 DefaultTextures



**Full Name:** Ops.Extension.WebGpu.DefaultTextures

**Description:** Outputs textures

**> Input Ports:**

- Visit *Ops.Extension.WebGpu.DefaultTextures documentation* for input port details

**< Output Ports:**

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.DefaultTextures>

## 59.1.11 FaceCulling



**Full Name:** Ops.Extension.WebGpu.FaceCulling

**Description:** cull (do not draw) back or front facing faces/triangles

**> Input Ports:**

- **Render** (Trigger)

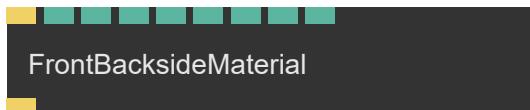
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.FaceCulling>

## 59.1.12 FrontBacksideMaterial



FrontBacksideMaterial

**Full Name:** Ops.Extension.WebGpu.FrontBacksideMaterial

**Description:** Show direction of faces as color

**> Input Ports:**

- **Render** (Trigger)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)
- **R 2** (Number)
- **G 2** (Number)
- **B 2** (Number)
- **A 2** (Number)

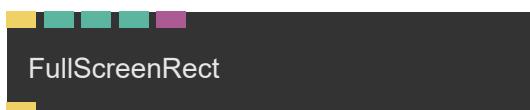
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.FrontBacksideMaterial>

## 59.1.13 FullScreenRect



FullScreenRect

**Full Name:** Ops.Extension.WebGpu.FullScreenRect

**Description:** Render a rectangle that fills the whole canvas

**> Input Ports:**

- **Render** (Trigger)
- **Flip Y** (Number: Boolean)
- **Flip X** (Number: Boolean)
- **Texture** (Object:Texture)

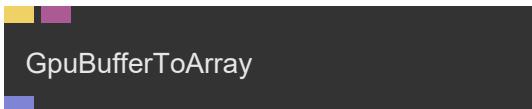
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.FullScreenRect>

## 59.1.14 GpuBufferToArray



**Full Name:** Ops.Extension.WebGpu.GpuBufferToArray

**Description:** Convert a GpuBuffer to a CPU Array

**> Input Ports:**

- **Trigger** (Trigger)
- **Pos Buffer** (Object)

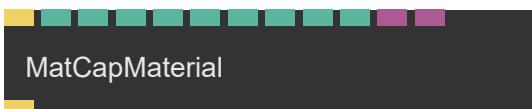
**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.GpuBufferToArray>

## 59.1.15 MatCapMaterial



**Full Name:** Ops.Extension.WebGpu.MatCapMaterial

**Description:** Image based material that uses a matcap environment texture

**> Input Ports:**

- **Render** (Trigger)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)
- **Colorize Texture** (Number: Boolean)
- **DiffuseRepeatX** (Number)

- **DiffuseRepeatY** (Number)

- **Tex Offset X** (Number)

- **Tex Offset Y** (Number)

- **Matcap** (Object:Texture)

- **Diffuse** (Object:Texture)

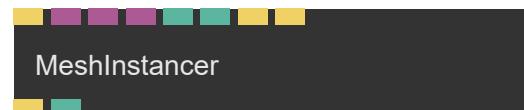
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.MatCapMaterial>

## 59.1.16 MeshInstancer



**Full Name:** Ops.Extension.WebGpu.MeshInstancer

**Description:** Draw the same mesh many times very fast

**> Input Ports:**

- **Render** (Trigger)
- **Geometry** (Object:Geometry)
- **Pos Buffer** (Object)
- **Scale Buffer** (Object)
- **Num Instances** (Number: Integer)
- **Reset** (Trigger)
- **Test** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **Total Instances** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.MeshInstancer>

## 59.1.17 Pipeline



**Full Name:** Ops.Extension.WebGpu.Pipeline

**Description:** show content of last used pipeline for debugging

**> Input Ports:**

- **Trigger** (Trigger)
- **Force Rebuild** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **Pipeline** (Object)
- **Shader Info** (Object)
- **Shader Source** (String)
- **Compile Count** (Number)
- **Shader Id** (String)
- **Defines** (Array)

**Example Patch:** Open in Editor

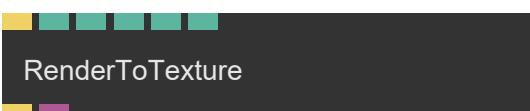
**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.Pipeline>

- **Texture** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.RenderToTexture>

## 59.1.18 RenderToTexture



**Full Name:** Ops.Extension.WebGpu.RenderToTexture

**Description:** render into a texture

**> Input Ports:**

- **Trigger** (Trigger)
- **Texture Width** (Number: Integer)
- **Texture Height** (Number: Integer)
- **Clear** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)

## 59.1.19 SaselHund



**Full Name:** Ops.Extension.WebGpu.SaselHund

**Description:** Visit documentation for details

**> Input Ports:**

- Visit *Ops.Extension.WebGpu.SaselHund* documentation for input port details

**< Output Ports:**

- Visit *Ops.Extension.WebGpu.SaselHund* documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.SaselHund>

## 59.1.20 Texture



**Full Name:** Ops.Extension.WebGpu.Texture

**Description:** Load an image file as a texture

**> Input Ports:**

- **File** (String)
- **Wrap Index** (Number: Integer)

**< Output Ports:**

- **Texture** (Object)
- **Width** (Number)
- **Height** (Number)

- **Pixelformat** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.Texture>

## 59.1.21 VizTexture



**Full Name:** Ops.Extension.WebGpu.VizTexture

**Description:** Vizualize a webgpu texture on the patchfield

**> Input Ports:**

- **Texture In** (Object:Texture)

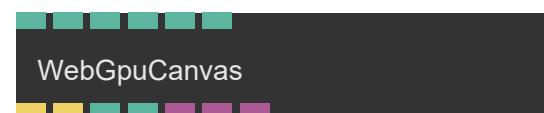
**< Output Ports:**

- Visit *Ops.Extension.WebGpu.VizTexture documentation* for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.VizTexture>

## 59.1.22 WebGpuCanvas



**Full Name:** Ops.Extension.WebGpu.WebGpuCanvas

**Description:** Create a canvas for WebGPU

**> Input Ports:**

- **Active** (Number: Boolean)
- **Catch Errors** (Number: Boolean)
- **Stop On Errors** (Number: Boolean)
- **Profile** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)
- **Next2** (Trigger)

- **Supported** (booleanNumber)
- **MS Frame** (Number)
- **Canvas** (Object)
- **Canvas Prev** (Object)
- **Profiler Data** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.WebGpuCanvas>

## 59.1.23 WebGpuInfo



**Full Name:** Ops.Extension.WebGpu.WebGpuInfo

**Description:** Output information about WebGPU adapter and implementation

**> Input Ports:**

- **Trigger** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **Limits** (Object)
- **Vendor** (String)
- **Architecture** (String)
- **Presentation Format** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Extension.WebGpu.WebGpuInfo>

# 60 Ops.Gl

## 60.1 Ops.Gl

### 60.1.1 ArrayToTexture\_v2



**Full Name:** Ops.Gl.ArrayToTexture\_v2

**Description:** create a texture from an array of number values

#### > Input Ports:

- **Update** (Trigger)
- **Array** (Array)
- **Width** (Number: Integer)
- **Height** (Number: Integer)
- **Fill Up** (Number: Boolean)
- **Flip** (Number: Boolean)
- **PixelFormat Index** (Number: Integer)
- **Wrap Index** (Number: Integer)

#### < Output Ports:

- **Next** (Trigger)
- **Texture Out** (Object)
- **Tex Width** (Number)
- **Tex Height** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ArrayToTexture\\_v2](https://cables.gl/op/Ops.Gl.ArrayToTexture_v2)

### 60.1.2 BlendMode



**Full Name:** Ops.Gl.BlendMode

**Description:** change how colors are mixed (blending/mixing modes)

#### > Input Ports:

- **Render** (Trigger)
- **Blendmode Index** (Number: Integer)
- **Premultiplied** (Number: Boolean)

#### < Output Ports:

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.BlendMode>

### 60.1.3 CanvasFocus



**Full Name:** Ops.Gl.CanvasFocus

**Description:** is canvas focussed ?

#### > Input Ports:

- **Focus** (Trigger)

#### < Output Ports:

- **Has Focus** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.CanvasFocus>

### 60.1.4 CanvasInfo\_v3



**Full Name:** Ops.Gl.CanvasInfo\_v3

**Description:** the size of the canvas in pixels, aspect ratio and pixel density

#### > Input Ports:

- Visit `Ops.Gl.CanvasInfo_v3` documentation for input port details

**< Output Ports:**

- **CSS Width** (Number)
- **CSS Height** (Number)
- **Pixel Ratio** (Number)
- **Pixel Width** (Number)
- **Pixel Height** (Number)
- **Aspect Ratio** (Number)
- **Landscape** (booleanNumber)
- **Canvas** (Object)
- **Canvas Parent** (Object)
- **Resized** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.CanvasInfo\\_v3](https://cables.gl/op/Ops.Gl.CanvasInfo_v3)

## 60.1.5 CanvasToTexture



**Full Name:** Ops.Gl.CanvasToTexture

**Description:** convert a canvas to texture

**> Input Ports:**

- **Canvas** (Object)
- **Filter Index** (Number: Integer)
- **Wrap Index** (Number: Integer)
- **Force Update** (Trigger)

**< Output Ports:**

- **Texture** (Object)
- **Width** (Number)
- **Height** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.CanvasToTexture>

## 60.1.6 ClearColor



**Full Name:** Ops.Gl.ClearColor

**Description:** sets all cleared pixels to one colour. Use to change the background colour.

**> Input Ports:**

- **Render** (Trigger)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ClearColor>

## 60.1.7 ClearDepth



**Full Name:** Ops.Gl.ClearDepth

**Description:** Clears the depth buffer (zbuffer, z buffer)

**> Input Ports:**

- **Render** (Trigger)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ClearDepth>

## 60.1.8 ColorMask



**Full Name:** Ops.Gl.ColorMask

**Description:** enable/disable RGBA color channels of your entire scene

**> Input Ports:**

- **Execute** (Trigger)
- **Red** (Number: Boolean)
- **Green** (Number: Boolean)
- **Blue** (Number: Boolean)
- **Alpha** (Number: Boolean)

**< Output Ports:**

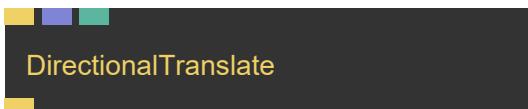
- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ColorMask>

**Docs:** <https://cables.gl/op/Ops.Gl.ColorPick>

## 60.1.10 DirectionalTranslate



**Full Name:** Ops.Gl.DirectionalTranslate

**Description:** translate away from a point in space

**> Input Ports:**

- **Exec** (Trigger)
- **Center Model Matrix** (Array)
- **Amount** (Number)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.DirectionalTranslate>

## 60.1.9 ColorPick



**Full Name:** Ops.Gl.ColorPick

**Description:** pick a color at x,y coordinates of canvas

**> Input Ports:**

- **Render** (Trigger)
- **X** (Number)
- **Y** (Number)

**< Output Ports:**

- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)

**Example Patch:** Open in Editor

## 60.1.11 DownloadTexture\_v3



**Full Name:** Ops.Gl.DownloadTexture\_v3

**Description:** Download a texture as an image file

**> Input Ports:**

- **Texture** (Object:Texture)
- **Quality** (Number)
- **Filename** (String)
- **Download** (Trigger)

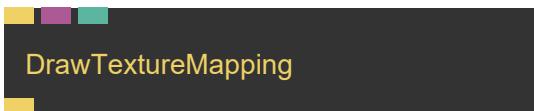
**< Output Ports:**

- **Jcrmz8mnz** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.DownloadTexture\\_v3](https://cables.gl/op/Ops.Gl.DownloadTexture_v3)

## 60.1.12 DrawTextureMapping



**DrawTextureMapping**

**Full Name:** Ops.Gl.DrawTextureMapping

**Description:** draw texture mapping coordinates

**> Input Ports:**

- **Render** (Trigger)
- **Geometry** (Object:Geometry)
- **Num Points** (Number)

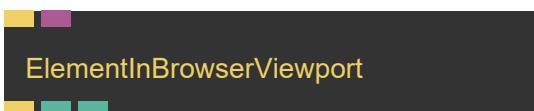
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.DrawTextureMapping>

## 60.1.13 ElementInBrowserViewport



**ElementInBrowserViewport**

**Full Name:** Ops.Gl.ElementInBrowserViewport

**Description:** check if webgl canvas element is in the current browser view-port

**> Input Ports:**

- **Update** (Trigger)
- **Element** (Object:Element)

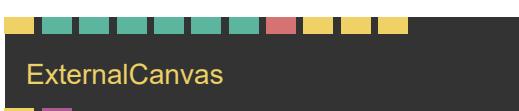
**< Output Ports:**

- **Next** (Trigger)
- **Fully Visible** (booleanNumber)
- **Partly Visible** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ElementInBrowserViewport>

## 60.1.14 ExternalCanvas



**ExternalCanvas**

**Full Name:** Ops.Gl.ExternalCanvas

**Description:** Open a new window that shows a copy of the patch canvas

**> Input Ports:**

- **Update** (Trigger)
- **Pos X** (Number: Integer)
- **Pos Y** (Number: Integer)
- **Width** (Number: Integer)
- **Height** (Number: Integer)
- **Smoothing** (Number: Boolean)
- **Stretch** (Number: Boolean)
- **Title** (String)
- **Open Window** (Trigger)
- **Fullscreen** (Trigger)
- **Close** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **Element** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ExternalCanvas>

## 60.1.15 FaceCulling\_v2



**FaceCulling**

**Full Name:** Ops.Gl.FaceCulling\_v2

**Description:** Disable the rendering of front or back facing triangles with culling

**> Input Ports:**

- **Render** (Trigger)

- **Active** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.FaceCulling\\_v2](https://cables.gl/op/Ops.Gl.FaceCulling_v2)

## 60.1.16 FontMSDF\_v2



**Full Name:** Ops.Gl.FontMSDF\_v2

**Description:** Load MSDF Font data and texture to use

**> Input Ports:**

- **Font Name** (String)
- **Font Data** (String)
- **Font Image** (String)
- **Font Image 1** (String)
- **Font Image 2** (String)
- **Font Image 3** (String)

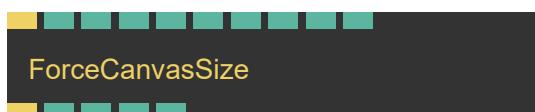
**< Output Ports:**

- **Loaded** (booleanNumber)
- **Total Chars** (Number)
- **Chars** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.FontMSDF\\_v2](https://cables.gl/op/Ops.Gl.FontMSDF_v2)

## 60.1.17 ForceCanvasSize



**Full Name:** Ops.Gl.ForceCanvasSize

**Description:** Resize canvas element to a specific pixel size or aspect ratio

**> Input Ports:**

- **Trigger** (Trigger)
- **Active** (Number: Boolean)
- **Center In Parent** (Number: Boolean)
- **Scale To Fit Parent** (Number: Boolean)
- **Set Width** (Number: Integer)
- **Set Height** (Number: Integer)
- **Aspect Ratio Index** (Number: Integer)
- **Ratio** (Number)
- **Fill Parent Index** (Number: Integer)

**< Output Ports:**

- **Next** (Trigger)
- **Width** (Number)
- **Height** (Number)
- **Margin Left** (Number)
- **Margin Top** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ForceCanvasSize>

## 60.1.18 GateTexture



**Full Name:** Ops.Gl.GateTexture

**Description:** Will only allow an Object to be output if the pass through parameter evaluates to true

**> Input Ports:**

- **Object In** (Object:Texture)
- **Pass Through** (Number: Boolean)
- **Only Valid Textures** (Number: Boolean)

**< Output Ports:**

- **Object Out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GateTexture>

## 60.1.19 GlBlendFunc



**Full Name:** Ops.Gl.GlBlendFunc

**Description:** set gl blendmodes directly

**> Input Ports:**

- **Exec** (Trigger)
- **Src RGB Index** (Number: Integer)
- **Dst RGB Index** (Number: Integer)
- **Src Alpha Index** (Number: Integer)
- **Dst Alpha Index** (Number: Integer)
- **Blend Equation Index** (Number: Integer)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GlBlendFunc>

## 60.1.20 GlInfo\_v2



**Full Name:** Ops.Gl.GlInfo\_v2

**Description:** information about the webgl context

**> Input Ports:**

- Visit *Ops.Gl.GlInfo\_v2 documentation* for input port details

**< Output Ports:**

- **WebGL Version Short** (Number)
- **WebGL Version** (String)
- **GLSL Version** (String)
- **Max Frag Uniforms** (Number)
- **Max Vert Uniforms** (Number)
- **Max Texture Size** (Number)

- **Max Texture Units** (Number)
- **Max Varying Vectors** (Number)
- **Max MSAA Samples** (Number)
- **Extensions** (Array)
- **Vendor** (String)
- **Renderer** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.GlInfo\\_v2](https://cables.gl/op/Ops.Gl.GlInfo_v2)

## 60.1.21 GlPrimitive



**Full Name:** Ops.Gl.GlPrimitive

**Description:** force rendering of meshes using points,lines or triangles

**> Input Ports:**

- **Execute** (Trigger)
- **Primitive Index** (Number: Integer)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GlPrimitive>

## 60.1.22 GradientTexture



**Full Name:** Ops.Gl.GradientTexture

**Description:** texture containing a colour gradient that can be altered with an editor

**> Input Ports:**

- **Gradient** (Number)

- **Direction Index** (Number: Integer)
- **Smoothstep** (Number: Boolean)
- **Step** (Number: Boolean)
- **Flip** (Number: Boolean)
- **SRGB** (Number: Boolean)
- **Oklab** (Number: Boolean)
- **Size** (Number: Integer)
- **Wrap Index** (Number: Integer)
- **Dither** (Number)
- **Gradient Array** (Array)
- **Randomize Colors** (Trigger)

**< Output Ports:**

- **Texture** (Object)
- **Alpha Mask** (Object)
- **Colors** (Array)
- **Colors Pos** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GradientTexture>

## 60.1.23 GridTransform



**Full Name:** Ops.Gl.GridTransform

**Description:** transform and arrange elements into a grid

**> Input Ports:**

- **Render** (Trigger)
- **Num X** (Number: Integer)
- **Num Y** (Number: Integer)
- **Space X** (Number)
- **Space Y** (Number)

**< Output Ports:**

- **Next** (Trigger)
- **Index** (Number)
- **X Index** (Number)

- **Y Index** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GridTransform>

## 60.1.24 Identity



**Full Name:** Ops.Gl.Identity

**Description:** reset all transforms (modelmatrix)

**> Input Ports:**

- **Exe** (Trigger)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Identity>

## 60.1.25 IdentityViewMatrix



**Full Name:** Ops.Gl.IdentityViewMatrix

**Description:** reset the view matrix (cameras etc.)

**> Input Ports:**

- **Exe** (Trigger)

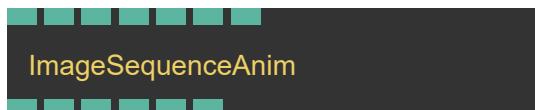
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.IdentityViewMatrix>

## 60.1.26 ImageSequenceAnim\_v2



**Full Name:** Ops.Gl.ImageSequenceAnim\_v2

**Description:** play a image sprite animation

**> Input Ports:**

- **Time** (Number)
- **FPS** (Number)
- **Num X** (Number)
- **Num Y** (Number)
- **Max Frames** (Number: Integer)
- **Flip Y** (Number: Boolean)

**< Output Ports:**

- **Repeat X** (Number)
- **Repeat Y** (Number)
- **Offset X** (Number)
- **Offset Y** (Number)
- **Frame** (Number)
- **Progress** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageSequenceAnim\\_v2](https://cables.gl/op/Ops.Gl.ImageSequenceAnim_v2)

- **ID** (String)
- **Class** (String)
- **Pivot X Index** (Number: Integer)
- **Pivot Y Index** (Number: Integer)
- **Axis Index** (Number: Integer)
- **Is Interactive** (Number: Boolean)
- **Render Rectangle** (Number: Boolean)
- **Show Boundings** (Number: Boolean)
- **Cursor Index** (Number: Integer)
- **Active** (Number: Boolean)

**< Output Ports:**

- **Trigger Out** (Trigger)
- **Geometry** (Object)
- **Pointer Hover** (booleanNumber)
- **Pointer Down** (booleanNumber)
- **Pointer X** (Number)
- **Pointer Y** (Number)
- **Top** (Number)
- **Left** (Number)
- **Right** (Number)
- **Bottom** (Number)
- **Left Click** (Trigger)
- **Dom Element** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.InteractiveRectangle\\_v2](https://cables.gl/op/Ops.Gl.InteractiveRectangle_v2)

## 60.1.27 InteractiveRectangle\_v2



**Full Name:** Ops.Gl.InteractiveRectangle\_v2

**Description:** An area which is interactive

**> Input Ports:**

- **Trigger In** (Trigger)
- **Width** (Number)
- **Height** (Number)

## 60.1.28 LayerSequence



**Full Name:** Ops.Gl.LayerSequence

**Description:** Render Multiple Layers in a specific order

**> Input Ports:**

- **Exe** (Trigger)

**< Output Ports:**

- **Trigger 0** (Trigger)
- **Trigger 1** (Trigger)
- **Trigger 2** (Trigger)
- **Trigger 3** (Trigger)
- **Trigger 4** (Trigger)
- **Trigger 5** (Trigger)
- **Trigger 6** (Trigger)
- **Trigger 7** (Trigger)
- **Trigger 8** (Trigger)
- **Trigger 9** (Trigger)
- **Trigger 10** (Trigger)
- **Trigger 11** (Trigger)
- **Trigger 12** (Trigger)
- **Trigger 13** (Trigger)
- **Trigger 14** (Trigger)
- **Trigger 15** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.LayerSequence>

## 60.1.29 LineFont\_v2



**Full Name:** Ops.Gl.LineFont\_v2

**Description:** A Simple way to write text on the screen.

**> Input Ports:**

- **Render** (Trigger)
- **Text** (String)
- **Letter Spacing** (Number)

**< Output Ports:**

- **Lines** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.LineFont\\_v2](https://cables.gl/op/Ops.Gl.LineFont_v2)

## 60.1.30 MainLoop\_v2



**Full Name:** Ops.Gl.MainLoop\_v2

**Description:** Trigger other ops once every frame to create smooth animations

**> Input Ports:**

- **FPS Limit** (Number)
- **Reduce FPS Unfocussed** (Number: Boolean)
- **Transparent** (Number: Boolean)
- **Active** (Number: Boolean)
- **Focus Canvas** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)
- **Width** (Number)
- **Height** (Number)
- **Pixel Density** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.MainLoop\\_v2](https://cables.gl/op/Ops.Gl.MainLoop_v2)

## 60.1.31 MediaRecorder\_v2



**Full Name:** Ops.Gl.MediaRecorder\_v2

**Description:** Record the renderer-output to video

**> Input Ports:**

- **Recording** (Number: Boolean)
- **Filename** (String)
- **Download Video** (Number: Boolean)
- **Mimetype Index** (Number: Integer)
- **MBit** (Number)

- **Max FPS** (Number)
- **Force FPS** (Number)
- **Audio In** (Object:AudioNode)
- **Video Canvas Id** (String)

**< Output Ports:**

- **State** (String)
- **Error** (String)
- **Final Mimetype** (String)
- **Valid Mimetypes** (Array)
- **Duration** (Number)
- **Finished Recording** (Trigger)
- **Video DataUrl** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.MediaRecorder\\_v2](https://cables.gl/op/Ops.Gl.MediaRecorder_v2)

## 60.1.32 MeshInstancer\_v4



**Full Name:** Ops.Gl.MeshInstancer\_v4

**Description:** Draw the same mesh multiple times on the GPU

**> Input Ports:**

- **Exe** (Trigger)
- **Geom** (Object:Geometry)
- **Scale** (Number)
- **Limit Instances** (Number: Boolean)
- **Limit** (Number: Integer)
- **Positions** (Array)
- **Scale Array** (Array)
- **Rotations** (Array)
- **Colors** (Array)
- **TexCoords** (Array)

**< Output Ports:**

- **Trigger Out** (Trigger)
- **Num** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.MeshInstancer\\_v4](https://cables.gl/op/Ops.Gl.MeshInstancer_v4)

## 60.1.33 MeshMorph



**Full Name:** Ops.Gl.MeshMorph

**Description:** morph from one geometry to another

**> Input Ports:**

- **Render** (Trigger)
- **Geometry** (Number: Integer)
- **Duration** (Number)
- **Index** (Number)
- **Index 2** (Number)
- **Fade** (Number)
- **Easing Index** (Number: Integer)
- **Geometry 0** (Object)
- **Geometry 1** (Object)
- **Geometry 2** (Object)
- **Geometry 3** (Object)
- **Geometry 4** (Object)
- **Geometry 5** (Object)
- **Geometry 6** (Object)
- **Geometry 7** (Object)
- **Geometry 8** (Object)
- **Geometry 9** (Object)
- **Geometry 10** (Object)
- **Geometry 11** (Object)
- **Geometry 12** (Object)
- **Geometry 13** (Object)
- **Geometry 14** (Object)
- **Geometry 15** (Object)

**< Output Ports:**

- **Finished** (booleanNumber)
- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.MeshMorph>

### 60.1.34 NormalizeScreenCoordinates



**Full Name:** Ops.Gl.NormalizeScreenCoordinates

**Description:** convert screen pixel coordinates to range 0-1

**> Input Ports:**

- **X** (Number)
- **Y** (Number)

**< Output Ports:**

- **Result X** (Number)
- **Result Y** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.NormalizeScreenCoordinates>

### 60.1.35 OrTexture



**Full Name:** Ops.Gl.OrTexture

**Description:** outputs the first valid texture of the

**> Input Ports:**

- **Texture 1** (Object:Texture)
- **Texture 2** (Object:Texture)
- **Texture 3** (Object:Texture)
- **Texture 4** (Object:Texture)
- **Texture 5** (Object:Texture)
- **Texture 6** (Object:Texture)
- **Texture 7** (Object:Texture)

- **Texture 8** (Object:Texture)

**< Output Ports:**

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.OrTexture>

### 60.1.36 Orthogonal\_v2



**Full Name:** Ops.Gl.Orthogonal\_v2

**Description:** Orthogonal projection / objects in distance don't appear smaller (isometric)

**> Input Ports:**

- **Render** (Trigger)
- **Bounds** (Number)
- **Axis Index** (Number: Integer)
- **Frustum Near** (Number)
- **Frustum Far** (Number)

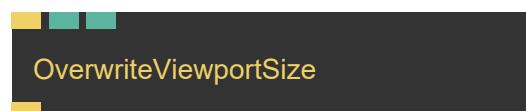
**< Output Ports:**

- **Trigger** (Trigger)
- **Ratio** (Number)
- **Width** (Number)
- **Height** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Orthogonal\\_v2](https://cables.gl/op/Ops.Gl.Orthogonal_v2)

### 60.1.37 OverwriteViewportSize



**Full Name:** Ops.Gl.OverwriteViewportSize

**Description:** Force a manually set viewport size for connected ops

**> Input Ports:**

- **Exec** (Trigger)
- **Width** (Number: Integer)
- **Height** (Number: Integer)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.OverwriteViewportSize>

## 60.1.38 Performance



**Full Name:** Ops.Gl.Performance

**Description:** Show WebGL Performance Statistics

**> Input Ports:**

- **Exe** (Trigger)
- **Active** (Number: Boolean)
- **Visible** (Number: Boolean)
- **Measure GPU** (Number: Boolean)
- **Open** (Number: Boolean)
- **Smooth Graph** (Number: Boolean)
- **Scale** (Number)
- **Size** (Number)

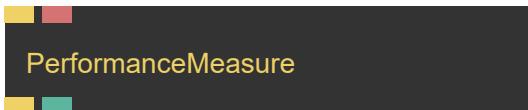
**< Output Ports:**

- **Childs** (Trigger)
- **Canvas** (Object)
- **FPS** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Performance>

## 60.1.39 PerformanceMeasure



**Full Name:** Ops.Gl.PerformanceMeasure

**Description:** Measure the time used to execute all child ops

**> Input Ports:**

- **Execute** (Trigger)
- **Name** (String)

**< Output Ports:**

- **Childs** (Trigger)
- **Time Used** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.PerformanceMeasure>

## 60.1.40 Perspective



**Full Name:** Ops.Gl.Perspective

**Description:** Adjust FOV, field of view, and frustum clipping

**> Input Ports:**

- **Render** (Trigger)
- **FOV Degrees** (Number)
- **Frustum Near** (Number)
- **Frustum Far** (Number)
- **Auto Aspect Ratio** (Number: Boolean)
- **Aspect Ratio** (Number)

**< Output Ports:**

- **Trigger** (Trigger)
- **Aspect** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Perspective>

### 60.1.41 PixelProjection\_v3



**Full Name:** Ops.Gl.PixelProjection\_v3

**Description:** Remaps world co-ordinates to a pixel co-ordinate system

#### > Input Ports:

- **Render** (Trigger)
- **Width** (Number)
- **Height** (Number)
- **Frustum Near** (Number)
- **Frustum Far** (Number)
- **Flip X** (Number: Boolean)
- **Flip Y** (Number: Boolean)
- **Zero Y** (Number: Boolean)

#### < Output Ports:

- **Trigger** (Trigger)
- **Size Width** (Number)
- **Size Height** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.PixelProjection\\_v3](https://cables.gl/op/Ops.Gl.PixelProjection_v3)

### 60.1.42 PointCollector



**Full Name:** Ops.Gl.PointCollector

**Description:** save points/coordinates in an array

#### > Input Ports:

- **Render** (Trigger)
- **Absolute** (Number: Boolean)

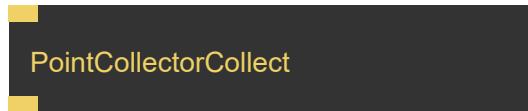
#### < Output Ports:

- **Trigger** (Trigger)
- **Points** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.PointCollector>

### 60.1.43 PointCollectorCollect



**Full Name:** Ops.Gl.PointCollectorCollect

**Description:** collect world space coordinates into an array

#### > Input Ports:

- **Render** (Trigger)

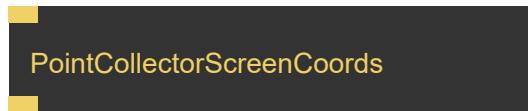
#### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.PointCollectorCollect>

### 60.1.44 PointCollectorScreenCoords



**Full Name:** Ops.Gl.PointCollectorScreenCoords

**Description:** collect screen pixel coordinates into an array

#### > Input Ports:

- **Render** (Trigger)

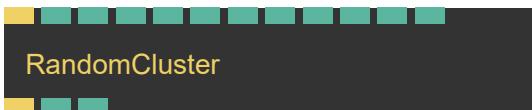
#### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.PointCollectorScreenCoords>

## 60.1.45 RandomCluster



RandomCluster

**Full Name:** Ops.Gl.RandomCluster

**Description:** Transforms objects randomly in space

### > Input Ports:

- **Exe** (Trigger)
- **Num** (Number: Integer)
- **Random Seed** (Number)
- **Round** (Number: Boolean)
- **Size** (Number)
- **ScaleX** (Number)
- **ScaleY** (Number)
- **ScaleZ** (Number)
- **Rotate X** (Number)
- **Rotate Y** (Number)
- **Rotate Z** (Number)
- **Scroll X** (Number)

### < Output Ports:

- **Trigger** (Trigger)
- **Index** (Number)
- **Rnd** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.RandomCluster>

## 60.1.46 RenderAnim\_v2



RenderAnim

**Full Name:** Ops.Gl.RenderAnim\_v2

**Description:** record, render an animation and save as webm video file or png image sequence

### > Input Ports:

- **Render** (Trigger)
- **File Type Index** (Number: Integer)
- **ZIP Multiple Files** (Number: Boolean)
- **Download Files** (Number: Boolean)
- **Filename** (String)
- **Quality** (Number)
- **Duration** (Number: Integer)
- **FPS** (Number: Integer)
- **Transparency** (Number: Boolean)
- **Use Canvas Size** (Number: Boolean)
- **Texture Width** (Number: Integer)
- **Texture Height** (Number: Integer)
- **Start** (Trigger)

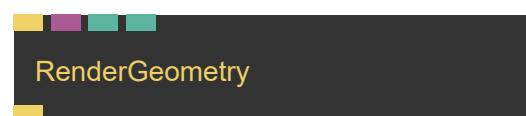
### < Output Ports:

- **Next** (Trigger)
- **Progress** (Number)
- **Frame** (Number)
- **Status** (String)
- **Started** (booleanNumber)
- **Data URL** (String)
- **Finished** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.RenderAnim\\_v2](https://cables.gl/op/Ops.Gl.RenderAnim_v2)

## 60.1.47 RenderGeometry\_v2



RenderGeometry

**Full Name:** Ops.Gl.RenderGeometry\_v2

**Description:** Render a geometry as mesh

### > Input Ports:

- **Render** (Trigger)
- **Geometry** (Object:Geometry)
- **Add Vertex Numbers** (Number: Boolean)
- **Render Mesh** (Number: Boolean)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.RenderGeometry\\_v2](https://cables.gl/op/Ops.Gl.RenderGeometry_v2)

## 60.1.48 RenderToTexture\_v3



**Full Name:** Ops.Gl.RenderToTexture\_v3

**Description:** Render into an Image

**> Input Ports:**

- **Render** (Trigger)
- **Texture Width** (Number: Integer)
- **Texture Height** (Number: Integer)
- **Auto Aspect** (Number: Boolean)
- **PixelFormat Index** (Number: Integer)
- **Depth** (Number: Boolean)
- **Clear** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)
- **Texture** (Object)
- **TextureDepth** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.RenderToTexture\\_v3](https://cables.gl/op/Ops.Gl.RenderToTexture_v3)

## 60.1.49 RenderToTextures\_v3



**Full Name:** Ops.Gl.RenderToTextures\_v3

**Description:** render to multiple textures at the same time

**> Input Ports:**

- **Render** (Trigger)
- **Texture Width** (Number: Integer)
- **Texture Height** (Number: Integer)
- **Auto Aspect** (Number: Boolean)
- **PixelFormat Index** (Number: Integer)
- **Wrap Index** (Number: Integer)
- **Clear** (Number: Boolean)
- **Texture 0 Index** (Number: Integer)
- **Texture 1 Index** (Number: Integer)
- **Texture 2 Index** (Number: Integer)
- **Texture 3 Index** (Number: Integer)
- **Texture 4 Index** (Number: Integer)
- **Texture 5 Index** (Number: Integer)
- **Texture 6 Index** (Number: Integer)
- **Texture 7 Index** (Number: Integer)

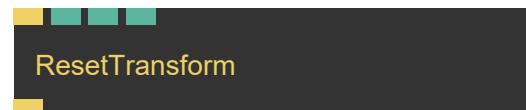
**< Output Ports:**

- **Next** (Trigger)
- **Result Texture 0** (Object)
- **Result Texture 1** (Object)
- **Result Texture 2** (Object)
- **Result Texture 3** (Object)
- **Result Texture 4** (Object)
- **Result Texture 5** (Object)
- **Result Texture 6** (Object)
- **Result Texture 7** (Object)
- **TextureDepth** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.RenderToTextures\\_v3](https://cables.gl/op/Ops.Gl.RenderToTextures_v3)

## 60.1.50 ResetTransform



**Full Name:** Ops.Gl.ResetTransform

**Description:** reset current transforms to initial value (identity)

**> Input Ports:**

- **Exe** (Trigger)
- **Reset Model Transform** (Number: Boolean)
- **Reset View Transform** (Number: Boolean)
- **Default View** (Number: Boolean)

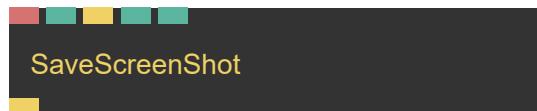
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ResetTransform>

### 60.1.51 SaveScreenShot\_v3



**Full Name:** Ops.Gl.SaveScreenShot\_v3

**Description:** Download the current screen content as png file

**> Input Ports:**

- **Filename** (String)
- **Use Canvas Size** (Number: Boolean)
- **Screenshot** (Trigger)
- **Width** (Number: Integer)
- **Height** (Number: Integer)

**< Output Ports:**

- **Finished** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.SaveScreenShot\\_v3](https://cables.gl/op/Ops.Gl.SaveScreenShot_v3)

### 60.1.52 ShowNormals\_v2



**Full Name:** Ops.Gl.ShowNormals\_v2

**Description:** visualize normals, tangents or bitangents

**> Input Ports:**

- **Render** (Trigger)
- **Draw** (Number: Boolean)
- **Geometry** (Object:Geometry)
- **Length** (Number)
- **Colorize** (Number: Boolean)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)

**< Output Ports:**

- **Trigger** (Trigger)
- **Line Geom** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShowNormals\\_v2](https://cables.gl/op/Ops.Gl.ShowNormals_v2)

### 60.1.53 SurfaceScatter\_v2



**Full Name:** Ops.Gl.SurfaceScatter\_v2

**Description:** Scatter an object on the surface of a mesh with different distribution methods

**> Input Ports:**

- **Render** (Trigger)
- **Draw** (Number: Boolean)
- **Num** (Number: Integer)
- **Geom Surface** (Object)
- **Distribution Index** (Number: Integer)
- **Selection Index** (Number: Integer)
- **Random Seed** (Number)
- **Size Min** (Number)
- **Size Max** (Number)
- **Limit** (Number: Boolean)
- **Limit Num** (Number: Integer)
- **Random Rotate** (Number: Boolean)

#### < Output Ports:

- **Next** (Trigger)
- **Positions** (Array)
- **Scale** (Array)
- **Quaternions** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.SurfaceScatter\\_v2](https://cables.gl/op/Ops.Gl.SurfaceScatter_v2)

## 60.1.54 TextMeshMSDF\_v2



**Full Name:** Ops.Gl.TextMeshMSDF\_v2

**Description:** draw text using the FontMSDF operator

#### > Input Ports:

- **Render** (Trigger)
- **Text** (String)
- **Scale** (Number)
- **Letter Spacing** (Number)
- **Line Height** (Number)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)
- **SDF** (Number: Boolean)
- **Smoothing** (Number)
- **Border** (Number: Boolean)
- **Border Width** (Number)
- **Smoothness** (Number)
- **Border R** (Number)
- **Border G** (Number)
- **Border B** (Number)
- **Shadow** (Number: Boolean)
- **Texture Color** (Object:Texture)
- **Texture Mask** (Object:Texture)
- **Positions** (Array)

#### > Output Ports:

- **Scalings** (Array)
- **Rotations** (Array)
- **Colors** (Array)
- **Premultiply** (Number: Boolean)

#### < Output Ports:

- **Next** (Trigger)
- **Positions Original** (Array)
- **Scales** (Array)
- **Num Lines** (Number)
- **Width** (Number)
- **Height** (Number)
- **Start Y** (Number)
- **Num Chars** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.TextMeshMSDF\\_v2](https://cables.gl/op/Ops.Gl.TextMeshMSDF_v2)

## 60.1.55 Texture\_v2



**Full Name:** Ops.Gl.Texture\_v2

**Description:** Load an image as a texture

#### > Input Ports:

- **File** (String)
- **Wrap Index** (Number: Integer)
- **Flip** (Number: Boolean)
- **Active** (Number: Boolean)
- **Save Memory** (Number: Boolean)
- **Add Cachebuster** (Number: Boolean)
- **Reload** (Trigger)

#### < Output Ports:

- **Texture** (Object)
- **Width** (Number)
- **Height** (Number)
- **Aspect Ratio** (Number)

- **Loaded** (booleanNumber)
- **Loading** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Texture\\_v2](https://cables.gl/op/Ops.Gl.Texture_v2)

## 60.1.56 TextureArray



**Full Name:** Ops.Gl.TextureArray

**Description:** create an array of textures

**> Input Ports:**

- **Texture 0** (Object:Texture)
- **Texture 1** (Object:Texture)
- **Texture 2** (Object:Texture)
- **Texture 3** (Object:Texture)
- **Texture 4** (Object:Texture)
- **Texture 5** (Object:Texture)
- **Texture 6** (Object:Texture)
- **Texture 7** (Object:Texture)
- **Texture 8** (Object:Texture)
- **Texture 9** (Object:Texture)
- **Texture 10** (Object:Texture)
- **Texture 11** (Object:Texture)
- **Texture 12** (Object:Texture)
- **Texture 13** (Object:Texture)
- **Texture 14** (Object:Texture)

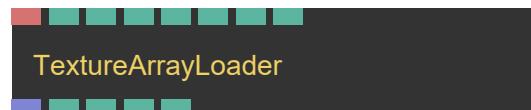
**< Output Ports:**

- **Array** (Array)
- **Count** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.TextureArray>

## 60.1.57 TextureArrayLoader\_v2



**Full Name:** Ops.Gl.TextureArrayLoader\_v2

**Description:** load multiple images into an array

**> Input Ports:**

- **Url** (String)
- **Left Pad** (Number: Boolean)
- **Index Start** (Number: Integer)
- **Index End** (Number: Integer)
- **Filter Index** (Number: Integer)
- **Wrap Index** (Number: Integer)
- **Flip** (Number: Boolean)
- **UnpackPreMultipliedAlpha** (Number: Boolean)

**< Output Ports:**

- **TextureArray** (Array)
- **Width** (Number)
- **Height** (Number)
- **Loading** (booleanNumber)
- **Aspect Ratio** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.TextureArrayLoader\\_v2](https://cables.gl/op/Ops.Gl.TextureArrayLoader_v2)

## 60.1.58 TextureArrayLoaderFromArray\_v3



**Full Name:** Ops.Gl.TextureArrayLoaderFromArray\_v3

**Description:** load multiple texture from filenames given as an array

**> Input Ports:**

- **Urls** (Array)
- **Filter Index** (Number: Integer)

- **Wrap Index** (Number: Integer)
- **Flip** (Number: Boolean)
- **UnpackPremultipliedAlpha** (Number: Boolean)
- **Caching** (Number: Boolean)
- **Asset In Patch** (Number: Boolean)

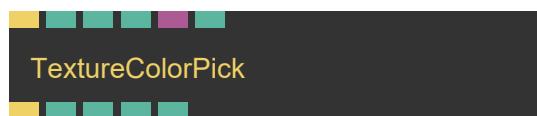
**< Output Ports:**

- **TextureArray** (Array)
- **Width** (Number)
- **Height** (Number)
- **Loading** (booleanNumber)
- **Aspect Ratio** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.TextureArrayLoaderFromArray\\_v3](https://cables.gl/op/Ops.Gl.TextureArrayLoaderFromArray_v3)

## 60.1.59 TextureColorPick



**Full Name:** Ops.Gl.TextureColorPick

**Description:** get the color of a pixel in a texture

**> Input Ports:**

- **Update** (Trigger)
- **X** (Number: Integer)
- **Y** (Number: Integer)
- **Texture** (Object:Texture)
- **Active** (Number: Boolean)

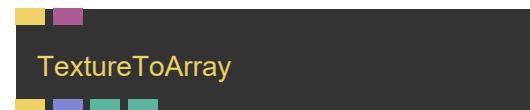
**< Output Ports:**

- **Trigger** (Trigger)
- **Red** (Number)
- **Green** (Number)
- **Blue** (Number)
- **Alpha** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.TextureColorPick>

## 60.1.60 TextureToArray\_v4



**Full Name:** Ops.Gl.TextureToArray\_v4

**Description:** extract colors from a texture

**> Input Ports:**

- **Update** (Trigger)
- **Texture** (Object)

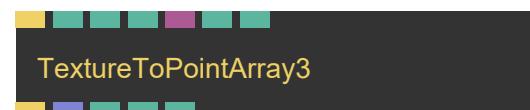
**< Output Ports:**

- **Trigger** (Trigger)
- **Colors** (Array)
- **Floating Point** (booleanNumber)
- **Num Pixel** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.TextureToArray\\_v4](https://cables.gl/op/Ops.Gl.TextureToArray_v4)

## 60.1.61 TextureToPointArray3



**Full Name:** Ops.Gl.TextureToPointArray3

**Description:** generate an array3 of grid positions from a texture

**> Input Ports:**

- **Update** (Trigger)
- **Center** (Number: Boolean)
- **Threshold Remove** (Number)
- **Z Multiply** (Number)
- **Texture** (Object)
- **Width** (Number)
- **Height** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

- **Points** (Array)
- **Total Points** (Number)
- **Min Z** (Number)
- **Max Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.TextureToPointArray3>

## 60.1.62 TextureToRandomPoints



**Full Name:** Ops.Gl.TextureToRandomPoints

**Description:** Create points by sampling texture

**> Input Ports:**

- **Update** (Trigger)
- **Num Points** (Number: Integer)
- **Seed** (Number)
- **Z Position Index** (Number: Integer)
- **Z Multiply** (Number)
- **Texture** (Object)

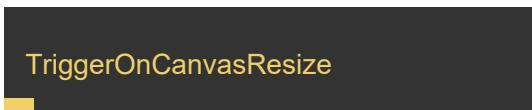
**< Output Ports:**

- **Trigger** (Trigger)
- **Points** (Array)
- **NumPoints** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.TextureToRandomPoints>

## 60.1.63 TriggerOnCanvasResize



**Full Name:** Ops.Gl.TriggerOnCanvasResize

**Description:** will trigger when canvas was resized

**> Input Ports:**

- Visit [Ops.Gl.TriggerOnCanvasResize documentation](#) for input port details

**< Output Ports:**

- **Resized** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.TriggerOnCanvasResize>

## 60.1.64 ValidTexture



**Full Name:** Ops.Gl.ValidTexture

**Description:** output current input texture or a default texture

**> Input Ports:**

- **Texture** (Object:Texture)

**< Output Ports:**

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ValidTexture>

## 60.1.65 ViewPortSize



**Full Name:** Ops.Gl.ViewPortSize

**Description:** Outputs current viewport size

**> Input Ports:**

- **Exec** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **X** (Number)

- **Y** (Number)
- **Width** (Number)
- **Height** (Number)

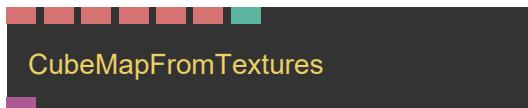
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ViewPortSize>

# 61 Ops.Gl.CubeMap

## 61.1 Ops.Gl.CubeMap

### 61.1.1 CubeMapFromTextures\_v2



**Full Name:** Ops.Gl.CubeMap.CubeMapFromTextures\_v2

**Description:** generate a cubemap from 6 textures

**> Input Ports:**

- **Posx** (String)
- **Negx** (String)
- **Posy** (String)
- **Negy** (String)
- **Posz** (String)
- **Negz** (String)
- **Flip Y** (Number: Boolean)

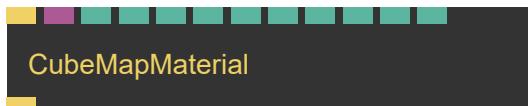
**< Output Ports:**

- **Cubemap** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.CubeMap.CubeMapFromTextures\\_v2](https://cables.gl/op/Ops.Gl.CubeMap.CubeMapFromTextures_v2)

### 61.1.2 CubeMapMaterial\_v2



**Full Name:** Ops.Gl.CubeMap.CubeMapMaterial\_v2

**Description:** use a cubemap or equirectangular texture as a material

**> Input Ports:**

- **Render** (Trigger)

- **Cubemap** (Object)
- **Use Reflection** (Number: Boolean)
- **Blur** (Number)
- **Rotation** (Number)
- **Flip X** (Number: Boolean)
- **Flip Y** (Number: Boolean)
- **Flip Z** (Number: Boolean)
- **Colorize** (Number: Boolean)
- **R** (Number)
- **G** (Number)
- **B** (Number)

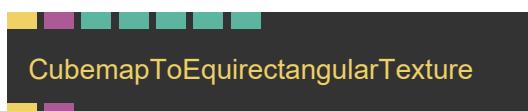
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.CubeMap.CubeMapMaterial\\_v2](https://cables.gl/op/Ops.Gl.CubeMap.CubeMapMaterial_v2)

## 61.1.3 CubemapToEquirectangularTexture\_v2



**Full Name:** Ops.Gl.CubeMap.CubemapToEquirectangularTexture\_v2

**Description:** visualize cubemap as folded texture or equirectangular texture

**> Input Ports:**

- **In Trigger** (Trigger)
- **Cubemap** (Object)
- **Projection Index** (Number: Integer)
- **Format Index** (Number: Integer)
- **Filter Index** (Number: Integer)
- **Width** (Number: Integer)
- **Height** (Number: Integer)

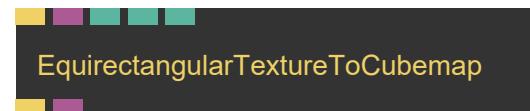
**< Output Ports:**

- **Out Trigger** (Trigger)
- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.CubeMap.CubemapToEquirectangularTexture\\_v2](https://cables.gl/op/Ops.Gl.CubeMap.CubemapToEquirectangularTexture_v2)

## 61.1.4 EquirectangularTextureToCubemap



**Full Name:** Ops.Gl.CubeMap.EquirectangularTextureToCubemap

**Description:** convert an equirectangular map to a cubemap

**> Input Ports:**

- **Trigger In** (Trigger)
- **Equirectangular Map** (Object:Texture)
- **Cubemap Size Index** (Number: Integer)
- **Advanced** (Number: Boolean)
- **Filter Index** (Number: Integer)

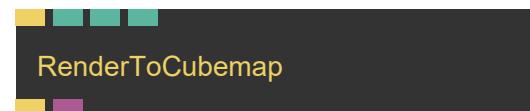
**< Output Ports:**

- **Trigger Out** (Trigger)
- **Cubemap Projection** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.CubeMap.EquirectangularTextureToCube map>

## 61.1.5 RenderToCubemap\_v3



**Full Name:** Ops.Gl.CubeMap.RenderToCubemap\_v3

**Description:** render a scene into a cubemap

**> Input Ports:**

- **Render** (Trigger)
- **Size Index** (Number: Integer)
- **PixelFormat Index** (Number: Integer)
- **MSAA Index** (Number: Integer)

#### < Output Ports:

- **Next** (Trigger)
- **Cubemap** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.CubeMap.RenderToCubemap\\_v3](https://cables.gl/op/Ops.Gl.CubeMap.RenderToCubemap_v3)

### 61.1.6 Skybox



**Full Name:** Ops.Gl.CubeMap.Skybox

**Description:** render an equirectangular map or a cubemap as scene background

#### > Input Ports:

- **Trigger In** (Trigger)
- **Render** (Number: Boolean)
- **Skybox** (Object:Texture)
- **Rotate** (Number)
- **RGBE Format** (Number: Boolean)
- **Exposure** (Number)
- **Gamma** (Number)

#### < Output Ports:

- **Trigger Out** (Trigger)

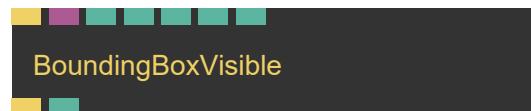
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.CubeMap.Skybox>

## 62 Ops.Gl.Geometry

### 62.1 Ops.Gl.Geometry

#### 62.1.1 BoundingBoxVisible



**Full Name:** Ops.Gl.Geometry.BoundingBoxVisible

**Description:** Test if a boundingbox could be visible in the current viewport

#### > Input Ports:

- **Exec** (Trigger)
- **Boundings** (Object)
- **Active** (Number: Boolean)
- **Draw** (Number: Boolean)
- **Width** (Number)
- **Height** (Number)
- **Length** (Number)

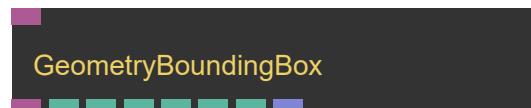
#### < Output Ports:

- **Next** (Trigger)
- **Visible** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Geometry.BoundingBoxVisible>

#### 62.1.2 GeometryBoundingBox



**Full Name:** Ops.Gl.Geometry.GeometryBoundingBox

**Description:** Calculate a bounding box from a geometry

#### > Input Ports:

- **Geometry** (Object)

**< Output Ports:**

- **Boundings** (Object)
- **Min X** (Number)
- **Min Y** (Number)
- **Min Z** (Number)
- **Max X** (Number)
- **Max Y** (Number)
- **Max Z** (Number)
- **MaxMin Points** (Array)

**Example Patch:** Open in Editor

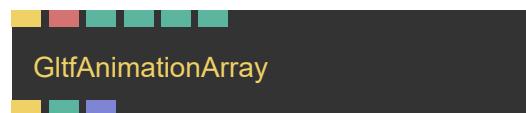
**Docs:** <https://cables.gl/op/Ops.Gl.Geometry.GeometryBoundingBox>

## 63 Ops.Gl.GLTF

---

### 63.1 Ops.Gl.GLTF

#### 63.1.1 GltfAnimationArray



GltfAnimationArray

**Full Name:** Ops.Gl.GLTF.GltfAnimationArray

**Description:** Convert an animation into an array of coordinates

**> Input Ports:**

- **Render** (Trigger)
- **Node Name** (String)
- **Steps** (Number: Integer)
- **Full Animation** (Number: Boolean)
- **Start** (Number)
- **Length** (Number)

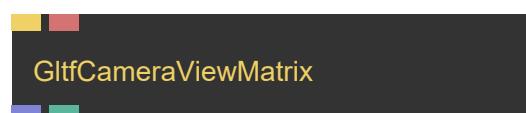
**< Output Ports:**

- **Next** (Trigger)
- **Found** (booleanNumber)
- **Positions** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GLTF.GltfAnimationArray>

#### 63.1.2 GltfCameraViewMatrix



GltfCameraViewMatrix

**Full Name:** Ops.Gl.GLTF.GltfCameraViewMatrix

**Description:** get view matrix from a gltf camera

**> Input Ports:**

- **Update** (Trigger)
- **Node Name** (String)

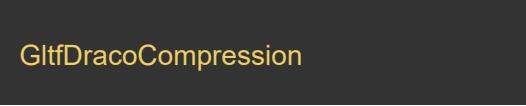
**< Output Ports:**

- **Matrix** (Array)
- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GLTF.GltfCameraViewMatrix>

### 63.1.3 GltfDracoCompression



GltfDracoCompression

**Full Name:** Ops.Gl.GLTF.GltfDracoCompression

**Description:** gltf draco compression library

**> Input Ports:**

- Visit *Ops.Gl.GLTF.GltfDracoCompression documentation* for input port details

**< Output Ports:**

- Visit *Ops.Gl.GLTF.GltfDracoCompression documentation* for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GLTF.GltfDracoCompression>

### 63.1.4 GltfGeometry



GltfGeometry

**Full Name:** Ops.Gl.GLTF.GltfGeometry

**Description:** expose geometry from gltf meshes, also possible to expose sub-material geometries

**> Input Ports:**

- **Update** (Trigger)

- **Name** (String)
- **Submesh** (Number: Integer)

**< Output Ports:**

- **Next** (Trigger)
- **Geometry** (Object)
- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GLTF.GltfGeometry>

### 63.1.5 GltfHierarchy



GltfHierarchy

**Full Name:** Ops.Gl.GLTF.GltfHierarchy

**Description:** export array of positions from a hierarchy of a branch structure in a gltf, e.g. a skeleton bones

**> Input Ports:**

- **Trigger** (Trigger)
- **Node Name** (String)

**< Output Ports:**

- **Next** (Trigger)
- **Bones Lines** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GLTF.GltfHierarchy>

### 63.1.6 GltfInfo



GltfInfo

**Full Name:** Ops.Gl.GLTF.GltfInfo

**Description:** output some infos about the current parent GLTF scene

**> Input Ports:**

- **Exec** (Trigger)

**< Output Ports:**

- **Num Nodes** (Number)
- **Num Cams** (Number)
- **FileUrl** (String)
- **FileName** (String)
- **Camera Names** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GLTF.GltfInfo>

### 63.1.7 GltfMeshSequence\_v2



**Full Name:** Ops.Gl.GLTF.GltfMeshSequence\_v2

**Description:** switch between meshes e.g. like a stop motion animation

**> Input Ports:**

- **Render** (Trigger)
- **Index** (Number: Integer)
- **Node Name** (String)
- **Transformation** (Number: Boolean)
- **Ignore Material** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)
- **Found** (Number)
- **Current Index** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.GLTF.GltfMeshSequence\\_v2](https://cables.gl/op/Ops.Gl.GLTF.GltfMeshSequence_v2)

### 63.1.8 GltfMorphTargets



**Full Name:** Ops.Gl.GLTF.GltfMorphTargets

**Description:** render weighted morph targets/shape keys from a gltf file

**> Input Ports:**

- **Render** (Trigger)
- **Node Name** (String)
- **Scene Time** (Number: Boolean)
- **Time** (Number)
- **Submesh** (Number: Integer)
- **Target Weights** (Array)

**< Output Ports:**

- **Found Node** (booleanNumber)
- **Found Skin** (booleanNumber)
- **Target Names** (Array)
- **MorphTargets Tex** (Object)
- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GLTF.GltfMorphTargets>

### 63.1.9 GltfNode\_v2



**Full Name:** Ops.Gl.GLTF.GltfNode\_v2

**Description:** Control a single node from the GLTFscene op

**> Input Ports:**

- **Render** (Trigger)
- **Node Name** (String)
- **Transformation** (Number: Boolean)
- **Draw Mesh** (Number: Boolean)
- **Draw Childs** (Number: Boolean)
- **Ignore Material** (Number: Boolean)
- **Use Scene Time** (Number: Boolean)
- **Time** (Number)

**< Output Ports:**

- **Next** (Trigger)

- **Geometry** (Object)
- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.GLTF.GltfNode\\_v2](https://cables.gl/op/Ops.Gl.GLTF.GltfNode_v2)

### 63.1.10 GltfNodeSineAnim



**Full Name:** Ops.Gl.GLTF.GltfNodeSineAnim

**Description:** sine animate gltf nodes by a filter

**> Input Ports:**

- **Update** (Trigger)
- **Filter** (String)
- **Time** (Number)
- **Offset** (Number)
- **Amplitude** (Number)
- **Axis X** (Number)
- **Axis Y** (Number)
- **Axis Z** (Number)

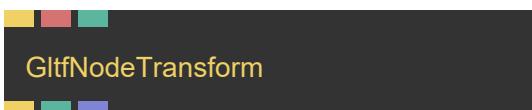
**< Output Ports:**

- **Next** (Trigger)
- **Found** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GLTF.GltfNodeSineAnim>

### 63.1.11 GltfNodeTransform\_v2



**Full Name:** Ops.Gl.GLTF.GltfNodeTransform\_v2

**Description:** Get the transform from the GLTFscene op

**> Input Ports:**

- **Render** (Trigger)
- **Node Name** (String)
- **Set Matrix** (Number: Boolean)

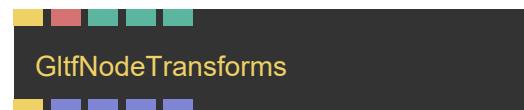
**< Output Ports:**

- **Next** (Trigger)
- **Found** (booleanNumber)
- **Matrix** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.GLTF.GltfNodeTransform\\_v2](https://cables.gl/op/Ops.Gl.GLTF.GltfNodeTransform_v2)

### 63.1.12 GltfNodeTransforms\_v3



**Full Name:** Ops.Gl.GLTF.GltfNodeTransforms\_v3

**Description:** output all transformations of nodes starting with [search]

**> Input Ports:**

- **Render** (Trigger)
- **Search** (String)
- **Order Index** (Number: Integer)
- **Space Index** (Number: Integer)
- **Time** (Number)

**< Output Ports:**

- **Next** (Trigger)
- **Positions** (Array)
- **Scale** (Array)
- **Rotation** (Array)
- **Names** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.GLTF.GltfNodeTransforms\\_v3](https://cables.gl/op/Ops.Gl.GLTF.GltfNodeTransforms_v3)

## 63.1.13 GltfScene\_v4



**Full Name:** Ops.Gl.GLTF.GltfScene\_v4

**Description:** Load GLTF/GLB 3d files

### > Input Ports:

- **Render** (Trigger)
- **Glb File** (String)
- **Draw** (Number: Boolean)
- **Camera Index** (Number: Integer)
- **Animation** (String)
- **Show Structure** (Trigger)
- **Rescale** (Number: Boolean)
- **Rescale Size** (Number)
- **Time** (Number)
- **Sync To Timeline** (Number: Boolean)
- **Loop** (Number: Boolean)
- **Materials** (Object)
- **Hide Nodes** (Array)
- **Use Material Properties** (Number: Boolean)
- **Active** (Number: Boolean)

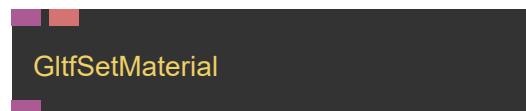
### < Output Ports:

- **Render Before** (Trigger)
- **Next** (Trigger)
- **Generator** (String)
- **GLTF Version** (Number)
- **GLTF Extensions Used** (Array)
- **Anim Length** (Number)
- **Anim Time** (Number)
- **Json** (Object)
- **Anims** (Array)
- **BoundingPoints** (Array)
- **Bounds** (Object)
- **Finished** (Trigger)
- **Loading** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.GLTF.GltfScene\\_v4](https://cables.gl/op/Ops.Gl.GLTF.GltfScene_v4)

## 63.1.14 GltfSetMaterial



**Full Name:** Ops.Gl.GLTF.GltfSetMaterial

**Description:** Assigns a material to a node inside of the gltfScene op

### > Input Ports:

- **Shader** (Object:Shader)
- **Material Name** (String)

### < Output Ports:

- **Material** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GLTF.GltfSetMaterial>

## 63.1.15 GltfSkin



**Full Name:** Ops.Gl.GLTF.GltfSkin

**Description:** render a skinned mesh (bone/rigging/rigged animation)

### > Input Ports:

- **Render** (Trigger)
- **Node Name** (String)
- **Scene Time** (Number: Boolean)
- **Time** (Number)
- **Blend Anims** (Array)

### < Output Ports:

- **Found Node** (booleanNumber)
- **Found Skin** (booleanNumber)
- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GLTF.GltfSkin>

### 63.1.16 GltfTexture



**Full Name:** Ops.Gl.GLTF.GltfTexture

**Description:** Load textures from inside a .glb file

#### > Input Ports:

- **Render** (Trigger)
- **Name** (String)
- **Filter Index** (Number: Integer)
- **Wrap Index** (Number: Integer)
- **Anisotropic Index** (Number: Integer)
- **Flip** (Number: Boolean)
- **Pre Multiplied Alpha** (Number: Boolean)

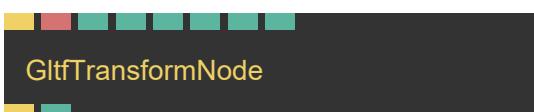
#### < Output Ports:

- **Texture** (Object)
- **Width** (Number)
- **Height** (Number)
- **Type** (String)
- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GLTF.GltfTexture>

### 63.1.17 GltfTransformNode



**Full Name:** Ops.Gl.GLTF.GltfTransformNode

**Description:** set transformation of a gltf node

#### > Input Ports:

- **Render** (Trigger)
- **Node Name** (String)
- **Translate X** (Number)
- **Translate Y** (Number)
- **Translate Z** (Number)
- **Rotation X** (Number)
- **Rotation Y** (Number)
- **Rotation Z** (Number)

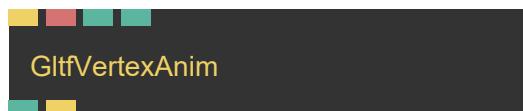
#### < Output Ports:

- **Next** (Trigger)
- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GLTF.GltfTransformNode>

### 63.1.18 GltfVertexAnim



**Full Name:** Ops.Gl.GLTF.GltfVertexAnim

**Description:** play gltf vertex anim directly with its own timing

#### > Input Ports:

- **Render** (Trigger)
- **Node Name** (String)
- **Scene Time** (Number: Boolean)
- **Time** (Number)

#### < Output Ports:

- **Found Node** (Number)
- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.GLTF.GltfVertexAnim>

# 64 Ops.Gl.ImageCompose

## 64.1 Ops.Gl.ImageCompose

### 64.1.1 Alpha



**Full Name:** Ops.Gl.ImageCompose.Alpha

**Description:** Modify current alpha-opacity

#### > Input Ports:

- **Render** (Trigger)
- **Amount** (Number)
- **Clamp** (Number: Boolean)

#### < Output Ports:

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Alpha>

### 64.1.2 AlphaMask\_v2



**Full Name:** Ops.Gl.ImageCompose.AlphaMask\_v2

**Description:** Set alphachannel of current imagecompose via a texture mask

#### > Input Ports:

- **Render** (Trigger)
- **Amount** (Number)
- **Invert** (Number: Boolean)
- **Image** (Object:Texture)
- **Method Index** (Number: Integer)

#### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.AlphaMask\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.AlphaMask_v2)

### 64.1.3 BarrelDistortion\_v3



**Full Name:** Ops.Gl.ImageCompose.BarrelDistortion\_v3

**Description:** Simulate fisheye effect

#### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Intensity** (Number)

#### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.BarrelDistortion\\_v3](https://cables.gl/op/Ops.Gl.ImageCompose.BarrelDistortion_v3)

### 64.1.4 Blur



**Full Name:** Ops.Gl.ImageCompose.Blur

**Description:** Blur the pixels of an image

#### > Input Ports:

- **Render** (Trigger)
- **Amount** (Number)
- **Direction Index** (Number: Integer)
- **Direction** (String)

- **Fast** (Number: Boolean)
- **Mask** (Object:Texture)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Blur>

## 64.1.5 Border\_v2



**Full Name:** Ops.Gl.ImageCompose.Border\_v2

**Description:** Draws a Border (rectangular frame) around the current Image-Compose

**> Input Ports:**

- **Render** (Trigger)
- **Width** (Number)
- **Blend Mode Index** (Number: Integer)
- **Alpha Mask Index** (Number: Integer)
- **Amount** (Number)
- **Smooth** (Number: Boolean)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **Side A** (Number)
- **Side B** (Number)
- **Side C** (Number)
- **Side D** (Number)

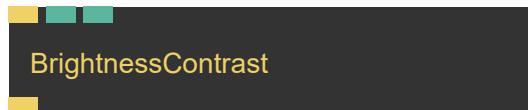
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Border\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Border_v2)

## 64.1.6 BrightnessContrast



**Full Name:** Ops.Gl.ImageCompose.BrightnessContrast

**Description:** adjust image brightness and contrast

**> Input Ports:**

- **Render** (Trigger)
- **Contrast** (Number)
- **Brightness** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.BrightnessContrast>

## 64.1.7 BulgePinch



**Full Name:** Ops.Gl.ImageCompose.BulgePinch

**Description:** bulge and pinch an image (deform/stretch/distort)

**> Input Ports:**

- **Render** (Trigger)
- **Radius** (Number)
- **Strength** (Number)
- **Center X** (Number)
- **Center Y** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.BulgePinch>

## 64.1.8 CheckerBoard\_v2



**Full Name:** Ops.Gl.ImageCompose.CheckerBoard\_v2

**Description:** Draw a checkerboard pattern

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Alpha Mask Index** (Number: Integer)
- **Amount** (Number)
- **Square** (Number: Boolean)
- **Num X** (Number)
- **Num Y** (Number)
- **Rotate** (Number)
- **Centered** (Number: Boolean)

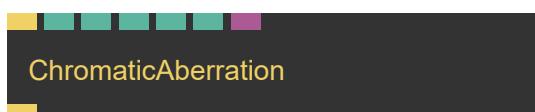
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.CheckerBoard\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.CheckerBoard_v2)

## 64.1.9 ChromaticAberration\_v2



**Full Name:** Ops.Gl.ImageCompose.CromaticAberration\_v2

**Description:** simulating lens effect by shifting rgb channels

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Pixel** (Number)
- **Lens Distort** (Number)

- **Smooth** (Number: Boolean)

- **Mask** (Object:Texture)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.CromaticAberration\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.CromaticAberration_v2)

## 64.1.10 CircleTexture\_v4



**Full Name:** Ops.Gl.ImageCompose.CircleTexture\_v4

**Description:** Draw 2d circle into texture

**> Input Ports:**

- **Render** (Trigger)
- **Amount** (Number)
- **Blend Mode Index** (Number: Integer)
- **Alpha Mask Index** (Number: Integer)
- **Size** (Number)
- **Inner** (Number)
- **Stretch X** (Number)
- **Stretch Y** (Number)
- **Pos X** (Number)
- **Pos Y** (Number)
- **FallOff Index** (Number: Integer)
- **Fade Out** (Number)
- **Warn Overflow** (Number: Boolean)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.CircleTexture\\_v4](https://cables.gl/op/Ops.Gl.ImageCompose.CircleTexture_v4)

## 64.1.11 ClampTexture\_v2



ClampTexture

**Full Name:** Ops.Gl.ImageCompose.ClampTexture\_v2

**Description:** Clamps a texture to min and max values - Also has remap modes

### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Mode Index** (Number: Integer)
- **R** (Number: Boolean)
- **R Min** (Number)
- **R Max** (Number)
- **G** (Number: Boolean)
- **G Min** (Number)
- **G Max** (Number)
- **B** (Number: Boolean)
- **B Min** (Number)
- **B Max** (Number)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.ClampTexture\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.ClampTexture_v2)

## 64.1.12 Clarity



Clarity

**Full Name:** Ops.Gl.ImageCompose.Clarity

**Description:** Increase contrast in midtones

### > Input Ports:

- **Render** (Trigger)

- **Amount** (Number)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Clarity>

## 64.1.13 Color\_v2



Color

**Full Name:** Ops.Gl.ImageCompose.Color\_v2

**Description:** fill image using a color (overlay)

### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Alpha Mask Index** (Number: Integer)
- **Mask** (Object:Texture)
- **Mask Invert** (Number: Boolean)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Color\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Color_v2)

## 64.1.14 ColorBalance\_v2



ColorBalance

**Full Name:** Ops.Gl.ImageCompose.ColorBalance\_v2

**Description:** change intensity of r,g,b channels

**> Input Ports:**

- **Render** (Trigger)
- **Tone Index** (Number: Integer)
- **R** (Number)
- **G** (Number)
- **B** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.ColorBalance\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.ColorBalance_v2)

## 64.1.15 ColorChannel\_v2



**Full Name:** Ops.Gl.ImageCompose.ColorChannel\_v2

**Description:** enable disable RGB color channels

**> Input Ports:**

- **Render** (Trigger)
- **ChannelR** (Number: Boolean)
- **ChannelG** (Number: Boolean)
- **ChannelB** (Number: Boolean)
- **ChannelA** (Number: Boolean)
- **Mono** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.ColorChannel\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.ColorChannel_v2)

## 64.1.16 ColorMap\_v2



**Full Name:** Ops.Gl.ImageCompose.ColorMap\_v2

**Description:** colorize a black and white image using a gradient texture

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Gradient** (Object:Texture)
- **Method Index** (Number: Integer)
- **Min** (Number)
- **Max** (Number)
- **Position** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.ColorMap\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.ColorMap_v2)

## 64.1.17 Denoise



**Full Name:** Ops.Gl.ImageCompose.Denoise

**Description:** Denoise texture effect - used to smooth out noisy images

**> Input Ports:**

- **Render** (Trigger)
- **Exponent** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Denoise>

### 64.1.18 DepthTexture\_v2



**Full Name:** Ops.Gl.ImageCompose.DepthTexture\_v2

**Description:** draw the content of a depth texture

#### > Input Ports:

- **Render** (Trigger)
- **Image** (Object:Texture)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Farplane** (Number)
- **Nearplane** (Number)
- **Invert** (Number: Boolean)

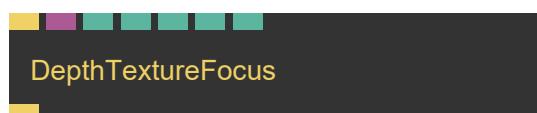
#### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.DepthTexture\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.DepthTexture_v2)

### 64.1.19 DepthTextureFocus\_v2



**Full Name:** Ops.Gl.ImageCompose.DepthTextureFocus\_v2

**Description:** draws a gradient from white to black back to white over distance of the scene

#### > Input Ports:

- **Render** (Trigger)
- **Depth Texture** (Object)
- **Focus** (Number)
- **focus distance** (in world space)

- **Width** (Number)
- **width of the focus** (in world space)
- **Invert** (Number: Boolean)
- **Nearplane** (Number)
- **Farplane** (Number)

#### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.DepthTextureFocus\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.DepthTextureFocus_v2)

### 64.1.20 Desaturate



**Full Name:** Ops.Gl.ImageCompose.Desaturate

**Description:** Remove colors from image / greyscale

#### > Input Ports:

- **Render** (Trigger)
- **Amount** (Number)
- **Mask** (Object)
- **Invert Mask** (Number: Boolean)

#### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Desaturate>

### 64.1.21 Dither\_v2



**Full Name:** Ops.Gl.ImageCompose.Dither\_v2

**Description:** convert color to black and white patterns

#### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Threshold** (Number)
- **Strength** (Number)
- **Mask** (Object:Texture)

#### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.DrawImage\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.DrawImage_v2)

### 64.1.22 DrawImage\_v3



**Full Name:** Ops.Gl.ImageCompose.DrawImage\_v3

**Description:** Draws an image into a composition

#### > Input Ports:

- **Render** (Trigger)
- **BlendMode Index** (Number: Integer)
- **Amount** (Number)
- **Image** (Object:Texture)
- **Premultiplied** (Number: Boolean)
- **Alpha Mask** (Number: Boolean)
- **RemoveAlphaSrc** (Number: Boolean)
- **Mask** (Object:Texture)
- **Mask Src Index** (Number: Integer)
- **Invert Alpha Channel** (Number: Boolean)
- **Aspect Ratio** (Number: Boolean)
- **Stretch Axis Index** (Number: Integer)
- **Position** (Number)
- **Crop** (Number: Boolean)
- **Flip X** (Number: Boolean)
- **Flip Y** (Number: Boolean)
- **Transform** (Number: Boolean)

#### > Input Ports:

- **Scale X** (Number)
- **Scale Y** (Number)
- **Position X** (Number)
- **Position Y** (Number)
- **Rotation** (Number)
- **Clip Repeat** (Number: Boolean)

#### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.DrawImage\\_v3](https://cables.gl/op/Ops.Gl.ImageCompose.DrawImage_v3)

### 64.1.23 EdgeDetection\_v4



**Full Name:** Ops.Gl.ImageCompose.EdgeDetection\_v4

**Description:** Draw only the edges of an image

#### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Strength** (Number)
- **Width** (Number)
- **Mul Color** (Number)

#### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.EdgeDetection\\_v4](https://cables.gl/op/Ops.Gl.ImageCompose.EdgeDetection_v4)

### 64.1.24 Emboss



**Full Name:** Ops.Gl.ImageCompose.Emboss

**Description:** Emboss / bevel effect

**> Input Ports:**

- **Render** (Trigger)
- **Strength** (Number)
- **Clear** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Emboss>

## 64.1.25 FastBlur\_v2



**Full Name:** Ops.Gl.ImageCompose.FastBlur\_v2

**Description:** Blurs a texture - simple and fast

**> Input Ports:**

- **Render** (Trigger)
- **Passes** (Number: Integer)
- **Clamp** (Number: Boolean)
- **Direction Index** (Number: Integer)
- **Mask** (Object:Texture)
- **Mask Invert** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.FastBlur\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.FastBlur_v2)

## 64.1.26 Flip



**Full Name:** Ops.Gl.ImageCompose.Flip

**Description:** flip the image on x or y axis

**> Input Ports:**

- **Render** (Trigger)
- **X** (Number: Boolean)
- **Y** (Number: Boolean)

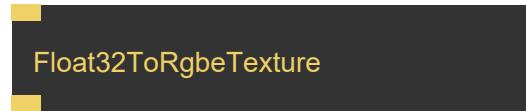
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Flip>

## 64.1.27 Float32ToRgbeTexture



**Full Name:** Ops.Gl.ImageCompose.Float32ToRgbeTexture

**Description:** Convert a Float32 bit/HDR texture to RGBE format (only positive numbers)

**> Input Ports:**

- **Render** (Trigger)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Float32ToRgbeTexture>

## 64.1.28 Fog\_v4



**Full Name:** Ops.Gl.ImageCompose.Fog\_v4

**Description:** add post processing fog (nebula) to a scene

### > Input Ports:

- **Render** (Trigger)
- **BlendMode Index** (Number: Integer)
- **Amount** (Number)
- **Depth Texture** (Object:Texture)
- **Gradient Texture** (Object:Texture)
- **Background Texture** (Object:Texture)
- **Fog Start** (Number)
- **Fog End** (Number)
- **Fog Density** (Number)
- **Ignore Infinity** (Number: Boolean)
- **Nearplane** (Number)
- **Farplane** (Number)
- **Fog R** (Number)
- **Fog G** (Number)
- **Fog B** (Number)
- **Fog A** (Number)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Fog\\_v4](https://cables.gl/op/Ops.Gl.ImageCompose.Fog_v4)

## 64.1.29 FXAA



**Full Name:** Ops.Gl.ImageCompose.FXAA

**Description:** post processing antialiasing

### > Input Ports:

- **Render** (Trigger)
- **Span Index** (Number: Integer)
- **ReduceMin** (Number)
- **ReduceMul** (Number)
- **Use Viewport Size** (Number: Boolean)
- **Width** (Number: Integer)
- **Height** (Number: Integer)

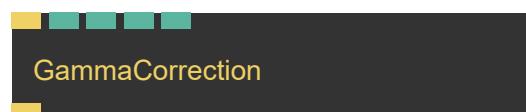
### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.FXAA>

## 64.1.30 GammaCorrection\_v2



**Full Name:** Ops.Gl.ImageCompose.GammaCorrection\_v2

**Description:** Allows for Gamma correction of a texture

### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Multiply Texture** (Number)
- **Gamma Correction** (Number)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.GammaCorrection\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.GammaCorrection_v2)

### 64.1.31 Gradient\_v2



**Full Name:** Ops.Gl.ImageCompose.Gradient\_v2

**Description:** Draws a simple gradient between three colors

#### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Alpha Mask Index** (Number: Integer)
- **Amount** (Number)
- **Width** (Number)
- **Type Index** (Number: Integer)
- **Pos** (Number)
- **Smoothstep** (Number: Boolean)
- **SRGB** (Number: Boolean)
- **Color Space Index** (Number: Integer)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **R2** (Number)
- **G2** (Number)
- **B2** (Number)
- **R3** (Number)
- **G3** (Number)
- **B3** (Number)
- **Randomize** (Trigger)

#### < Output Ports:

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Gradient\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Gradient_v2)

### 64.1.32 GridTexture\_v2



**Full Name:** Ops.Gl.ImageCompose.GridTexture\_v2

**Description:** Creates a grid texture

#### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Alpha Mask Index** (Number: Integer)
- **Amount** (Number)
- **Line Thickness X** (Number)
- **Line Thickness Y** (Number)
- **Cells X** (Number)
- **Cells Y** (Number)
- **Rotate** (Number)
- **Offset X** (Number)
- **Offset Y** (Number)
- **Invert Color** (Number: Boolean)
- **Line Red** (Number)
- **Line Green** (Number)
- **Line Blue** (Number)

#### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.GridTexture\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.GridTexture_v2)

### 64.1.33 GrowPixels\_v2



**Full Name:** Ops.Gl.ImageCompose.GrowPixels\_v2

**Description:** Make one pixel lines thicker via postprocessing

#### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Strength** (Number)
- **Iterations** (Number: Integer)
- **R** (Number)
- **G** (Number)
- **B** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.GrowPixels\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.GrowPixels_v2)

#### 64.1.34 Hue



**Full Name:** Ops.Gl.ImageCompose.Hue

**Description:** Adjust Hue of current ImageCompose

**> Input Ports:**

- **Render** (Trigger)
- **Hue** (Number)
- **Mask** (Object:Texture)
- **Offset** (Object:Texture)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Hue>

#### 64.1.35 ImageCompose\_v4



**Full Name:** Ops.Gl.ImageCompose.ImageCompose\_v4

**Description:** Compose Images and effects as layers to generate new Images

**> Input Ports:**

- **Render** (Trigger)
- **Base Texture** (Object:Texture)
- **UV Texture** (Object:Texture)
- **Width** (Number: Integer)
- **Height** (Number: Integer)
- **Wrap Index** (Number: Integer)
- **PixelFormat Index** (Number: Integer)
- **Clear** (Number: Boolean)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)

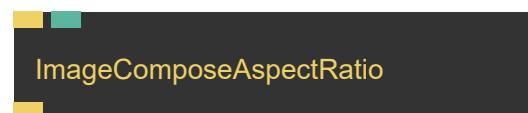
**< Output Ports:**

- **Next** (Trigger)
- **Texture\_out** (Object)
- **Aspect Ratio** (Number)
- **Texture Width** (Number)
- **Texture Height** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.ImageCompose\\_v4](https://cables.gl/op/Ops.Gl.ImageCompose.ImageCompose_v4)

#### 64.1.36 ImageComposeAspectRatio



**Full Name:** Ops.Gl.ImageCompose.ImageComposeAspectRatio

**Description:** Adjust aspect ratio of an image compose branch

**> Input Ports:**

- **Update** (Trigger)
- **Aspect** (Number)

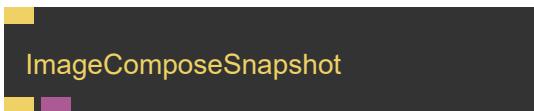
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.ImageComposeAspectRatio>

### 64.1.37 ImageComposeSnapshot



**Full Name:** Ops.Gl.ImageCompose.ImageComposeSnapshot

**Description:** capture the current state of an imageCompose branch by copying the texture

**> Input Ports:**

- **Update** (Trigger)

**< Output Ports:**

- **Trigger** (Trigger)
- **Texture** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.ImageComposeSnapshot>

### 64.1.38 Interlace



**Full Name:** Ops.Gl.ImageCompose.Interlace

**Description:** Tv scanlines effect

**> Input Ports:**

- **Render** (Trigger)
- **Amount** (Number)
- **Lumi Scale** (Number)
- **X Or Y** (Number: Boolean)
- **Line Size** (Number)
- **Displacement** (Number)

- **Add** (Number)

- **Scroll** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Interlace>

### 64.1.39 Invert\_v2



**Full Name:** Ops.Gl.ImageCompose.Invert\_v2

**Description:** Invert image colors

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Mask Invert** (Number: Boolean)
- **Mask** (Object:Texture)
- **Invert R** (Number: Boolean)
- **Invert G** (Number: Boolean)
- **Invert B** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Invert\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Invert_v2)

### 64.1.40 Kaleidoscope\_v2



**Full Name:** Ops.Gl.ImageCompose.Kaleidoscope\_v2

**Description:** Kaleidoscope effect

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Sides** (Number)
- **Angle** (Number)
- **Slide X** (Number)
- **Slide Y** (Number)
- **Center X** (Number)
- **Center Y** (Number)
- **Aspect Ratio** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Kaleidoscope\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Kaleidoscope_v2)

## 64.1.41 LensDirt\_v2



**Full Name:** Ops.Gl.ImageCompose.LensDirt\_v2

**Description:** Creates a lens dirt like texture

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Alpha Mask Index** (Number: Integer)
- **Offset X** (Number)
- **Offset Y** (Number)
- **Zoom** (Number)
- **Iterations** (Number: Integer)
- **Seed** (Number: Integer)
- **Spot Edge** (Number)

- **Gamma** (Number)

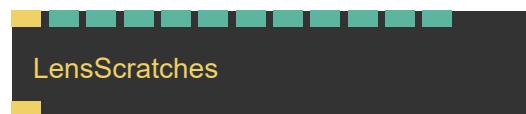
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.LensDirt\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.LensDirt_v2)

## 64.1.42 LensScratches\_v2



**Full Name:** Ops.Gl.ImageCompose.LensScratches\_v2

**Description:** Creates a procedural texture simulating scratches on a lens

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Alpha Mask Index** (Number: Integer)
- **Offset X** (Number)
- **Offset Y** (Number)
- **Wavyness** (Number)
- **Scale** (Number)
- **Layers** (Number: Integer)
- **AA Iterations** (Number)
- **Frequency** (Number)
- **Frequency Step Size** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.LensScratches\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.LensScratches_v2)

## 64.1.43 Levels\_v2



**Full Name:** Ops.Gl.ImageCompose.Levels\_v2

**Description:** adjust levels to correct the tonal range of an image

**> Input Ports:**

- **Render** (Trigger)
- **In Min** (Number)
- **Midpoint** (Number)
- **In Max** (Number)
- **Out Min** (Number)
- **Out Max** (Number)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Levels\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Levels_v2)

## 64.1.44 LumaKey\_v3



**Full Name:** Ops.Gl.ImageCompose.LumaKey\_v3

**Description:** Remove darkest or brightest parts of the image

**> Input Ports:**

- **Render** (Trigger)
- **Invert** (Number: Boolean)
- **Black White** (Number: Boolean)
- **Remove Alpha** (Number: Boolean)
- **Remap** (Number: Boolean)
- **Threshold Low** (Number)
- **Threshold High** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.LumaKey\\_v3](https://cables.gl/op/Ops.Gl.ImageCompose.LumaKey_v3)

## 64.1.45 LUTMap



**Full Name:** Ops.Gl.ImageCompose.LUTMap

**Description:** apply color filter/effects by using a lookup texture

**> Input Ports:**

- **Render** (Trigger)
- **LUT Image** (Object:Texture)
- **Amount** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.LUTMap>

## 64.1.46 Mirror



**Full Name:** Ops.Gl.ImageCompose.Mirror

**Description:** mirroring image effect

**> Input Ports:**

- **Render** (Trigger)
- **Axis Index** (Number: Integer)
- **Width** (Number)
- **Offset** (Number)
- **Flip** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Mirror>

#### 64.1.47 Mix



**Full Name:** Ops.Gl.ImageCompose.Mix

**Description:** simple mix/fade of two input images

**> Input Ports:**

- **Render** (Trigger)
- **Texture 1** (Object:Texture)
- **Fade** (Number)
- **Texture 2** (Object:Texture)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Mix>

#### 64.1.48 MultiDrawImage



**Full Name:** Ops.Gl.ImageCompose.MultiDrawImage

**Description:** draw multiple images at once

**> Input Ports:**

- **Render** (Trigger)
- **Mask Invert** (Number: Boolean)
- **Texture 1** (Object:Texture)
- **Blendmode 1 Index** (Number: Integer)
- **Mask 1** (Object:Texture)
- **Mask Source 1 Index** (Number: Integer)
- **Opacity 1 Index** (Number: Integer)

- **Amount 1** (Number)
- **Texture 2** (Object:Texture)
- **Blendmode 2 Index** (Number: Integer)
- **Mask 2** (Object:Texture)
- **Mask Source 2 Index** (Number: Integer)
- **Opacity 2 Index** (Number: Integer)
- **Amount 2** (Number)
- **Texture 3** (Object:Texture)
- **Blendmode 3 Index** (Number: Integer)
- **Mask 3** (Object:Texture)
- **Mask Source 3 Index** (Number: Integer)
- **Opacity 3 Index** (Number: Integer)
- **Amount 3** (Number)
- **Texture 4** (Object:Texture)
- **Blendmode 4 Index** (Number: Integer)
- **Mask 4** (Object:Texture)
- **Mask Source 4 Index** (Number: Integer)
- **Opacity 4 Index** (Number: Integer)
- **Amount 4** (Number)
- **Texture 5** (Object:Texture)
- **Blendmode 5 Index** (Number: Integer)
- **Mask 5** (Object:Texture)
- **Mask Source 5 Index** (Number: Integer)
- **Opacity 5 Index** (Number: Integer)
- **Amount 5** (Number)
- **Texture 6** (Object:Texture)
- **Blendmode 6 Index** (Number: Integer)
- **Mask 6** (Object:Texture)
- **Mask Source 6 Index** (Number: Integer)
- **Opacity 6 Index** (Number: Integer)
- **Amount 6** (Number)
- **Texture 7** (Object:Texture)
- **Blendmode 7 Index** (Number: Integer)
- **Mask 7** (Object:Texture)
- **Mask Source 7 Index** (Number: Integer)
- **Opacity 7 Index** (Number: Integer)
- **Amount 7** (Number)
- **Texture 8** (Object:Texture)
- **Blendmode 8 Index** (Number: Integer)
- **Mask 8** (Object:Texture)

- **Mask Source 8 Index** (Number: Integer)
- **Opacity 8 Index** (Number: Integer)
- **Amount 8** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.MultiDrawImage>

#### 64.1.49 OnePassBlur



**Full Name:** Ops.Gl.ImageCompose.OnePassBlur

**Description:** Visit documentation for details

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Radius** (Number)
- **Mask** (Object:Texture)
- **Mask Invert** (Number: Boolean)

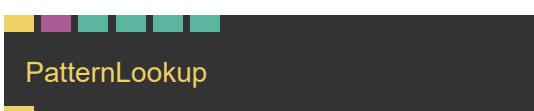
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.OnePassBlur>

#### 64.1.50 PatternLookup



**Full Name:** Ops.Gl.ImageCompose.PatternLookup

**Description:** map a pattern to value levels of your texture

**> Input Ports:**

- **Render** (Trigger)
- **Multiplier** (Object)
- **Blend Mode** (Number: String)
- **Amount** (Number)
- **Width** (Number)
- **Height** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.PatternLookup>

#### 64.1.51 Pixelate\_v2



**Full Name:** Ops.Gl.ImageCompose.Pixelate\_v2

**Description:** Pixelate an image

**> Input Ports:**

- **Render** (Trigger)
- **Multiplier** (Object:Texture)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Width** (Number)
- **Height** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Pixelate\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Pixelate_v2)

## 64.1.52 PixelColor



**Full Name:** Ops.Gl.ImageCompose.PixelColor

**Description:** fill image with one color picked at a position

### > Input Ports:

- **Render** (Trigger)
- **Source Texture** (Object:Texture)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Pos X** (Number)
- **Pos Y** (Number)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.PixelColor>

## 64.1.53 PixelDifference



**Full Name:** Ops.Gl.ImageCompose.PixelDifference

**Description:** visualize the difference of neighbouring pixels (slope)

### > Input Ports:

- **Render** (Trigger)
- **Strength** (Number)
- **Step** (Number)
- **Red Index** (Number: Integer)
- **Red Flip** (Number: Boolean)
- **Green Index** (Number: Integer)
- **Green Flip** (Number: Boolean)
- **Blue Index** (Number: Integer)

- **Blue Flip** (Number: Boolean)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.PixelDifference>

## 64.1.54 PixelDisplacement\_v4



**Full Name:** Ops.Gl.ImageCompose.PixelDisplacement\_v4

**Description:** Changes color lookup for every pixel using a displacement map

### > Input Ports:

- **Render** (Trigger)
- **DisplaceTex** (Object:Texture)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Amount X** (Number)
- **Amount Y** (Number)
- **Input Index** (Number: Integer)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.PixelDisplacement\\_v4](https://cables.gl/op/Ops.Gl.ImageCompose.PixelDisplacement_v4)

## 64.1.55 Plasma\_v2



**Full Name:** Ops.Gl.ImageCompose.Plasma\_v2

**Description:** Renders a plasma effect

### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Alpha Mask Index** (Number: Integer)
- **Amount** (Number)
- **Width** (Number)
- **Height** (Number)
- **Aspect** (Number: Boolean)
- **Mul** (Number)
- **X** (Number)
- **Y** (Number)
- **Time** (Number)
- **Greyscale** (Number: Boolean)
- **Offset** (Object:Texture)
- **Offset Multiply** (Number)
- **Offset X Index** (Number: Integer)
- **Offset Y Index** (Number: Integer)
- **Offset Time Index** (Number: Integer)
- **Mask** (Object:Texture)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Plasma\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Plasma_v2)

## 64.1.56 PolarCoords



**Full Name:** Ops.Gl.ImageCompose.PolarCoords

**Description:** display texture using polar/radial coordinate system

**> Input Ports:**

- **Render** (Trigger)
- **Radius Inner** (Number)
- **Radius Outer** (Number)
- **Crop** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.PolarCoords>

## 64.1.57 Posterize\_v2



**Full Name:** Ops.Gl.ImageCompose.Posterize\_v2

**Description:** reduce number of colors

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Levels** (Number)

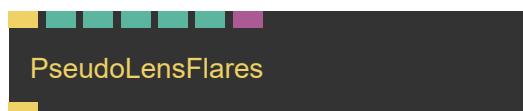
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Posterize\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Posterize_v2)

## 64.1.58 PseudoLensFlares



**Full Name:** Ops.Gl.ImageCompose.PseudoLensFlares

**Description:** simulate lens flare effect

**> Input Ports:**

- **Render** (Trigger)
- **Ghosts** (Number)
- **Num Ghosts** (Number: Integer)
- **Dispersal** (Number)
- **Halo** (Number)

- **Halo Width** (Number)
- **Color Lookup** (Object)

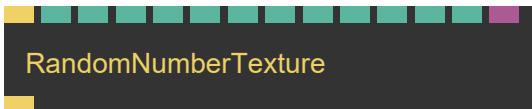
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.PseudoLensFlares>

## 64.1.59 RandomNumberTexture



**Full Name:** Ops.Gl.ImageCompose.RandomNumberTexture

**Description:** Set random numbers into an imagecompose

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Seed** (Number)
- **Min R** (Number)
- **Max R** (Number)
- **Min G** (Number)
- **Max G** (Number)
- **Min B** (Number)
- **Max B** (Number)
- **Min A** (Number)
- **Max A** (Number)
- **Multiply** (Object:Texture)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.RandomNumberTexture>

## 64.1.60 RectangleTexture\_v5



**Full Name:** Ops.Gl.ImageCompose.RectangleTexture\_v5

**Description:** draws a 2d rectangle into a texture.

**> Input Ports:**

- **Render** (Trigger)
- **Amount** (Number)
- **Blend Mode Index** (Number: Integer)
- **Center** (Number: Boolean)
- **X** (Number)
- **Y** (Number)
- **Inner** (Number)
- **Width** (Number)
- **Height** (Number)
- **Rotate** (Number)
- **Roundness** (Number)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)
- **Map Texture** (Object:Texture)
- **Start X** (Number)
- **Start Y** (Number)
- **Map Width** (Number)
- **Map Height** (Number)
- **Mask** (Object:Texture)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.RectangleTexture\\_v5](https://cables.gl/op/Ops.Gl.ImageCompose.RectangleTexture_v5)

## 64.1.61 RemoveAlpha



**Full Name:** Ops.Gl.ImageCompose.RemoveAlpha

**Description:** Remove alpha information from image

**> Input Ports:**

- **Render** (Trigger)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.RemoveAlpha>

## 64.1.62 RepeatTexture\_v2



**Full Name:** Ops.Gl.ImageCompose.RepeatTexture\_v2

**Description:** Visit documentation for details

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **X** (Number)
- **Y** (Number)
- **Clear** (Number: Boolean)
- **Multiply** (Object:Texture)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.RepeatTexture\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.RepeatTexture_v2)

## 64.1.63 RgbMultiply



**Full Name:** Ops.Gl.ImageCompose.RgbMultiply

**Description:** multiply image colors by color channel

**> Input Ports:**

- **Render** (Trigger)
- **R** (Number)
- **G** (Number)
- **B** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.RgbMultiply>

## 64.1.64 RGBOffset\_v2



**Full Name:** Ops.Gl.ImageCompose.RGBOffset\_v2

**Description:** Offsets the xy components of an RGB texture

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Red Offset X** (Number)
- **Red Offset Y** (Number)
- **Red Amount** (Number)
- **amount of red** (fade, hide, show)
- **Green Offset X** (Number)
- **Green Offset Y** (Number)
- **Green Amount** (Number)

- **amount of green** (fade, hide, show)
- **Blue Offset X** (Number)
- **Blue Offset Y** (Number)
- **Blue Amount** (Number)
- **amount of blue** (fade, hide, show)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.RGBOffset\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.RGBOffset_v2)

## 64.1.65 RgbToHsvTexture



**Full Name:** Ops.Gl.ImageCompose.RgbToHsvTexture

**Description:** Convert a RGB Texture to Hue/Saturation/Lightness values as RGB colors

**> Input Ports:**

- **Render** (Trigger)
- **Output RGB Index** (Number: Integer)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.RgbToHsvTexture>

## 64.1.66 RotateTexture\_v2



**Full Name:** Ops.Gl.ImageCompose.RotateTexture\_v2

**Description:** Rotates a texture

**> Input Ports:**

- **Render** (Trigger)
- **Multiplier** (Object:Texture)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Rotate** (Number)
- **Crop** (Number: Boolean)
- **Clear** (Number: Boolean)

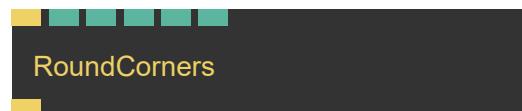
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.RotateTexture\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.RotateTexture_v2)

## 64.1.67 RoundCorners



**Full Name:** Ops.Gl.ImageCompose.RoundCorners

**Description:** Draw round corners around image (border)

**> Input Ports:**

- **Render** (Trigger)
- **Radius** (Number)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.RoundCorners>

## 64.1.68 ScaleTexture\_v3



**Full Name:** Ops.Gl.ImageCompose.ScaleTexture\_v3

**Description:** Scales a texture

### > Input Ports:

- **Render** (Trigger)
- **Multiplier** (Object:Texture)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Scale X** (Number)
- **Scale Y** (Number)
- **Offset X** (Number)
- **Offset Y** (Number)
- **Center X** (Number)
- **Center Y** (Number)
- **Clear** (Number: Boolean)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.ScaleTexture\\_v3](https://cables.gl/op/Ops.Gl.ImageCompose.ScaleTexture_v3)

## 64.1.69 ScrollTexture



**Full Name:** Ops.Gl.ImageCompose.ScrollTexture

**Description:** Visit documentation for details

### > Input Ports:

- **Render** (Trigger)
- **AmountX** (Number)
- **AmountY** (Number)

- **Mask** (Object:Texture)
- **Repeat** (Number: Boolean)

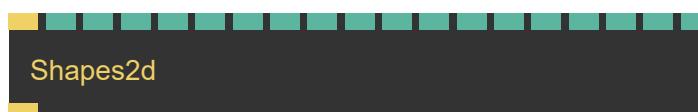
### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.ScrollTexture>

## 64.1.70 Shapes2d\_v2



**Full Name:** Ops.Gl.ImageCompose.Shapes2d\_v2

**Description:** Generates different 2d shapes to use as a texture

### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Alpha Mask Index** (Number: Integer)
- **Amount** (Number)
- **Shape Index** (Number: Integer)
- **Mirror X** (Number: Boolean)
- **Mirror Y** (Number: Boolean)
- **Offset X** (Number)
- **Offset Y** (Number)
- **FillShape** (Number: Boolean)
- **Line Thickness** (Number)
- **Invert Color** (Number: Boolean)
- **Width** (Number)
- **Height** (Number)
- **Rotate** (Number)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Shapes2d\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Shapes2d_v2)

## 64.1.71 Sharpen



**Full Name:** Ops.Gl.ImageCompose.Sharpen

**Description:** Adjust image sharpness

### > Input Ports:

- **Render** (Trigger)
- **Amount** (Number)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Sharpen>

## 64.1.72 SkewStretchImage\_v2



**Full Name:** Ops.Gl.ImageCompose.SkewStretchImage\_v2

**Description:** skew / stretch an image by rendering scaled sides

### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Clamp** (Number: Boolean)
- **Stretch Top** (Number)
- **Stretch Bottom** (Number)
- **Stretch Left** (Number)
- **Stretch Right** (Number)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.SkewStretchImage\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.SkewStretchImage_v2)

## 64.1.73 Stripes\_v4



**Full Name:** Ops.Gl.ImageCompose.Stripes\_v4

**Description:** Create a texture of stripes /lines

### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Alpha Mask Index** (Number: Integer)
- **Num** (Number)
- **Width** (Number)
- **Rotate** (Number)
- **Offset** (Number)
- **Gradients** (Number: Boolean)
- **Circular** (Number: Boolean)
- **Invert** (Number: Boolean)
- **R** (Number)
- **G** (Number)
- **B** (Number)

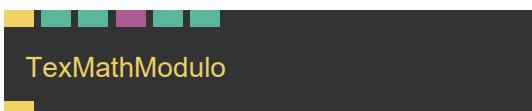
### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Stripes\\_v4](https://cables.gl/op/Ops.Gl.ImageCompose.Stripes_v4)

## 64.1.74 TexMathModulo



TexMathModulo

**Full Name:** Ops.Gl.ImageCompose.TexMathModulo

**Description:** modulo pixel color values

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Mask Invert** (Number: Boolean)
- **Mask** (Object:Texture)
- **Amount** (Number)
- **Modulo** (Number)

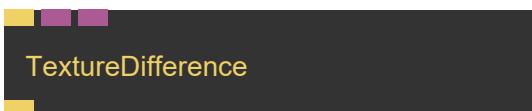
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.TexMathModulo>

## 64.1.75 TextureDifference



TextureDifference

**Full Name:** Ops.Gl.ImageCompose.TextureDifference

**Description:** render the difference of two textures

**> Input Ports:**

- **Render** (Trigger)
- **Texture 1** (Object:Texture)
- **Texture 2** (Object:Texture)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.TextureDifference>

## 64.1.76 ToNormalMap\_v2



ToNormalMap

**Full Name:** Ops.Gl.ImageCompose.ToNormalMap\_v2

**Description:** Convert a black and white map to a normal map

**> Input Ports:**

- **Render** (Trigger)
- **Strength** (Number)
- **Step Multiplier** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.ToNormalMap\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.ToNormalMap_v2)

## 64.1.77 Twirl\_v4



Twirl

**Full Name:** Ops.Gl.ImageCompose.Twirl\_v4

**Description:** Creates a twirl/swirl/spiral effect in a texture

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Twist Amount** (Number)
- **Radius** (Number)
- **Center X** (Number)
- **Center Y** (Number)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Twirl\\_v4](https://cables.gl/op/Ops.Gl.ImageCompose.Twirl_v4)

### 64.1.78 Vibrance



**Full Name:** Ops.Gl.ImageCompose.Vibrance

**Description:** adjust vibrance/saturation

#### > Input Ports:

- **Render** (Trigger)
- **Amount** (Number)

#### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Vibrance>

### 64.1.79 Vignette\_v3



**Full Name:** Ops.Gl.ImageCompose.Vignette\_v3

**Description:** Simulating an old camera effect of fading away the edges of the image

#### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Alpha Mask Index** (Number: Integer)
- **Amount** (Number)
- **Strength** (Number)
- **Radius** (Number)
- **Sharp** (Number)
- **Aspect** (Number)
- **R** (Number)

- **G** (Number)
- **B** (Number)
- **Alpha** (Number: Boolean)

#### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Vignette\\_v3](https://cables.gl/op/Ops.Gl.ImageCompose.Vignette_v3)

### 64.1.80 Waveform\_v3



**Full Name:** Ops.Gl.ImageCompose.Waveform\_v3

**Description:** Generates 4 different waveform textures. Sine, saw-tooth, Triangle, Square.

#### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Alpha Mask Index** (Number: Integer)
- **Waveform Index** (Number: Integer)
- **Amplitude** (Number)
- **Frequency** (Number)
- **Line Width** (Number)
- **Line Glow** (Number)
- **Invert Color** (Number: Boolean)
- **Solid Fill** (Number: Boolean)
- **Offset X** (Number)
- **Offset Y** (Number)
- **Rotate** (Number)
- **R** (Number)
- **G** (Number)
- **B** (Number)

#### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Waveform\\_v3](https://cables.gl/op/Ops.Gl.ImageCompose.Waveform_v3)

### 64.1.81 WaveformGradient\_v4



**Full Name:** Ops.Gl.ImageCompose.WaveformGradient\_v4

**Description:** Generate different texture waveforms. Sine, sawtooth and triangle.

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Alpha Mask Index** (Number: Integer)
- **Amount** (Number)
- **Mode Index** (Number: Integer)
- **Frequency** (Number)
- **Pow Factor** (Number)
- **Offset** (Number)
- **Rotate** (Number)
- **R** (Number)
- **G** (Number)
- **B** (Number)

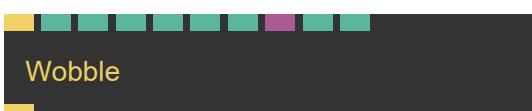
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.WaveformGradient\\_v4](https://cables.gl/op/Ops.Gl.ImageCompose.WaveformGradient_v4)

### 64.1.82 Wobble\_v2



**Full Name:** Ops.Gl.ImageCompose.Wobble\_v2

**Description:** waving wobble motion effect

**> Input Ports:**

- **Render** (Trigger)
- **Time** (Number)
- **SpeedX** (Number)
- **SpeedY** (Number)
- **RepeatX** (Number)
- **RepeatY** (Number)
- **Multiply** (Number)
- **Amount Map** (Object:Texture)
- **Source Amount Map Index** (Number: Integer)
- **Invert Amount Map** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Wobble\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Wobble_v2)

### 64.1.83 ZoomBlur\_v2



**Full Name:** Ops.Gl.ImageCompose.ZoomBlur\_v2

**Description:** Directional blur effect

**> Input Ports:**

- **Render** (Trigger)
- **Strength** (Number)
- **Samples** (Number: Integer)
- **X** (Number)
- **Y** (Number)
- **Strength Map** (Object:Texture)
- **Source Strength Map Index** (Number: Integer)
- **Invert Strength Map** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.ZoomBlur\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.ZoomBlur_v2)

# 65 Ops.Gl.ImageCompose.Math

## 65.1 Ops.Gl.ImageCompose.Math

### 65.1.1 ColorMapRange



**Full Name:** Ops.Gl.ImageCompose.Math.ColorMapRange

**Description:** Map the range of color number values to another

**> Input Ports:**

- **Render** (Trigger)
- **Old Min** (Number)
- **Old Max** (Number)
- **New Min** (Number)
- **New Max** (Number)
- **Clamp** (Number: Boolean)
- **R** (Number: Boolean)
- **G** (Number: Boolean)
- **B** (Number: Boolean)
- **A** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Math.ColorMapRange>

### 65.1.2 Normalize



**Full Name:** Ops.Gl.ImageCompose.Math.Normalize

**Description:** normalize texture rgb values

**> Input Ports:**

- **Render** (Trigger)
- **Fade** (Number)
- **Size** (Number)

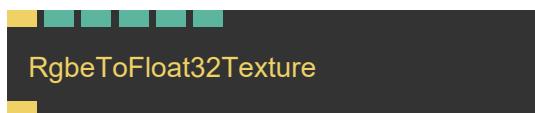
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Math.Normalize>

### 65.1.3 RgbeToFloat32Texture



**Full Name:** Ops.Gl.ImageCompose.Math.RgbeToFloat32Texture

**Description:** Convert a RGBE texture to HDR/floating point texture

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Mode Index** (Number: Integer)
- **Min** (Number)
- **Max** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Math.RgbeToFloat32Texture>

### 65.1.4 RgbMath



**Full Name:** Ops.Gl.ImageCompose.Math.RgbMath

**Description:** This OP enables you to use precise values to modify the pixels in your texture. For example adjusting texture values that are modifying your geometry or array values, or even your post processing compositions.

**> Input Ports:**

- **Render** (Trigger)
- **Operation Index** (Number: Integer)
- **R Active** (Number: Boolean)
- **G Active** (Number: Boolean)
- **B Active** (Number: Boolean)
- **A Active** (Number: Boolean)
- **Texture** (Object:Texture)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)
- **Multiply Texture** (Number)
- **Mask** (Object:Texture)

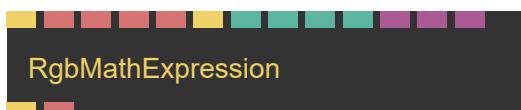
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Math.RgbMath>

### 65.1.5 RgbMathExpression



**Full Name:** Ops.Gl.ImageCompose.Math.RgbMathExpression

**Description:** Execute a glsl code math expression in a image compose

**> Input Ports:**

- **Render** (Trigger)
- **Update Shader** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **W** (Number)
- **TexA** (Object:Texture)
- **TexB** (Object:Texture)
- **TexC** (Object:Texture)

**< Output Ports:**

- **Trigger** (Trigger)
- **Code** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Math.RgbTransform>

## 65.1.6 RgbTransform



**Full Name:** Ops.Gl.ImageCompose.Math.RgbTransform

**Description:** transform RGB values interpreted as XYZ coordinates

**> Input Ports:**

- **Render** (Trigger)
- **Translate** (Number: Boolean)
- **Pos X** (Number)
- **Pos Y** (Number)
- **Pos Z** (Number)
- **Scale** (Number: Boolean)
- **Scale X** (Number)
- **Scale Y** (Number)
- **Scale Z** (Number)
- **Rotate** (Number: Boolean)
- **Rotation X** (Number)
- **Rotation Y** (Number)
- **Rotation Z** (Number)

- **Mask** (Object:Texture)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Math.Round>

## 65.1.7 Round



**Full Name:** Ops.Gl.ImageCompose.Math.Round

**Description:** Round number values of texture color channels

**> Input Ports:**

- **Render** (Trigger)
- **Amount** (Number)
- **Multiplier** (Number)

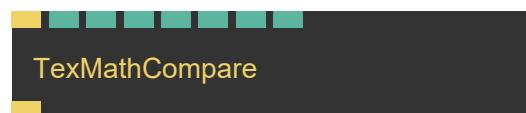
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Math.Round>

## 65.1.8 TexMathCompare



**Full Name:** Ops.Gl.ImageCompose.Math.TexMathCompare

**Description:** compare and pass through of color channel values

**> Input Ports:**

- **Render** (Trigger)
- **Comparison Index** (Number: Integer)
- **Result Index** (Number: Integer)
- **Number** (Number)

- **R Active** (Number: Boolean)
- **G Active** (Number: Boolean)
- **B Active** (Number: Boolean)
- **A Active** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Math.TexMathCompare>

## 66 Ops.Gl.ImageCompose.Noise

---

### 66.1 Ops.Gl.ImageCompose.Noise

#### 66.1.1 CellularNoise\_v2



**Full Name:** Ops.Gl.ImageCompose.Noise.CellularNoise\_v2

**Description:** Visit documentation for details

**> Input Ports:**

- **Render** (Trigger)
- **Mask** (Object:Texture)
- **Blend Mode Index** (Number: Integer)
- **Alpha Mask Index** (Number: Integer)
- **Amount** (Number)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Scale** (Number)
- **Harmonics Index** (Number: Integer)
- **Tileable** (Number: Boolean)
- **Offset** (Object:Texture)
- **Offset Multiply** (Number)
- **Offset X Index** (Number: Integer)
- **Offset Y Index** (Number: Integer)
- **Offset Z Index** (Number: Integer)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Noise.CellularNoise\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Noise.CellularNoise_v2)

## 66.1.2 FBMNoise\_v2



**Full Name:** Ops.Gl.ImageCompose.Noise.FBMNoise\_v2

**Description:** fractional brownian motion noise

### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Alpha Mask Index** (Number: Integer)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **Scale** (Number)
- **Anim** (Number)
- **ScrollX** (Number)
- **ScrollY** (Number)
- **Repeat** (Number)
- **Aspect** (Number)
- **Layer 1** (Number: Boolean)
- **Layer 2** (Number: Boolean)
- **Layer 3** (Number: Boolean)
- **Layer 4** (Number: Boolean)
- **Tileable** (Number: Boolean)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Noise.FBMNoise\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Noise.FBMNoise_v2)

## 66.1.3 GaborNoise



**Full Name:** Ops.Gl.ImageCompose.Noise.GaborNoise

**Description:** Render “gabor noise” into a texture

### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Phase** (Number)
- **Scale** (Number)
- **X** (Number)
- **Y** (Number)

### < Output Ports:

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Noise.GaborNoise>

## 66.1.4 GlitchNoise\_v2



**Full Name:** Ops.Gl.ImageCompose.Noise.GlitchNoise\_v2

**Description:** Creates a black and white glitched texture to use for displacement

### > Input Ports:

- **Render** (Trigger)
- **Amount** (Number)
- **Blend Mode Index** (Number: Integer)
- **Alpha Mask Index** (Number: Integer)
- **Seed** (Number)
- **Frequency** (Number)
- **Strength** (Number)
- **Block Size Small X** (Number)
- **Block Size Small Y** (Number)
- **Block Size Large X** (Number)
- **Block Size Large Y** (Number)
- **Scroll X** (Number)

- **Scroll Y** (Number)

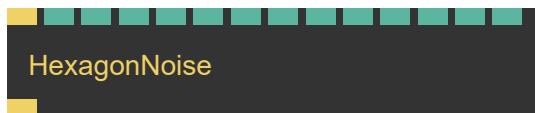
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Noise.GlitchNoise\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Noise.GlitchNoise_v2)

## 66.1.5 HexagonNoise\_v2



**Full Name:** Ops.Gl.ImageCompose.Noise.HexagonNoise\_v2

**Description:** Creates a hexagonal noise

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Alpha Mask Index** (Number: Integer)
- **Loop** (Number: Boolean)
- **RGB** (Number: Boolean)
- **Minimum Value** (Number)
- **Maximum Value** (Number)
- **Scale** (Number)
- **Orientation** (Number: Boolean)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Seed** (Number)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Noise.HexagonNoise\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Noise.HexagonNoise_v2)

## 66.1.6 LayerNoise\_v3



**Full Name:** Ops.Gl.ImageCompose.Noise.LayerNoise\_v3

**Description:** Multilayer perlin noise variation

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Alpha Mask Index** (Number: Integer)
- **Mode Index** (Number: Integer)
- **RGBA** (Number: Boolean)
- **Scale** (Number)
- **Layers** (Number: Integer)
- **Factor** (Number)
- **Exponent** (Number)
- **ScrollX** (Number)
- **ScrollY** (Number)
- **ScrollZ** (Number)
- **Tileable** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Noise.LayerNoise\\_v3](https://cables.gl/op/Ops.Gl.ImageCompose.Noise.LayerNoise_v3)

## 66.1.7 Noise\_v2



**Full Name:** Ops.Gl.ImageCompose.Noise.Noise\_v2

**Description:** White noise pixel effect

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Threshold** (Number)
- **Animated** (Number: Boolean)
- **RGB** (Number: Boolean)
- **Normalize** (Number: Boolean)
- **Multiply** (Object:Texture)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Noise.Noise\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Noise.Noise_v2)

## 66.1.8 PerlinNoise\_v2



**Full Name:** Ops.Gl.ImageCompose.Noise.PerlinNoise\_v2

**Description:** Draw perlin noise into an image

**> Input Ports:**

- **Render** (Trigger)
- **Mask** (Object:Texture)
- **Blend Mode Index** (Number: Integer)
- **Alpha Mask Index** (Number: Integer)
- **Amount** (Number)
- **Color Index** (Number: Integer)
- **Scale** (Number)
- **Multiply** (Number)
- **Harmonics Index** (Number: Integer)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Offset** (Object:Texture)
- **Offset Multiply** (Number)
- **Offset X Index** (Number: Integer)
- **Offset Y Index** (Number: Integer)

- **Offset Z Index** (Number: Integer)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Noise.PerlinNoise\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Noise.PerlinNoise_v2)

## 66.1.9 PixelNoise\_v3



**Full Name:** Ops.Gl.ImageCompose.Noise.PixelNoise\_v3

**Description:** Amount of blend mode to apply

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Alpha Mask Index** (Number: Integer)
- **Amount** (Number)
- **Loop** (Number: Boolean)
- **RGB** (Number: Boolean)
- **Minimum Value** (Number)
- **Maximum Value** (Number)
- **Num X** (Number)
- **Num Y** (Number)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Seed** (Number)
- **Centered** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Noise.PixelNoise\\_v3](https://cables.gl/op/Ops.Gl.ImageCompose.Noise.PixelNoise_v3)

## 66.1.10 PolkaDotNoise\_v2



PolkaDotNoise

**Full Name:** Ops.Gl.ImageCompose.Noise.PolkaDotNoise\_v2

**Description:** Visit documentation for details

### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Alpha Mask Index** (Number: Integer)
- **Square Look** (Number: Boolean)
- **Threshold** (Number)
- **Radius Low** (Number)
- **Radius High** (Number)
- **Scale** (Number)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)

### < Output Ports:

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Noise.PolkaDotNoise\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Noise.PolkaDotNoise_v2)

## 66.1.11 Shardnoise



Shardnoise

**Full Name:** Ops.Gl.ImageCompose.Noise.Shardnoise

**Description:** Render “shard noise” into a texture

### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)

- **Amount** (Number)
- **Sharpness** (Number)
- **Scale** (Number)
- **Round** (Number: Boolean)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)

### < Output Ports:

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ImageCompose.Noise.Shardnoise>

## 66.1.12 SimplexNoise\_v2



SimplexNoise

**Full Name:** Ops.Gl.ImageCompose.Noise.SimplexNoise\_v2

**Description:** simplex noise generator

### > Input Ports:

- **Render** (Trigger)
- **Mask** (Object:Texture)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Alpha Mask Index** (Number: Integer)
- **Smoothness** (Number)
- **Harmonics Index** (Number: Integer)
- **Scale** (Number)
- **X** (Number)
- **Y** (Number)
- **Time** (Number)
- **Offset** (Object:Texture)
- **Offset Multiply** (Number)
- **Offset X Index** (Number: Integer)
- **Offset Y Index** (Number: Integer)
- **Offset Z Index** (Number: Integer)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Noise.SimplexNoise\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Noise.SimplexNoise_v2)

### 66.1.13 TriangleNoise\_v2



**Full Name:** Ops.Gl.ImageCompose.Noise.TriangleNoise\_v2

**Description:** noise made from triangles

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Alpha Mask Index** (Number: Integer)
- **Scale** (Number)
- **Angle** (Number)
- **Add** (Number)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Noise.TriangleNoise\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Noise.TriangleNoise_v2)

### 66.1.14 ValueNoise\_v2



**Full Name:** Ops.Gl.ImageCompose.Noise.ValueNoise\_v2

**Description:** Visit documentation for details

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)

- **Amount** (Number)

- **Scale** (Number)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Noise.ValueNoise\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Noise.ValueNoise_v2)

### 66.1.15 Voronoise\_v2



**Full Name:** Ops.Gl.ImageCompose.Noise.Voronoise\_v2

**Description:** Voronoi Noise function

**> Input Ports:**

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Amount** (Number)
- **Alpha Mask Index** (Number: Integer)
- **Time** (Number)
- **Movement** (Number)
- **Num** (Number)
- **Seed** (Number)
- **Fill Index** (Number: Integer)
- **Draw Isolines** (Number: Boolean)
- **Draw Distance** (Number: Boolean)
- **Draw Center** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Noise.Voronoise\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Noise.Voronoise_v2)

## 66.1.16 WorleyNoise\_v2



**Full Name:** Ops.Gl.ImageCompose.Noise.WorleyNoise\_v2

**Description:** Visit documentation for details

### > Input Ports:

- **Render** (Trigger)
- **Blend Mode Index** (Number: Integer)
- **Alpha Mask Index** (Number: Integer)
- **Amount** (Number)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Scale** (Number)
- **Harmonics Index** (Number: Integer)
- **Invert** (Number: Boolean)
- **RangeA** (Number)
- **RangeB** (Number)
- **Tileable** (Number: Boolean)
- **Amount Map** (Object:Texture)
- **Source Strength Map Index** (Number: Integer)
- **Invert Strength Map** (Number: Boolean)
- **Offset** (Object:Texture)
- **Offset Multiply** (Number)
- **Offset X Index** (Number: Integer)
- **Offset Y Index** (Number: Integer)
- **Offset Z Index** (Number: Integer)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ImageCompose.Noise.WorleyNoise\\_v2](https://cables.gl/op/Ops.Gl.ImageCompose.Noise.WorleyNoise_v2)

# 67 Ops.Gl.Matrix

## 67.1 Ops.Gl.Matrix

### 67.1.1 AnimMatrix



**Full Name:** Ops.Gl.Matrix.AnimMatrix

**Description:** animate values in a matrix to a new matrix

### > Input Ports:

- **Update** (Trigger)
- **Next Matrix** (Array)
- **Duration** (Number)
- **Easing Index** (Number: Integer)

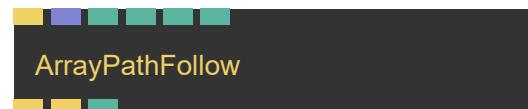
### < Output Ports:

- **Next** (Trigger)
- **Matrix** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.AnimMatrix>

### 67.1.2 ArrayPathFollow



**Full Name:** Ops.Gl.Matrix.ArrayPathFollow

**Description:** interpolate position on a spline/array3x

### > Input Ports:

- **Exe** (Trigger)
- **Array** (Array)
- **Time** (Number)

- **Duration** (Number)
- **Offset** (Number)
- **Look Ahead** (Number)

**< Output Ports:**

- **Trigger** (Trigger)
- **Transform Lookat** (Trigger)
- **Index** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.ArrayPathFollow>

### 67.1.3 ArrayPathFollowParticles\_v2



**Full Name:** Ops.Gl.Matrix.ArrayPathFollowParticles\_v2

**Description:** render lots of particles following a path/spline/array3x

**> Input Ports:**

- **Exec** (Trigger)
- **Points** (Array)
- **Num Particles** (Number)
- **Length** (Number)
- **Spread** (Number)
- **Offset** (Number)
- **Max Distance** (Number)
- **RandomSpeed** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Matrix.ArrayPathFollowParticles\\_v2](https://cables.gl/op/Ops.Gl.Matrix.ArrayPathFollowParticles_v2)

### 67.1.4 Billboard



**Full Name:** Ops.Gl.Matrix.Billboard

**Description:** rotate an object to always face the camera

**> Input Ports:**

- **Exec** (Trigger)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.Billboard>

### 67.1.5 Camera\_v2



**Full Name:** Ops.Gl.Matrix.Camera\_v2

**Description:** Transforms and projects the scene from the point of view of the camera.

**> Input Ports:**

- **Render** (Trigger)
- **Identity** (Number: Boolean)
- **Projection Mode Index** (Number: Integer)
- **Frustum Near** (Number)
- **Frustum Far** (Number)
- **Fov** (Number)
- **Auto Aspect Ratio** (Number: Boolean)
- **Aspect Ratio** (Number)
- **Eye X** (Number)
- **Eye Y** (Number)
- **Eye Z** (Number)
- **Center X** (Number)

- **Center Y** (Number)
- **Center Z** (Number)
- **Truck** (Number)
- **Move sideways** (in local x axis)
- **Boom** (Number)
- **Dolly** (Number)
- **Tilt** (Number)
- **Pan** (Number)
- **Roll** (Number)

**< Output Ports:**

- **Trigger** (Trigger)
- **Aspect** (Number)
- **Look At Array** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Matrix.Camera\\_v2](https://cables.gl/op/Ops.Gl.Matrix.Camera_v2)

## 67.1.6 CameraInfo



**Full Name:** Ops.Gl.Matrix.CameraInfo

**Description:** get camera attributes from current camera/orbit controls

**> Input Ports:**

- **Render** (Trigger)
- **Camera Type Index** (Number: Integer)

**< Output Ports:**

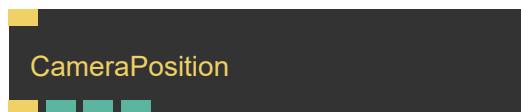
- **Trigger** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Right X** (Number)
- **Right Y** (Number)
- **Right Z** (Number)
- **Up X** (Number)
- **Up Y** (Number)

- **Up Z** (Number)
- **Forward X** (Number)
- **Forward Y** (Number)
- **Forward Z** (Number)
- **Near Frustum** (Number)
- **Far Frustum** (Number)
- **Bottom Frustum** (Number)
- **Top Frustum** (Number)
- **Left Frustum** (Number)
- **Right Frustum** (Number)
- **FOV** (Number)
- **Aspect Ratio** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.CameraInfo>

## 67.1.7 CameraPosition



**Full Name:** Ops.Gl.Matrix.CameraPosition

**Description:** get the current position of viewmatrix/camera eye

**> Input Ports:**

- **Render** (Trigger)

**< Output Ports:**

- **Trigger** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.CameraPosition>

## 67.1.8 Coordinates



**Full Name:** Ops.Gl.Matrix.Coordinates

**Description:** current xyz coordinates (modelmatrix)

**> Input Ports:**

- **Render** (Trigger)

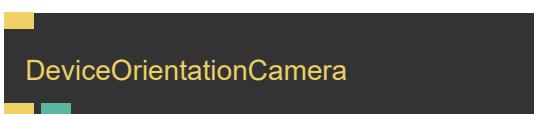
**< Output Ports:**

- **Trigger** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.Coordinates>

## 67.1.9 DeviceOrientationCamera



**Full Name:** Ops.Gl.Matrix.DeviceOrientationCamera

**Description:** gyroscope motionsensor camera

**> Input Ports:**

- **Render** (Trigger)

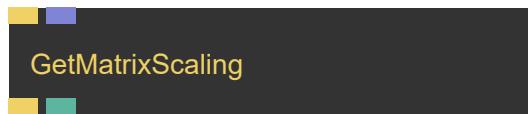
**< Output Ports:**

- **Next** (Trigger)
- **Window Orientation** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.DeviceOrientationCamera>

## 67.1.10 GetMatrixScaling



**Full Name:** Ops.Gl.Matrix.GetMatrixScaling

**Description:** Get the scalar scaling of a matrix

**> Input Ports:**

- **Render** (Trigger)
- **Matrix** (Array)

**< Output Ports:**

- **Trigger** (Trigger)
- **Scaling** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.GetMatrixScaling>

## 67.1.11 GetModelMatrix



**Full Name:** Ops.Gl.Matrix.GetModelMatrix

**Description:** Get current modelmatrix

**> Input Ports:**

- **Render** (Trigger)

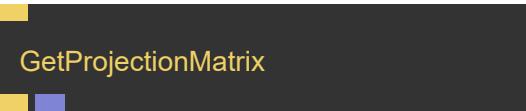
**< Output Ports:**

- **Trigger** (Trigger)
- **Matrix** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.GetModelMatrix>

## 67.1.12 GetProjectionMatrix



GetProjectionMatrix

**Full Name:** Ops.Gl.Matrix.GetProjectionMatrix

**Description:** get current projectionmatrix

**> Input Ports:**

- **Render** (Trigger)

**< Output Ports:**

- **Trigger** (Trigger)
- **Matrix** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.GetProjectionMatrix>

## 67.1.13 GetViewMatrix



GetViewMatrix

**Full Name:** Ops.Gl.Matrix.GetViewMatrix

**Description:** get current viewmatrix

**> Input Ports:**

- **Render** (Trigger)

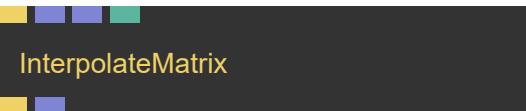
**< Output Ports:**

- **Trigger** (Trigger)
- **Matrix** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.GetViewMatrix>

## 67.1.14 InterpolateMatrix



InterpolateMatrix

**Full Name:** Ops.Gl.Matrix.InterpolateMatrix

**Description:** interpolate between two matrices

**> Input Ports:**

- **Exe** (Trigger)
- **Array 1** (Array)
- **Array 2** (Array)
- **Perc** (Number)

**< Output Ports:**

- **Next** (Trigger)
- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.InterpolateMatrix>

## 67.1.15 InvertMatrix



InvertMatrix

**Full Name:** Ops.Gl.Matrix.InvertMatrix

**Description:** outputs an inverted matrix

**> Input Ports:**

- **Matrix** (Array)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.InvertMatrix>

## 67.1.16 LookatCamera



**Full Name:** Ops.Gl.Matrix.LookatCamera

**Description:** transforms view to look from eye to center

**> Input Ports:**

- **Render** (Trigger)
- **EyeX** (Number)
- **EyeY** (Number)
- **EyeZ** (Number)
- **CenterX** (Number)
- **CenterY** (Number)
- **CenterZ** (Number)
- **UpX** (Number)
- **UpY** (Number)
- **UpZ** (Number)

**< Output Ports:**

- **Trigger** (Trigger)
- **Array** (Array)

**Example Patch:** Open in Editor

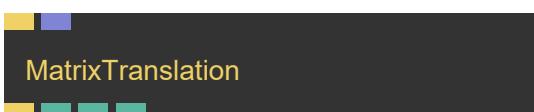
**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.LookatCamera>

- **Trigger** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.MatrixTranslation>

## 67.1.17 MatrixTranslation



**Full Name:** Ops.Gl.Matrix.MatrixTranslation

**Description:** get translation of a matrix

**> Input Ports:**

- **Render** (Trigger)
- **Matrix** (Array)

**< Output Ports:**

## 67.1.18 MultiplyModelMatrix



**Full Name:** Ops.Gl.Matrix.MultiplyModelMatrix

**Description:** multiply model matrix

**> Input Ports:**

- **Render** (Trigger)
- **Identity** (Number: Boolean)
- **Matrix** (Array)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.MultiplyModelMatrix>

## 67.1.19 MulViewMatrix



**Full Name:** Ops.Gl.Matrix.MulViewMatrix

**Description:** multiply view matrix

**> Input Ports:**

- **Render** (Trigger)
- **Matrix** (Array)
- **Identity** (Number: Boolean)

#### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.MulViewMatrix>

## 67.1.20 Quaternion



**Full Name:** Ops.Gl.Matrix.Quaternion

**Description:** multiplies current modelmatrix with a quaternion

#### > Input Ports:

- **Render** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **W** (Number)

#### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.Quaternion>

## 67.1.21 QuaternionCamera



**Full Name:** Ops.Gl.Matrix.QuaternionCamera

**Description:** Set up a camera, rotated by a quaternion

#### > Input Ports:

- **Render** (Trigger)
- **EyeX** (Number)
- **EyeY** (Number)

#### > Output Ports:

- **EyeZ** (Number)
- **QuatX** (Number)
- **QuatY** (Number)
- **QuatZ** (Number)
- **QuatW** (Number)
- **UpX** (Number)
- **UpY** (Number)
- **UpZ** (Number)

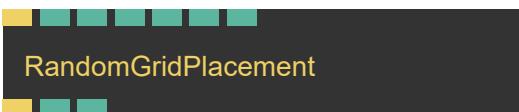
#### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.QuaternionCamera>

## 67.1.22 RandomGridPlacement



**Full Name:** Ops.Gl.Matrix.RandomGridPlacement

**Description:** place random objects on a grid

#### > Input Ports:

- **Exe** (Trigger)
- **Max Depth** (Number)
- **Possibility** (Number)
- **Seed** (Number)
- **Scale** (Number)
- **Width** (Number)
- **Height** (Number)

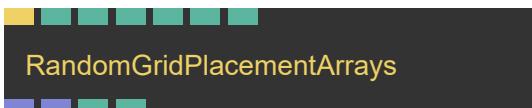
#### < Output Ports:

- **Next** (Trigger)
- **Index** (Number)
- **Depth** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.RandomGridPlacement>

## 67.1.23 RandomGridPlacementArrays



**Full Name:** Ops.Gl.Matrix.RandomGridPlacementArrays

**Description:** Place random objects on a grid

**> Input Ports:**

- **Exe** (Trigger)
- **Max Depth** (Number)
- **Possibility** (Number)
- **Seed** (Number)
- **Scale** (Number)
- **Width** (Number)
- **Height** (Number)

**< Output Ports:**

- **Positions** (Array)
- **Scalings** (Array)
- **Array Length** (Number)
- **Total Points** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.RandomGridPlacementArrays>

## 67.1.24 Scale



**Full Name:** Ops.Gl.Matrix.Scale

**Description:** Scale all child objects (scaleXYZ)

**> Input Ports:**

- **Render** (Trigger)
- **Scale** (Number)
- **X** (Number)
- **Y** (Number)

- **Z** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.Scale>

## 67.1.25 ScaleXYZViewMatrix



**Full Name:** Ops.Gl.Matrix.ScaleXYZViewMatrix

**Description:** scale xyz of viewmatrix

**> Input Ports:**

- **Render** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.ScaleXYZViewMatrix>

## 67.1.26 ScreenCoordinates\_v2



**Full Name:** Ops.Gl.Matrix.ScreenCoordinates\_v2

**Description:** screen/pixel coordinates of the current transform

**> Input Ports:**

- **Execute** (Trigger)
- **Pixel Unit Index** (Number: Integer)

**< Output Ports:**

- **Trigger** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Visible** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Matrix.ScreenCoordinates\\_v2](https://cables.gl/op/Ops.Gl.Matrix.ScreenCoordinates_v2)

### 67.1.27 ScreenPosTo3d\_v3



**Full Name:** Ops.Gl.Matrix.ScreenPosTo3d\_v3

**Description:** convert screen coordinates to a 3d position

**> Input Ports:**

- **Exec** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Input Type Index** (Number: Integer)

**< Output Ports:**

- **Trigger Out** (Trigger)
- **Result X** (Number)
- **Result Y** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Matrix.ScreenPosTo3d\\_v3](https://cables.gl/op/Ops.Gl.Matrix.ScreenPosTo3d_v3)

### 67.1.28 SetProjectionMatrix



**Full Name:** Ops.Gl.Matrix.SetProjectionMatrix

**Description:** set a projection matrix

**> Input Ports:**

- **Exe** (Trigger)
- **Matrix** (Array)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.SetProjectionMatrix>

### 67.1.29 Shear



**Full Name:** Ops.Gl.Matrix.Shear

**Description:** displaces each point of a mesh in fixed direction

**> Input Ports:**

- **Render** (Trigger)
- **ShearX** (Number)
- **ShearY** (Number)

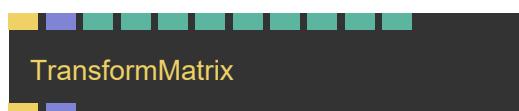
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.Shear>

### 67.1.30 TransformMatrix



**Full Name:** Ops.Gl.Matrix.TransformMatrix

**Description:** transform a matrix (mat4)

**> Input Ports:**

- **Transform** (Trigger)

- **Matrix** (Array)
- **Translate X** (Number)
- **Translate Y** (Number)
- **Translate Z** (Number)
- **Scale X** (Number)
- **Scale Y** (Number)
- **Scale Z** (Number)
- **Rotation X** (Number)
- **Rotation Y** (Number)
- **Rotation Z** (Number)

**< Output Ports:**

- **Next** (Trigger)
- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.TransformMatrix>

## 67.1.31 TransformMul



**Full Name:** Ops.Gl.Matrix.TransformMul

**Description:** multiply current modelmatrix

**> Input Ports:**

- **Render** (Trigger)
- **Mul** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.TransformMul>

## 67.1.32 Translate



**Full Name:** Ops.Gl.Matrix.Translate

**Description:** Translate objects (move / position in 3D space)

**> Input Ports:**

- **Render** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.Translate>

## 67.1.33 TranslateView



**Full Name:** Ops.Gl.Matrix.TranslateView

**Description:** translate the view/camera matrix

**> Input Ports:**

- **Render** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)

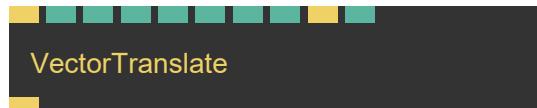
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.TranslateView>

## 67.1.34 VectorTranslate



VectorTranslate

**Full Name:** Ops.Gl.Matrix.VectorTranslate

**Description:** Translate any geometry underneath it using vectors and speed.

### > Input Ports:

- **Exec** (Trigger)
- **Speed** (Number)
- **Vector X** (Number)
- **Vector Y** (Number)
- **Vector Z** (Number)
- **Reset Position X** (Number)
- **Reset Position Y** (Number)
- **Reset Position Z** (Number)
- **Reset** (Trigger)
- **Max** (Number)

### < Output Ports:

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Matrix.VectorTranslate>

- **Allow Flying** (Number: Boolean)
- **Active** (Number: Boolean)
- **Move X-** (Number: Boolean)
- **Move Y-** (Number: Boolean)
- **Reset** (Trigger)

### < Output Ports:

- **Trigger** (Trigger)
- **IsLocked** (booleanNumber)
- **PosX** (Number)
- **PosY** (Number)
- **PosZ** (Number)
- **Mouse Left** (Trigger)
- **Mouse Right** (Trigger)
- **Dir X** (Number)
- **Dir Y** (Number)
- **Dir Z** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Matrix.WASDCamera\\_v2](https://cables.gl/op/Ops.Gl.Matrix.WASDCamera_v2)

## 67.1.35 WASDCamera\_v2



**Full Name:** Ops.Gl.Matrix.WASDCamera\_v2

**Description:** simple camera you control with W,A,S,D keys like in a FPS game

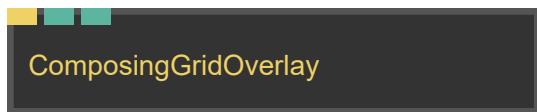
### > Input Ports:

- **Render** (Trigger)
- **Enable Pointer Lock** (Number: Boolean)
- **Speed** (Number)
- **Mouse Speed** (Number)

# 68 Ops.Gl.Meshes

## 68.1 Ops.Gl.Meshes

### 68.1.1 ComposingGridOverlay



**Full Name:** Ops.Gl.Meshes.ComposingGridOverlay

**Description:** Rule of thirds image composition helper

#### > Input Ports:

- **Render** (Trigger)
- **Scale** (Number)
- **Show Center** (Number: Boolean)

#### < Output Ports:

- Visit *Ops.Gl.Meshes.ComposingGridOverlay documentation* for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Meshes.ComposingGridOverlay>

### 68.1.2 Cone



**Full Name:** Ops.Gl.Meshes.Cone

**Description:** number of horizontal segments

#### > Input Ports:

- **Render** (Trigger)
- **Slices** (Number)
- **Stacks** (Number)
- **Radius** (Number)

- **Height** (Number)
- **Active** (Number: Boolean)

#### < Output Ports:

- **Trigger** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Meshes.Cone>

### 68.1.3 Corner



**Full Name:** Ops.Gl.Meshes.Corner

**Description:** render a rectangular corner

#### > Input Ports:

- **Render** (Trigger)
- **Width** (Number)
- **Height** (Number)
- **Thickness** (Number)
- **Draw** (Number: Boolean)
- **Pivot X Index** (Number: Integer)
- **Pivot Y Index** (Number: Integer)

#### < Output Ports:

- **Trigger** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Meshes.Corner>

### 68.1.4 Cylinder\_v2



**Full Name:** Ops.Gl.Meshes.Cylinder\_v2

**Description:** draw parameterizable cylinder (aka tube,pipe,round,circle)

**> Input Ports:**

- **Render** (Trigger)
- **Render Mesh** (Number: Boolean)
- **Segments** (Number: Integer)
- **Stacks** (Number: Integer)
- **Length** (Number)
- **Outer Radius** (Number)
- **Inner Radius** (Number)
- **Flip Mapping** (Number: Boolean)
- **Caps** (Number: Boolean)
- **Flat Normals** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Meshes.Cylinder\\_v2](https://cables.gl/op/Ops.Gl.Meshes.Cylinder_v2)

## 68.1.5 FloorGrid



**Full Name:** Ops.Gl.Meshes.FloorGrid

**Description:** draw a grid on the floor

**> Input Ports:**

- **Render** (Trigger)
- **Active** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Meshes.FloorGrid>

## 68.1.6 FreeFormPlane



**Full Name:** Ops.Gl.Meshes.FreeFormPlane

**Description:** A freely deformable plane, rectangle, polygon

**> Input Ports:**

- **Render** (Trigger)
- **X 1** (Number)
- **Y 1** (Number)
- **Z 1** (Number)
- **X 2** (Number)
- **Y 2** (Number)
- **Z 2** (Number)
- **X 3** (Number)
- **Y 3** (Number)
- **Z 3** (Number)
- **X 4** (Number)
- **Y 4** (Number)
- **Z 4** (Number)
- **Tc X 1** (Number)
- **Tc Y 1** (Number)
- **Tc X 2** (Number)
- **Tc Y 2** (Number)
- **Tc X 3** (Number)
- **Tc Y 3** (Number)
- **Tc X 4** (Number)
- **Tc Y 4** (Number)

**< Output Ports:**

- **Trigger** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Meshes.FreeFormPlane>

## 68.1.7 FullscreenRectangle\_v2



**Full Name:** Ops.Gl.Meshes.FullscreenRectangle\_v2

**Description:** Draws a rectangle using the full WebGL canvas size

### > Input Ports:

- **Render** (Trigger)
- **Flip Y** (Number: Boolean)
- **Flip X** (Number: Boolean)
- **Texture** (Object:Texture)

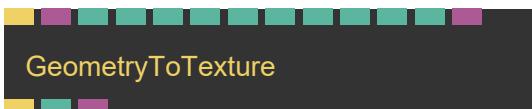
### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Meshes.FullscreenRectangle\\_v2](https://cables.gl/op/Ops.Gl.Meshes.FullscreenRectangle_v2)

## 68.1.8 GeometryToTexture\_v3



**Full Name:** Ops.Gl.Meshes.GeometryToTexture\_v3

**Description:** Convert vertices of a geometry to a data texture

### > Input Ports:

- **Render** (Trigger)
- **Geometry** (Object:Geometry)
- **Continously Update** (Number: Boolean)
- **Order Index** (Number: Integer)
- **Content Index** (Number: Integer)
- **New Size** (Number)
- **Tex Width** (Number: Integer)
- **Filter Index** (Number: Integer)
- **Wrap Index** (Number: Integer)
- **PixelFormat Index** (Number: Integer)

- **Color Texture** (Object:Texture)

### < Output Ports:

- **Next** (Trigger)
- **Total Vertices** (Number)
- **Texture** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Meshes.GeometryToTexture\\_v3](https://cables.gl/op/Ops.Gl.Meshes.GeometryToTexture_v3)

## 68.1.9 Grid



**Full Name:** Ops.Gl.Meshes.Grid

**Description:** Draw a simple grid of lines

### > Input Ports:

- **Render** (Trigger)
- **Num** (Number: Integer)
- **Spacing** (Number)
- **Center** (Number: Boolean)
- **Axis Index** (Number: Integer)

### < Output Ports:

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Meshes.Grid>

## 68.1.10 HeightMap



**Full Name:** Ops.Gl.Meshes.HeightMap

**Description:** generate a rectangular mesh where the height is defined by the luminance of an image

#### > Input Ports:

- **Render** (Trigger)
- **File** (String)
- **Extrude** (Number)
- **Width** (Number)
- **Height** (Number)
- **Rows** (Number: Integer)
- **Columns** (Number: Integer)
- **TexCoords Slice** (Number: Boolean)
- **Flat** (Number: Boolean)

#### < Output Ports:

- **Trigger** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Meshes.HeightMap>

## 68.1.11 Helix



**Full Name:** Ops.Gl.Meshes.Helix

**Description:** generates a helix, spiral spline

#### > Input Ports:

- **Render** (Trigger)
- **Draw** (Number: Boolean)
- **Segments** (Number)
- **Frequency** (Number)
- **Radius** (Number)
- **Radius End** (Number)
- **Height** (Number)

#### < Output Ports:

- **Next** (Trigger)
- **Points** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Meshes.Helix>

## 68.1.12 Icosahedron\_v2



**Full Name:** Ops.Gl.Meshes.Icosahedron\_v2

**Description:** Renders a icosahedron (polyhedron with 20 faces)

#### > Input Ports:

- **Render** (Trigger)
- **Smooth** (Number: Boolean)
- **Render Mesh** (Number: Boolean)

#### < Output Ports:

- **Trigger** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Meshes.Icosahedron\\_v2](https://cables.gl/op/Ops.Gl.Meshes.Icosahedron_v2)

## 68.1.13 Line



**Full Name:** Ops.Gl.Meshes.Line

**Description:** Draw a line between two points

#### > Input Ports:

- **Render** (Trigger)
- **X 1** (Number)
- **Y 1** (Number)
- **Z 1** (Number)
- **X 2** (Number)
- **Y 2** (Number)
- **Z 2** (Number)

#### < Output Ports:

- **Next** (Trigger)
- **Array** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Meshes.Line>

## 68.1.14 LinesArray



**Full Name:** Ops.Gl.Meshes.LinesArray

**Description:** an array of lines

### > Input Ports:

- **Render** (Trigger)
- **Width** (Number)
- **Height** (Number)
- **Logarithmic** (Number: Boolean)
- **Pivot X Index** (Number: Integer)
- **Pivot Y Index** (Number: Integer)
- **Num Columns** (Number: Integer)
- **Num Rows** (Number: Integer)
- **Axis Index** (Number: Integer)

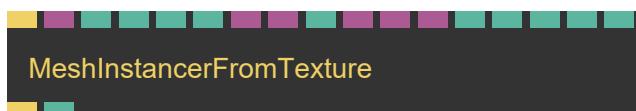
### < Output Ports:

- **Trigger** (Trigger)
- **Point Arrays** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Meshes.LinesArray>

## 68.1.15 MeshInstancerFromTexture\_v3



**Full Name:** Ops.Gl.Meshes.MeshInstancerFromTexture\_v3

**Description:** Draw the same mesh multiple times on the GPU

### > Input Ports:

- **Exe** (Trigger)
- **Geometry** (Object:Geometry)
- **Scale** (Number)
- **Limit Instances** (Number: Boolean)
- **Num Instances** (Number: Integer)
- **Position Texture** (Object:Texture)
- **Rotation Texture** (Object:Texture)
- **Scale Texture** (Object:Texture)
- **Color Texture** (Object:Texture)
- **TexCoord Texture** (Object:Texture)
- **Ignore Alpha Less Than** (Number)
- **Multiply Pos X** (Number)
- **Multiply Pos Y** (Number)
- **Multiply Pos Z** (Number)

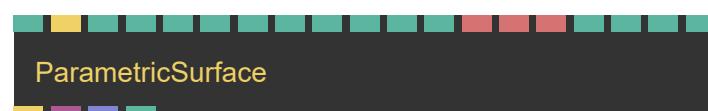
### < Output Ports:

- **Trigger Out** (Trigger)
- **Num** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Meshes.MeshInstancerFromTexture\\_v3](https://cables.gl/op/Ops.Gl.Meshes.MeshInstancerFromTexture_v3)

## 68.1.16 ParametricSurface



**Full Name:** Ops.Gl.Meshes.ParametricSurface

**Description:** Creates a 3d mesh from a 2d area expressions

### > Input Ports:

- **Shapes Index** (Number: Integer)
- **Render** (Trigger)
- **U Segments** (Number: Integer)
- **V Segments** (Number: Integer)
- **Multiple Of PI - U** (Number: Boolean)
- **UMin** (Number)
- **UMax** (Number)
- **Displace U** (Number)
- **Multiple Of PI - V** (Number: Boolean)

- **VMin** (Number)
- **VMax** (Number)
- **Displace V** (Number)
- **X Function** (String)
- **Y Function** (String)
- **Z Function** (String)
- **Scale X** (Number)
- **Scale Y** (Number)
- **Scale Z** (Number)
- **Draw** (Number: Boolean)

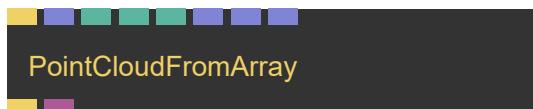
**< Output Ports:**

- **Trigger** (Trigger)
- **Geometry** (Object)
- **Position** (Array)
- **outputs the vertices of the surface** (as an xyz-Array)
- **Position Amount** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Meshes.ParametricSurface>

## 68.1.17 PointCloudFromArray\_v2



**Full Name:** Ops.Gl.Meshes.PointCloudFromArray\_v2

**Description:** visualize an array of coordinates as points

**> Input Ports:**

- **Exe** (Trigger)
- **Positions** (Array)
- **Num Points** (Number: Integer)
- **Scramble Texcoords** (Number: Boolean)
- **Seed** (Number)
- **Texture Coordinates** (Array)
- **Point Sizes** (Array)
- **Vertex Colors** (Array)

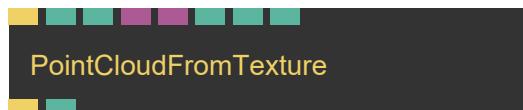
**< Output Ports:**

- **Trigger Out** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Meshes.PointCloudFromArray\\_v2](https://cables.gl/op/Ops.Gl.Meshes.PointCloudFromArray_v2)

## 68.1.18 PointCloudFromTexture



**Full Name:** Ops.Gl.Meshes.PointCloudFromTexture

**Description:** Visualize a RGB texture as XYZ coordinates as points

**> Input Ports:**

- **Render** (Trigger)
- **Num Points** (Number: Integer)
- **Texture** (Object:Texture)
- **Point Size** (Object:Texture)
- **Normalize** (Number: Boolean)
- **Remove Point At 0** (Number: Boolean)
- **Ignore Alpha 0** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)
- **Total Points** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Meshes.PointCloudFromTexture>

## 68.1.19 Polyhedron\_v2



**Full Name:** Ops.Gl.Meshes.Polyhedron\_v2

**Description:** Generate polyhedron meshes

**> Input Ports:**

- **Receipt** (String)

**< Output Ports:**

- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Meshes.Polyhedron\\_v2](https://cables.gl/op/Ops.Gl.Meshes.Polyhedron_v2)

## 68.1.20 Pyramid\_v2



**Full Name:** Ops.Gl.Meshes.Pyramid\_v2

**Description:** render a pyramid mesh

**> Input Ports:**

- **Render** (Trigger)
- **Width** (Number)
- **Length** (Number)
- **Height** (Number)
- **Smooth** (Number: Boolean)
- **Draw** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Meshes.Pyramid\\_v2](https://cables.gl/op/Ops.Gl.Meshes.Pyramid_v2)

## 68.1.21 QuadWarpTexture



**Full Name:** Ops.Gl.Meshes.QuadWarpTexture

**Description:** Warp a texture mapped quad (projection mapping)

**> Input Ports:**

- **Render** (Trigger)

- **A X** (Number)
- **A Y** (Number)
- **B X** (Number)
- **B Y** (Number)
- **C X** (Number)
- **C Y** (Number)
- **D X** (Number)
- **D Y** (Number)
- **Flip Y** (Number: Boolean)
- **Flip X** (Number: Boolean)
- **Texture** (Object)

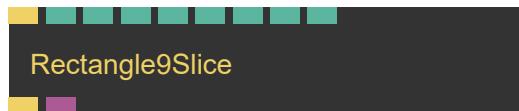
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Meshes.QuadWarpTexture>

## 68.1.22 Rectangle9Slice



**Full Name:** Ops.Gl.Meshes.Rectangle9Slice

**Description:** nine slice image format texture mapped rectangle

**> Input Ports:**

- **Render** (Trigger)
- **Width** (Number)
- **Height** (Number)
- **Border Width** (Number)
- **Scale X** (Number)
- **Scale Y** (Number)
- **Draw** (Number: Boolean)
- **Pivot X Index** (Number: Integer)
- **Pivot Y Index** (Number: Integer)

**< Output Ports:**

- **Trigger** (Trigger)

- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Meshes.Rectangle9Slice>

### 68.1.23 RectangleFrame\_v2



**Full Name:** Ops.Gl.Meshes.RectangleFrame\_v2

**Description:** Draws a rectangle frame

**> Input Ports:**

- **Render** (Trigger)
- **Width** (Number)
- **Height** (Number)
- **Thickness** (Number)
- **Draw Top** (Number: Boolean)
- **Draw Bottom** (Number: Boolean)
- **Draw Left** (Number: Boolean)
- **Draw Right** (Number: Boolean)
- **Active** (Number: Boolean)

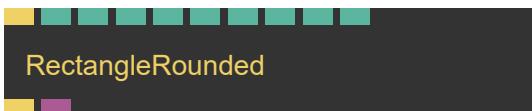
**< Output Ports:**

- **Trigger** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Meshes.RectangleFrame\\_v2](https://cables.gl/op/Ops.Gl.Meshes.RectangleFrame_v2)

### 68.1.24 RectangleRounded\_v2



**Full Name:** Ops.Gl.Meshes.RectangleRounded\_v2

**Description:** Draws a rectangle with rounded corners

**> Input Ports:**

- **Render** (Trigger)
- **Segments** (Number: Integer)
- **Width** (Number)
- **Height** (Number)
- **Border Radius** (Number)
- **Top Left** (Number: Boolean)
- **Top Right** (Number: Boolean)
- **Bottom Left** (Number: Boolean)
- **Bottom Right** (Number: Boolean)
- **Draw** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Meshes.RectangleRounded\\_v2](https://cables.gl/op/Ops.Gl.Meshes.RectangleRounded_v2)

### 68.1.25 SimpleSpline\_v2



**Full Name:** Ops.Gl.Meshes.SimpleSpline\_v2

**Description:** Draws a simple spline only one pixel wide

**> Input Ports:**

- **Render** (Trigger)
- **Points** (Array)
- **Num Points** (Number: Integer)
- **Line Strip** (Number: Boolean)
- **TexCoords Array** (Array)
- **Vertex Colors** (Array)

**< Output Ports:**

- **Geometry** (Object)
- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Meshes.SimpleSpline\\_v2](https://cables.gl/op/Ops.Gl.Meshes.SimpleSpline_v2)

## 68.1.26 SimpleWireframe



**Full Name:** Ops.Gl.Meshes.SimpleWireframe

**Description:** Simple Wireframe Line Renderer

**> Input Ports:**

- **Render** (Trigger)
- **Geometry** (Object:Geometry)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Meshes.SimpleWireframe>

## 68.1.27 SplineMesh\_v2



**Full Name:** Ops.Gl.Meshes.SplineMesh\_v2

**Description:** draw splines/lines

**> Input Ports:**

- **Render** (Trigger)
- **Points** (Array)
- **Tesselate Edges** (Number: Boolean)
- **Render Mesh** (Number: Boolean)

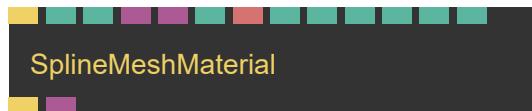
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Meshes.SplineMesh\\_v2](https://cables.gl/op/Ops.Gl.Meshes.SplineMesh_v2)

## 68.1.28 SplineMeshMaterial\_v2



**Full Name:** Ops.Gl.Meshes.SplineMeshMaterial\_v2

**Description:** material for splinemesh

**> Input Ports:**

- **Render** (Trigger)
- **Width** (Number)
- **Width Perspective** (Number: Boolean)
- **Texture** (Object:Texture)
- **Texture Mask** (Object:Texture)
- **Mapping Index** (Number: Integer)
- **Mapping** (String)
- **Colorize Texture** (Number: Boolean)
- **Offset** (Number)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)

**< Output Ports:**

- **Trigger** (Trigger)
- **Shader** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Meshes.SplineMeshMaterial\\_v2](https://cables.gl/op/Ops.Gl.Meshes.SplineMeshMaterial_v2)

## 68.1.29 TextMesh\_v2



**Full Name:** Ops.Gl.Meshes.TextMesh\_v2

**Description:** Draws text in 3d space using one of the font ops

**> Input Ports:**

- **Render** (Trigger)
- **Text** (String)
- **Scale Text** (Number)
- **Line Scale** (Number)
- **Font** (String)
- **Align Index** (Number: Integer)
- **Vertical Align Index** (Number: Integer)
- **Line Height** (Number)
- **Letter Spacing** (Number)
- **Texture Color** (Object:Texture)
- **Texture Mask** (Object:Texture)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)

**< Output Ports:**

- **Next** (Trigger)
- **Total Lines** (Number)
- **Width** (Number)
- **Font Available** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Meshes.TextMesh\\_v2](https://cables.gl/op/Ops.Gl.Meshes.TextMesh_v2)

## 68.1.30 Torus\_v3



**Full Name:** Ops.Gl.Meshes.Torus\_v3

**Description:** Draw a torus (doughnut, donut, ring mesh)

**> Input Ports:**

- **Render** (Trigger)
- **Sides** (Number)
- **Rings** (Number)
- **InnerRadius** (Number)
- **OuterRadius** (Number)
- **Render Mesh** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Meshes.Torus\\_v3](https://cables.gl/op/Ops.Gl.Meshes.Torus_v3)

## 68.1.31 TriangleSphere



**Full Name:** Ops.Gl.Meshes.TriangleSphere

**Description:** A sphere mesh with uniform distributed vertices

**> Input Ports:**

- **Render** (Trigger)
- **Iterations** (Number)
- **Flat** (Number: Boolean)
- **Draw** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Meshes.TriangleSphere>

# 69 Ops.Gl.Pbr

## 69.1 Ops.Gl.Pbr

### 69.1.1 PbrEnvironmentLight



**Full Name:** Ops.Gl.Pbr.PbrEnvironmentLight

**Description:** PBR image based lighting setup

#### > Input Ports:

- **Render** (Trigger)
- **Intensity** (Number)
- **RGBE Environment Map** (Object:Texture)
- **Size Irradiance Map Index** (Number: Integer)
- **Size Pre-Filtered Environment Index** (Number: Integer)
- **Size IBL LUT Index** (Number: Integer)
- **Force 8bit IBL** (Number: Boolean)
- **Rotation** (Number)
- **Use Parallax Correction** (Number: Boolean)
- **Center X** (Number)
- **Center Y** (Number)
- **Center Z** (Number)
- **Box Min X** (Number)
- **Box Min Y** (Number)
- **Box Min Z** (Number)
- **Box Max X** (Number)
- **Box Max Y** (Number)
- **Box Max Z** (Number)

#### < Output Ports:

- **Render** (Trigger)
- **Intensity** (Number)
- **RGBE Environment Map** (Object:Texture)
- **Size Irradiance Map Index** (Number: Integer)

- **Size Pre-Filtered Environment Index** (Number: Integer)
- **Size IBL LUT Index** (Number: Integer)
- **Force 8bit IBL** (Number: Boolean)
- **Rotation** (Number)
- **Use Parallax Correction** (Number: Boolean)
- **Center X** (Number)
- **Center Y** (Number)
- **Center Z** (Number)
- **Box Min X** (Number)
- **Box Min Y** (Number)
- **Box Min Z** (Number)
- **Box Max X** (Number)
- **Box Max Y** (Number)
- **Box Max Z** (Number)
- **Next** (Trigger)
- **IBL LUT** (Object)
- **for PBR Material** (not required)
- **Number Of Pre-Filtered Mip Levels** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Pbr.PbrEnvironmentLight>

### 69.1.2 PbrMaterial



**Full Name:** Ops.Gl.Pbr.PbrMaterial

**Description:** PBR/Physical Based Rendering Material for realistic materials

#### > Input Ports:

- **Render** (Trigger)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)
- **Roughness** (Number)
- **Metalness** (Number)
- **Use Clear Coat** (Number: Boolean)
- **Clear Coat Intensity** (Number)

- **Clear Coat Roughness** (Number)
- **Use Normal Map For Clear Coat** (Number: Boolean)
- **Clear Coat Normal Map** (Object:Texture)
- **Use Thin Film** (Number: Boolean)
- **Thin Film Intensity** (Number)
- **Thin Film IOR** (Number)
- **Thickness Tex Min** (Number)
- **Thickness Tex Max** (Number)
- **Exposure** (Number)
- **Emission Intensity** (Number)
- **Disable Geometric Roughness** (Number: Boolean)
- **Use Roughness From Normal Map** (Number: Boolean)
- **Use Vertex Colours** (Number: Boolean)
- **Height Intensity** (Number)
- **Faster Heightmapping** (Number: Boolean)
- **Double Sided** (Number: Boolean)
- **IBL LUT** (Object:Texture)
- **Diffuse Irradiance** (Object:Texture)
- **Pre-Filtered Envmap** (Object:Texture)
- **Num Mip Levels** (Number: Integer)
- **Albedo** (Object:Texture)
- **AORM** (Object:Texture)
- **Normal Map** (Object:Texture)
- **Emission** (Object:Texture)
- **Height** (Object:Texture)
- **Lightmap** (Object:Texture)
- **Thin Film** (Object:Texture)
- **Diffuse Intensity** (Number)
- **Specular Intensity** (Number)
- **Lightmap Is RGBE** (Number: Boolean)
- **Lightmap Intensity** (Number)

**< Output Ports:**

- **Next** (Trigger)
- **Shader** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Pbr.PbrMaterial>

# 70 Ops.Gl.Phong

---

## 70.1 Ops.Gl.Phong

### 70.1.1 AmbientLight\_v4



**Full Name:** Ops.Gl.Phong.AmbientLight\_v4

**Description:** ambient light for phong material shading

**> Input Ports:**

- **Trigger In** (Trigger)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **Intensity** (Number)

**< Output Ports:**

- **Trigger Out** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Phong.AmbientLight\\_v4](https://cables.gl/op/Ops.Gl.Phong.AmbientLight_v4)

### 70.1.2 DirectionalLight\_v5



**Full Name:** Ops.Gl.Phong.DirectionalLight\_v5

**Description:** Directional light for phong shading

**> Input Ports:**

- **Trigger In** (Trigger)
- **Cast Light** (Number: Boolean)
- **Intensity** (Number)

- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **Specular R** (Number)
- **Specular G** (Number)
- **Specular B** (Number)
- **Cast Shadow** (Number: Boolean)
- **Rendering Active** (Number: Boolean)
- **Map Size Index** (Number: Integer)
- **Map Size** (String)
- **Shadow Strength** (Number)
- **LR-BottomTop** (Number)
- **Near** (Number)
- **Far** (Number)
- **Bias** (Number)
- **Polygon Offset** (Number: Integer)
- **Normal Offset** (Number)
- **Blur Amount** (Number)
- **Enable Advanced** (Number: Boolean)
- **MSAA Index** (Number: Integer)
- **MSAA** (String)
- **Texture Filter Index** (Number: Integer)
- **Texture Filter** (String)
- **Anisotropic Index** (Number: Integer)
- **Anisotropic** (String)

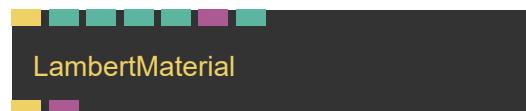
**< Output Ports:**

- **Trigger Out** (Trigger)
- **Shadow Map** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Phong.Directionallight\\_v5](https://cables.gl/op/Ops.Gl.Phong.Directionallight_v5)

### 70.1.3 LambertMaterial\_v2



**Full Name:** Ops.Gl.Phong.LambertMaterial\_v2

**Description:** a simple shaded material

**> Input Ports:**

- **Execute** (Trigger)
- **Diffuse R** (Number)
- **Diffuse G** (Number)
- **Diffuse B** (Number)
- **Diffuse A** (Number)
- **Diffuse Texture** (Object:Texture)
- **Colorize Texture** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)
- **Shader** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Phong.LambertMaterial\\_v2](https://cables.gl/op/Ops.Gl.Phong.LambertMaterial_v2)

### 70.1.4 PhongMaterial\_v6



**Full Name:** Ops.Gl.Phong.PhongMaterial\_v6

**Description:** A shaded material for lighting objects

**> Input Ports:**

- **Trigger In** (Trigger)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)
- **Albedo** (Number)
- **Roughness** (Number)
- **Fresnel Intensity** (Number)

- **Fresnel Width** (Number)
- **Fresnel Exponent** (Number)
- **Fresnel R** (Number)
- **Fresnel G** (Number)
- **Fresnel B** (Number)
- **Emissive Active** (Number: Boolean)
- **Color Intensity** (Number)
- **Emissive R** (Number)
- **Emissive G** (Number)
- **Emissive B** (Number)
- **Shininess** (Number)
- **Specular Amount** (Number)
- **Diffuse Texture** (Object:Texture)
- **Specular Texture** (Object:Texture)
- **Normal Map** (Object:Texture)
- **AO Texture** (Object:Texture)
- **Emissive Texture** (Object:Texture)
- **Emissive Mask** (Object:Texture)
- **Opacity Texture** (Object:Texture)
- **Environment Map** (Object:Texture)
- **Env Map Mask** (Object:Texture)
- **Diffuse Repeat X** (Number)
- **Diffuse Repeat Y** (Number)
- **Texture Offset X** (Number)
- **texture pixel offset on the C axis** (applied to all textures)
- **Texture Offset Y** (Number)
- **texture pixel offset on the Y axis** (applied to all textures)
- **Specular Intensity** (Number)
- **Normal Map Intensity** (Number)
- **AO Intensity** (Number)
- **Emissive Intensity** (Number)
- **Emissive Mask Intensity** (Number)
- **Env Map Intensity** (Number)
- **Env Mask Intensity** (Number)

#### < Output Ports:

- **Trigger Out** (Trigger)
- **Shader** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Phong.PhongMaterial\\_v6](https://cables.gl/op/Ops.Gl.Phong.PhongMaterial_v6)

## 70.1.5 PointLight\_v5



**Full Name:** Ops.Gl.Phong.PointLight\_v5

**Description:** Point light for phong shading

#### > Input Ports:

- **Trigger In** (Trigger)
- **Cast Light** (Number: Boolean)
- **Intensity** (Number)
- **Radius** (Number)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **Specular R** (Number)
- **Specular G** (Number)
- **Specular B** (Number)
- **Falloff** (Number)
- **Cast Shadow** (Number: Boolean)
- **Rendering Active** (Number: Boolean)
- **Shadow Strength** (Number)
- **Near** (Number)
- **Far** (Number)
- **Bias** (Number)
- **Polygon Offset** (Number: Integer)

#### < Output Ports:

- **Trigger Out** (Trigger)
- **Cubemap** (Object)
- **World Position X** (Number)
- **World Position Y** (Number)
- **World Position Z** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Phong.PointLight\\_v5](https://cables.gl/op/Ops.Gl.Phong.PointLight_v5)

## 70.1.6 ResetLights



**Full Name:** Ops.Gl.Phong.ResetLights

**Description:** reset lights for everything triggered after

**> Input Ports:**

- **Trigger In** (Trigger)
- **Reset Lights** (Number: Boolean)

**< Output Ports:**

- **Trigger Out** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Phong.ResetLights>

- **B** (Number)
- **Specular R** (Number)
- **Specular G** (Number)
- **Specular B** (Number)
- **Cone Angle** (Number)
- **Inner Cone Angle** (Number)
- **Spot Exponent** (Number)
- **Falloff** (Number)
- **Cast Shadow** (Number: Boolean)
- **Rendering Active** (Number: Boolean)
- **Shadow Strength** (Number)

**< Output Ports:**

- **Trigger Out** (Trigger)
- **Shadow Map** (Object)
- **World Position X** (Number)
- **World Position Y** (Number)
- **World Position Z** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Phong.SpotLight\\_v5](https://cables.gl/op/Ops.Gl.Phong.SpotLight_v5)



**Full Name:** Ops.Gl.Phong.SpotLight\_v5

**Description:** spot light that emits a cone of light

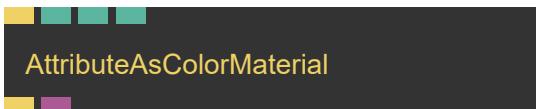
**> Input Ports:**

- **Trigger In** (Trigger)
- **Cast Light** (Number: Boolean)
- **Intensity** (Number)
- **Radius** (Number)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Point At X** (Number)
- **Point At Y** (Number)
- **Point At Z** (Number)
- **R** (Number)
- **G** (Number)

# 71 Ops.Gl.Shader

## 71.1 Ops.GlShader

### 71.1.1 AttributeAsColorMaterial



**Full Name:** Ops.Gl.Shader.AttributeAsColorMaterial

**Description:** render mesh normals as colors

#### > Input Ports:

- **Render** (Trigger)
- **Absolute** (Number: Boolean)
- **World Space** (Number: Boolean)

#### < Output Ports:

- **Trigger** (Trigger)
- **Shader** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.GlShader.AttributeAsColorMaterial>

### 71.1.2 BasicMaterial\_v3



**Full Name:** Ops.Gl.Shader.BasicMaterial\_v3

**Description:** A material without shading

#### > Input Ports:

- **Render** (Trigger)
- **R** (Number)
- **G** (Number)
- **B** (Number)

- **A** (Number)
- **Texture** (Object:Texture)
- **ColorizeTexture** (Number: Boolean)
- **Vertex Colors** (Number: Boolean)
- **TextureOpacity** (Object:Texture)
- **Opacity TexCoords Transform** (Number: Boolean)
- **Discard Transparent Pixels** (Number: Boolean)
- **DiffuseRepeatX** (Number)
- **DiffuseRepeatY** (Number)
- **Tex Offset X** (Number)
- **Tex Offset Y** (Number)
- **Crop TexCoords** (Number: Boolean)
- **Billboard** (Number: Boolean)

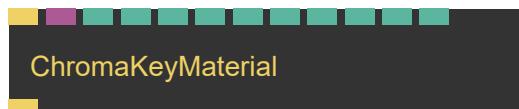
#### < Output Ports:

- **Trigger** (Trigger)
- **Shader** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.GlShader.BasicMaterial\\_v3](https://cables.gl/op/Ops.GlShader.BasicMaterial_v3)

### 71.1.3 ChromaKeyMaterial



**Full Name:** Ops.Gl.Shader.ChromaKeyMaterial

**Description:** display texture and replace a color with transparency

#### > Input Ports:

- **Render** (Trigger)
- **Texture** (Object)
- **Mode Index** (Number: Integer)
- **WeightMul** (Number)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **White** (Number)
- **DiffuseRepeatX** (Number)
- **DiffuseRepeatY** (Number)

- **Tex Offset X** (Number)
- **Tex Offset Y** (Number)

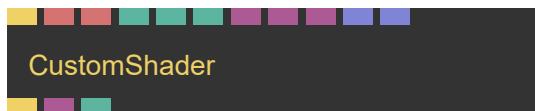
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.GlShader.ChromaKeyMaterial>

## 71.1.4 CustomShader\_v2



**Full Name:** Ops.Gl.Shader.CustomShader\_v2

**Description:** Write your own custom shader

**> Input Ports:**

- **Render** (Trigger)
- **Fragment Code** (String)
- **Vertex Code** (String)
- **Use As Material** (Number: Boolean)
- **W** (Number)
- **H** (Number)
- **GPosition** (Object)
- **GNormal** (Object)
- **TexNoise** (Object)
- **Samples** (Array)
- **Projection** (Array)

**< Output Ports:**

- **Trigger** (Trigger)
- **Shader** (Object)
- **Has Errors** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.GlShader.CustomShader\\_v2](https://cables.gl/op/Ops.GlShader.CustomShader_v2)

## 71.1.5 ErrorMaterial

ErrorMaterial

**Full Name:** Ops.Gl.Shader.ErrorMaterial

**Description:** draw meshes using the cables error material shader

**> Input Ports:**

- **Render** (Trigger)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.GlShader.ErrorMaterial>

## 71.1.6 FrontBacksideMaterial

FrontBacksideMaterial

**Full Name:** Ops.Gl.Shader.FrontBacksideMaterial

**Description:** visualize which faces are facing the camera

**> Input Ports:**

- **Render** (Trigger)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.GlShader.FrontBacksideMaterial>

## 71.1.7 GetShader

GetShader

**Full Name:** Ops.Gl.Shader.GetShader

**Description:** get current set shader

**> Input Ports:**

- **Update** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **Shader** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Shader.GetShader>

## 71.1.8 MatCapMaterial\_v3



**Full Name:** Ops.Gl.Shader.MatCapMaterial\_v3

**Description:** Easy to use image based lighting Material

**> Input Ports:**

- **Render** (Trigger)
- **MatCap** (Object:Texture)
- **Diffuse** (Object:Texture)
- **Normal** (Object:Texture)
- **Specular Mask** (Object:Texture)
- **Specular MatCap** (Object:Texture)
- **AO Texture** (Object:Texture)
- **Opacity Texture** (Object:Texture)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **Opacity** (Number)
- **AO Intensity** (Number)
- **Normal Map Intensity** (Number)
- **Repeat X** (Number)
- **Repeat Y** (Number)
- **Offset X** (Number)
- **Offset Y** (Number)

• **Double Sided** (Number: Boolean)

• **Screen Space Normals** (Number: Boolean)

• **check to use screen space normals** (flat shading)

• **Calc Normal Tangents** (Number: Boolean)

• **Opacity TexCoords Transform** (Number: Boolean)

• **Discard Transparent Pixels** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)
- **Shader** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Shader.MatCapMaterial\\_v3](https://cables.gl/op/Ops.Gl.Shader.MatCapMaterial_v3)

## 71.1.9 MinifyGlsl



**Full Name:** Ops.Gl.Shader.MinifyGlsl

**Description:** Minify GLSL shader source code

**> Input Ports:**

- **Shader Source** (String)

**< Output Ports:**

- **Minified Shader Source** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Shader.MinifyGlsl>

## 71.1.10 PointMaterial\_v6



**Full Name:** Ops.Gl.Shader.PointMaterial\_v6

**Description:** Draw all vertices as points / circles

**> Input Ports:**

- **Render** (Trigger)
- **PointSize** (Number)
- **Size In Pixels** (Number: Boolean)
- **Random Size** (Number)
- **Round** (Number: Boolean)
- **Round Antialias** (Number: Boolean)
- **Scale By Distance** (Number: Boolean)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)
- **Vertex Colors** (Number: Boolean)
- **Texture** (Object:Texture)
- **Colorize Texture** (Number: Boolean)
- **Texture Mask** (Object:Texture)
- **Texture Colorize** (Object:Texture)
- **Colorize Randomize** (Number: Boolean)
- **Texture Opacity** (Object:Texture)
- **Texture Point Size** (Object:Texture)
- **Texture Point Size Mul** (Number)
- **Flip Texture** (Number: Boolean)
- **Atlas Cross Fade** (Number: Boolean)
- **Atlas Repeat X** (Number)
- **Atlas Lookup** (Object:Texture)
- **Rotate Texture** (Object:Texture)
- **Min Point Size** (Number)

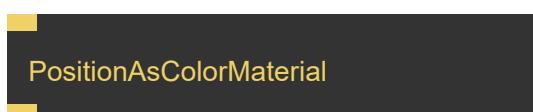
**< Output Ports:**

- **Trigger** (Trigger)
- **Shader** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Shader.PointMaterial\\_v6](https://cables.gl/op/Ops.Gl.Shader.PointMaterial_v6)

## 71.1.11 PositionAsColorMaterial



**Full Name:** Ops.Gl.Shader.PositionAsColorMaterial  
**Description:** draw meshes using XYZ position coordinates as RGB color  
**> Input Ports:**

- **Render** (Trigger)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor  
**Docs:** <https://cables.gl/op/Ops.Gl.Shader.PositionAsColorMaterial>

## 71.1.12 SetShader



**Full Name:** Ops.Gl.Shader.SetShader  
**Description:** Reuse another shader at different points in the patch.  
**> Input Ports:**

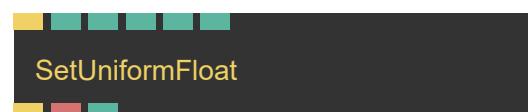
- **Render** (Trigger)
- **Shader** (Object)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor  
**Docs:** <https://cables.gl/op/Ops.Gl.Shader.SetShader>

## 71.1.13 SetUniformFloat\_v2



**Full Name:** Ops.Gl.Shader.SetUniformFloat\_v2  
**Description:** set a uniform value of the current shader  
**> Input Ports:**

- **Render** (Trigger)

- **Uniform Index** (Number: Integer)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **W** (Number)

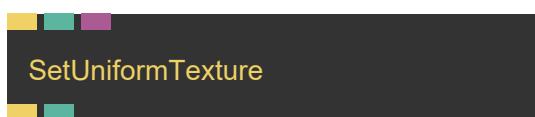
**< Output Ports:**

- **Next** (Trigger)
- **Type** (String)
- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.GlShader.SetUniformFloat\\_v2](https://cables.gl/op/Ops.GlShader.SetUniformFloat_v2)

## 71.1.14 SetUniformTexture\_v2



**Full Name:** Ops.Gl.Shader.SetUniformTexture\_v2

**Description:** set a uniform value of the current shader

**> Input Ports:**

- **Render** (Trigger)
- **Uniform Index** (Number: Integer)
- **Texture** (Object:Texture)

**< Output Ports:**

- **Next** (Trigger)
- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.GlShader.SetUniformTexture\\_v2](https://cables.gl/op/Ops.GlShader.SetUniformTexture_v2)

## 71.1.15 ShaderDefine



**Full Name:** Ops.Gl.Shader.ShaderDefine

**Description:** Set shader defines

**> Input Ports:**

- **Shader** (Object)
- **Name** (String)
- **Value** (String)
- **Active** (Number: Boolean)
- **Public** (4): 1

**< Output Ports:**

• Visit *Ops.GlShader.ShaderDefine* documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.GlShader.ShaderDefine>

## 71.1.16 ShaderInfo



**Full Name:** Ops.Gl.Shader.ShaderInfo

**Description:** view current shader source code

**> Input Ports:**

- **Exec** (Trigger)
- **Show Fragment** (Trigger)
- **Show Vertex** (Trigger)
- **Show Modules** (Trigger)
- **Show Uniforms** (Trigger)
- **State Info** (Trigger)

**< Output Ports:**

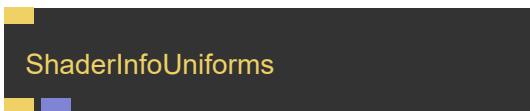
- **Next** (Trigger)
- **Source Frag** (String)
- **Source Vert** (String)
- **Name** (String)
- **Id** (String)
- **NeedsBarycentric** (booleanNumber)
- **Num Uniforms** (Number)
- **Num Attributes** (Number)

- **Attributes Names** (Array)
- **Num Defines** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Shader.ShaderInfo>

## 71.1.17 ShaderInfoUniforms\_v2



**Full Name:** Ops.Gl.Shader.ShaderInfoUniforms\_v2

**Description:** read back all uniforms values of the current bound shader

**> Input Ports:**

- **Exec** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **Uniforms** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Shader.ShaderInfoUniforms\\_v2](https://cables.gl/op/Ops.Gl.Shader.ShaderInfoUniforms_v2)

## 71.1.18 ShaderToTexture\_v2



**Full Name:** Ops.Gl.Shader.ShaderToTexture\_v2

**Description:** render a shader into a texture

**> Input Ports:**

- **Render** (Trigger)
- **Shader** (Object:Shader)
- **Width** (Number: Integer)
- **Height** (Number: Integer)
- **Filter Index** (Number: Integer)
- **Wrap Index** (Number: Integer)

- **Pixel Format Index** (Number: Integer)

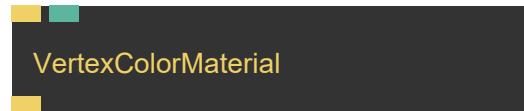
**< Output Ports:**

- **Next** (Trigger)
- **Texture** (Object)
- **Texture 2** (Object)
- **Texture 3** (Object)
- **Texture 4** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Shader.ShaderToTexture\\_v2](https://cables.gl/op/Ops.Gl.Shader.ShaderToTexture_v2)

## 71.1.19 VertexColorMaterial



**Full Name:** Ops.Gl.Shader.VertexColorMaterial

**Description:** Draw a mesh, showing only its vertex colors

**> Input Ports:**

- **Render** (Trigger)
- **Opacity** (Number)

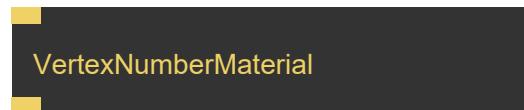
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Shader.VertexColorMaterial>

## 71.1.20 VertexNumberMaterial



**Full Name:** Ops.Gl.Shader.VertexNumberMaterial

**Description:** visually debug vertices of your 3D geometry

**> Input Ports:**

- **Render** (Trigger)

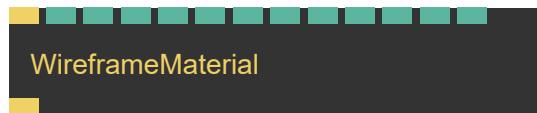
< Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.GlShader.VertexNumberMaterial>

### 71.1.21 WireframeMaterial\_v2



**Full Name:** Ops.Gl.Shader.WireframeMaterial\_v2

**Description:** Renders following meshes as wireframes

> Input Ports:

- **Render** (Trigger)
- **Enable Depth Testing** (Number: Boolean)
- **Width** (Number)
- **AntiAlias** (Number)
- **Diffuse R** (Number)
- **Diffuse G** (Number)
- **Diffuse B** (Number)
- **Diffuse A** (Number)
- **Fill** (Number: Boolean)
- **Fill R** (Number)
- **Fill G** (Number)
- **Fill B** (Number)
- **Fill A** (Number)

< Output Ports:

- **Trigger** (Trigger)

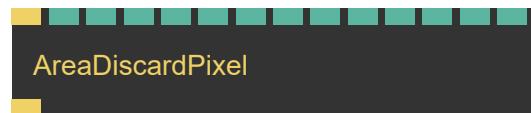
**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.GlShader.WireframeMaterial\\_v2](https://cables.gl/op/Ops.GlShader.WireframeMaterial_v2)

## 72 Ops.Gl.ShaderEffects

### 72.1 Ops.Gl.ShaderEffects

#### 72.1.1 AreaDiscardPixel\_v2



**Full Name:** Ops.Gl.ShaderEffects.AreaDiscardPixel\_v2

**Description:** do not draw pixels inside a defined 3d area

> Input Ports:

- **Render** (Trigger)
- **Invert** (Number: Boolean)
- **Area Index** (Number: Integer)
- **Area** (Number: String)
- **Size** (Number)
- **Size X** (Number)
- **Size Y** (Number)
- **Size Z** (Number)
- **Repeat** (Number: Boolean)
- **Repeat Distance** (Number)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **WorldSpace** (Number: Boolean)

< Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.GlShaderEffects.AreaDiscardPixel\\_v2](https://cables.gl/op/Ops.GlShaderEffects.AreaDiscardPixel_v2)

## 72.1.2 AreaRotate\_v2



**Full Name:** Ops.Gl.ShaderEffects.AreaRotate\_v2

**Description:** rotate vertices in an area around a center point

### > Input Ports:

- **Render** (Trigger)
- **Size** (Number)
- **Strength** (Number)
- **Smooth** (Number: Boolean)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.AreaRotate\\_v2](https://cables.gl/op/Ops.Gl.ShaderEffects.AreaRotate_v2)

## 72.1.3 AreaScaler\_v3



**Full Name:** Ops.Gl.ShaderEffects.AreaScaler\_v3

**Description:** Scales the size of meshes within the area of influence

### > Input Ports:

- **Render** (Trigger)
- **Area Size** (Number)
- **Source Index** (Number: Integer)
- **Strength** (Number)
- **Smoothstep** (Number: Boolean)
- **Min Size Original** (Number: Boolean)
- **Clamp Size** (Number: Boolean)

- **Clamp Min** (Number)
- **Clamp Max** (Number)
- **Pos X** (Number)
- **Pos Y** (Number)
- **Pos Z** (Number)

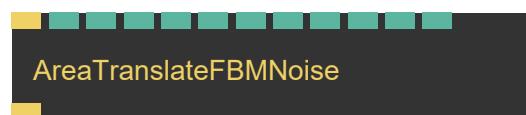
### < Output Ports:

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.AreaScaler\\_v3](https://cables.gl/op/Ops.Gl.ShaderEffects.AreaScaler_v3)

## 72.1.4 AreaTranslateFBMNoise



**Full Name:** Ops.Gl.ShaderEffects.AreaTranslateFBMNoise

**Description:** Area size of noise

### > Input Ports:

- **Render** (Trigger)
- **Mode Index** (Number: Integer)
- **Size** (Number)
- **Strength** (Number)
- **Smooth** (Number: Boolean)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Noise Scale** (Number)
- **Noise X** (Number)
- **Noise Y** (Number)
- **Noise Z** (Number)

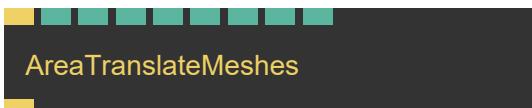
### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ShaderEffects.AreaTranslateFBMNoise>

## 72.1.5 AreaTranslateMeshes\_v3



AreaTranslateMeshes

**Full Name:** Ops.Gl.ShaderEffects.AreaTranslateMeshes\_v3

**Description:** Change the position of all meshes inside of the area of influence

### > Input Ports:

- **Render** (Trigger)
- **Size** (Number)
- **Strength** (Number)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Multiply X** (Number)
- **Multiply Y** (Number)
- **Multiply Z** (Number)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.AreaTranslateMeshes\\_v3](https://cables.gl/op/Ops.Gl.ShaderEffects.AreaTranslateMeshes_v3)

## 72.1.6 Bend\_v2



Bend

**Full Name:** Ops.Gl.ShaderEffects.Bend\_v2

**Description:** bend objects along an axis

### > Input Ports:

- **Render** (Trigger)
- **Amount** (Number)
- **RotX** (Number)
- **RotY** (Number)
- **RotZ** (Number)

- **Scale** (Number)
- **Offset** (Number)
- **Limited** (Number: Boolean)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.Bend\\_v2](https://cables.gl/op/Ops.Gl.ShaderEffects.Bend_v2)

## 72.1.7 ClampVertexPosition\_v2



ClampVertexPosition

**Full Name:** Ops.Gl.ShaderEffects.ClampVertexPosition\_v2

**Description:** clamp/restrict the vertex position to min/max values per axis

### > Input Ports:

- **Render** (Trigger)
- **Axis Index** (Number: Integer)
- **Min** (Number)
- **Max** (Number)
- **Update Normals** (Number: Boolean)

### < Output Ports:

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.ClampVertexPosition\\_v2](https://cables.gl/op/Ops.Gl.ShaderEffects.ClampVertexPosition_v2)

## 72.1.8 ColorArea\_v5



ColorArea

**Full Name:** Ops.Gl.ShaderEffects.ColorArea\_v5

**Description:** Colorize all meshes around current position

### > Input Ports:

- **Render** (Trigger)
- **Area Index** (Number: Integer)
- **Size** (Number)
- **Roundness** (Number)
- **Amount** (Number)
- **Falloff** (Number)
- **Invert** (Number: Boolean)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Change Size** (Number: Boolean)
- **Size X** (Number)
- **Size Y** (Number)
- **Size Z** (Number)
- **Texture** (Object:Texture)
- **Priority** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.ColorArea\\_v5](https://cables.gl/op/Ops.Gl.ShaderEffects.ColorArea_v5)

## 72.1.9 DeformArea



**Full Name:** Ops.Gl.ShaderEffects.DeformArea

**Description:** deform a spherical area of a mesh

**> Input Ports:**

- **Render** (Trigger)
- **Size** (Number)
- **Strength** (Number)
- **Smooth** (Number: Boolean)
- **WorldSpace** (Number: Boolean)

- **X** (Number)
- **Y** (Number)
- **Z** (Number)

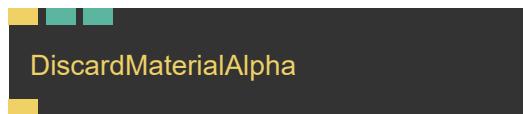
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ShaderEffects.DeformArea>

## 72.1.10 DiscardMaterialAlpha



**Full Name:** Ops.Gl.ShaderEffects.DiscardMaterialAlpha

**Description:** discard transparent pixels in material textures

**> Input Ports:**

- **Render** (Trigger)
- **Method Index** (Number: Integer)
- **Threshold** (Number)

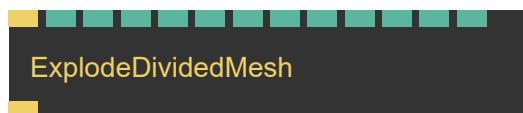
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ShaderEffects.DiscardMaterialAlpha>

## 72.1.11 ExplodeDividedMesh\_v2



**Full Name:** Ops.Gl.ShaderEffects.ExplodeDividedMesh\_v2

**Description:** explode a (divided) mesh in the direction of faces normals

**> Input Ports:**

- **Render** (Trigger)
- **Distance** (Number)

- **Size** (Number)
- **Absolute** (Number: Boolean)
- **Add X** (Number)
- **Add Y** (Number)
- **Add Z** (Number)
- **Mul X** (Number)
- **Mul Y** (Number)
- **Mul Z** (Number)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.ExplodeDividedMesh\\_v2](https://cables.gl/op/Ops.Gl.ShaderEffects.ExplodeDividedMesh_v2)

## 72.1.12 FogEffect



**Full Name:** Ops.Gl.ShaderEffects.FogEffect

**Description:** Fog as a shadereffect applied to a material

**> Input Ports:**

- **Render** (Trigger)
- **Mode Index** (Number: Integer)
- **Start** (Number)
- **End** (Number)
- **Amount** (Number)
- **R** (Number)
- **G** (Number)
- **B** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ShaderEffects.FogEffect>

## 72.1.13 FresnelGlow



**Full Name:** Ops.Gl.ShaderEffects.FresnelGlow

**Description:** add fresnel glow to any material

**> Input Ports:**

- **Trigger In** (Trigger)
- **Active** (Number: Boolean)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **Fresnel Intensity** (Number)
- **Fresnel Exponent** (Number)

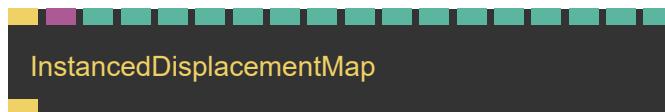
**< Output Ports:**

- **Trigger Out** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ShaderEffects.FresnelGlow>

## 72.1.14 InstancedDisplacementMap\_v2



**Full Name:** Ops.Gl.ShaderEffects.InstancedDisplacementMap\_v2

**Description:** displace positions of instanced meshes using a texture

**> Input Ports:**

- **Trigger** (Trigger)
- **Texture** (Object:Texture)
- **Source Index** (Number: Integer)
- **Mode Index** (Number: Integer)
- **Strength** (Number)
- **Min** (Number)
- **Scale** (Number)

- **Clamp** (Number: Boolean)
- **Colorize** (Number: Boolean)
- **Debug Bounds** (Number: Boolean)
- **Normalize** (Number: Boolean)
- **Offset X** (Number)
- **Offset Y** (Number)
- **Abs** (Number: Boolean)
- **Channel Index** (Number: Integer)
- **X** (Number: Boolean)
- **Y** (Number: Boolean)
- **Z** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.InstancedDisplacementMap\\_v2](https://cables.gl/op/Ops.Gl.ShaderEffects.InstancedDisplacementMap_v2)

## 72.1.15 InstancedPerlinPosition\_v2



**Full Name:** Ops.Gl.ShaderEffects.InstancedPerlinPosition\_v2

**Description:** displace position of instanced object by perlin noise value

**> Input Ports:**

- **Render** (Trigger)
- **Strength** (Number)
- **Scroll X** (Number)
- **Scroll Y** (Number)
- **Scroll Z** (Number)
- **Scale** (Number)
- **Method Index** (Number: Integer)
- **Method** (String)
- **Mul X** (Number)
- **Mul Y** (Number)
- **Mul Z** (Number)
- **Min Scale** (Number)

- **WorldSpace** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.InstancedPerlinPosition\\_v2](https://cables.gl/op/Ops.Gl.ShaderEffects.InstancedPerlinPosition_v2)

## 72.1.16 InstancedTextureColorize



**Full Name:** Ops.Gl.ShaderEffects.InstancedTextureColorize

**Description:** colorize instanced meshes using a texture

**> Input Ports:**

- **Trigger** (Trigger)
- **Texture** (Object:Texture)
- **Strength** (Number)
- **Scale** (Number)
- **Clamp** (Number: Boolean)
- **Debug Bounds** (Number: Boolean)
- **Offset X** (Number)
- **Offset Y** (Number)
- **Method Index** (Number: Integer)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ShaderEffects.InstancedTextureColorize>

## 72.1.17 LimitMeshByTexCoord





- **Flip Normals** (Number: Boolean)
- **Falloff** (Number)
- **Output Index** (Number: Integer)
- **Source Index** (Number: Integer)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Scroll X** (Number)
- **Scroll Y** (Number)
- **Scroll Z** (Number)
- **WorldSpace** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.PerlinAreaDeform\\_v4](https://cables.gl/op/Ops.Gl.ShaderEffects.PerlinAreaDeform_v4)

## 72.1.21 ScaleByNormal\_v2



**Full Name:** Ops.Gl.ShaderEffects.ScaleByNormal\_v2

**Description:** Scale vertices of an object in the direction of face normals

**> Input Ports:**

- **Render** (Trigger)
- **Strength** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.ScaleByNormal\\_v2](https://cables.gl/op/Ops.Gl.ShaderEffects.ScaleByNormal_v2)

## 72.1.22 Shadow\_v3



**Full Name:** Ops.Gl.ShaderEffects.Shadow\_v3

**Description:** add shadow capabilities to any material

**> Input Ports:**

- **Trigger In** (Trigger)
- **Cast Shadow** (Number: Boolean)
- **Receive Shadow** (Number: Boolean)
- **Sample Distribution** (Number: Integer)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **Discard Transparent** (Number: Boolean)
- **Opacity Threshold** (Number)
- **Opacity Texture** (Object:Texture)
- **Cull Backfacing** (Number: Boolean)

**< Output Ports:**

- **Trigger Out** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.Shadow\\_v3](https://cables.gl/op/Ops.Gl.ShaderEffects.Shadow_v3)

## 72.1.23 SplineDeform\_v2



**Full Name:** Ops.Gl.ShaderEffects.SplineDeform\_v2

**Description:** Deform a mesh along a spline

**> Input Ports:**

- **Render** (Trigger)
- **Size** (Number)
- **Offset** (Number)

- Points (Array)

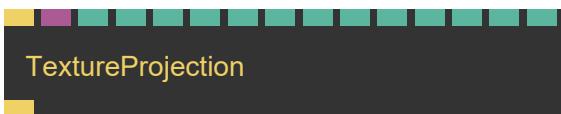
**< Output Ports:**

- Next (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.SplineDeform\\_v2](https://cables.gl/op/Ops.Gl.ShaderEffects.SplineDeform_v2)

## 72.1.24 TextureProjection\_v2



TextureProjection

**Full Name:** Ops.Gl.ShaderEffects.TextureProjection\_v2

**Description:** texture projection on meshes

**> Input Ports:**

- Render (Trigger)
- Texture (Object:Texture)
- BlendMode Index (Number: Integer)
- Amount (Number)
- Scale (Number)
- Use Texture Alpha (Number: Boolean)
- Pos X (Number)
- Pos Y (Number)
- Rot X (Number)
- Rot Y (Number)
- Rot Z (Number)
- Mapping Index (Number: Integer)
- Discard (Number: Boolean)
- WorldSpace (Number: Boolean)

**< Output Ports:**

- Trigger (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.TextureProjection\\_v2](https://cables.gl/op/Ops.Gl.ShaderEffects.TextureProjection_v2)

## 72.1.25 TransformTextureCoordinates



TransformTextureCoordinates

**Full Name:** Ops.Gl.ShaderEffects.TransformTextureCoordinates

**Description:** Transform and repeat texture coordinates of a mesh via vertex shader

**> Input Ports:**

- Render (Trigger)
- Translate X (Number)
- Translate Y (Number)
- Repeat X (Number)
- Repeat Y (Number)
- Flip X (Number: Boolean)
- Flip Y (Number: Boolean)
- Rotation (Number)

**< Output Ports:**

- Trigger (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ShaderEffects.TransformTextureCoordinates>

## 72.1.26 TransformVertex



TransformVertex

**Full Name:** Ops.Gl.ShaderEffects.TransformVertex

**Description:** transform vertices of a mesh via vertex shader

**> Input Ports:**

- Render (Trigger)
- Translate X (Number)
- Translate Y (Number)
- Translate Z (Number)
- Scale X (Number)

- **Scale Y** (Number)
- **Scale Z** (Number)
- **Rotation X** (Number)
- **Rotation Y** (Number)
- **Rotation Z** (Number)
- **Transform Normals** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ShaderEffects.TransformVertex>

## 72.1.27 Twist\_v3



**Full Name:** Ops.Gl.ShaderEffects.Twist\_v3

**Description:** twist a mesh around an axis

**> Input Ports:**

- **Render** (Trigger)
- **Degree** (Number)
- **Height** (Number)
- **Axis Index** (Number: Integer)
- **Axis** (Number: String)

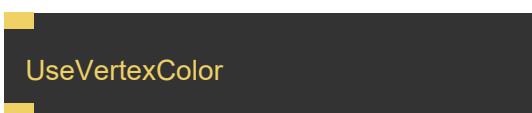
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.Twist\\_v3](https://cables.gl/op/Ops.Gl.ShaderEffects.Twist_v3)

## 72.1.28 UseVertexColor



**Full Name:** Ops.Gl.ShaderEffects.UseVertexColor

**Description:** Use vertex color as basecolor/diffuse color

**> Input Ports:**

- **Render** (Trigger)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ShaderEffects.UseVertexColor>

## 72.1.29 VertexArea



**Full Name:** Ops.Gl.ShaderEffects.VertexArea

**Description:** transform an area of a mesh

**> Input Ports:**

- **Render** (Trigger)
- **Area Index** (Number: Integer)
- **Visualize Area** (Number: Boolean)
- **WorldSpace** (Number: Boolean)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Radius** (Number)
- **Area Size X** (Number)
- **Area Size Y** (Number)
- **Area Size Z** (Number)
- **Translate X** (Number)
- **Translate Y** (Number)
- **Translate Z** (Number)
- **Scale X** (Number)
- **Scale Y** (Number)
- **Scale Z** (Number)

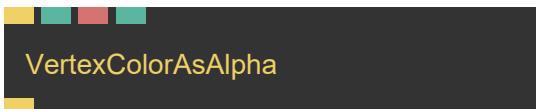
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ShaderEffects.VertexArea>

### 72.1.30 VertexColorAsAlpha



**Full Name:** Ops.Gl.ShaderEffects.VertexColorAsAlpha

**Description:** Use mesh vertexcolor as Alpha/Opacity

**> Input Ports:**

- **Render** (Trigger)
- **Input Index** (Number: Integer)
- **Invert** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.ShaderEffects.VertexColorAsAlpha>

### 72.1.31 VertexDisplacementMap\_v5



**Full Name:** Ops.Gl.ShaderEffects.VertexDisplacementMap\_v5

**Description:** Displace the vertices of a mesh with the pixels brightness values from a texture

**> Input Ports:**

- **Render** (Trigger)
- **Extrude** (Number)
- **Texture** (Object:Texture)
- **Offset X** (Number)
- **Offset Y** (Number)
- **Scale** (Number)

- **Calc Normals** (Number: Boolean)

- **Discard Zero Values** (Number: Boolean)

- **Colorize** (Number: Boolean)

- **Colorize Min** (Number)

- **Colorize Max** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.VertexDisplacementMap\\_v5](https://cables.gl/op/Ops.Gl.ShaderEffects.VertexDisplacementMap_v5)

### 72.1.32 VertexNumberLimit\_v2



**Full Name:** Ops.Gl.ShaderEffects.VertexNumberLimit\_v2

**Description:** only draw the first X vertices of a mesh

**> Input Ports:**

- **Render** (Trigger)
- **Min** (Number: Integer)
- **Max** (Number: Integer)
- **Invert** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.VertexNumberLimit\\_v2](https://cables.gl/op/Ops.Gl.ShaderEffects.VertexNumberLimit_v2)

### 72.1.33 VertexPositionFromTexture\_v2



**Full Name:** Ops.Gl.ShaderEffects.VertexPositionFromTexture\_v2

**Description:** set vertex positions of a mesh from a texture

**> Input Ports:**

- **Render** (Trigger)
- **Texture** (Object:Texture)
- **Mode Index** (Number: Integer)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.VertexPositionFromTexture\\_v2](https://cables.gl/op/Ops.Gl.ShaderEffects.VertexPositionFromTexture_v2)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.ShaderEffects.VertexWobble\\_v2](https://cables.gl/op/Ops.Gl.ShaderEffects.VertexWobble_v2)

## 72.1.34 VertexWobble\_v2



**Full Name:** Ops.Gl.ShaderEffects.VertexWobble\_v2

**Description:** sine wave vertex displacement

**> Input Ports:**

- **Render** (Trigger)
- **Source Index** (Number: Integer)
- **Amount** (Number)
- **Time** (Number)
- **Scale** (Number)
- **AxisX** (Number: Boolean)
- **AxisY** (Number: Boolean)
- **AxisZ** (Number: Boolean)
- **Area Index** (Number: Integer)
- **Size** (Number)
- **Falloff** (Number)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **WorldSpace** (Number: Boolean)
- **Invert** (Number: Boolean)

# 73 Ops.Gl.Textures

## 73.1 Ops.Gl.Textures

### 73.1.1 Base64ToTexture



**Full Name:** Ops.Gl.Textures.Base64ToTexture

**Description:** Converts a base-64 image string into a texture

#### > Input Ports:

- **Wrap Index** (Number: Integer)
- **Pre Multiplied Alpha** (Number: Boolean)

#### < Output Ports:

- **Texture** (Object)
- **Has Error** (booleanNumber)
- **Loading** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Textures.Base64ToTexture>

### 73.1.2 ColorTexture



**Full Name:** Ops.Gl.Textures.ColorTexture

**Description:** Simple texture filled with one color

#### > Input Ports:

- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)

#### < Output Ports:

- **Texture\_out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Textures.ColorTexture>

### 73.1.3 CombineTextures



**Full Name:** Ops.Gl.Textures.CombineTextures

**Description:** combine multiple textures into one by copying colorchannels

#### > Input Ports:

- **Execute** (Trigger)
- **Filter Index** (Number: Integer)
- **Wrap Index** (Number: Integer)
- **PixelFormat Index** (Number: Integer)
- **Size Index** (Number: Integer)
- **R** (Object:Texture)
- **R Source Index** (Number: Integer)
- **R Value Index** (Number: Integer)
- **R Default** (Number)
- **G** (Object:Texture)
- **G Source Index** (Number: Integer)
- **G Value Index** (Number: Integer)
- **G Default** (Number)
- **B** (Object:Texture)
- **B Source Index** (Number: Integer)
- **B Value Index** (Number: Integer)
- **B Default** (Number)
- **A** (Object:Texture)
- **A Source Index** (Number: Integer)
- **A Value Index** (Number: Integer)
- **A Default** (Number)

#### < Output Ports:

- **Next** (Trigger)
- **Texture** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Textures.CombineTextures>

### 73.1.4 CopyTexture\_v3



**Full Name:** Ops.Gl.Textures.CopyTexture\_v3

**Description:** copy a texture and optionally resize it

**> Input Ports:**

- **Render** (Trigger)
- **Texture** (Object:Texture)
- **Alpha Mask** (Object:Texture)
- **Use Original Size** (Number: Boolean)
- **Width** (Number: Integer)
- **Height** (Number: Integer)
- **Pixel Format Index** (Number: Integer)
- **Wrap Index** (Number: Integer)
- **Invert A** (Number: Boolean)
- **Flip X** (Number: Boolean)
- **Flip Y** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)
- **Texture\_out** (Object)
- **Aspect Ratio** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Textures.CopyTexture\\_v3](https://cables.gl/op/Ops.Gl.Textures.CopyTexture_v3)

### 73.1.5 EmptyTexture



**Full Name:** Ops.Gl.Textures.EmptyTexture

**Description:** A very simple empty transparent texture with an opacity of 0

**> Input Ports:**

- **Width** (Number)
- **Height** (Number)

**< Output Ports:**

- **Texture** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Textures.EmptyTexture>

### 73.1.6 ExrTexture



**Full Name:** Ops.Gl.Textures.ExrTexture

**Description:** load .exr floating point texture files

**> Input Ports:**

- **EXR File** (String)
- **Remove Alpha** (Number: Boolean)
- **Flip** (Number: Boolean)

**< Output Ports:**

- **Texture** (Object)
- **Width** (Number)
- **Height** (Number)
- **Channels** (String)
- **Loading** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Textures.ExrTexture>

### 73.1.7 GraphTexture



**Full Name:** Ops.Gl.Textures.GraphTexture

**Description:** draw a graph of a value into a texture

**> Input Ports:**

- **Trigger** (Trigger)
- **Value** (Number)
- **Index** (Number: Integer)
- **Reset** (Trigger)
- **Color Random Seed** (Number)
- **Texture Width** (Number: Integer)
- **Texture Height** (Number: Integer)

**< Output Ports:**

- **Texture** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Textures.GraphTexture>

### 73.1.8 Histogram



**Full Name:** Ops.Gl.Textures.Histogram

**Description:** graphical representation of distribution of color in a texture

**> Input Ports:**

- **Trigger** (Trigger)
- **Texture** (Object:Texture)

**< Output Ports:**

- **Histogram Texture** (Object)
- **Histogram Data** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Textures.Histogram>

### 73.1.9 MontageTextures\_v2



**Full Name:** Ops.Gl.Textures.MontageTextures\_v2

**Description:** combine multiple textures into one by copying colorchannels

**> Input Ports:**

- **Execute** (Trigger)
- **Flip Order** (Number: Boolean)
- **Width** (Number: Integer)
- **Height** (Number: Integer)
- **Wrap Index** (Number: Integer)
- **PixelFormat Index** (Number: Integer)
- **Texture 0** (Object:Texture)
- **Texture 1** (Object:Texture)
- **Texture 2** (Object:Texture)
- **Texture 3** (Object:Texture)
- **Texture 4** (Object:Texture)
- **Texture 5** (Object:Texture)
- **Texture 6** (Object:Texture)
- **Texture 7** (Object:Texture)
- **Texture 8** (Object:Texture)
- **Texture 9** (Object:Texture)
- **Texture 10** (Object:Texture)
- **Texture 11** (Object:Texture)
- **Texture 12** (Object:Texture)
- **Texture 13** (Object:Texture)
- **Texture 14** (Object:Texture)
- **Texture 15** (Object:Texture)

**< Output Ports:**

- **Next** (Trigger)
- **Texture** (Object)
- **Columns** (Number)
- **Rows** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Textures.MontageTextures\\_v2](https://cables.gl/op/Ops.Gl.Textures.MontageTextures_v2)

### 73.1.10 NoiseTexture



**Full Name:** Ops.Gl.Textures.NoiseTexture

**Description:** Simple noisetexture

**> Input Ports:**

- **Width** (Number: Integer)
- **Height** (Number: Integer)
- **Wrap Index** (Number: Integer)
- **Color** (Number: Boolean)
- **PixelFormat Index** (Number: Integer)
- **Integer** (Number: Boolean)
- **Seed** (Number)
- **Channel R** (Number: Boolean)
- **Min R** (Number)
- **Max R** (Number)
- **Channel G** (Number: Boolean)
- **Min G** (Number)
- **Max G** (Number)
- **Channel B** (Number: Boolean)
- **Min B** (Number)
- **Max B** (Number)
- **Channel A** (Number: Boolean)
- **Min A** (Number)
- **Max A** (Number)

**< Output Ports:**

- **Texture** (Object)
- **Total Pixel** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Textures.NoiseTexture>

### 73.1.11 PaletteTexture



**Full Name:** Ops.Gl.Textures.PaletteTexture

**Description:** Create a RGB color palette using an array

**> Input Ports:**

- **Palette Array** (Array)
- **Smooth** (Number: Boolean)

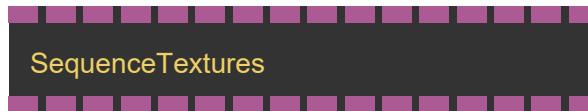
**< Output Ports:**

- **Color Array** (Array)
- **Texture** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Textures.PaletteTexture>

### 73.1.12 SequenceTextures



**Full Name:** Ops.Gl.Textures.SequenceTextures

**Description:** control order and flow of objects

**> Input Ports:**

- **Texture 0** (Object:Texture)
- **Texture 1** (Object:Texture)
- **Texture 2** (Object:Texture)
- **Texture 3** (Object:Texture)
- **Texture 4** (Object:Texture)
- **Texture 5** (Object:Texture)
- **Texture 6** (Object:Texture)
- **Texture 7** (Object:Texture)
- **Texture 8** (Object:Texture)
- **Texture 9** (Object:Texture)
- **Texture 10** (Object:Texture)

- **Texture 11** (Object:Texture)
- **Texture 12** (Object:Texture)
- **Texture 13** (Object:Texture)
- **Texture 14** (Object:Texture)
- **Texture 15** (Object:Texture)

**< Output Ports:**

- **Output 0** (Object)
- **Output 1** (Object)
- **Output 2** (Object)
- **Output 3** (Object)
- **Output 4** (Object)
- **Output 5** (Object)
- **Output 6** (Object)
- **Output 7** (Object)
- **Output 8** (Object)
- **Output 9** (Object)
- **Output 10** (Object)
- **Output 11** (Object)
- **Output 12** (Object)
- **Output 13** (Object)
- **Output 14** (Object)
- **Output 15** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Textures.SequenceTextures>

### 73.1.13 SSAO



**Full Name:** Ops.Gl.Textures.SSAO

**Description:** screen space ambient occlusion from depth texture

**> Input Ports:**

- **Execute** (Trigger)
- **Depth Texture** (Object:Texture)
- **Radius** (Number)
- **Max Dist** (Number)

- **Begin** (Number)
- **End** (Number)
- **Strength** (Number)
- **Base** (Number)
- **Filter Index** (Number: Integer)

**< Output Ports:**

- **Next** (Trigger)
- **SSAO** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Textures.SSAO>

### 73.1.14 SwitchTextureMultiPort\_v2



**Full Name:** Ops.Gl.Textures.SwitchTextureMultiPort\_v2

**Description:** Switch between multiple textures

**> Input Ports:**

- **Index** (Number: Integer)
- **Textures\_0** (Object)
- **Add Port** (Object)

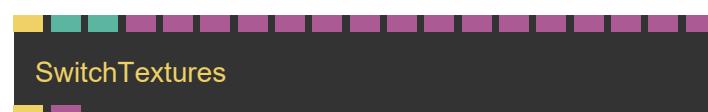
**< Output Ports:**

- **Texture** (Object)
- **Num Textures** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Textures.SwitchTextureMultiPort\\_v2](https://cables.gl/op/Ops.Gl.Textures.SwitchTextureMultiPort_v2)

### 73.1.15 SwitchTextures\_v2



**Full Name:** Ops.Gl.Textures.SwitchTextures\_v2

**Description:** Switch between different textures

**> Input Ports:**

- **Exec** (Trigger)
- **Num** (Number: Integer)
- **Default Texture Transparent** (Number: Boolean)
- **Texture0** (Object:Texture)
- **Texture1** (Object:Texture)
- **Texture2** (Object:Texture)
- **Texture3** (Object:Texture)
- **Texture4** (Object:Texture)
- **Texture5** (Object:Texture)
- **Texture6** (Object:Texture)
- **Texture7** (Object:Texture)
- **Texture8** (Object:Texture)
- **Texture9** (Object:Texture)
- **Texture10** (Object:Texture)
- **Texture11** (Object:Texture)
- **Texture12** (Object:Texture)
- **Texture13** (Object:Texture)
- **Texture14** (Object:Texture)
- **Texture15** (Object:Texture)

**< Output Ports:**

- **Next** (Trigger)
- **Texture** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Textures.SwitchTextures\\_v2](https://cables.gl/op/Ops.Gl.Textures.SwitchTextures_v2)

### 73.1.16 TextTexture\_v6



**Full Name:** Ops.Gl.Textures.TextTexture\_v6

**Description:** Generates a texture of Text using one of the font ops

**> Input Ports:**

- **Render** (Trigger)

- **Text** (String)
- **Draw Mesh** (Number: Boolean)
- **Scale Mesh** (Number)
- **Width** (Number: Integer)
- **Height** (Number: Integer)
- **Auto Height** (Number: Boolean)
- **Auto Line Breaks** (Number: Boolean)
- **Font** (String)
- **Weight** (String)
- **FontSize** (Number)
- **Letter Spacing** (Number)
- **Line Height Add** (Number)
- **Padding Y Top** (Number: Integer)
- **Padding Y Bottom** (Number: Integer)
- **Padding X** (Number: Integer)
- **Wrap Index** (Number: Integer)
- **Reuse Texture** (Number: Boolean)
- **Show Debug** (Number: Boolean)
- **Redraw On Font Load** (Number: Boolean)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **Opacity** (Number)
- **Background R** (Number)
- **Background G** (Number)
- **Background B** (Number)
- **Background A** (Number)
- **Force Redraw** (Trigger)

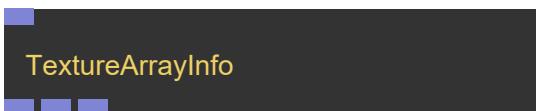
**< Output Ports:**

- **Next** (Trigger)
- **Ratio** (Number)
- **Texture** (Object)
- **Canvas** (Object)
- **Aspect** (Number)
- **Num Lines** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Textures.TextTexture\\_v6](https://cables.gl/op/Ops.Gl.Textures.TextTexture_v6)

### 73.1.17 TextureArrayInfo



**Full Name:** Ops.Gl.Textures.TextureArrayInfo

**Description:** Information about Textures in an array

**> Input Ports:**

- **Texture Array** (Array)

**< Output Ports:**

- **Names** (Array)
- **Widths** (Array)
- **Heights** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Textures.TextureArrayInfo>

- **Is Empty Default Texture** (booleanNumber)
- **Is Default Texture** (booleanNumber)
- **Is Cubemap** (booleanNumber)
- **Id** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Textures.TextureInfo\\_v2](https://cables.gl/op/Ops.Gl.Textures.TextureInfo_v2)

### 73.1.18 TextureInfo\_v2



**Full Name:** Ops.Gl.Textures.TextureInfo\_v2

**Description:** Outputs information about the connected texture

**> Input Ports:**

- **Texture** (Object:Texture)

**< Output Ports:**

- **Name** (String)
- **PixelFormat** (String)
- **Width** (Number)
- **Height** (Number)
- **Ratio** (Number)
- **Filter** (Number)
- **Wrap** (Number)
- **Flipped** (booleanNumber)
- **HDR** (booleanNumber)

### 73.1.19 TextureSVG\_v2



**Full Name:** Ops.Gl.Textures.TextureSVG\_v2

**Description:** Load a SVG image and convert to a texture of pixels

**> Input Ports:**

- **File** (String)
- **Texture Width** (Number: Integer)
- **Texture Height** (Number: Integer)
- **Wrap Index** (Number: Integer)
- **Filter Index** (Number: Integer)

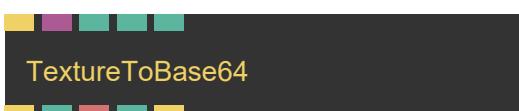
**< Output Ports:**

- **Texture** (Object)
- **Loaded** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Textures.TextureSVG\\_v2](https://cables.gl/op/Ops.Gl.Textures.TextureSVG_v2)

### 73.1.20 TextureToBase64\_v5



**Full Name:** Ops.Gl.Textures.TextureToBase64\_v5

**Description:** Converts a texture into a base-64 image string

**> Input Ports:**

- **Trigger** (Trigger)
- **Texture** (Object:Texture)
- **Quality** (Number)
- **Output DataUrl** (Number: Boolean)

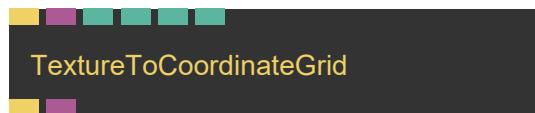
**< Output Ports:**

- **Next** (Trigger)
- **Binary Size** (Number)
- **Base64 String** (String)
- **Loading** (booleanNumber)
- **Finished** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Textures.TextureToBase64\\_v5](https://cables.gl/op/Ops.Gl.Textures.TextureToBase64_v5)

## 73.1.21 TextureToCoordinateGrid



**Full Name:** Ops.Gl.Textures.TextureToCoordinateGrid

**Description:** convert a texture to a 3d coordinate grid storing coordinates in texture RGB channels

**> Input Ports:**

- **Execute** (Trigger)
- **Texture** (Object:Texture)
- **Aspect** (Number)
- **Threshold** (Number)
- **Repeats** (Number: Integer)
- **Repeats Spacing** (Number)

**< Output Ports:**

- **Next** (Trigger)
- **HDR Texture** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Gl.Textures.TextureToCoordinateGrid>

## 73.1.22 VideoTexture\_v3



**Full Name:** Ops.Gl.Textures.VideoTexture\_v3

**Description:** Play a video file and use it as a texture

**> Input Ports:**

- **Update** (Trigger)
- **File** (String)
- **Play** (Number: Boolean)
- **Loop** (Number: Boolean)
- **Volume** (Number)
- **Mute** (Number: Boolean)
- **Update FPS** (Number)
- **Wrap Index** (Number: Integer)
- **Flip** (Number: Boolean)
- **Speed** (Number)
- **Set Time** (Number)
- **Rewind** (Trigger)
- **Preload** (Number: Boolean)
- **Show Interaction Needed Button** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)
- **Texture** (Object)
- **Duration** (Number)
- **Progress** (Number)
- **Interaction Needed** (booleanNumber)
- **CurrentTime** (Number)
- **Loading** (booleanNumber)
- **Playing** (booleanNumber)
- **Can Play Through** (booleanNumber)
- **Width** (Number)
- **Height** (Number)
- **Aspect Ratio** (Number)
- **Has Error** (booleanNumber)
- **Auto FPS** (booleanNumber)
- **Error Message** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Textures.VideoTexture\\_v3](https://cables.gl/op/Ops.Gl.Textures.VideoTexture_v3)

### 73.1.23 WebcamTexture\_v3



**Full Name:** Ops.Gl.Textures.WebcamTexture\_v3

**Description:** Use your webcam camera as a texture

**> Input Ports:**

- **Render** (Trigger)
- **Active** (Number: Boolean)
- **Generate Texture** (Number: Boolean)
- **Webcam Input Index** (Number: Integer)
- **Requested Width** (Number: Integer)
- **Requested Height** (Number: Integer)
- **Flip X** (Number: Boolean)
- **Flip Y** (Number: Boolean)
- **Show HTML Element** (Number: Boolean)
- **CSS** (String)
- **Element Flip X** (Number: Boolean)
- **Element Flip Y** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)
- **Texture** (Object)
- **Ratio** (Number)
- **Available** (booleanNumber)
- **Size Width** (Number)
- **Size Height** (Number)
- **Error** (String)
- **HTML Element** (Object)
- **Available Devices** (Array)
- **Active Device** (String)
- **Texture Updated** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Gl.Textures.WebcamTexture\\_v3](https://cables.gl/op/Ops.Gl.Textures.WebcamTexture_v3)

## 74 Ops.Graphics

### 74.1 Ops.Graphics

#### 74.1.1 ArrayToExr



**Full Name:** Ops.Graphics.ArrayToExr

**Description:** convert and download an array of numbers as an .exr image file

**> Input Ports:**

- **Array** (Array)
- **Width** (Number: Integer)
- **Height** (Number: Integer)
- **ZIP Compression** (Number: Boolean)
- **Filename** (String)
- **Download** (Trigger)

**< Output Ports:**

- Visit *Ops.Graphics.ArrayToExr documentation* for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.ArrayToExr>

#### 74.1.2 DepthTest



**Full Name:** Ops.Graphics.DepthTest

**Description:** change depth testing method (depthMask,depthWrite,depthFunc)

**> Input Ports:**

- **Render** (Trigger)

- **Enable Depth Testing** (Number: Boolean)
- **Depth Test Method Index** (Number: Integer)
- **Write To Depth Buffer** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.DepthTest>

### 74.1.3 GeometryMergeSimple



**Full Name:** Ops.Graphics.GeometryMergeSimple

**Description:** merge two geometries into one

**> Input Ports:**

- **Geometry** (Object)
- **Geometry 2** (Object)

**< Output Ports:**

- **Geometry Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.GeometryMergeSimple>

### 74.1.4 GetMaterialId



**Full Name:** Ops.Graphics.GetMaterialId

**Description:** get the id/index of the current set material

**> Input Ports:**

- **Update** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **Material Id** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.GetMaterialId>

### 74.1.5 GetObjectId



**Full Name:** Ops.Graphics.GetObjectId

**Description:** get the id/index of the current object/mesh

**> Input Ports:**

- **Update** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **Material Id** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.GetObjectId>

### 74.1.6 OrbitControls\_v3



**Full Name:** Ops.Graphics.OrbitControls\_v3

**Description:** rotate your object by clicking and dragging the mouse

**> Input Ports:**

- **Render** (Trigger)
- **Min Distance** (Number)
- **Max Distance** (Number)
- **Min Rot Y** (Number)
- **Max Rot Y** (Number)
- **Initial Radius** (Number)

- **Initial Axis Y** (Number)
- **Initial Axis X** (Number)
- **Smoothness** (Number)
- **Speed X** (Number)
- **Speed Y** (Number)
- **Active** (Number: Boolean)
- **Allow Panning** (Number: Boolean)
- **Allow Zooming** (Number: Boolean)
- **Allow Rotation** (Number: Boolean)
- **Restricted** (Number: Boolean)
- **Identity** (Number: Boolean)
- **Reset** (Trigger)

**< Output Ports:**

- **Trigger** (Trigger)
- **Radius** (Number)
- **Rot X** (Number)
- **Rot Y** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Graphics.OrbitControls\\_v3](https://cables.gl/op/Ops.Graphics.OrbitControls_v3)

## 74.1.7 Transform



**Full Name:** Ops.Graphics.Transform

**Description:** Transform objects in 3d space (rotate, translate, scale)

**> Input Ports:**

- **Render** (Trigger)
- **PosX** (Number)
- **PosY** (Number)
- **PosZ** (Number)
- **Scale** (Number)
- **RotX** (Number)
- **RotY** (Number)
- **RotZ** (Number)

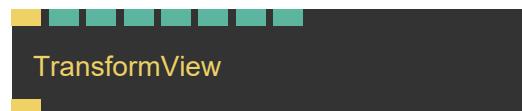
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Transform>

## 74.1.8 TransformView



**Full Name:** Ops.Graphics.TransformView

**Description:** the most simple camera op / transform the viewmatrix

**> Input Ports:**

- **Render** (Trigger)
- **PosX** (Number)
- **PosY** (Number)
- **PosZ** (Number)
- **Scale** (Number)
- **RotX** (Number)
- **RotY** (Number)
- **RotZ** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

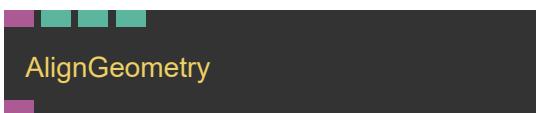
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.TransformView>

# 75 Ops.Graphics.Geometry

## 75.1 Ops.Graphics.Geometry

### 75.1.1 AlignGeometry



**Full Name:** Ops.Graphics.Geometry.AlignGeometry

**Description:** align a geometry / change its pivot / center / origin point

#### > Input Ports:

- **Geometry** (Object)
- **X Index** (Number: Integer)
- **Y Index** (Number: Integer)
- **Z Index** (Number: Integer)

#### < Output Ports:

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.AlignGeometry>

### 75.1.2 BoundingBox



**Full Name:** Ops.Graphics.Geometry.BoundingBox

**Description:** create a simple bounding box from width,height,depth

#### > Input Ports:

- **Width** (Number)
- **Height** (Number)
- **Depth** (Number)

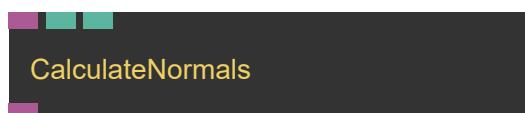
#### < Output Ports:

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.BoundingBox>

### 75.1.3 CalculateNormals



**Full Name:** Ops.Graphics.Geometry.CalculateNormals

**Description:** calculate normals of a geometry

#### > Input Ports:

- **Geometry** (Object)
- **Smooth** (Number: Boolean)
- **Force Z Up** (Number: Boolean)

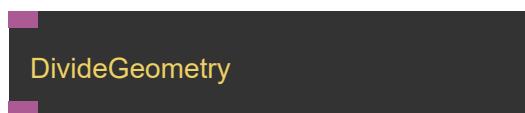
#### < Output Ports:

- **Geometry Out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.CalculateNormals>

### 75.1.4 DivideGeometry



**Full Name:** Ops.Graphics.Geometry.DivideGeometry

**Description:** disconnect faces/polylines of a mesh

#### > Input Ports:

- **Geometry** (Object)

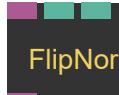
#### < Output Ports:

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.DivideGeometry>

## 75.1.5 FlipNormals



FlipNormals

**Full Name:** Ops.Graphics.Geometry.FlipNormals

**Description:** flip all normals of a geometry

**> Input Ports:**

- **Geometry** (Object)
- **Flip** (Number: Boolean)
- **Normalize** (Number: Boolean)

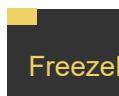
**< Output Ports:**

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.FlipNormals>

## 75.1.6 FreezeMeshes



FreezeMeshes

**Full Name:** Ops.Graphics.Geometry.FreezeMeshes

**Description:** capture all following meshes into one geometry

**> Input Ports:**

- **Capture** (Trigger)

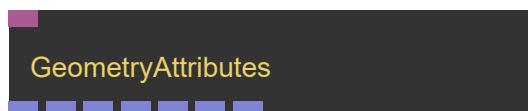
**< Output Ports:**

- **Geometry** (Object)
- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.FreezeMeshes>

## 75.1.7 GeometryAttributes



GeometryAttributes

**Full Name:** Ops.Graphics.Geometry.GeometryAttributes

**Description:** Get vertices of a geometry as array3x (vertex vertices)

**> Input Ports:**

- **Geometry** (Object)

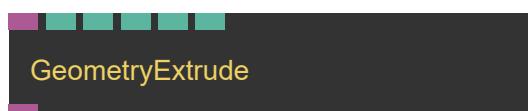
**< Output Ports:**

- **Faces** (Array)
- **Vertices** (Array)
- **Normals** (Array)
- **TexCoords** (Array)
- **Vertex Colors** (Array)
- **Tangents** (Array)
- **BiTangents** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.GeometryAttributes>

## 75.1.8 GeometryExtrude



GeometryExtrude

**Full Name:** Ops.Graphics.Geometry.GeometryExtrude

**Description:** basic extrusion of flat geometry

**> Input Ports:**

- **Geometry** (Object:Geometry)
- **Height** (Number)
- **Smooth** (Number: Boolean)
- **Walls** (Number: Boolean)
- **Top** (Number: Boolean)
- **Bottom** (Number: Boolean)

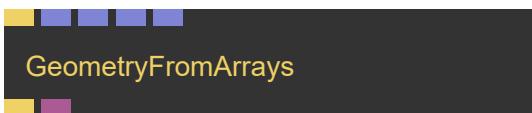
**< Output Ports:**

- **Result Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.GeometryExtrude>

## 75.1.9 GeometryFromArrays



**Full Name:** Ops.Graphics.Geometry.GeometryFromArrays

**Description:** Create a geometry from array data

**> Input Ports:**

- **Render** (Trigger)
- **Vertices** (Array)
- **Faces** (Array)
- **Texture Coords** (Array)
- **Normals** (Array)

**< Output Ports:**

- **Next** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.GeometryFromArrays>

## 75.1.10 GeometryInfo



**Full Name:** Ops.Graphics.Geometry.GeometryInfo

**Description:** information about a geometry

**> Input Ports:**

- **Geometry** (Object:Geometry)

**< Output Ports:**

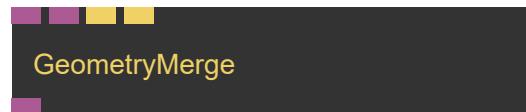
- **Indexed** (Number)

- **Faces** (Number)
- **Indices** (Number)
- **Vertices** (Number)
- **Normals** (Number)
- **TexCoords** (Number)
- **Tangents** (Number)
- **BiTangents** (Number)
- **VertexColors** (Number)
- **Other Attributes** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.GeometryInfo>

## 75.1.11 GeometryMerge



**Full Name:** Ops.Graphics.Geometry.GeometryMerge

**Description:** merge two geometries to one

**> Input Ports:**

- **Geometry** (Object)
- **Geometry 2** (Object)
- **Merge** (Trigger)
- **Reset** (Trigger)

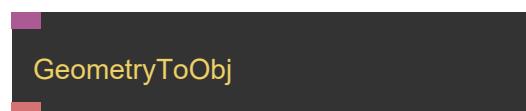
**< Output Ports:**

- **Geometry Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.GeometryMerge>

## 75.1.12 GeometryToObj



**Full Name:** Ops.Graphics.Geometry.GeometryToObj  
**Description:** Generate an .obj file as string from a geometry

- > Input Ports:**
- **Geometry** (Object:Geometry)
- < Output Ports:**
- **Obj** (String)

**Example Patch:** Open in Editor  
**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.GeometryToObj>

- **Geometry** (Object:Geometry)

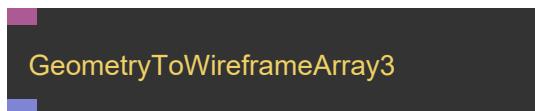
**< Output Ports:**

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.GeometryUnIndex>

### 75.1.13 GeometryToWireframeArray3



**Full Name:** Ops.Graphics.Geometry.GeometryToWireframeArray3  
**Description:** generate an array of lines from a mesh to render a wireframe  
**> Input Ports:**

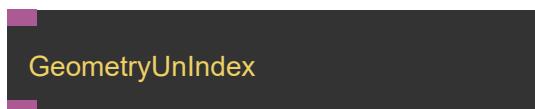
- **Geometry** (Object)

**< Output Ports:**

- **Array** (Array)

**Example Patch:** Open in Editor  
**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.GeometryToWireframeArray3>

### 75.1.14 GeometryUnIndex



**Full Name:** Ops.Graphics.Geometry.GeometryUnIndex  
**Description:** convert geometry to only flat triangles without reusing vertices positions  
**> Input Ports:**

### 75.1.15 ObjGeometry



**Full Name:** Ops.Graphics.Geometry.ObjGeometry  
**Description:** parse an obj string to a geometry object

**> Input Ports:**

- **Obj** (String)

**< Output Ports:**

- **Geometry** (Object)
- **Status** (String)

**Example Patch:** Open in Editor  
**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.ObjGeometry>

### 75.1.16 RandomizeTriangles



**Full Name:** Ops.Graphics.Geometry.RandomizeTriangles  
**Description:** randomize order of triangles in a geometry

**> Input Ports:**

- **Geometry** (Object)
- **Seed** (Number)

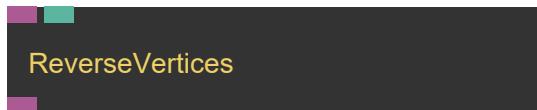
**< Output Ports:**

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.RandomizeTriangles>

## 75.1.17 ReverseVertices



**Full Name:** Ops.Graphics.Geometry.ReverseVertices

**Description:** Reverses the order of vertices in a geometry, back facing triangles become front facing ones

**> Input Ports:**

- **Geometry** (Object)
- **Flip** (Number: Boolean)

**< Output Ports:**

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.ReverseVertices>

## 75.1.18 ScaleGeometry



**Full Name:** Ops.Graphics.Geometry.ScaleGeometry

**Description:** uniform scaling of geometry vertices

**> Input Ports:**

- **Geometry** (Object)
- **Scale** (Number)

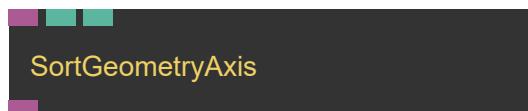
**< Output Ports:**

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.ScaleGeometry>

## 75.1.19 SortGeometryAxis



**Full Name:** Ops.Graphics.Geometry.SortGeometryAxis

**Description:** sort geometry triangles by position

**> Input Ports:**

- **Geometry** (Object)
- **Sort Index** (Number: Integer)
- **Reverse** (Number: Boolean)

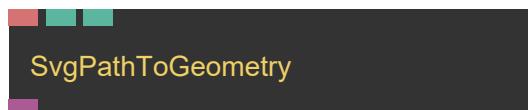
**< Output Ports:**

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.SortGeometryAxis>

## 75.1.20 SvgPathToGeometry\_v2



**Full Name:** Ops.Graphics.Geometry.SvgPathToGeometry\_v2

**Description:** Generate a SVG path string of a string using an opentype font

**> Input Ports:**

- **SVG Path** (String)
- **Bezier Stepsize** (Number)
- **Rescale** (Number)

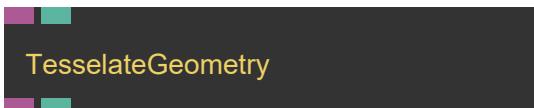
**< Output Ports:**

- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Graphics.Geometry.SvgPathToGeometry\\_v2](https://cables.gl/op/Ops.Graphics.Geometry.SvgPathToGeometry_v2)

## 75.1.21 TesselateGeometry



**Full Name:** Ops.Graphics.Geometry.TesselateGeometry

**Description:** create new triangles in a mesh (subdivide)

**> Input Ports:**

- **Geometry** (Object)
- **Iterations** (Number: Integer)

**< Output Ports:**

- **Result** (Object)
- **Num Vertices** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.TesselateGeometry>

## 75.1.22 TransformGeometry



**Full Name:** Ops.Graphics.Geometry.TransformGeometry

**Description:** transform vertices of geometry

**> Input Ports:**

- **Geometry** (Object)
- **Translate X** (Number)
- **Translate Y** (Number)
- **Translate Z** (Number)
- **Scale X** (Number)
- **Scale Y** (Number)
- **Scale Z** (Number)
- **Rotation X** (Number)
- **Rotation Y** (Number)
- **Rotation Z** (Number)

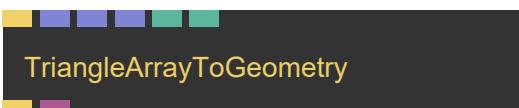
**< Output Ports:**

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.TransformGeometry>

## 75.1.23 TriangleArrayToGeometry\_v2



**Full Name:** Ops.Graphics.Geometry.TriangleArrayToGeometry\_v2

**Description:** Draws multiple triangles using coordinates from an array

**> Input Ports:**

- **Render** (Trigger)
- **Points** (Array)
- **Vertex Colors** (Array)
- **TexCoords** (Array)
- **Flat** (Number: Boolean)
- **Render Mesh** (Number: Boolean)

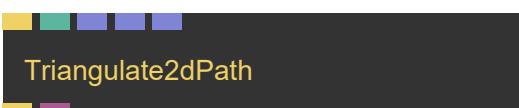
**< Output Ports:**

- **Next** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Graphics.Geometry.TriangleArrayToGeometry\\_v2](https://cables.gl/op/Ops.Graphics.Geometry.TriangleArrayToGeometry_v2)

## 75.1.24 Triangulate2dPath



**Full Name:** Ops.Graphics.Geometry.Triangulate2dPath

**Description:** Triangulate a 2d path to a flat and filled 3d geometry

**> Input Ports:**

- **Update** (Trigger)

- **Combine Index** (Number: Integer)
- **Path 2** (Array)
- **Path 3** (Array)

**< Output Ports:**

- **Next** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

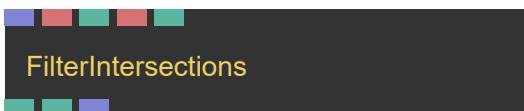
**Docs:** <https://cables.gl/op/Ops.Graphics.Geometry.Triangulate2dPath>

# 76 Ops.Graphics.Intersection

---

## 76.1 Ops.Graphics.Intersection

### 76.1.1 FilterIntersections



**Full Name:** Ops.Graphics.Intersection.FilterIntersections

**Description:** Define filters to get colliding and intersecting bodies

**> Input Ports:**

- **Collisions** (Array)
- **Name 1** (String)
- **Match Name 1 Index** (Number: Integer)
- **Name 2** (String)
- **Match Name 2 Index** (Number: Integer)

**< Output Ports:**

- **Colliding** (booleanNumber)
- **Num Collisions** (Number)
- **Result Collisions** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Intersection.FilterIntersections>

### 76.1.2 IntersectBody



**Full Name:** Ops.Graphics.Intersection.IntersectBody

**Description:** Add Bodies and check if they intersect/collide with each other

**> Input Ports:**

- **Trigger** (Trigger)

- **Name** (String)
- **Radius** (Number)
- **Size X** (Number)
- **Size Y** (Number)
- **Size Z** (Number)
- **Positions** (Array)
- **Append Index To Name** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.IntersectBody>

### 76.1.3 IntersectTestBody



**Full Name:** Ops.Graphics.Intersection.IntersectTestBody

**Description:** test one body against all bodies in the world

**> Input Ports:**

- **Trigger** (Trigger)
- **Name** (String)
- **Active** (Number: Boolean)

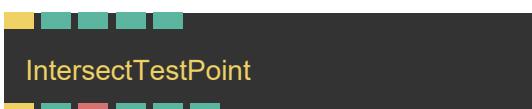
**< Output Ports:**

- **Next** (Trigger)
- **Has Hit** (booleanNumber)
- **Hit Body Name** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.IntersectTestBody>

### 76.1.4 IntersectTestPoint



**Full Name:** Ops.Graphics.Intersection.IntersectTestPoint

**Description:** test intersect bodies collision against a point/coordinate

**> Input Ports:**

- **Trigger** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Active** (Number: Boolean)

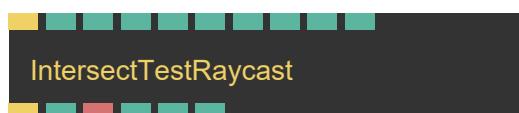
**< Output Ports:**

- **Next** (Trigger)
- **Has Hit** (booleanNumber)
- **Hit Body Name** (String)
- **Hit X** (Number)
- **Hit Y** (Number)
- **Hit Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.IntersectTestPoint>

### 76.1.5 IntersectTestRaycast



**Full Name:** Ops.Graphics.Intersection.IntersectTestRaycast

**Description:** Cast a ray and check if it intersect/collide with bodies

**> Input Ports:**

- **Trigger** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **To X** (Number)
- **To Y** (Number)
- **To Z** (Number)
- **Active** (Number: Boolean)
- **Change Cursor** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)
- **Has Hit** (booleanNumber)
- **Hit Body Name** (String)
- **Hit X** (Number)
- **Hit Y** (Number)
- **Hit Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Intersection.IntersectTestRaycast>

## 76.1.6 IntersectWorld



**Full Name:** Ops.Graphics.Intersection.IntersectWorld

**Description:** Define a world to check for intersections and collisions

**> Input Ports:**

- **Trigger** (Trigger)
- **Check Body Collisions** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)
- **Total Bodies** (Number)
- **Collisions** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Intersection.IntersectWorld>

# 77 Ops.Graphics.Meshes

## 77.1 Ops.Graphics.Meshes

### 77.1.1 CablesLogo



**Full Name:** Ops.Graphics.Meshes.CablesLogo

**Description:** cables logo mesh/geometry

**> Input Ports:**

- **Render** (Trigger)
- **Scale** (Number)
- **Draw** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Meshes.CablesLogo>

### 77.1.2 Circle\_v3



**Full Name:** Ops.Graphics.Meshes.Circle\_v3

**Description:** Draws a circle to the canvas.

**> Input Ports:**

- **Render** (Trigger)
- **Radius** (Number)
- **InnerRadius** (Number)
- **Segments** (Number: Integer)

- **Percent** (Number)
- **Steps** (Number)
- **InvertSteps** (Number: Boolean)
- **Spline** (Number: Boolean)
- **Render Mesh** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Graphics.Meshes.Circle\\_v3](https://cables.gl/op/Ops.Graphics.Meshes.Circle_v3)

### 77.1.3 Cross



**Full Name:** Ops.Graphics.Meshes.Cross

**Description:** Draws a cross with controllable thickness and length.

**> Input Ports:**

- **Render** (Trigger)
- **Size** (Number)
- **Thickness** (Number)
- **Crosshair** (Number: Boolean)
- **Left** (Number: Boolean)
- **Right** (Number: Boolean)
- **Top** (Number: Boolean)
- **Bottom** (Number: Boolean)
- **Active** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Graphics.Meshes.Cross>

### 77.1.4 Cube\_v2



**Full Name:** Ops.Graphics.Meshes.Cube\_v2

**Description:** Draws a cube to the canvas. Please note that without doing a rotation you will only see a rectangle.

**> Input Ports:**

- **Render** (Trigger)
- **Render Mesh** (Number: Boolean)
- **Width** (Number)
- **Length** (Number)
- **Height** (Number)
- **Center** (Number: Boolean)
- **Bias** (Number)
- **Flip X** (Number: Boolean)
- **Top** (Number: Boolean)
- **Bottom** (Number: Boolean)
- **Left** (Number: Boolean)
- **Right** (Number: Boolean)
- **Front** (Number: Boolean)
- **Back** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Graphics.Meshes.Cube\\_v2](https://cables.gl/op/Ops.Graphics.Meshes.Cube_v2)

### 77.1.5 Rectangle\_v4



**Full Name:** Ops.Graphics.Meshes.Rectangle\_v4

**Description:** draw a rectangle (plane, square)

#### > Input Ports:

- **Trigger** (Trigger)
- **Render** (Number: Boolean)
- **Width** (Number)
- **Height** (Number)
- **Flip TexCoord X** (Number: Boolean)
- **Flip TexCoord Y** (Number: Boolean)
- **Num Columns** (Number: Integer)
- **Num Rows** (Number: Integer)

#### < Output Ports:

- **Trigger** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Graphics.Meshes.Rectangle\\_v4](https://cables.gl/op/Ops.Graphics.Meshes.Rectangle_v4)

## 77.1.6 Sphere\_v3



**Full Name:** Ops.Graphics.Meshes.Sphere\_v3

**Description:** Draw parameterizable sphere

#### > Input Ports:

- **Render** (Trigger)
- **Radius** (Number)
- **Stacks** (Number)
- **Slices** (Number)
- **Filloffset** (Number)

#### < Output Ports:

- **Trigger** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Graphics.Meshes.Sphere\\_v3](https://cables.gl/op/Ops.Graphics.Meshes.Sphere_v3)

## 77.1.7 Star\_v2



**Full Name:** Ops.Graphics.Meshes.Star\_v2

**Description:** draw a star mesh (saw,gear)

#### > Input Ports:

- **Render** (Trigger)
- **Segments** (Number)
- **Radius** (Number)
- **Shape Index** (Number: Integer)
- **Length** (Number)
- **Peak Z Pos** (Number)
- **Percent** (Number)
- **Fill** (Number: Boolean)
- **Render Mesh** (Number: Boolean)

#### < Output Ports:

- **Trigger** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Graphics.Meshes.Star\\_v2](https://cables.gl/op/Ops.Graphics.Meshes.Star_v2)

## 77.1.8 Triangle\_v2



**Full Name:** Ops.Graphics.Meshes.Triangle\_v2

**Description:** Renders a triangle to the canvas.

#### > Input Ports:

- **Render** (Trigger)
- **Width** (Number)
- **Height** (Number)
- **Draw** (Number: Boolean)

< Output Ports:

- **Trigger** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Graphics.Meshes.Triangle\\_v2](https://cables.gl/op/Ops.Graphics.Meshes.Triangle_v2)

# 78 Ops.Html

---

## 78.1 Ops.Html

### 78.1.1 ActiveElement



**Full Name:** Ops.Html.ActiveElement

**Description:** Outputs the currently active/focused element

> Input Ports:

- **Trigger** (Trigger)

< Output Ports:

- **Active Element** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.ActiveElement>

### 78.1.2 AlignElement



**Full Name:** Ops.Html.AlignElement

**Description:** Align a HTML element to another, keep positioning

> Input Ports:

- **Element** (Object:Element)
- **Align Element** (Object:Element)
- **Force Update** (Trigger)
- **Offset X** (Number)
- **Offset Y** (Number)

< Output Ports:

- **Element Passthrough** (Object)
- **Aligned Element** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.AlignElement>

### 78.1.3 AppendChild\_v2



**Full Name:** Ops.Html.AppendChild\_v2

**Description:** Appends a HTML DOM Element to another

**> Input Ports:**

- **Parent** (Object:Element)
- **Child** (Object:Element)

**< Output Ports:**

- **Parent Out** (Object)
- **Child Out** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html.AppendChild\\_v2](https://cables.gl/op/Ops.Html.AppendChild_v2)

### 78.1.4 BrowserSpecificFile\_v2



**Full Name:** Ops.Html.BrowserSpecificFile\_v2

**Description:** set file dependant on browser

**> Input Ports:**

- **Chrome File** (String)
- **Firefox File** (String)
- **Safari File** (String)
- **Edge File** (String)
- **Opera File** (String)

- **Default File** (String)

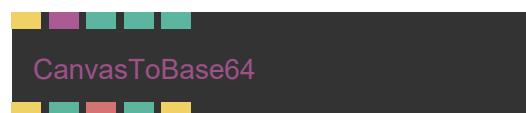
**< Output Ports:**

- **Browser Specific File** (String)
- **Detected Browser** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html.BrowserSpecificFile\\_v2](https://cables.gl/op/Ops.Html.BrowserSpecificFile_v2)

### 78.1.5 CanvasToBase64



**Full Name:** Ops.Html.CanvasToBase64

**Description:** Create an image file from a canvas

**> Input Ports:**

- **Trigger** (Trigger)
- **Texture** (Object)
- **Quality** (Number)
- **Output DataUrl** (Number: Boolean)

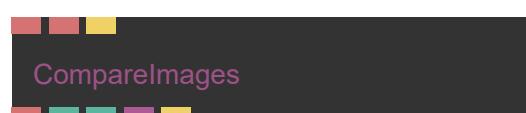
**< Output Ports:**

- **Next** (Trigger)
- **Binary Size** (Number)
- **Base64 String** (String)
- **Loading** (booleanNumber)
- **Finished** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.CanvasToBase64>

### 78.1.6 CompareImages\_v2



**Full Name:** Ops.Html.CompareImages\_v2

**Description:** compares two images and shows the difference as a pink color

**> Input Ports:**

- **Image 1** (String)
- **Image 2** (String)
- **Start** (Trigger)

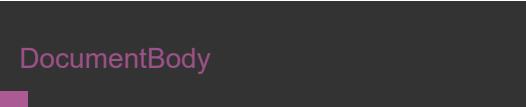
**< Output Ports:**

- **Difference Image** (String)
- **Mismatch Percentage** (Number)
- **Same Dimensions** (booleanNumber)
- **Resemble Data** (Object)
- **Finished** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html.CompareImages\\_v2](https://cables.gl/op/Ops.Html.CompareImages_v2)

## 78.1.7 DocumentBody



DocumentBody

**Full Name:** Ops.Html.DocumentBody

**Description:** Outputs the current document body element

**> Input Ports:**

- Visit *Ops.Html.DocumentBody documentation* for input port details

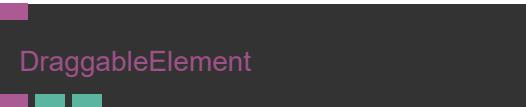
**< Output Ports:**

- **Body** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.DocumentBody>

## 78.1.8 DraggableElement



DraggableElement

**Full Name:** Ops.Html.DraggableElement

**Description:** Make a HTML element draggable to move it around with the mouse

**> Input Ports:**

- **Element** (Object:Element)

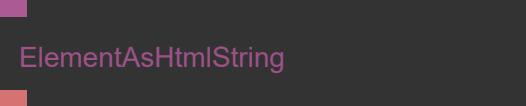
**< Output Ports:**

- **Element Out** (Object)
- **X** (Number)
- **Y** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.DraggableElement>

## 78.1.9 ElementAsHtmlString



ElementAsHtmlString

**Full Name:** Ops.Html.ElementAsHtmlString

**Description:** Serialize HTML/SVG elements to a string

**> Input Ports:**

- **Parent** (Object:Element)

**< Output Ports:**

- **HTML String** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.ElementAsHtmlString>

## 78.1.10 ElementChilds\_v2



ElementChilds

**Full Name:** Ops.Html.ElementChilds\_v2

**Description:** Set childs of a HTML Element

#### > Input Ports:

- **Parent** (Object:Element)
- **Child 1** (Object:Element)
- **Child 2** (Object:Element)
- **Child 3** (Object:Element)
- **Child 4** (Object:Element)
- **Child 5** (Object:Element)
- **Child 6** (Object:Element)
- **Child 7** (Object:Element)
- **Child 8** (Object:Element)
- **Child 9** (Object:Element)
- **Child 10** (Object:Element)

#### < Output Ports:

- **Parent Out** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html.ElementChilds\\_v2](https://cables.gl/op/Ops.Html.ElementChilds_v2)

## 78.1.11 ElementChildsMultiPort\_v2



**Full Name:** Ops.Html.ElementChildsMultiPort\_v2

**Description:** add child elements to another HTML Element

#### > Input Ports:

- **Parent** (Object:Element)
- **Childs\_0** (Object)
- **Add Port** (Object)

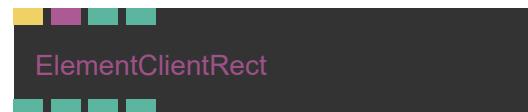
#### < Output Ports:

- **Parent Out** (Object)
- **Num Values** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html.ElementChildsMultiPort\\_v2](https://cables.gl/op/Ops.Html.ElementChildsMultiPort_v2)

## 78.1.12 ElementClientRect



**Full Name:** Ops.Html.ElementClientRect

**Description:** get html element absolute position and size in pixels on screen

#### > Input Ports:

- **Update** (Trigger)
- **Element** (Object:Element)

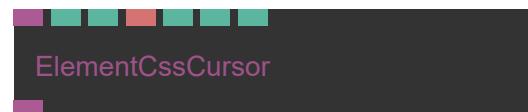
#### < Output Ports:

- **X** (Number)
- **Y** (Number)
- **Width** (Number)
- **Height** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.ElementClientRect>

## 78.1.13 ElementCssCursor\_v3



**Full Name:** Ops.Html.ElementCssCursor\_v3

**Description:** Set the mouse cursor

#### > Input Ports:

- **Element** (Object:Element)
- **CSS Cursors Index** (Number: Integer)
- **File** (String)
- **Offset X** (Number: Integer)
- **Offset Y** (Number: Integer)

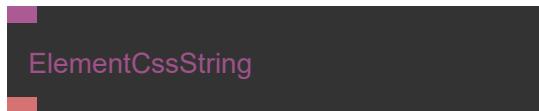
#### < Output Ports:

- **HTML Element** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html.ElementCssCursor\\_v3](https://cables.gl/op/Ops.Html.ElementCssCursor_v3)

## 78.1.14 ElementCssString



**Full Name:** Ops.Html.ElementCssString

**Description:** Output css attributes of an element as a string

**> Input Ports:**

- **Element** (Object:Element)

**< Output Ports:**

- **CSS** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.ElementCssString>

## 78.1.15 ElementDataSet



**Full Name:** Ops.Html.ElementDataSet

**Description:** Get the data-attributes and values of an HTML element

**> Input Ports:**

- **HTML Element** (Object:Element)

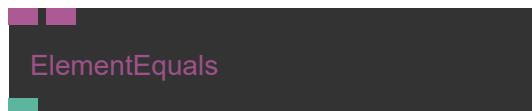
**< Output Ports:**

- **Dataset** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.ElementDataSet>

## 78.1.16 ElementEquals



**Full Name:** Ops.Html.ElementEquals

**Description:** Check if two HTML element objects are equal

**> Input Ports:**

- **HTML Element** (Object:Element)
- **HTML Element 2** (Object:Element)

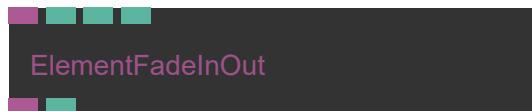
**< Output Ports:**

- **Equal** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.ElementEquals>

## 78.1.17 ElementFadeInOut\_v2



**Full Name:** Ops.Html.ElementFadeInOut\_v2

**Description:** fade html elements in or out

**> Input Ports:**

- **HTML Element** (Object)
- **Visible** (Number: Boolean)
- **Duration** (Number)
- **Opacity** (Number)

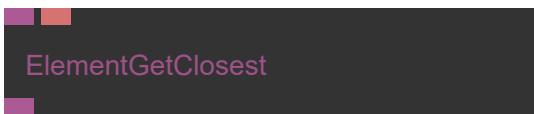
**< Output Ports:**

- **PassThrough** (Object)
- **Is Showing** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html.ElementFadeInOut\\_v2](https://cables.gl/op/Ops.Html.ElementFadeInOut_v2)

## 78.1.18 ElementGetClosest



**Full Name:** Ops.Html.ElementGetClosest

**Description:** get the closest parent element matching the query selector

**> Input Ports:**

- **HTML Element** (Object:Element)
- **Query** (String)

**< Output Ports:**

- **Element** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.ElementGetClosest>

## 78.1.19 ElementGradientBg



**Full Name:** Ops.Html.ElementGradientBg

**Description:** Use a cables gradient as HTML element background

**> Input Ports:**

- **Element** (Object:Element)
- **Rect Color Space Index** (Number: Integer)
- **Angle** (Number)
- **Gradient Object** (Object:Gradient)

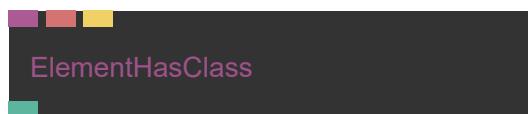
**< Output Ports:**

- **HTML Element** (Object)
- **CSS String** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.ElementGradientBg>

## 78.1.20 ElementHasClass



**Full Name:** Ops.Html.ElementHasClass

**Description:** Does the element currently have a specific class set

**> Input Ports:**

- **Element** (Object:Element)
- **Classname** (String)
- **Update** (Trigger)

**< Output Ports:**

- **Has Class** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.ElementHasClass>

## 78.1.21 ElementInfo



**Full Name:** Ops.Html.ElementInfo

**Description:** Get information about an element

**> Input Ports:**

- **Element** (Object)

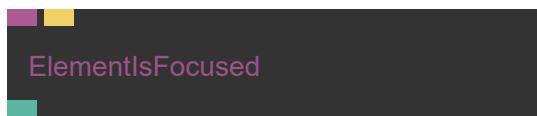
**< Output Ports:**

- **Tagname** (String)
- **Id** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.ElementInfo>

## 78.1.22 ElementIsFocused



ElementIsFocused

**Full Name:** Ops.Html.ElementIsFocused

**Description:** Is the connected element currently focused

**> Input Ports:**

- **Element** (Object:Element)
- **Update** (Trigger)

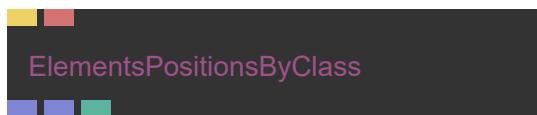
**< Output Ports:**

- **Has Focus** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.ElementIsFocused>

## 78.1.23 ElementsPositionsByClass



ElementsPositionsByClass

**Full Name:** Ops.Html.ElementsPositionsByClass

**Description:** get html element absolute positions and sizes by classname

**> Input Ports:**

- **Update** (Trigger)
- **Classname** (String)

**< Output Ports:**

- **Position** (Array)
- **Size** (Array)
- **Total Elements** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.ElementsPositionsByClass>

## 78.1.24 FontFile\_v2



FontFile

**Full Name:** Ops.Html.FontFile\_v2

**Description:** Load a font file like .otf, .ttf, .woff via css

**> Input Ports:**

- **File** (String)
- **Family** (String)
- **Active** (Number: Boolean)

**< Output Ports:**

- **Loaded** (booleanNumber)
- **Loaded Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html.FontFile\\_v2](https://cables.gl/op/Ops.Html.FontFile_v2)

## 78.1.25 FontsLoaded



FontsLoaded

**Full Name:** Ops.Html.FontsLoaded

**Description:** triggers when asynchronous requests finished loading

**> Input Ports:**

- Visit *Ops.Html.FontsLoaded documentation* for input port details

**< Output Ports:**

- **Font Loaded** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.FontsLoaded>

## 78.1.26 FullscreenMode



**Full Name:** Ops.Html.FullscreenMode

**Description:** Switch webgl to fullscreen

**> Input Ports:**

- **Request Fullscreen** (Trigger)
- **Exit Fullscreen** (Trigger)

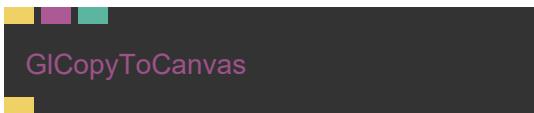
**< Output Ports:**

- **Is Fullscreen** (booleanNumber)
- **Supported** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.FullscreenMode>

## 78.1.27 GLCopyToCanvas



**Full Name:** Ops.Html.GLCopyToCanvas

**Description:** Copy GL canvas content to another canvas

**> Input Ports:**

- **Render** (Trigger)
- **Canvas** (Object:Element)
- **Smooth** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.GLCopyToCanvas>

## 78.1.28 HyperLink\_v3



**Full Name:** Ops.Html.HyperLink\_v3

**Description:** Open another website

**> Input Ports:**

- **Open** (Trigger)
- **URL** (String)
- **Frame Name** (String)
- **Win Specs** (String)
- **Rel Attribute** (String)

**< Output Ports:**

- Visit *Ops.Html.HyperLink\_v3 documentation* for output port details

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html.HyperLink\\_v3](https://cables.gl/op/Ops.Html.HyperLink_v3)

## 78.1.29 InnerHTML



**Full Name:** Ops.Html.InnerHTML

**Description:** Set innerHTML or innerTEXT of an HTML element

**> Input Ports:**

- **Element** (Object)
- **Value** (String)
- **Active** (Number: Boolean)
- **Clear** (Trigger)

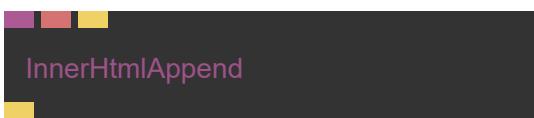
**< Output Ports:**

- **HTML Element** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.InnerHTML>

## 78.1.30 InnerHtmlAppend



**Full Name:** Ops.Html.InnerHtmlAppend

**Description:** Append string to the inner html or an element

**> Input Ports:**

- **Element** (Object:Element)
- **Html** (String)
- **Trigger** (Trigger)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.InnerHtmlAppend>

## 78.1.31 MailtoLink



**Full Name:** Ops.Html.MailtoLink

**Description:** creates a mailto: link to open the default email app

**> Input Ports:**

- **Email** (String)
- **Subject** (String)
- **Execute** (Trigger)

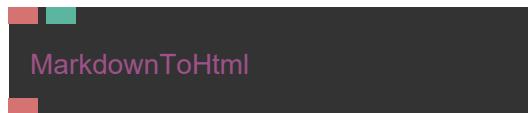
**< Output Ports:**

- *Visit Ops.Html.MailtoLink documentation for output port details*

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.MailtoLink>

## 78.1.32 MarkdownToHtml



**MarkdownToHtml**

**Full Name:** Ops.Html.MarkdownToHtml

**Description:** markdown markup language to html parser

**> Input Ports:**

- **Markdown** (String)
- **Active** (Number: Boolean)

**< Output Ports:**

- **Html** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.MarkdownToHtml>

## 78.1.33 ModalOverlay



**ModalOverlay**

**Full Name:** Ops.Html.ModalOverlay

**Description:** create a modal HTML overlay with a darkened background

**> Input Ports:**

- **Content Element** (Object)
- **Show** (Trigger)
- **Close** (Trigger)
- **Show Closebutton** (Number: Boolean)
- **Opacity** (Number)

**< Output Ports:**

- **Visible** (booleanNumber)
- **Closed** (Trigger)
- **Element** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.ModalOverlay>

## 78.1.34 QuerySelector\_v3



**Full Name:** Ops.Html.QuerySelector\_v3

**Description:** Selects an element in the DOM

**> Input Ports:**

- **Update** (Trigger)
- **Query** (String)
- **Type Index** (Number: Integer)
- **Document** (String)
- **Input Element** (Object:Element)

**< Output Ports:**

- **Element** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html.QuerySelector\\_v3](https://cables.gl/op/Ops.Html.QuerySelector_v3)

**Docs:** [https://cables.gl/op/Ops.Html.QuerySelectorAll\\_v2](https://cables.gl/op/Ops.Html.QuerySelectorAll_v2)

## 78.1.36 ReloadPage



**Full Name:** Ops.Html.ReloadPage

**Description:** reload the website

**> Input Ports:**

- **Exec** (Trigger)

**< Output Ports:**

- Visit *Ops.Html.ReloadPage documentation* for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.ReloadPage>

## 78.1.35 QuerySelectorAll\_v2



**Full Name:** Ops.Html.QuerySelectorAll\_v2

**Description:** Selects all matching elements in the DOM

**> Input Ports:**

- **Query** (String)
- **Mode Index** (Number: Integer)
- **Type Index** (Number: Integer)
- **Document** (String)
- **Element** (Object:Element)
- **Update** (Trigger)

**< Output Ports:**

- **Elements** (Array)

**Example Patch:** Open in Editor

## 78.1.37 ScrollIntoView



**Full Name:** Ops.Html.ScrollIntoView

**Description:** Scroll an area, so the html element is visible/in view

**> Input Ports:**

- **Element** (Object:Element)
- **Scroll Into View** (Trigger)

**< Output Ports:**

- **HTML Element** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.ScrollIntoView>

## 78.1.38 ScrollPosition\_v2



**Full Name:** Ops.Html.ScrollPosition\_v2

**Description:** the current x y top left scrolling position of html page or element

**> Input Ports:**

- **Update** (Trigger)
- **Element** (Object:Element)
- **Scroll To Top** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **Left** (Number)
- **Top** (Number)
- **Percentage X** (Number)
- **Percentage Y** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html.ScrollPosition\\_v2](https://cables.gl/op/Ops.Html.ScrollPosition_v2)

**Docs:** <https://cables.gl/op/Ops.Html.ScrollTo>

## 78.1.40 WindowClose



**Full Name:** Ops.Html.WindowClose

**Description:** close current window

**> Input Ports:**

- **Close** (Trigger)

**< Output Ports:**

- Visit *Ops.Html.WindowClose documentation* for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.WindowClose>

## 78.1.39 ScrollTo



**Full Name:** Ops.Html.ScrollTo

**Description:** Trigger the browser to scroll to top or bottom of an element

**> Input Ports:**

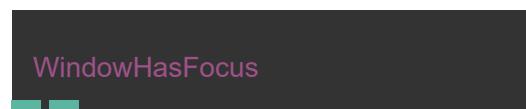
- **Element** (Object:Element)
- **Scroll To Top** (Trigger)
- **Scroll To Bottom** (Trigger)

**< Output Ports:**

- Visit *Ops.Html.ScrollTo documentation* for output port details

**Example Patch:** Open in Editor

## 78.1.41 WindowHasFocus



**Full Name:** Ops.Html.WindowHasFocus

**Description:** detect if the browser window/tab has focus

**> Input Ports:**

- Visit *Ops.Html.WindowHasFocus documentation* for input port details

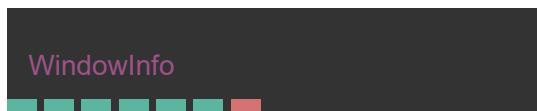
**< Output Ports:**

- **Has Focus** (booleanNumber)
- **Tab Visible** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.WindowHasFocus>

## 78.1.42 WindowInfo



**Full Name:** Ops.Html.WindowInfo

**Description:** size of browser window in pixels

**> Input Ports:**

- Visit *Ops.Html.WindowInfo documentation for input port details*

**< Output Ports:**

- **ClientWidth** (Number)
- **ClientHeight** (Number)
- **Body Scroll Height** (Number)
- **Device Pixel Ratio** (Number)
- **Iframe Parent** (booleanNumber)
- **Orientation Angle** (Number)
- **Orientation Type** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.WindowInfo>

## 78.1.43 WindowScroll



**Full Name:** Ops.Html.WindowScroll

**Description:** Get the current scroll position of the window

**> Input Ports:**

- **Update** (Trigger)

**< Output Ports:**

- **Scoll X** (Number)
- **Scoll Y** (Number)

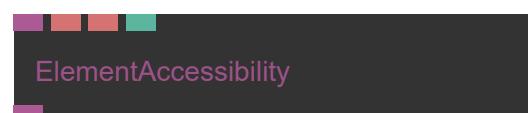
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.WindowScroll>

# 79 Ops.Html.Attributes

## 79.1 Ops.Html.Attributes

### 79.1.1 ElementAccessibility



**Full Name:** Ops.Html.Attributes.ElementAccessibility

**Description:** Element Accessibility properties for screen reader

**> Input Ports:**

- **Element** (Object)
- **Aria Label** (String)
- **Aria Labeled By** (String)
- **Aria Hidden** (Number: Boolean)

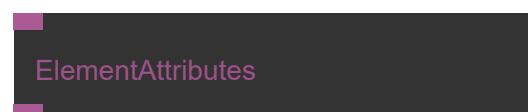
**< Output Ports:**

- **HTML Element** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.Attributes.ElementAccessibility>

### 79.1.2 ElementAttributes



**Full Name:** Ops.Html.Attributes.ElementAttributes

**Description:** Get all attributes from an element an object

**> Input Ports:**

- **Element** (Object)

**< Output Ports:**

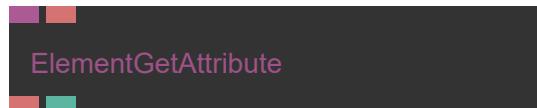
- **Attribs** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.Attributes.ElementAttributes>

**Docs:** <https://cables.gl/op/Ops.Html.Attributes.ElementSetAttribute>

### 79.1.3 ElementGetAttribute



**Full Name:** Ops.Html.Attributes.ElementGetAttribute

**Description:** Read or Get the value of an HTML element Attribute

**> Input Ports:**

- **Element** (Object)
- **Attribute Name** (String)

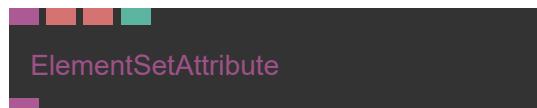
**< Output Ports:**

- **Value** (String)
- **Has Attribute** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.Attributes.ElementGetAttribute>

### 79.1.4 ElementSetAttribute



**Full Name:** Ops.Html.Attributes.ElementSetAttribute

**Description:** Write or Set the value of an HTML element Attribute

**> Input Ports:**

- **Element** (Object)
- **Attribute** (String)
- **Value** (String)
- **Active** (Number: Boolean)

**< Output Ports:**

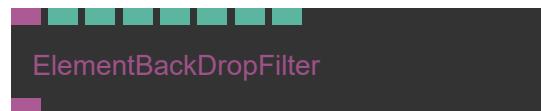
- **HTML Element** (Object)

**Example Patch:** Open in Editor

# 80 Ops.Html.Css

## 80.1 Ops.Html.Css

### 80.1.1 ElementBackDropFilter



**Full Name:** Ops.Html.Css.ElementBackDropFilter

**Description:** Set CSS backdrop filter like blur, contrast, brightness, saturation

#### > Input Ports:

- **Element** (Object)
- **Blur** (Number)
- **Contrast** (Number)
- **Brightness** (Number)
- **Hue** (Number)
- **Invert** (Number)
- **Saturate** (Number)
- **Sepia** (Number)

#### < Output Ports:

- **HTML Element** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.Css.ElementBackDropFilter>

### 80.1.2 ElementMargin



**Full Name:** Ops.Html.Css.ElementMargin

**Description:** Set CSS margins of a html element

#### > Input Ports:

- **Element** (Object:Element)
- **Margin** (Number)
- **Margin Top** (Number)
- **Margin Bottom** (Number)
- **Margin Left** (Number)
- **Margin Right** (Number)

#### < Output Ports:

- **HTML Element** (Object)

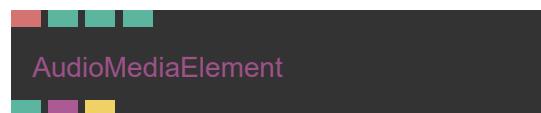
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.Css.ElementMargin>

# 81 Ops.Html.Elements

## 81.1 Ops.Html.Elements

### 81.1.1 AudioMediaElement



**Full Name:** Ops.Html.Elements.AudioMediaElement

**Description:** Simple Audio Player, using HTML5 Audio, does not need WebAudio

#### > Input Ports:

- **File** (String)
- **Play** (Number: Boolean)
- **Volume** (Number)
- **Loop** (Number: Boolean)

#### < Output Ports:

- **Playing** (Number)
- **Element** (Object)
- **Has Ended** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.Elements.AudioMediaElement>

### 81.1.2 Element\_v2



**Full Name:** Ops.Html.Elements.Element\_v2

**Description:** A more convenient version of div element op, that can be used for creating html without writing much css code

#### > Input Ports:

- **Text** (String)
- **Set Size** (Number: Boolean)
- **Width** (Number)
- **Height** (Number)
- **Inline Style** (String)
- **CSS Class** (String)
- **Disable CSS Props** (String)
- **Display Index** (Number: Integer)
- **Tag Name** (String)
- **Opacity** (Number)
- **Propagate Click-Events** (Number: Boolean)
- **Add To DOM** (Number: Boolean)

#### < Output Ports:

- **DOM Element** (Object)
- **Hovering** (booleanNumber)
- **Clicked** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html.Elements.Element\\_v2](https://cables.gl/op/Ops.Html.Elements.Element_v2)

### 81.1.3 IFrame\_v3



**Full Name:** Ops.Html.Elements.IFrame\_v3

**Description:** Show another website in an iframe element

#### > Input Ports:

- **URL** (String)
- **ID** (String)
- **Active** (Number: Boolean)
- **Style** (String)

#### < Output Ports:

- **Element** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html.Elements.IFrame\\_v3](https://cables.gl/op/Ops.Html.Elements.IFrame_v3)

## 81.1.4 ImageElement\_v3



**Full Name:** Ops.Html.Elements.ImageElement\_v3

**Description:** create an image(img) html element

**> Input Ports:**

- **File** (String)
- **Class** (String)
- **Style** (String)
- **Alt Text** (String)

**< Output Ports:**

- **Image Element** (Object)
- **Width** (Number)
- **Height** (Number)
- **Loading** (booleanNumber)
- **Error** (booleanNumber)
- **Loaded** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html.Elements.ImageElement\\_v3](https://cables.gl/op/Ops.Html.Elements.ImageElement_v3)

## 81.1.5 InputElement



**Full Name:** Ops.Html.Elements.InputElement

**Description:** HTML input/textarea element to allow the user to enter text

**> Input Ports:**

- **Default Value** (String)
- **Placeholder** (String)
- **Id** (String)
- **Class** (String)
- **Style** (String)

- **Autocomplete** (Number: Boolean)
- **Max Length** (Number: Integer)
- **Enter Key Prevent Default** (Number: Boolean)
- **Visible** (Number: Boolean)
- **Focus** (Trigger)
- **Blur** (Trigger)
- **Clear** (Trigger)
- **Select** (Trigger)

**< Output Ports:**

- **DOM Element** (Object)
- **Value** (String)
- **Hover** (booleanNumber)
- **Enter Pressed** (Trigger)
- **Escape Pressed** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.Elements.InputElement>

## 81.1.6 VideoElement



**Full Name:** Ops.Html.Elements.VideoElement

**Description:** html video player element

**> Input Ports:**

- **File** (String)
- **ID** (String)
- **Play** (Number: Boolean)
- **Autoplay** (Number: Boolean)
- **Controls** (Number: Boolean)
- **Active** (Number: Boolean)
- **Loop** (Number: Boolean)
- **Muted** (Number: Boolean)
- **Style** (String)
- **Rewind** (Trigger)

**< Output Ports:**

- **Element** (Object)
- **Playing** (booleanNumber)
- **Can Play Through** (booleanNumber)
- **Time** (Number)
- **Ended** (Trigger)
- **Has Error** (booleanNumber)
- **Error Message** (String)
- **Video Width** (Number)
- **Video Height** (Number)

**Example Patch:** Open in Editor

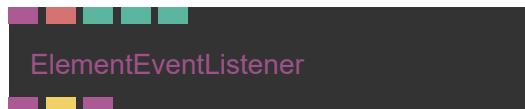
**Docs:** <https://cables.gl/op/Ops.Html.Elements.VideoElement>

## 82 Ops.Html.Event

---

### 82.1 Ops.Html.Event

#### 82.1.1 ElementEventListener\_v2



**Full Name:** Ops.Html.Event.ElementEventListener\_v2

**Description:** Add a custom event listener

**> Input Ports:**

- **Element** (Object)
- **Event Name** (String)
- **Use Capture** (Number: Boolean)
- **Prevent Default** (Number: Boolean)
- **Stop Propagation** (Number: Boolean)

**< Output Ports:**

- **Element Passthrough** (Object)
- **Event Trigger** (Trigger)
- **Event Object** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html.Event.ElementEventListener\\_v2](https://cables.gl/op/Ops.Html.Event.ElementEventListener_v2)

#### 82.1.2 ElementPointerEvents



**Full Name:** Ops.Html.Event.ElementPointerEvents

**Description:** Listen to events of an element

**> Input Ports:**

- **Dom Element** (Object)

- **Mouse Down Active** (Number: Boolean)
- **Mouse Up Active** (Number: Boolean)
- **Click Active** (Number: Boolean)
- **Mouse Move Active** (Number: Boolean)
- **Touch Start Active** (Number: Boolean)
- **Touch Move Active** (Number: Boolean)
- **Touch End Active** (Number: Boolean)
- **Touch Cancel Active** (Number: Boolean)

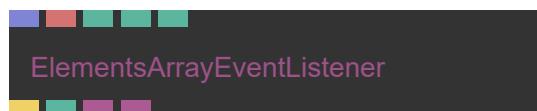
**< Output Ports:**

- **Event Object** (Object)
- **Mouse Down** (Trigger)
- **Mouse Up** (Trigger)
- **Click** (Trigger)
- **Mouse Move** (Trigger)
- **Touch Start** (Trigger)
- **Touch Move** (Trigger)
- **Touch End** (Trigger)
- **Touch Cancel** (Trigger)
- **Event Name** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.Event.ElementPointerEvents>

## 82.1.3 ElementsArrayEventListener



**Full Name:** Ops.Html.Event.ElementsArrayEventListener

**Description:** listen to events on multiple html elements

**> Input Ports:**

- **Elements** (Array)
- **Event Name** (String)
- **Use Capture** (Number: Boolean)
- **Prevent Default** (Number: Boolean)
- **Stop Propagation** (Number: Boolean)

**< Output Ports:**

- **Event Trigger** (Trigger)
- **Index** (Number)
- **Event Object** (Object)
- **Event Element** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.Event.ElementsArrayEventListener>

## 82.1.4 PreventDefault



**Full Name:** Ops.Html.Event.PreventDefault

**Description:** Prevents the default on a JavaScript event

**> Input Ports:**

- **Execute** (Trigger)
- **Event In** (Object)

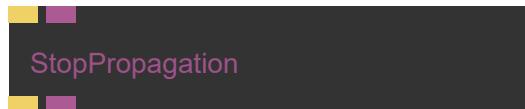
**< Output Ports:**

- **Next** (Trigger)
- **Event Out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.Event.PreventDefault>

## 82.1.5 StopPropagation



**Full Name:** Ops.Html.Event.StopPropagation

**Description:** Stop a JavaScript event (bubbling / capturing)

**> Input Ports:**

- **Execute** (Trigger)
- **Event In** (Object)

**< Output Ports:**

- **Next** (Trigger)
- **Event Out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.Event.StopPropagation>

## 83 Ops.Html\_Utils

---

### 83.1 Ops.Html\_Utils

#### 83.1.1 CablesLink



**Full Name:** Ops.Html\_Utils.CablesLink

**Description:** create a cables logo which links to cables gl

**> Input Ports:**

- **Size** (Number)
- **Opacity** (Number)

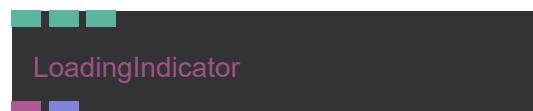
**< Output Ports:**

- Visit *Ops.Html\_Utils.CablesLink documentation for output port details*

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html\\_Utils.CablesLink](https://cables.gl/op/Ops.Html_Utils.CablesLink)

#### 83.1.2 LoadingIndicator\_v2



**Full Name:** Ops.Html\_Utils.LoadingIndicator\_v2

**Description:** show a typical web loading/progress indicator animation

**> Input Ports:**

- **Center Position** (Number: Boolean)

**< Output Ports:**

- **Elment** (Object)
- **Requests** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html.Utils.LoadingIndicator\\_v2](https://cables.gl/op/Ops.Html.Utils.LoadingIndicator_v2)

### 83.1.3 Notification



**Full Name:** Ops.Html.Utils.Notification

**Description:** Trigger a simple pop up notification on the screen

#### > Input Ports:

- **Trigger Animation** (Trigger)
- **Text** (String)
- **Class** (String)
- **Style** (String)
- **Active** (Number: Boolean)
- **Convert Line Breaks** (Number: Boolean)
- **Fade In** (Number)
- **Hold** (Number)
- **Fade Out** (Number)
- **Mode Index** (Number: Integer)
- **Side Index** (Number: Integer)
- **Starting Position** (Number)
- **Ending Position** (Number)

#### < Output Ports:

- **Finished Trigger** (Trigger)
- **Finished** (booleanNumber)
- **DOM Element** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.Utils.Notification>

### 83.1.4 PlayButton



PlayButton

**Full Name:** Ops.Html.Utils.PlayButton

**Description:** shows a playbutton for forcing a simple user interaction

#### > Input Ports:

- **Trigger** (Trigger)
- **Only If Audio Suspended** (Number: Boolean)
- **Reset** (Trigger)
- **Style Outer** (String)
- **Style Inner** (String)
- **Active** (Number: Boolean)

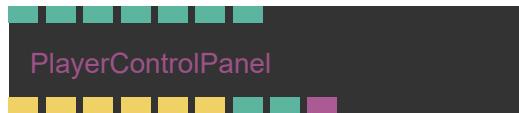
#### < Output Ports:

- **Next** (Trigger)
- **Not Clicked** (Trigger)
- **Audiocontext State** (String)
- **Element** (Object)
- **Clicked** (booleanNumber)
- **Clicked Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.Utils.PlayButton>

### 83.1.5 PlayerControlPanel\_v2



**Full Name:** Ops.Html.Utils.PlayerControlPanel\_v2

**Description:** simple html ui for timeline/mediaplayers (was: TimeLineUI)

#### > Input Ports:

- **Length** (Number)
- **Current** (Number)
- **Clamp** (Number: Boolean)
- **Is Playing** (Number: Boolean)
- **Visible** (Number: Boolean)
- **Show Time** (Number: Boolean)
- **Show Skip Buttons** (Number: Boolean)

#### < Output Ports:

- **Play Clicked** (Trigger)

- **Pause Clicked** (Trigger)
- **Rewind Clicked** (Trigger)
- **Skip Back Clicked** (Trigger)
- **Skip Forward Clicked** (Trigger)
- **Dragged** (Trigger)
- **Current Value** (Number)
- **Dragging** (booleanNumber)
- **DOM Element** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Html.Utils.PlayerControlPanel\\_v2](https://cables.gl/op/Ops.Html.Utils.PlayerControlPanel_v2)

### 83.1.6 QrCode



**Full Name:** Ops.Html.Utils.QrCode

**Description:** Generate a qr code as a texture

**> Input Ports:**

- **Text** (String)

**< Output Ports:**

- **Image DataUrl** (String)
- **Element** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.Utils.QrCode>

### 83.1.7 YoutubePlayer



**Full Name:** Ops.Html.Utils.YoutubePlayer

**Description:** play a youtube video in a HTML element

**> Input Ports:**

- **Video Id** (String)
- **Active** (Number: Boolean)
- **Style** (String)
- **ElementID** (String)
- **Autoplay** (Number: Boolean)
- **Display Captions** (Number: Boolean)
- **Loop** (Number: Boolean)
- **Allow Fullscreen** (Number: Boolean)
- **Hide Controls** (Number: Boolean)
- **Start At Second** (Number: Integer)

**< Output Ports:**

- **Element** (Object)
- **Direct Link** (String)

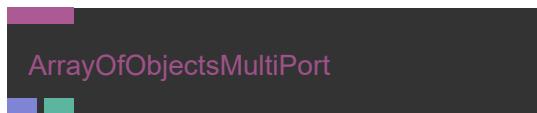
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Html.Utils.YoutubePlayer>

# 84 Ops.Json

## 84.1 Ops.Json

### 84.1.1 ArrayOfObjectsMultiPort\_v2



**Full Name:** Ops.Json.ArrayOfObjectsMultiPort\_v2

**Description:** create an array with multiple objects

**> Input Ports:**

- **Objects\_0** (Object)
- **Add Port** (Object)

**< Output Ports:**

- **Array** (Array)
- **Num Values** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Json.ArrayOfObjectsMultiPort\\_v2](https://cables.gl/op/Ops.Json.ArrayOfObjectsMultiPort_v2)

### 84.1.2 CopyObject



**Full Name:** Ops.Json.CopyObject

**Description:** Creates a copy of a JSON object

**> Input Ports:**

- Visit *Ops.Json.CopyObject documentation for input port details*

**< Output Ports:**

- **Valid** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.CopyObject>

### 84.1.3 CsvArray



**Full Name:** Ops.Json.CsvArray

**Description:** parse CSV files as array

**> Input Ports:**

- **File** (String)

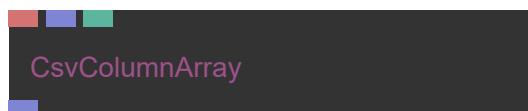
**< Output Ports:**

- **Result** (Array)
- **Num Items** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.CsvArray>

### 84.1.4 CsvColumnArray\_v2



**Full Name:** Ops.Json.CsvColumnArray\_v2

**Description:** get all values of a CSV column as array of strings

**> Input Ports:**

- **Column Name** (String)
- **CSV Array** (Array)
- **Numbers** (Number: Boolean)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Json.CsvColumnArray\\_v2](https://cables.gl/op/Ops.Json.CsvColumnArray_v2)

## 84.1.5 EmptyObject



EmptyObject

**Full Name:** Ops.Json.EmptyObject

**Description:** Visit documentation for details

**> Input Ports:**

- Visit *Ops.Json.EmptyObject* documentation for input port details

**< Output Ports:**

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.EmptyObject>

**Full Name:** Ops.Json.GateObject

**Description:** Will only allow an Object to be output if the the pass through parameter evaluates to true

**> Input Ports:**

- **Object In** (Object)
- **Pass Through** (Number: Boolean)
- **Only Valid Objects** (Number: Boolean)

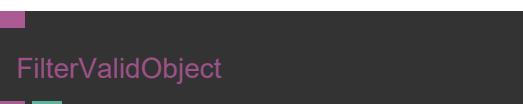
**< Output Ports:**

- **Object Out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.GateObject>

## 84.1.6 FilterValidObject



FilterValidObject

**Full Name:** Ops.Json.FilterValidObject

**Description:** Filter valid objects

**> Input Ports:**

- **Object** (Object)

**< Output Ports:**

- **Last Valid Object** (Object)
- **Is Valid** (booleanNumber)

**Example Patch:** Open in Editor

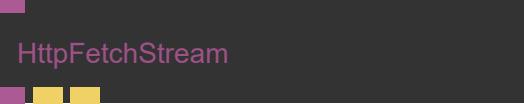
**Docs:** <https://cables.gl/op/Ops.Json.FilterValidObject>

## 84.1.7 GateObject



GateObject

## 84.1.8 HttpFetchStream



HttpFetchStream

**Full Name:** Ops.Json.HttpFetchStream

**Description:** HttpRequest/Fetch Streaming

**> Input Ports:**

- **Fetch Response** (Object)

**< Output Ports:**

- **Result** (Object)
- **Received Result** (Trigger)
- **Started** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.HttpFetchStream>

## 84.1.9 HttpRequest\_v4



HttpRequest

**Full Name:** Ops.Json.HttpRequest\_v4

**Description:** Request a json file and output an object (ajax, url, json,fetch)

**> Input Ports:**

- **URL** (String)
- **HTTP Method Index** (Number: Integer)
- **Request Body** (String)
- **Content-Type** (String)
- **the content type of the body sent** (if any)
- **Send Credentials** (Number: Boolean)
- **Headers** (Object)
- **Auto Request** (Number: Boolean)
- **trigger the request on any value change** (or on pagereload)
- **Empty Output On Change** (Number: Boolean)
- **Retry On Error** (Number: Boolean)
- **Reload** (Trigger)

**< Output Ports:**

- **Response Json Object** (Object)
- **Response String** (String)
- **Response Data Url** (String)
- **Status Code** (Number)
- **Is Loading** (booleanNumber)
- **Has Error** (booleanNumber)
- **Error** (String)
- **Duration MS** (Number)
- **Fetch Response** (Object)
- **Loaded** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Json.HttpRequest\\_v4](https://cables.gl/op/Ops.Json.HttpRequest_v4)

## 84.1.10 Object



**Full Name:** Ops.Json.Object

**Description:** Visit documentation for details

**> Input Ports:**

- **Object** (Object)

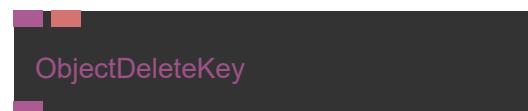
**< Output Ports:**

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.Object>

## 84.1.11 ObjectDeleteKey



**Full Name:** Ops.Json.ObjectDeleteKey

**Description:** Remove a Property from an Object by Key

**> Input Ports:**

- **Object** (Object)
- **Key** (String)

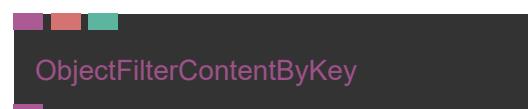
**< Output Ports:**

- **Object Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.ObjectDeleteKey>

## 84.1.12 ObjectFilterContentByKey



**Full Name:** Ops.Json.ObjectFilterContentByKey

**Description:** filter values from an object if key starts with input string

**> Input Ports:**

- **Object** (Object)
- **Name** (String)
- **Remove Null** (Number: Boolean)

**< Output Ports:**

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.ObjectFilterContentByKey>

### 84.1.13 ObjectFunnel



**Full Name:** Ops.Json.ObjectFunnel

**Description:** outputs the last changed object

**> Input Ports:**

- **Object1** (Object)
- **Object2** (Object)
- **Object3** (Object)
- **Object4** (Object)
- **Object5** (Object)

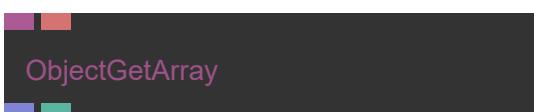
**< Output Ports:**

- **Out Object** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.ObjectFunnel>

### 84.1.14 ObjectGetArray\_v2



**Full Name:** Ops.Json.ObjectGetArray\_v2

**Description:** Returns an array from a JSON-object

**> Input Ports:**

- **Data** (Object)
- **Key** (String)

**< Output Ports:**

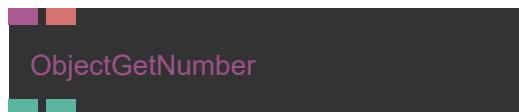
- **Result** (Array)

- **Length** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Json.ObjectGetArray\\_v2](https://cables.gl/op/Ops.Json.ObjectGetArray_v2)

### 84.1.15 ObjectGetNumber\_v2



**Full Name:** Ops.Json.ObjectGetNumber\_v2

**Description:** Get a number from an object

**> Input Ports:**

- **Data** (Object)
- **Key** (String)

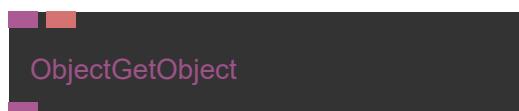
**< Output Ports:**

- **Result** (Number)
- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Json.ObjectGetNumber\\_v2](https://cables.gl/op/Ops.Json.ObjectGetNumber_v2)

### 84.1.16 ObjectGetObject\_v2



**Full Name:** Ops.Json.ObjectGetObject\_v2

**Description:** Get an object from an object

**> Input Ports:**

- **Object** (Object)
- **Key** (String)

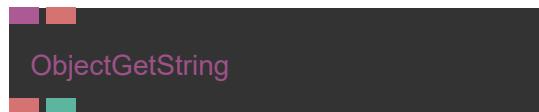
**< Output Ports:**

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Json.ObjectGetObject\\_v2](https://cables.gl/op/Ops.Json.ObjectGetObject_v2)

## 84.1.17 ObjectGetString\_v2



**Full Name:** Ops.Json.ObjectGetString\_v2

**Description:** Get string from object by key

**> Input Ports:**

- **Data** (Object)
- **Key** (String)

**< Output Ports:**

- **Result** (String)
- **Found** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Json.ObjectGetString\\_v2](https://cables.gl/op/Ops.Json.ObjectGetString_v2)

## 84.1.18 ObjectIsNull



**Full Name:** Ops.Json.ObjectIsNull

**Description:** check if object is null or a valid object

**> Input Ports:**

- **Object** (Object)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.ObjectIsNull>

## 84.1.19 ObjectKeys



**Full Name:** Ops.Json.ObjectKeys

**Description:** returns an array of strings, which contain the keys of the object

**> Input Ports:**

- **Object** (Object)

**< Output Ports:**

- **Keys** (Array)
- **Num Keys** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.ObjectKeys>

## 84.1.20 ObjectMerge



**Full Name:** Ops.Json.ObjectMerge

**Description:** merge key+values of two objects

**> Input Ports:**

- **Object 1** (Object)
- **Object 2** (Object)

**< Output Ports:**

- **Object Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.ObjectMerge>

## 84.1.21 ObjectOr



**Full Name:** Ops.Json.ObjectOr

**Description:** result is first connected valid object

**> Input Ports:**

- **Object 1** (Object)
- **Object 2** (Object)
- **Object 3** (Object)
- **Object 4** (Object)
- **Object 5** (Object)
- **Object 6** (Object)
- **Object 7** (Object)
- **Object 8** (Object)

**< Output Ports:**

- **Result** (Object)

**Example Patch:** Open in Editor

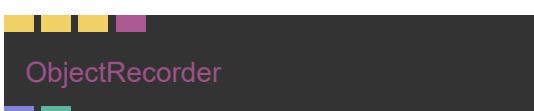
**Docs:** <https://cables.gl/op/Ops.Json.ObjectOr>

- **Num Objects** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.ObjectRecorder>

## 84.1.22 ObjectRecorder



**Full Name:** Ops.Json.ObjectRecorder

**Description:** record objects and download as json file

**> Input Ports:**

- **Exec** (Trigger)
- **Reset** (Trigger)
- **Download** (Trigger)
- **Object** (Object)

**< Output Ports:**

- **Result** (Array)

## 84.1.23 ObjectSetArray\_v2



**Full Name:** Ops.Json.ObjectSetArray\_v2

**Description:** Set array by key in an object

**> Input Ports:**

- **Object** (Object)
- **Key** (String)
- **Value** (Array)

**< Output Ports:**

- **Result Object** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Json.ObjectSetArray\\_v2](https://cables.gl/op/Ops.Json.ObjectSetArray_v2)

## 84.1.24 ObjectSetBool



**Full Name:** Ops.Json.ObjectSetBool

**Description:** set number at key in an object

**> Input Ports:**

- **Object** (Object)
- **Key** (String)
- **Boolean** (Number: Boolean)

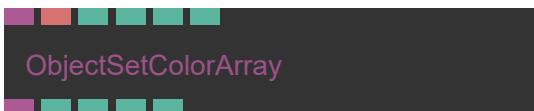
**< Output Ports:**

- **Result Object** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.ObjectSetBool>

### 84.1.25 ObjectSetColorArray



**Full Name:** Ops.Json.ObjectSetColorArray

**Description:** Set rgba array by key in an object

**> Input Ports:**

- **Object** (Object)
- **Key** (String)
- **R** (Number)
- **G** (Number)
- **B** (Number)
- **A** (Number)

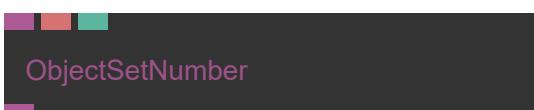
**< Output Ports:**

- **Result Object** (Object)
- **Out R** (Number)
- **Out G** (Number)
- **Out B** (Number)
- **Out A** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.ObjectSetColorArray>

### 84.1.26 ObjectSetNumber\_v2



**Full Name:** Ops.Json.ObjectSetNumber\_v2

**Description:** set number at key in an object

**> Input Ports:**

- **Object** (Object)
- **Key** (String)
- **Number** (Number)

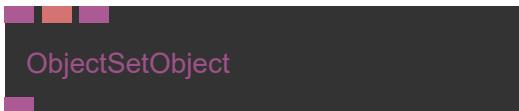
**< Output Ports:**

- **Result Object** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Json.ObjectSetNumber\\_v2](https://cables.gl/op/Ops.Json.ObjectSetNumber_v2)

### 84.1.27 ObjectSetObject\_v2



**Full Name:** Ops.Json.ObjectSetObject\_v2

**Description:** set object as value in an object

**> Input Ports:**

- **Object** (Object)
- **Key** (String)
- **Object Value** (Object)

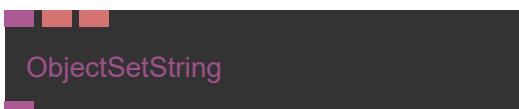
**< Output Ports:**

- **Result Object** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Json.ObjectSetObject\\_v2](https://cables.gl/op/Ops.Json.ObjectSetObject_v2)

### 84.1.28 ObjectSetString\_v2



**Full Name:** Ops.Json.ObjectSetString\_v2

**Description:** set a string value by key in an object

**> Input Ports:**

- **Object** (Object)
- **Key** (String)

- **Value** (String)

**< Output Ports:**

- **Result Object** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Json.ObjectSetString\\_v2](https://cables.gl/op/Ops.Json.ObjectSetString_v2)

## 84.1.29 ObjectStringify\_v2



**Full Name:** Ops.Json.ObjectStringify\_v2

**Description:** Convert object to string

**> Input Ports:**

- **Object** (Object)
- **Beautify** (Number: Boolean)

**< Output Ports:**

- **Result** (String)
- **Error** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Json.ObjectStringify\\_v2](https://cables.gl/op/Ops.Json.ObjectStringify_v2)

## 84.1.30 ObjectToArray



**Full Name:** Ops.Json.ObjectToArray

**Description:** cast an object port to an array port

**> Input Ports:**

- **Object** (Object)

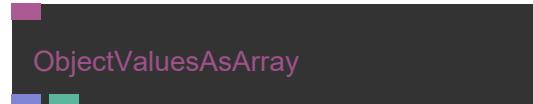
**< Output Ports:**

- **Array** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.ObjectToArray>

## 84.1.31 ObjectValuesAsArray



**Full Name:** Ops.Json.ObjectValuesAsArray

**Description:** extract all object values as an array

**> Input Ports:**

- **Object** (Object)

**< Output Ports:**

- **Values** (Array)
- **Num Values** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.ObjectValuesAsArray>

## 84.1.32 ParseObject\_v2



**Full Name:** Ops.Json.ParseObject\_v2

**Description:** Parses a string to a JSON object

**> Input Ports:**

- **JSON String** (String)

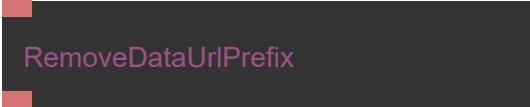
**< Output Ports:**

- **Result** (Object)
- **Valid** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Json.ParseObject\\_v2](https://cables.gl/op/Ops.Json.ParseObject_v2)

### 84.1.33 RemoveDataUrlPrefix



RemoveDataUrlPrefix

**Full Name:** Ops.Json.RemoveDataUrlPrefix

**Description:** Removes data URL prefix from a string

**> Input Ports:**

- **String Input** (String)

**< Output Ports:**

- **String Output** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.RemoveDataUrlPrefix>

### 84.1.34 RouteObject



RouteObject

**Full Name:** Ops.Json.RouteObject

**Description:** Route an object to an output port

**> Input Ports:**

- **Index** (Number: Integer)
- **Object In** (Object)
- **Default Object** (Object)

**< Output Ports:**

- **Index 0 Object** (Object)
- **Index 1 Object** (Object)
- **Index 2 Object** (Object)
- **Index 3 Object** (Object)
- **Index 4 Object** (Object)
- **Index 5 Object** (Object)
- **Index 6 Object** (Object)
- **Index 7 Object** (Object)
- **Index 8 Object** (Object)

- **Index 9 Object** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.RouteObject>

### 84.1.35 SaveJsonFile



SaveJsonFile

**Full Name:** Ops.Json.SaveJsonFile

**Description:** save/download an object as json file

**> Input Ports:**

- **Download** (Trigger)
- **Filename** (String)
- **Object** (Object)

**< Output Ports:**

- Visit *Ops.Json.SaveJsonFile documentation for output port details*

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.SaveJsonFile>

### 84.1.36 SequenceObjects\_v2



SequenceObjects

**Full Name:** Ops.Json.SequenceObjects\_v2

**Description:** control order and flow of objects

**> Input Ports:**

- **Number 0** (Object)
- **Number 1** (Object)
- **Number 2** (Object)
- **Number 3** (Object)
- **Number 4** (Object)
- **Number 5** (Object)

- **Number 6** (Object)
- **Number 7** (Object)
- **Number 8** (Object)
- **Number 9** (Object)
- **Number 10** (Object)
- **Number 11** (Object)
- **Number 12** (Object)
- **Number 13** (Object)
- **Number 14** (Object)
- **Number 15** (Object)

**< Output Ports:**

- **Output 0** (Object)
- **Output 1** (Object)
- **Output 2** (Object)
- **Output 3** (Object)
- **Output 4** (Object)
- **Output 5** (Object)
- **Output 6** (Object)
- **Output 7** (Object)
- **Output 8** (Object)
- **Output 9** (Object)
- **Output 10** (Object)
- **Output 11** (Object)
- **Output 12** (Object)
- **Output 13** (Object)
- **Output 14** (Object)
- **Output 15** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Json.SequenceObjects\\_v2](https://cables.gl/op/Ops.Json.SequenceObjects_v2)

### 84.1.37 SwitchObject



**Full Name:** Ops.Json.SwitchObject

**Description:** Allows switching between objects

**> Input Ports:**

- **Object Index** (Number: Integer)
- **Object Port 0** (Object)
- **Object Port 1** (Object)
- **Object Port 2** (Object)
- **Object Port 3** (Object)
- **Object Port 4** (Object)
- **Object Port 5** (Object)
- **Object Port 6** (Object)
- **Object Port 7** (Object)

**< Output Ports:**

- **Object Out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.SwitchObject>

### 84.1.38 SwitchObjectMultiPort\_v2



**Full Name:** Ops.Json.SwitchObjectMultiPort\_v2

**Description:** Switch between multiple object inputs

**> Input Ports:**

- **Index** (Number: Integer)
- **Objects\_0** (Object)
- **Add Port** (Object)

**< Output Ports:**

- **Object** (Object)
- **Num Values** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Json.SwitchObjectMultiPort\\_v2](https://cables.gl/op/Ops.Json.SwitchObjectMultiPort_v2)

## 84.1.39 TriggerObject



**Full Name:** Ops.Json.TriggerObject

**Description:** set output object when triggered

**> Input Ports:**

- **Trigger** (Trigger)
- **Object** (Object)

**< Output Ports:**

- **Next** (Trigger)
- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.TriggerObject>

## 84.1.40 TriggerObjectSetNumber



**Full Name:** Ops.Json.TriggerObjectSetNumber

**Description:** set a number value of an object using trigger

**> Input Ports:**

- **Trigger** (Trigger)
- **Object** (Object)
- **Key** (String)
- **Number** (Number)

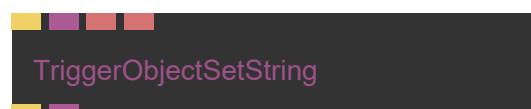
**< Output Ports:**

- **Next** (Trigger)
- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.TriggerObjectSetNumber>

## 84.1.41 TriggerObjectSetString



**Full Name:** Ops.Json.TriggerObjectSetString

**Description:** set a string value of an object using trigger

**> Input Ports:**

- **Trigger** (Trigger)
- **Object** (Object)
- **Key** (String)
- **String** (String)

**< Output Ports:**

- **Next** (Trigger)
- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Json.TriggerObjectSetString>

# 85 Ops.Math

## 85.1 Ops.Math

### 85.1.1 Abs



**Full Name:** Ops.Math.Abs

**Description:** Returns the absolute, positive value

#### > Input Ports:

- **Number** (Number)

#### < Output Ports:

- **Result** (Number)
- **The absolute value of Number** (always positive)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Abs>

### 85.1.2 Accumulator



**Full Name:** Ops.Math.Accumulator

**Description:** Add to and multiply a number, set to current value

#### > Input Ports:

- **Trigger In** (Trigger)
- **Add To Number** (Number)
- **Multiplier To Add Number** (Number)
- **Default Value** (Number)
- **Set Default Value** (Trigger)

#### < Output Ports:

- **Current Value** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Accumulator>

### 85.1.3 AddUp



**Full Name:** Ops.Math.AddUp

**Description:** Visit documentation for details

#### > Input Ports:

- **Number** (Number)
- **Add** (Trigger)
- **Reset** (Trigger)

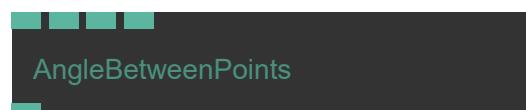
#### < Output Ports:

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.AddUp>

### 85.1.4 AngleBetweenPoints



**Full Name:** Ops.Math.AngleBetweenPoints

**Description:** outputs the angle between two points (degree)

#### > Input Ports:

- **Point 1 X** (Number)
- **Point 1 Y** (Number)
- **Point 2 X** (Number)
- **Point 2 Y** (Number)

#### < Output Ports:

- **Angle** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.AngleBetweenPoints>

## 85.1.5 Array3MultiplyMatrix



**Full Name:** Ops.Math.Array3MultiplyMatrix

**Description:** multiply every XYZ coordinate with a matrix

**> Input Ports:**

- **Update** (Trigger)
- **Array** (Array)
- **Matrix** (Array)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Array3MultiplyMatrix>

## 85.1.6 Array3To2dProjection



**Full Name:** Ops.Math.Array3To2dProjection

**Description:** calculate 2d positions of an array3x

**> Input Ports:**

- **Exec** (Trigger)
- **Array3x** (Array)
- **Fov** (Number)
- **W** (Number)
- **H** (Number)
- **Pos X** (Number)
- **Pos Y** (Number)

- **Mul** (Number)

**< Output Ports:**

- **Next** (Trigger)
- **Array2x** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Array3To2dProjection>

## 85.1.7 Atan2



**Full Name:** Ops.Math.Atan2

**Description:** Calculates the angle from a specified point to the coordinate origin.

**> Input Ports:**

- **X** (Number)
- **Y** (Number)
- **Phase** (Number)
- **Frequency** (Number)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Atan2>

## 85.1.8 Average



**Full Name:** Ops.Math.Average

**Description:** average of last two values

**> Input Ports:**

- **Number** (Number)

- **Influence** (Number)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Average>

## 85.1.9 ButterflyCurve



**Full Name:** Ops.Math.ButterflyCurve

**Description:** generate coordinates of a butterfly curve

**> Input Ports:**

- **Value** (Number)

**< Output Ports:**

- **X** (Number)
- **Y** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.ButterflyCurve>

## 85.1.10 Ceil



**Full Name:** Ops.Math.Ceil

**Description:** Returns the smallest integer greater than or equal to a given number

**> Input Ports:**

- **Number** (Number)

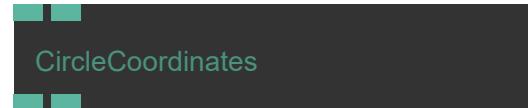
**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Ceil>

## 85.1.11 CircleCoordinates



**Full Name:** Ops.Math.CircleCoordinates

**Description:** x and y coordinates of a circle

**> Input Ports:**

- **Position** (Number)
- **Radius** (Number)

**< Output Ports:**

- **X** (Number)
- **Y** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.CircleCoordinates>

## 85.1.12 Clamp



**Full Name:** Ops.Math.Clamp

**Description:** Makes sure a value is within range cuts off the rest

**> Input Ports:**

- **Val** (Number)
- **Min** (Number)
- **Max** (Number)
- **Ignore Outside Values** (Number: Boolean)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Clamp>

### 85.1.13 Cosine



**Full Name:** Ops.Math.Cosine

**Description:** Calculates the cosine of an angle.

#### > Input Ports:

- **Value** (Number)
- **Phase** (Number)
- **Frequency** (Number)
- **Amplitude** (Number)
- **Asine** (Number: Boolean)

#### < Output Ports:

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Cosine>

### 85.1.14 Cross



**Full Name:** Ops.Math.Cross

**Description:** Computes the cross product of two vec3's

#### > Input Ports:

- **Exec** (Trigger)
- **X1** (Number)
- **Y1** (Number)
- **Z1** (Number)
- **X2** (Number)
- **Y2** (Number)
- **Z2** (Number)

#### < Output Ports:

- **Next** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Cross>

### 85.1.15 Degrees



**Full Name:** Ops.Math.Degrees

**Description:** Converts a radian measurement to its corresponding value in degrees.

#### > Input Ports:

- **Radians** (Number)

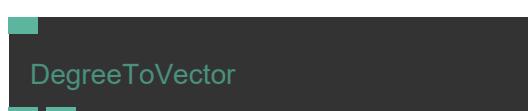
#### < Output Ports:

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Degrees>

### 85.1.16 DegreeToVector



**Full Name:** Ops.Math.DegreeToVector

**Description:** Calculates a vector (x and y) based on an angle in degrees

#### > Input Ports:

- **Degree** (Number)
- **The angle you want to convert** (in degrees)

#### < Output Ports:

- **X** (Number)
- **Y** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.DegreeToVector>

## 85.1.17 Delta



**Full Name:** Ops.Math.Delta

**Description:** difference to the last value (previous, store)

### > Input Ports:

- **Value** (Number)
- **Change Always** (Number: Boolean)
- **Reset** (Trigger)

### < Output Ports:

- **Delta** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Delta>

## 85.1.18 DeltaSum



**Full Name:** Ops.Math.DeltaSum

**Description:** add delta values to an clamped absolute value

### > Input Ports:

- **Delta Value** (Number)
- **Default Value** (Number)
- **Multiply** (Number)
- **Reset** (Trigger)
- **Limit** (Number: Boolean)

- **Min** (Number)
- **Max** (Number)
- **Rubberband** (Number)

### < Output Ports:

- **Absolute Value** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.DeltaSum>

## 85.1.19 Difference



**Full Name:** Ops.Math.Difference

**Description:** Difference between two numbers

### > Input Ports:

- **Number A** (Number)
- **Number B** (Number)

### < Output Ports:

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Difference>

## 85.1.20 Distance2d



**Full Name:** Ops.Math.Distance2d

**Description:** Calculates the Distance between two 2d points

### > Input Ports:

- **X1** (Number)
- **Y1** (Number)
- **X2** (Number)

- **Y2** (Number)

**< Output Ports:**

- **Distance** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Distance2d>

## 85.1.21 Distance3d\_v2



**Full Name:** Ops.Math.Distance3d\_v2

**Description:** distance between two 3d points, calculated when triggered

**> Input Ports:**

- **Calc** (Trigger)
- **X1** (Number)
- **Y1** (Number)
- **Z1** (Number)
- **X2** (Number)
- **Y2** (Number)
- **Z2** (Number)

**< Output Ports:**

- **Next** (Trigger)
- **Distance** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Math.Distance3d\\_v2](https://cables.gl/op/Ops.Math.Distance3d_v2)

## 85.1.22 Divide



**Full Name:** Ops.Math.Divide

**Description:** Divides a number by another

**> Input Ports:**

- **Number1** (Number)
- **Number2** (Number)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Divide>

## 85.1.23 Ease



**Full Name:** Ops.Math.Ease

**Description:** map a value to an easing curve

**> Input Ports:**

- **Value** (Number)
- **Min** (Number)
- **Max** (Number)
- **Easing Index** (Number: Integer)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Ease>

## 85.1.24 Exp



**Full Name:** Ops.Math.Exp

**Description:** Calculates the power of Euler's number

**> Input Ports:**

- **Number** (Number)

#### < Output Ports:

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Exp>

## 85.1.25 FlipSign



**Full Name:** Ops.Math.FlipSign

**Description:** positive numbers become negative and vice versa (negate)

#### > Input Ports:

- **Value** (Number)

#### < Output Ports:

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.FlipSign>

## 85.1.26 Floor



**Full Name:** Ops.Math.Floor

**Description:** returns the largest integer less than or equal to a given number

#### > Input Ports:

- **Number** (Number)

#### < Output Ports:

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Floor>

## 85.1.27 Fract



**Full Name:** Ops.Math.Fract

**Description:** returns the fractional part of a number

#### > Input Ports:

- **Value** (Number)

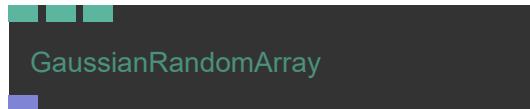
#### < Output Ports:

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Fract>

## 85.1.28 GaussianRandomArray



**Full Name:** Ops.Math.GaussianRandomArray

**Description:** random numbers fitting a Gaussian, or normal, distribution

#### > Input Ports:

- **Num** (Number: Integer)
- **Deviation** (Number)
- **Random Seed** (Number)

#### < Output Ports:

- **Array** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.GaussianRandomArray>

## 85.1.29 Incrementor



**Full Name:** Ops.Math.Incrementor

**Description:** increment a number by triggering

**> Input Ports:**

- **Increment** (Trigger)
- **Decrement** (Trigger)
- **Limit** (Number: Boolean)
- **Length** (Number: Integer)
- **Default** (Number: Integer)
- **Reset** (Trigger)

**< Output Ports:**

- **Changed** (Trigger)
- **Value** (Number)
- **Restarted** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Incrementor>

## 85.1.30 IndexFraction



**Full Name:** Ops.Math.IndexFraction

**Description:** return fraction of value by index

**> Input Ports:**

- **Number** (Number)
- **Index** (Number: Integer)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.IndexFraction>

## 85.1.31 Interpolate



**Full Name:** Ops.Math.Interpolate

**Description:** Interpolate between values, lerp, linear interpolate

**> Input Ports:**

- **Value 1** (Number)
- **Value 2** (Number)
- **Percentage** (Number)

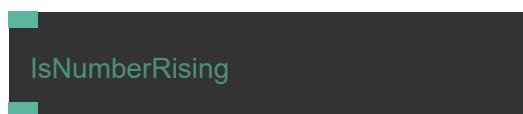
**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Interpolate>

## 85.1.32 IsNumberRising



**Full Name:** Ops.Math.IsNumberRising

**Description:** detect if a number rising or falling

**> Input Ports:**

- **Number** (Number)

**< Output Ports:**

- **Rising** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.IsNumberRising>

### 85.1.33 Log



**Full Name:** Ops.Math.Log

**Description:** Calculates the logarithm of Number

**> Input Ports:**

- **Number** (Number)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Log>

### 85.1.34 MapGeoCoordsSpherical



**Full Name:** Ops.Math.MapGeoCoordsSpherical

**Description:** map geo locations (latitude - longitude) to spherical coordinates

**> Input Ports:**

- **Coordinates** (Array)
- **Radius** (Number)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.MapGeoCoordsSpherical>

### 85.1.35 MapRange



**Full Name:** Ops.Math.MapRange

**Description:** Maps a value from one range into another.

**> Input Ports:**

- **Value** (Number)
- **Old Min** (Number)
- **Old Max** (Number)
- **New Min** (Number)
- **New Max** (Number)
- **Easing Index** (Number: Integer)
- **Clamp** (Number: Boolean)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.MapRange>

### 85.1.36 Math



**Full Name:** Ops.Math.Math

**Description:** Allows different mathematical operations to be applied to two numbers

**> Input Ports:**

- **Number 0** (Number)
- **Number 1** (Number)
- **Math Mode Index** (Number: Integer)

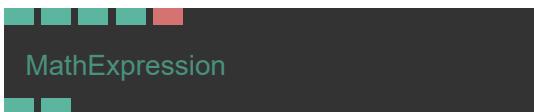
**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Math>

### 85.1.37 MathExpression



**Full Name:** Ops.Math.MathExpression

**Description:** calculates a user defined mathematical expression

**> Input Ports:**

- **A** (Number)
- **B** (Number)
- **C** (Number)
- **D** (Number)
- **Expression** (String)

**< Output Ports:**

- **Result** (Number)
- **Expression Valid** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.MathExpression>

### 85.1.38 Max



**Full Name:** Ops.Math.Max

**Description:** Sets the output to the input value which is higher

**> Input Ports:**

- **Value** (Number)
- **Maximum** (Number)

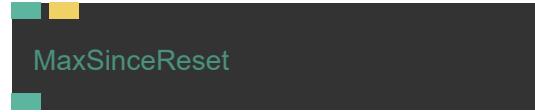
**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Max>

### 85.1.39 MaxSinceReset



**Full Name:** Ops.Math.MaxSinceReset

**Description:** Outputs the maximum value since reset has been triggered

**> Input Ports:**

- **Value** (Number)
- **Reset** (Trigger)

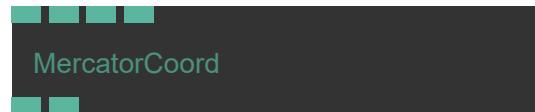
**< Output Ports:**

- **Maximum** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.MaxSinceReset>

### 85.1.40 MercatorCoord



**Full Name:** Ops.Math.MercatorCoord

**Description:** project mercator coordinates

**> Input Ports:**

- **Latitude** (Number)
- **Longitude** (Number)
- **MapWidth** (Number)
- **MapHeight** (Number)

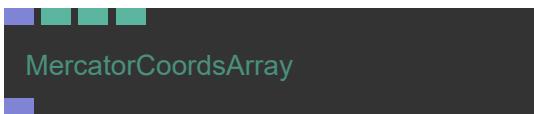
**< Output Ports:**

- **X** (Number)
- **Y** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.MercatorCoord>

## 85.1.41 MercatorCoordsArray



**Full Name:** Ops.Math.MercatorCoordsArray

**Description:** Mercator map and center an array of latitudes and longitudes to a local coordinate system

**> Input Ports:**

- **LatLon Array** (Array)
- **MapWidth** (Number)
- **Center Lat** (Number)
- **Center Lon** (Number)

**< Output Ports:**

- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.MercatorCoordsArray>

## 85.1.42 Min\_v3



**Full Name:** Ops.Math.Min\_v3

**Description:** Result will be the smaller number of the inputs

**> Input Ports:**

- **Value 1** (Number)
- **Value 2** (Number)

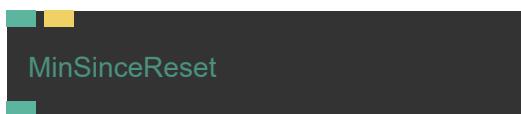
**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Math.Min\\_v3](https://cables.gl/op/Ops.Math.Min_v3)

## 85.1.43 MinSinceReset



**Full Name:** Ops.Math.MinSinceReset

**Description:** Outputs the minimum value since reset has been triggered

**> Input Ports:**

- **Value** (Number)
- **Reset** (Trigger)

**< Output Ports:**

- **Minimum** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.MinSinceReset>

## 85.1.44 Modulo



**Full Name:** Ops.Math.Modulo

**Description:** outputs the remainder after division of one number by another

**> Input Ports:**

- **Number1** (Number)
- **Number2** (Number)
- **Pingpong** (Number: Boolean)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Modulo>

## 85.1.45 MulMatrixXyz



**Full Name:** Ops.Math.MulMatrixXyz

**Description:** multiply XYZ values with a gl matrix vec3 x mat4

**> Input Ports:**

- **Update** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Matrix** (Array)

**< Output Ports:**

- **Next** (Trigger)
- **Result X** (Number)
- **Result Y** (Number)
- **Result Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.MulMatrixXyz>

## 85.1.46 Multiply



**Full Name:** Ops.Math.Multiply

**Description:** Multiplies two numbers

**> Input Ports:**

- **Number1** (Number)
- **Number2** (Number)

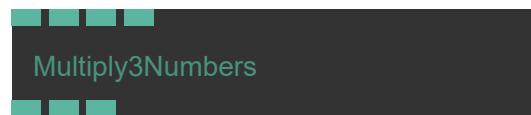
**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Multiply>

## 85.1.47 Multiply3Numbers



**Full Name:** Ops.Math.Multiply3Numbers

**Description:** multiply three numbers

**> Input Ports:**

- **R** (Number)
- **G** (Number)
- **B** (Number)
- **Multiply** (Number)

**< Output Ports:**

- **ResultR** (Number)
- **ResultG** (Number)
- **ResultB** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Multiply3Numbers>

## 85.1.48 Normalize



**Full Name:** Ops.Math.Normalize

**Description:** normalize a vector

**> Input Ports:**

- **X** (Number)
- **Y** (Number)
- **Z** (Number)

**< Output Ports:**

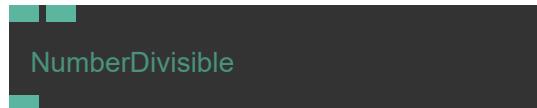
- **Result X** (Number)

- **Result Y** (Number)
- **Result Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Normalize>

## 85.1.49 NumberDivisible



**Full Name:** Ops.Math.NumberDivisible

**Description:** is a number capable of being divided.

**> Input Ports:**

- **Number** (Number)
- **Divisor** (Number)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.NumberDivisible>

## 85.1.50 OneMinus



**Full Name:** Ops.Math.OneMinus

**Description:** subtract a number from one

**> Input Ports:**

- **Value** (Number)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.OneMinus>

## 85.1.51 PerlinNoise\_v2



**Full Name:** Ops.Math.PerlinNoise\_v2

**Description:** outputs a perlin noise value like random

**> Input Ports:**

- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **Scale** (Number)
- **Seed** (Number)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Math.PerlinNoise\\_v2](https://cables.gl/op/Ops.Math.PerlinNoise_v2)

## 85.1.52 Pi



**Full Name:** Ops.Math.Pi

**Description:** returns PI (3.141592653589793) \* multiply amount

**> Input Ports:**

- **Multiply Amount** (Number)

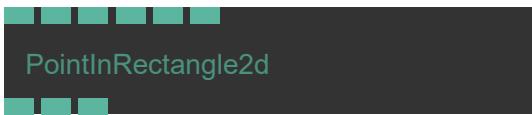
**< Output Ports:**

- **Pi** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Pi>

## 85.1.53 PointInRectangle2d



**Full Name:** Ops.Math.PointInRectangle2d

**Description:** test if a point is in or outside of a rectangle

**> Input Ports:**

- **X** (Number)
- **Y** (Number)
- **Rect Top** (Number)
- **Rect Left** (Number)
- **Rect Right** (Number)
- **Rect Bottom** (Number)

**< Output Ports:**

- **Result** (Number)
- **Pos X** (Number)
- **Pos Y** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.PointInRectangle2d>

## 85.1.54 Pow



**Full Name:** Ops.Math.Pow

**Description:** value of x to the power of y

**> Input Ports:**

- **Base** (Number)
- **Exponent** (Number)

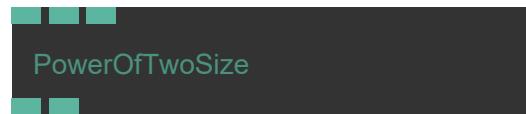
**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Pow>

## 85.1.55 PowerOfTwoSize



**Full Name:** Ops.Math.PowerOfTwoSize

**Description:** Return the next values as power of two

**> Input Ports:**

- **Width** (Number: Integer)
- **Height** (Number: Integer)
- **Strategy Index** (Number: Integer)

**< Output Ports:**

- **Width Result** (Number)
- **Height Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.PowerOfTwoSize>

## 85.1.56 Radians



**Full Name:** Ops.Math.Radians

**Description:** Converts a degree measurement to its corresponding value in radians.

**> Input Ports:**

- **Degrees** (Number)

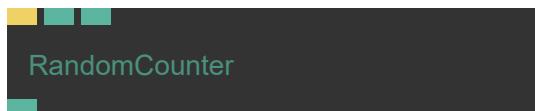
**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Radians>

## 85.1.57 RandomCounter



**Full Name:** Ops.Math.RandomCounter

**Description:** add up random numbers by triggering

**> Input Ports:**

- **Count** (Trigger)
- **Step Min** (Number)
- **Step Max** (Number)

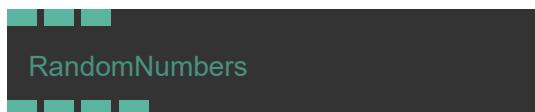
**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.RandomCounter>

## 85.1.58 RandomNumbers\_v3



**Full Name:** Ops.Math.RandomNumbers\_v3

**Description:** Simple way to get random numbers without using arrays

**> Input Ports:**

- **Seed** (Number)
- **Min** (Number)
- **Max** (Number)

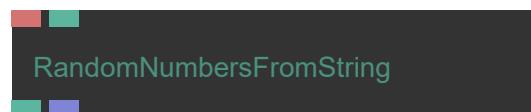
**< Output Ports:**

- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **W** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Math.RandomNumbers\\_v3](https://cables.gl/op/Ops.Math.RandomNumbers_v3)

## 85.1.59 RandomNumbersFromString



**Full Name:** Ops.Math.RandomNumbersFromString

**Description:** Random number generator from a string seed

**> Input Ports:**

- **Input String** (String)
- **Random Number Count** (Number: Integer)

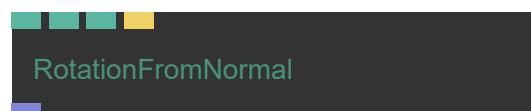
**< Output Ports:**

- **Random Value** (Number)
- **Random Numbers** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.RandomNumbersFromString>

## 85.1.60 RotationFromNormal



**Full Name:** Ops.Math.RotationFromNormal

**Description:** Create rotation matrix from normal

**> Input Ports:**

- **Normal X** (Number)
- **Normal Y** (Number)
- **Normal Z** (Number)
- **Recalculate** (Trigger)

**< Output Ports:**

- **RotationMatrix** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.RotationFromNormal>

## 85.1.61 Round



**Full Name:** Ops.Math.Round

**Description:** Outputs number rounded to the nearest integer

**> Input Ports:**

- **Number** (Number)
- **Decimal Places** (Number: Integer)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Round>

## 85.1.62 RoundEven



**Full Name:** Ops.Math.RoundEven

**Description:** round to the next even number

**> Input Ports:**

- **Number** (Number)
- **Mode Index** (Number: Integer)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.RoundEven>

## 85.1.63 SchlickBias



**Full Name:** Ops.Math.SchlickBias

**Description:** Custom easing curve via schlick bias and gain

**> Input Ports:**

- **Value** (Number)
- **Gain** (Number)
- **Bias** (Number)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.SchlickBias>

## 85.1.64 Sign



**Full Name:** Ops.Math.Sign

**Description:** get sign of value

**> Input Ports:**

- **Value** (Number)
- **Remove Zero** (Number: Boolean)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Sign>

## 85.1.65 SimpleMovingAverage



**Full Name:** Ops.Math.SimpleMovingAverage

**Description:** Calculate the Average of the last X values

**> Input Ports:**

- **Value** (Number)
- **Number Of Values** (Number: Integer)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.SimpleMovingAverage>

## 85.1.66 Sine



**Full Name:** Ops.Math.Sine

**Description:** Calculates the sine of an angle.

**> Input Ports:**

- **Value** (Number)
- **Phase** (Number)
- **Frequency** (Number)
- **Amplitude** (Number)
- **Asine** (Number: Boolean)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Sine>

## 85.1.67 Speed



**Full Name:** Ops.Math.Speed

**Description:** measure speed of how much a value changes

**> Input Ports:**

- **Update** (Trigger)
- **Value** (Number)

**< Output Ports:**

- **Speed** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Speed>

## 85.1.68 Sqrt



**Full Name:** Ops.Math.Sqrt

**Description:** square root of a number

**> Input Ports:**

- **Number** (Number)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Sqrt>

## 85.1.69 Subtract



**Full Name:** Ops.Math.Subtract

**Description:** Subtracts Number2 from Number1 (minus, -)

**> Input Ports:**

- **Number1** (Number)
- **Number2** (Number)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Subtract>

## 85.1.70 Sum



**Full Name:** Ops.Math.Sum

**Description:** Result of the addition

**> Input Ports:**

- **Number1** (Number)
- **Number2** (Number)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Sum>

## 85.1.71 Tangent



**Full Name:** Ops.Math.Tangent

**Description:** Calculates the ratio of the sine and cosine of an angle.

**> Input Ports:**

- **Value** (Number)
- **Phase** (Number)
- **Frequency** (Number)
- **Amplitude** (Number)
- **Asine** (Number: Boolean)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Tangent>

## 85.1.72 TriggerMathExpression



**Full Name:** Ops.Math.TriggerMathExpression

**Description:** calculates a user defined mathematical expression

**> Input Ports:**

- **Calculate** (Trigger)
- **Expression** (String)
- **X** (Number)
- **Y** (Number)
- **Z** (Number)
- **W** (Number)
- **A** (Number)
- **B** (Number)
- **C** (Number)
- **D** (Number)
- **I** (Number)

**< Output Ports:**

- **Next** (Trigger)
- **Result** (Number)
- **Expression Valid** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.TriggerMathExpression>

## 85.1.73 TriggerRandomNumber\_v3



**Full Name:** Ops.Math.TriggerRandomNumber\_v3

**Description:** Generate random number between min and max

**> Input Ports:**

- **Generate** (Trigger)
- **Min** (Number)
- **Max** (Number)
- **Integer** (Number: Boolean)
- **No Consecutive Duplicates** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)
- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Math.TriggerRandomNumber\\_v3](https://cables.gl/op/Ops.Math.TriggerRandomNumber_v3)

## 85.1.74 VectorLength



**Full Name:** Ops.Math.VectorLength

**Description:** length of a vector

**> Input Ports:**

- **X** (Number)
- **Y** (Number)
- **Z** (Number)

**< Output Ports:**

- **Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.VectorLength>

# 86 Ops.Math.Compare

## 86.1 Ops.Math.Compare

### 86.1.1 Between



**Full Name:** Ops.Math.Compare.Between

**Description:** result is true if value is between number1 and number2

**> Input Ports:**

- **Value** (Number)
- **Number1** (Number)
- **Number2** (Number)

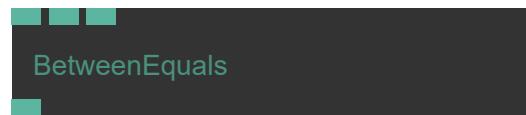
**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Compare.Between>

### 86.1.2 BetweenEquals



**Full Name:** Ops.Math.Compare.BetweenEquals

**Description:** result is true if value is between or equal number1 and number2

**> Input Ports:**

- **Value** (Number)
- **Range 1** (Number)
- **Range 2** (Number)

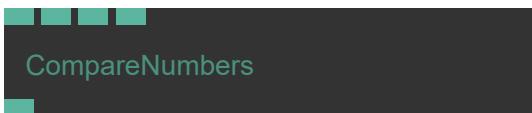
**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Compare.BetweenEquals>

### 86.1.3 CompareNumbers



**Full Name:** Ops.Math.Compare.CompareNumbers

**Description:** Performs logical comparisons on numbers (compare, operators)

**> Input Ports:**

- **Value In** (Number)
- **Comparison Mode Index** (Number: Integer)
- **Condition Value** (Number)
- **Max** (Number)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Compare.CompareNumbers>

### 86.1.4 Equals



**Full Name:** Ops.Math.Compare.Equals

**Description:** result is true if number1 and number2 are equal

**> Input Ports:**

- **Number1** (Number)
- **Number2** (Number)

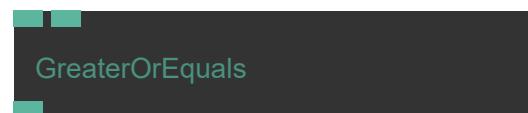
**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Compare.Equals>

### 86.1.5 GreaterOrEquals



**Full Name:** Ops.Math.Compare.GreaterOrEquals

**Description:** result is true if number 1 is greater or equals number 2

**> Input Ports:**

- **Number1** (Number)
- **Number2** (Number)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Compare.GreaterOrEquals>

### 86.1.6 GreaterThan



**Full Name:** Ops.Math.Compare.GreaterThan

**Description:** result is true if number1 is greater than number2

**> Input Ports:**

- **Number1** (Number)
- **Number2** (Number)

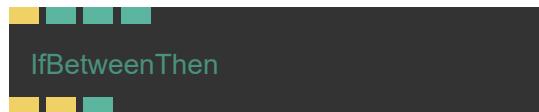
**< Output Ports:**

- **Result** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Compare.GreaterThan>

## 86.1.7 IfBetweenThen



**Full Name:** Ops.Math.Compare.IfBetweenThen

**Description:** triggers when value is between min and max

**> Input Ports:**

- **Exe** (Trigger)
- **Number** (Number)
- **Min** (Number)
- **Max** (Number)

**< Output Ports:**

- **Then** (Trigger)
- **Else** (Trigger)
- **Bs Between** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Compare.IfBetweenThen>

## 86.1.8 IsEven



**Full Name:** Ops.Math.Compare.IsEven

**Description:** Checks if Value is even or not

**> Input Ports:**

- **Number1** (Number)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Compare.IsEven>

## 86.1.9 LessThan



**Full Name:** Ops.Math.Compare.LessThan

**Description:** Is n1 smaller than n2? (lesser, less)

**> Input Ports:**

- **Number1** (Number)
- **Number2** (Number)

**< Output Ports:**

- **Result** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Math.Compare.LessThan>

# 87 Ops.Net

## 87.1 Ops.Net

### 87.1.1 CorsProxy\_v3



**Full Name:** Ops.Net.CorsProxy\_v3

**Description:** create a cables.gl CORS proxy URL

#### > Input Ports:

- URL (String)
- Use In Export (Number: Boolean)
- Active (Number: Boolean)

#### < Output Ports:

- CORS URL (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Net.CorsProxy\\_v3](https://cables.gl/op/Ops.Net.CorsProxy_v3)

# 88 Ops.Net.WebSocket

## 88.1 Ops.Net.WebSocket

### 88.1.1 WebSocket\_v2



**Full Name:** Ops.Net.WebSocket.WebSocket\_v2

**Description:** Create a websocket connection and receive data from it

#### > Input Ports:

- URL (String)

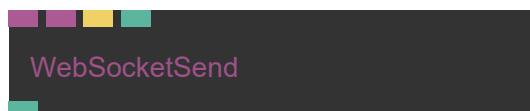
#### < Output Ports:

- Result (Object)
- Valid JSON (booleanNumber)
- Connection (Object)
- Connected (booleanNumber)
- Received Data (Trigger)
- Raw Data (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Net.WebSocket.WebSocket\\_v2](https://cables.gl/op/Ops.Net.WebSocket.WebSocket_v2)

### 88.1.2 WebSocketSend



**Full Name:** Ops.Net.WebSocket.WebSocketSend

**Description:** send an object to a websocket connection

#### > Input Ports:

- Connection (Object:Websocket)
- Object (Object)

- **Send** (Trigger)
- **Send String** (Number: Boolean)

◀ **Output Ports:**

- **Sent** (Number)

**Example Patch:** Open in Editor

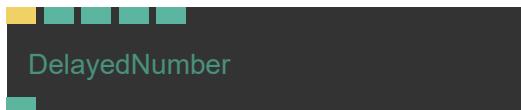
**Docs:** <https://cables.gl/op/Ops.Net.WebSocket.WebSocketSend>

## 89 Ops.Number

---

### 89.1 Ops.Number

#### 89.1.1 DelayedNumber



**Full Name:** Ops.Number.DelayedNumber

**Description:** delay a value by seconds

➢ **Input Ports:**

- **Update** (Trigger)
- **Value** (Number)
- **Delay** (Number)
- **Clear On Change** (Number: Boolean)
- **Easing Index** (Number: Integer)

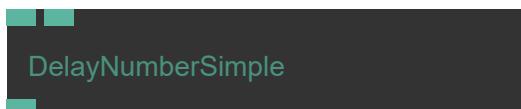
◀ **Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Number.DelayedNumber>

#### 89.1.2 DelayNumberSimple



**Full Name:** Ops.Number.DelayNumberSimple

**Description:** delay the value data flow by x seconds

➢ **Input Ports:**

- **Value** (Number)
- **Delay** (Number)

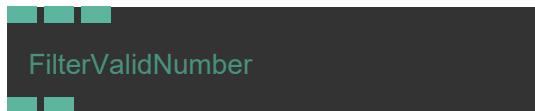
◀ **Output Ports:**

- **Out Value** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Number.DelayNumberSimple>

### 89.1.3 FilterValidNumber



**Full Name:** Ops.Number.FilterValidNumber

**Description:** Filter valid numbers

➢ **Input Ports:**

- **Number** (Number)
- **Invalid When 0** (Number: Boolean)

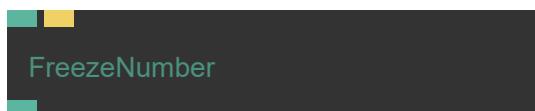
◀ **Output Ports:**

- **Last Valid Number** (Number)
- **Is Valid** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Number.FilterValidNumber>

### 89.1.4 FreezeNumber



**Full Name:** Ops.Number.FreezeNumber

**Description:** capture the current input and copy it to the output, even after a reload

➢ **Input Ports:**

- **Number** (Number)
- **Button** (Trigger)

◀ **Output Ports:**

- **Frozen Number** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Number.FreezeNumber>

### 89.1.5 GateNumber



**Full Name:** Ops.Number.GateNumber

**Description:** Let's a number through only if control bool is true, like a gate

➢ **Input Ports:**

- **Value In** (Number)
- **Pass Through** (Number: Boolean)
- **Custom Value** (Number)

◀ **Output Ports:**

- **Value Out** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Number.GateNumber>

### 89.1.6 Integer



**Full Name:** Ops.Number.Integer

**Description:** Number op which only outputs integers

➢ **Input Ports:**

- **Integer** (Number: Integer)

◀ **Output Ports:**

- **Number Out** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Number.Integer>

## 89.1.7 MaximumSafeInteger

MaximumSafeInteger

**Full Name:** Ops.Number.MaximumSafeInteger

**Description:** Returns the maximum safe integer (number, constant)

**> Input Ports:**

- Visit *Ops.Number.MaximumSafeInteger documentation for input port details*

**< Output Ports:**

- Max Int (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Number.MaximumSafeInteger>

## 89.1.8 MinimumSafeInteger

MinimumSafeInteger

**Full Name:** Ops.Number.MinimumSafeInteger

**Description:** Returns the minimum safe integer (number, constant)

**> Input Ports:**

- Visit *Ops.Number.MinimumSafeInteger documentation for input port details*

**< Output Ports:**

- Min Int (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Number.MinimumSafeInteger>

## 89.1.9 Number

Number

**Full Name:** Ops.Number.Number

**Description:** Stores a value, use the same value in different places (was: value.value)

**> Input Ports:**

- Value (Number)

**< Output Ports:**

- Result (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Number.Number>

## 89.1.10 NumberSequence

NumberSequence

**Full Name:** Ops.Number.NumberSequence

**Description:** Copies the input value to the (value sequence)

**> Input Ports:**

- In Value (Number)

**< Output Ports:**

- In Value (Number)
- Value Changed (Trigger)
- Out Value 0 (Number)
- Out Value 1 (Number)
- Out Value 2 (Number)
- Out Value 3 (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Number.NumberSequence>

## 89.1.11 Preset



**Full Name:** Ops.Number.Preset

**Description:** State management of all parameters connected to it - Create presets of multiple ops

**> Input Ports:**

- **Data** (String)
- **Sets** (String)
- **Presetid** (String)
- **Interpolation Index** (Number: Integer)
- **Interpolation** (String)
- **Preset A** (Number)
- **Preset B** (Number)
- **Fade** (Number)
- **Preset Index** (Number: Integer)
- **Preset** (Number: String)
- **Create New** (Trigger)
- **Update** (Trigger)
- **Move** (Trigger)
- **Delete** (Trigger)
- **Rename** (Trigger)

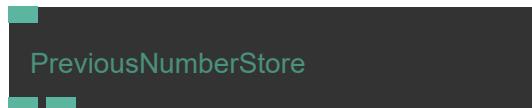
**< Output Ports:**

- **Create Variable** (Dynamic)
- **Num Presets** (Number)
- **Current Preset** (Number)
- **Dbg\_data** (Array)
- **Dbg\_sets** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Number.Preset>

## 89.1.12 PreviousNumberStore



**Full Name:** Ops.Number.PreviousNumberStore

**Description:** remember/store last set number

**> Input Ports:**

- **Value** (Number)

**< Output Ports:**

- **Current Value** (Number)
- **Previous Value** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Number.PreviousNumberStore>

## 89.1.13 RouteNumber



**Full Name:** Ops.Number.RouteNumber

**Description:** Routes the value to one of the (based on index, relay)

**> Input Ports:**

- **Index** (Number: Integer)
- **Value** (Number)

**< Output Ports:**

- **Index** (Number: Integer)
- **Value** (Number)
- **Default VaonlyOnePortlue** (Number)
- **Set Inactive To Default** (Number: Boolean)
- **Index 0 Value** (Number)
- **Index 1 Value** (Number)
- **Index 2 Value** (Number)
- **Index 3 Value** (Number)
- **Index 4 Value** (Number)

- **Index 5 Value** (Number)
- **Index 6 Value** (Number)
- **Index 7 Value** (Number)
- **Index 8 Value** (Number)
- **Index 9 Value** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Number.RouteNumber>

### 89.1.14 SequenceNumbers



**Full Name:** Ops.Number.SequenceNumbers

**Description:** control order and flow of numbers

**> Input Ports:**

- **Number 0** (Number)
- **Number 1** (Number)
- **Number 2** (Number)
- **Number 3** (Number)
- **Number 4** (Number)
- **Number 5** (Number)
- **Number 6** (Number)
- **Number 7** (Number)
- **Number 8** (Number)
- **Number 9** (Number)
- **Number 10** (Number)
- **Number 11** (Number)
- **Number 12** (Number)
- **Number 13** (Number)
- **Number 14** (Number)
- **Number 15** (Number)

**< Output Ports:**

- **Output 0** (Number)
- **Output 1** (Number)
- **Output 2** (Number)
- **Output 3** (Number)

- **Output 4** (Number)
- **Output 5** (Number)
- **Output 6** (Number)
- **Output 7** (Number)
- **Output 8** (Number)
- **Output 9** (Number)
- **Output 10** (Number)
- **Output 11** (Number)
- **Output 12** (Number)
- **Output 13** (Number)
- **Output 14** (Number)
- **Output 15** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Number.SequenceNumbers>

### 89.1.15 SumMultiPort\_v2



**Full Name:** Ops.Number.SumMultiPort\_v2

**Description:** Switch between multiple number inputs

**> Input Ports:**

- **Numbers\_0** (Number)
- **Add Port** (Number)

**< Output Ports:**

- **Number** (Number)
- **Num Values** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Number.SumMultiPort\\_v2](https://cables.gl/op/Ops.Number.SumMultiPort_v2)

## 89.1.16 SwitchNumber



**Full Name:** Ops.Number.SwitchNumber

**Description:** switch between number values by index

### > Input Ports:

- **Index** (Number: Integer)
- **Value 0** (Number)
- **Value 1** (Number)
- **Value 2** (Number)
- **Value 3** (Number)
- **Value 4** (Number)
- **Value 5** (Number)
- **Value 6** (Number)
- **Value 7** (Number)
- **Value 8** (Number)
- **Value 9** (Number)
- **Value 10** (Number)
- **Value 11** (Number)
- **Value 12** (Number)
- **Value 13** (Number)
- **Value 14** (Number)
- **Value 15** (Number)

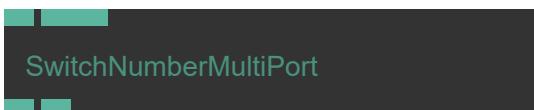
### < Output Ports:

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Number.SwitchNumber>

## 89.1.17 SwitchNumberMultiPort\_v2



**Full Name:** Ops.Number.SwitchNumberMultiPort\_v2

**Description:** Switch between multiple number inputs

### > Input Ports:

- **Index** (Number: Integer)
- **Numbers\_0** (Number)
- **Add Port** (Number)

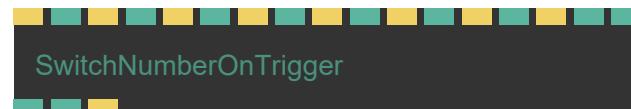
### < Output Ports:

- **Number** (Number)
- **Num Values** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Number.SwitchNumberMultiPort\\_v2](https://cables.gl/op/Ops.Number.SwitchNumberMultiPort_v2)

## 89.1.18 SwitchNumberOnTrigger



**Full Name:** Ops.Number.SwitchNumberOnTrigger

**Description:** Sets a specific output value on trigger

### > Input Ports:

- **Trigger 0** (Trigger)
- **Value 0** (Number)
- **Trigger 1** (Trigger)
- **Value 1** (Number)
- **Trigger 2** (Trigger)
- **Value 2** (Number)
- **Trigger 3** (Trigger)
- **Value 3** (Number)
- **Trigger 4** (Trigger)
- **Value 4** (Number)
- **Trigger 5** (Trigger)
- **Value 5** (Number)
- **Trigger 6** (Trigger)
- **Value 6** (Number)
- **Trigger 7** (Trigger)
- **Value 7** (Number)

- **Default Value** (Number: String)

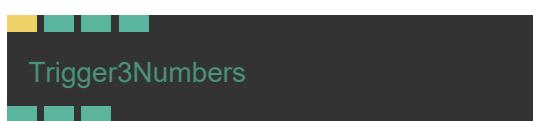
**< Output Ports:**

- **Value** (Number)
- **Last Value** (Number)
- **Triggered** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Number.SwitchNumberOnTrigger>

## 89.1.20 TriggerOnChangeNumber\_v2



**Full Name:** Ops.Number.Trigger3Numbers

**Description:** Stores a 3D coordinate (was Value3)

**> Input Ports:**

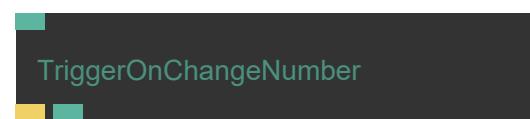
- **Exe** (Trigger)
- **Value X** (Number)
- **Value Y** (Number)
- **Value Z** (Number)

**< Output Ports:**

- **Exe** (Trigger)
- **Value X** (Number)
- **Value Y** (Number)
- **Value Z** (Number)
- **Result X** (Number)
- **Result Y** (Number)
- **Result Z** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Number.Trigger3Numbers>



**Full Name:** Ops.Number.TriggerOnChangeNumber\_v2

**Description:** triggers every time the input value changed

**> Input Ports:**

- **Value** (Number)

**< Output Ports:**

- **Next** (Trigger)
- **Number** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Number.TriggerOnChangeNumber\\_v2](https://cables.gl/op/Ops.Number.TriggerOnChangeNumber_v2)

# 90 OpsSidebar

## 90.1 OpsSidebar

### 90.1.1 Button\_v2



**Full Name:** OpsSidebar.Button\_v2

**Description:** sidebar push button/trigger element

#### > Input Ports:

- **Link** (Object)
- **Text** (String)
- **Grey Out** (Number: Boolean)
- **Visible** (Number: Boolean)

#### < Output Ports:

- **Childs** (Object)
- **Pressed Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/OpsSidebar.Button\\_v2](https://cables.gl/op/OpsSidebar.Button_v2)

### 90.1.2 ColorPicker\_v3



**Full Name:** OpsSidebar.ColorPicker\_v3

**Description:** Shows a color-picker in the sidebar

#### > Input Ports:

- **Link** (Object)
- **Text** (String)
- **Input Red** (Number)

- **Input Green** (Number)
- **Input Blue** (Number)
- **Input Opacity** (Number)
- **Set Default** (Trigger)
- **Show Opacity** (Number: Boolean)

#### < Output Ports:

- **Children** (Object)
- **Red** (Number)
- **Green** (Number)
- **Blue** (Number)
- **Opacity** (Number)
- **Hex** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/OpsSidebar.ColorPicker\\_v3](https://cables.gl/op/OpsSidebar.ColorPicker_v3)

### 90.1.3 DisplayValue\_v2



**Full Name:** OpsSidebar.DisplayValue\_v2

**Description:** display a value or string

#### > Input Ports:

- **Link** (Object)
- **Text** (String)
- **Value** (String)

#### < Output Ports:

- **Childs** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/OpsSidebar.DisplayValue\\_v2](https://cables.gl/op/OpsSidebar.DisplayValue_v2)

## 90.1.4 DropDown\_v2



**Full Name:** Ops.Sidebar.DropDown\_v2

**Description:** Shows a drop-down (select) element in the sidebar

### > Input Ports:

- **Link** (Object)
- **Text** (String)
- **Values** (Array)
- **Grey Out** (Number: Boolean)
- **Visible** (Number: Boolean)
- **Multiple Selection** (Number: Boolean)
- **Lines** (Number: Integer)
- **Set Default** (Trigger)

### < Output Ports:

- **Children** (Object)
- **Result** (String)
- **Index** (Number)
- **Selected Values** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops Sidebar.DropDown\\_v2](https://cables.gl/op/Ops Sidebar.DropDown_v2)

## 90.1.5 Group



**Full Name:** Ops.Sidebar.Group

**Description:** organize sidebar elements into groups

### > Input Ports:

- **Link** (Object)
- **Text** (String)
- **Show Title** (Number: Boolean)

- **Default Minimized** (Number: Boolean)

- **Visible** (Number: Boolean)

### < Output Ports:

- **Next** (Object)
- **Childs** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops Sidebar.Group>

## 90.1.6 Incrementor\_v3



**Full Name:** Ops.Sidebar.Incrementor\_v3

**Description:** steps through numerical or array values one by one

### > Input Ports:

- **Link** (Object)
- **Label** (String)
- **Min** (Number)
- **Max** (Number)
- **Stepsize** (Number)
- **Default** (Number)
- **Grey Out** (Number: Boolean)
- **Increment** (Trigger)
- **Decrement** (Trigger)
- **Set Default** (Trigger)
- **Reset** (Trigger)

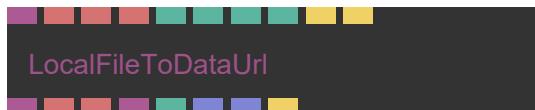
### < Output Ports:

- **Childs** (Object)
- **Value** (Number)
- **Changed** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops Sidebar.Incrementor\\_v3](https://cables.gl/op/Ops Sidebar.Incrementor_v3)

## 90.1.7 LocalFileToDataUrl



**Full Name:** Ops.Sidebar.LocalFileToDataUrl

**Description:** load a local file and output as data url

### > Input Ports:

- **Link** (Object)
- **Text** (String)
- **Button Text** (String)
- **Accept Files** (String)
- **Allow Multiple Files** (Number: Boolean)
- **Id** (Number: String)
- **Visible** (Number: Boolean)
- **Grey Out** (Number: Boolean)
- **Show Dialog** (Trigger)
- **Reset** (Trigger)

### < Output Ports:

- **Childs** (Object)
- **Data URL** (String)
- **Filename** (String)
- **File Object** (Object)
- **Num Files** (Number)
- **Data URLs** (Array)
- **Filenames** (Array)
- **File Changed** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/OpsSidebar.LocalFileToDataUrl>

## 90.1.8 NumberInput\_v2



**Full Name:** Ops.Sidebar.NumberInput\_v2

**Description:** Enter a number in the sidebar

### > Input Ports:

- **Link** (Object)
- **Text** (String)
- **Set Default** (Trigger)

### < Output Ports:

- **Children** (Object)
- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/OpsSidebar.NumberInput\\_v2](https://cables.gl/op/OpsSidebar.NumberInput_v2)

## 90.1.9 Presets\_v2



**Full Name:** Ops.Sidebar.Presets\_v2

**Description:** manage sidebar presets

### > Input Ports:

- **Link** (Object)
- **Text** (String)
- **Add Preset** (Trigger)
- **Update Current Preset** (Trigger)
- **Preset Title 0** (String)
- **Preset 0** (Object)
- **Preset Title 1** (String)
- **Preset 1** (Object)
- **Preset Title 2** (String)
- **Preset 2** (Object)
- **Preset Title 3** (String)
- **Preset 3** (Object)
- **Preset Title 4** (String)
- **Preset 4** (Object)
- **Preset Title 5** (String)
- **Preset 5** (Object)

- **Preset Title 6** (String)
- **Preset 6** (Object)
- **Preset Title 7** (String)
- **Preset 7** (Object)

**< Output Ports:**

- **Children** (Object)
- **Index** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops Sidebar.Presets\\_v2](https://cables.gl/op/Ops Sidebar.Presets_v2)

## 90.1.10 Sidebar



**Full Name:** Ops Sidebar Sidebar

**Description:** Sidebar overlay to control values

**> Input Ports:**

- **Visible** (Number: Boolean)
- **Opacity** (Number)
- **Default Minimized** (Number: Boolean)
- **Minimized Opacity** (Number)
- **Show Undo Button** (Number: Boolean)
- **Show Minimize** (Number: Boolean)
- **Title** (String)
- **Side** (Number: Boolean)
- **Default CSS** (Number: Boolean)

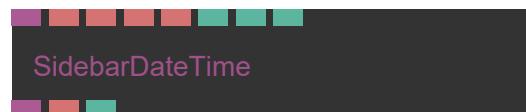
**< Output Ports:**

- **Childs** (Object)
- **Opened** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops Sidebar Sidebar>

## 90.1.11 SidebarDateTime



**Full Name:** Ops Sidebar SidebarDateTime

**Description:** date or datetime picker in the sidebar

**> Input Ports:**

- **Link** (Object)
- **Text** (String)
- **Default** (String)
- **Min** (String)
- **Max** (String)
- **Type Index** (Number: Integer)
- **Grey Out** (Number: Boolean)
- **Visible** (Number: Boolean)

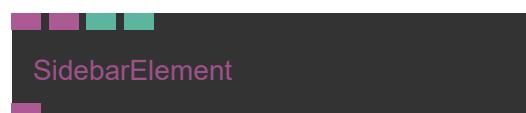
**< Output Ports:**

- **Children** (Object)
- **Result** (String)
- **Focus** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops Sidebar SidebarDateTime>

## 90.1.12 SidebarElement



**Full Name:** Ops Sidebar SidebarElement

**Description:** Add custom HTML Elements into the sidebar

**> Input Ports:**

- **Link** (Object)
- **Child Element** (Object)
- **Border** (Number: Boolean)
- **Visible** (Number: Boolean)

#### < Output Ports:

- **Childs** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops Sidebar SidebarElement>

### 90.1.13 SideBarImage



**Full Name:** Ops Sidebar SidebarImage

**Description:** Display an image in the sidebar

#### > Input Ports:

- **Link** (Object)
- **File** (String)

#### < Output Ports:

- **Childs** (Object)
- **Image Element** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops Sidebar SideBarImage>

### 90.1.14 SideBarStyle



**Full Name:** Ops Sidebar SidebarStyle

**Description:** adjust appearance of sidebar

#### > Input Ports:

- **Link** (Object)
- **Width** (Number: Integer)
- **Round Corners** (Number)
- **Special Color** (String)

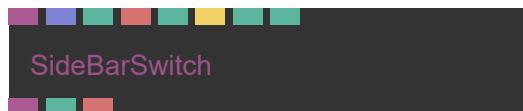
#### < Output Ports:

- **Childs** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops Sidebar SidebarStyle>

### 90.1.15 SideBarSwitch



**Full Name:** Ops Sidebar SideBarSwitch

**Description:** add tabs or switchbar to a sidebar

#### > Input Ports:

- **Link** (Object)
- **Names** (Array)
- **Text** (String)
- **Set Default** (Trigger)
- **Grey Out** (Number: Boolean)
- **Default** (Number)

#### < Output Ports:

- **Childs** (Object)
- **Index** (Number)
- **String** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops Sidebar SideBarSwitch>

### 90.1.16 SidebarText\_v3



**Full Name:** Ops Sidebar SidebarText\_v3

**Description:** Display text in the sidebar

#### > Input Ports:

- **Link** (Object)

- **Text** (String)
- **Id** (String)
- **Visible** (Number: Boolean)

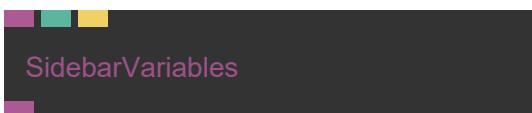
**< Output Ports:**

- **Childs** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops Sidebar SidebarText\\_v3](https://cables.gl/op/Ops Sidebar SidebarText_v3)

## 90.1.17 SidebarVariables



**Full Name:** Ops Sidebar SidebarVariables

**Description:** show values of all variables in a sidebar

**> Input Ports:**

- **Link** (Object)
- **Id** (Number: String)
- **Update** (Trigger)

**< Output Ports:**

- **Childs** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops Sidebar SidebarVariables>

## 90.1.18 Slider\_v3



**Full Name:** Ops Sidebar Slider\_v3

**Description:** Sidebar slider element (range)

**> Input Ports:**

- **Link** (Object)
- **Text** (String)

- **Min** (Number)
- **Max** (Number)
- **Step** (Number)
- **Suffix** (String)
- **Grey Out** (Number: Boolean)
- **Visible** (Number: Boolean)
- **Set Default** (Trigger)
- **Reset** (Trigger)

**< Output Ports:**

- **Childs** (Object)
- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops Sidebar Slider\\_v3](https://cables.gl/op/Ops Sidebar Slider_v3)

## 90.1.19 TextInput\_v2



**Full Name:** Ops Sidebar TextInput\_v2

**Description:** Get a string from an sidebar input field

**> Input Ports:**

- **Link** (Object)
- **Where to attach the sidebar item to** (Sidebar / Sidebar Group)
- **Text** (String)
- **Default** (String)
- **Placeholder** (String)
- **TextArea** (Number: Boolean)
- **Grey Out** (Number: Boolean)
- **Visible** (Number: Boolean)
- **Spellcheck** (Number: Boolean)
- **Enter Key Prevent Default** (Number: Boolean)
- **Clear** (Trigger)
- **Focus Input** (Trigger)

**< Output Ports:**

- **Children** (Object)

- **Result** (String)
- **Focus** (booleanNumber)
- **Keypress Enter** (Trigger)
- **Keypress ESC** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops Sidebar TextInput\\_v2](https://cables.gl/op/Ops Sidebar TextInput_v2)

## 90.1.20 Toggle\_v4



**Full Name:** Ops Sidebar.Toggle\_v4

**Description:** sidebar boolean toggle/switch element

### > Input Ports:

- **Link** (Object)
- **Text** (String)
- **Set Default** (Trigger)
- **Grey Out** (Number: Boolean)
- **Visible** (Number: Boolean)

### < Output Ports:

- **Childs** (Object)
- **Value** (booleanNumber)
- **Toggled** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops Sidebar.Toggle\\_v4](https://cables.gl/op/Ops Sidebar.Toggle_v4)

## 90.1.21 XYPad



**Full Name:** Ops Sidebar.XYPad

**Description:** 2d coordinate input element

### > Input Ports:

- **Link** (Object)
- **Text** (String)
- **Input X** (Number)
- **Input Y** (Number)
- **Flip X** (Number: Boolean)
- **Flip Y** (Number: Boolean)
- **Set Default** (Trigger)
- **Visible** (Number: Boolean)

### < Output Ports:

- **Children** (Object)
- **X** (Number)
- **Y** (Number)
- **HTML Element** (Object)

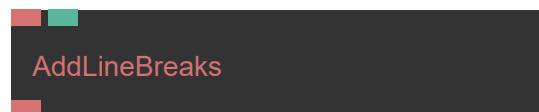
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops Sidebar.XYPad>

# 91 Ops.String

## 91.1 Ops.String

### 91.1.1 AddLineBreaks\_v2



**Full Name:** Ops.String.AddLineBreaks\_v2

**Description:** Insert a line break in a string of words

**> Input Ports:**

- **String** (String)
- **Max Characters Per Line** (Number: Integer)

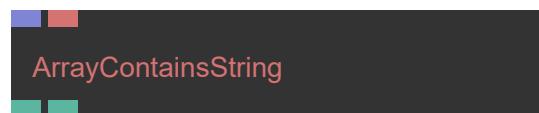
**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.AddLineBreaks\\_v2](https://cables.gl/op/Ops.String.AddLineBreaks_v2)

### 91.1.2 ArrayContainsString



**Full Name:** Ops.String.ArrayContainsString

**Description:** Check if an array contains a string which can also be a number (find,search,indexOf)

**> Input Ports:**

- **Array** (Array)
- **SearchValue** (String)

**< Output Ports:**

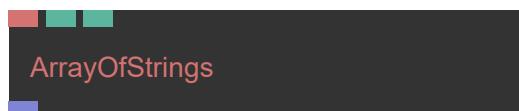
- **Found** (booleanNumber)

- **Index** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.ArrayContainsString>

### 91.1.3ArrayOfStrings



**Full Name:** Ops.String.ArrayOfStrings

**Description:** Create an array of strings and optionally attach index-number

**> Input Ports:**

- **String** (String)
- **Length** (Number: Integer)
- **Attach Number** (Number: Boolean)

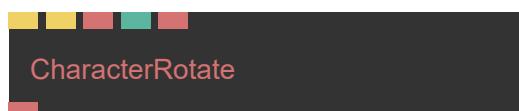
**< Output Ports:**

- **Array** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.ArrayOfStrings>

### 91.1.4 CharacterRotate



**Full Name:** Ops.String.CharacterRotate

**Description:** String rotate characters like a split-flap display

**> Input Ports:**

- **Update** (Trigger)
- **Reset** (Trigger)
- **Text** (String)
- **Random Seed** (Number)
- **Characters** (String)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.CharacterRotate>

## 91.1.5 Concat\_v2



**Full Name:** Ops.String.Concat\_v2

**Description:** Joins two strings together

**> Input Ports:**

- **String1** (String)
- **String2** (String)
- **New Line** (Number: Boolean)
- **Active** (Number: Boolean)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.Concat\\_v2](https://cables.gl/op/Ops.String.Concat_v2)

## 91.1.6 ConcatMulti\_v2



**Full Name:** Ops.String.ConcatMulti\_v2

**Description:** Joins multiple strings together

**> Input Ports:**

- **String 0** (String)
- **String 1** (String)
- **String 2** (String)
- **String 3** (String)
- **String 4** (String)

- **String 5** (String)

- **String 6** (String)

- **String 7** (String)

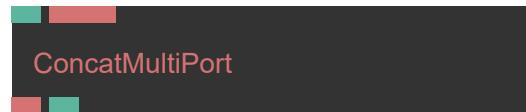
**< Output Ports:**

- **Concat String** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.ConcatMulti\\_v2](https://cables.gl/op/Ops.String.ConcatMulti_v2)

## 91.1.7 ConcatMultiPort\_v2



**Full Name:** Ops.String.ConcatMultiPort\_v2

**Description:** concatenate/join multiple string inputs

**> Input Ports:**

- **Strings\_0** (String)
- **Add Port** (String)

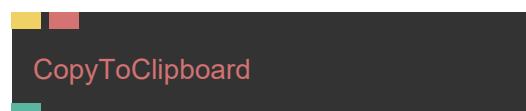
**< Output Ports:**

- **String** (String)
- **Num Strings** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.ConcatMultiPort\\_v2](https://cables.gl/op/Ops.String.ConcatMultiPort_v2)

## 91.1.8 CopyToClipboard



**Full Name:** Ops.String.CopyToClipboard

**Description:** Copy string to clipboard on trigger

**> Input Ports:**

- **Copy** (Trigger)
- **String** (String)

#### < Output Ports:

- **Success** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.CopyToClipboard>

## 91.1.9 DelayStringSimple



**Full Name:** Ops.String.DelayStringSimple

**Description:** delay the output of a string by n seconds

#### > Input Ports:

- **Value** (String)
- **Delay** (Number)

#### < Output Ports:

- **Out Value** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.DelayStringSimple>

## 91.1.10 EndsWith



**Full Name:** Ops.String.EndsWith

**Description:** does a string starts with another string?

#### > Input Ports:

- **String** (String)
- **Search** (String)

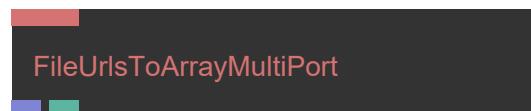
#### < Output Ports:

- **Ends With** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.EndsWith>

## 91.1.11 FileUrlsToArrayMultiPort\_v2



**Full Name:** Ops.String.FileUrlsToArrayMultiPort\_v2

**Description:** create an array from multiple string

#### > Input Ports:

- **Strings\_0** (String)
- **Add Port** (String)

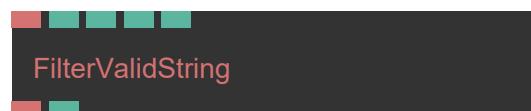
#### < Output Ports:

- **Result** (Array)
- **Num Values** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.FileUrlsToArrayMultiPort\\_v2](https://cables.gl/op/Ops.String.FileUrlsToArrayMultiPort_v2)

## 91.1.12 FilterValidString



**Full Name:** Ops.String.FilterValidString

**Description:** filter valid strings (not null,undefined or empty)

#### > Input Ports:

- **String** (String)
- **Invalid If Null** (Number: Boolean)
- **Invalid If Undefined** (Number: Boolean)
- **Invalid If Empty** (Number: Boolean)
- **Invalid If 0** (Number: Boolean)

#### < Output Ports:

- **Last Valid String** (String)
- **Is Valid** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.FilterValidString>

### 91.1.13 FreezeString



**Full Name:** Ops.String.FreezeString

**Description:** capture the current input and copy it to the output, even after a reload

**> Input Ports:**

- **String** (String)
- **Button** (Trigger)

**< Output Ports:**

- **Frozen String** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.FreezeString>

### 91.1.14 GateString



**Full Name:** Ops.String.GateString

**Description:** Output string if pass through is true

**> Input Ports:**

- **String In** (String)
- **Pass Through** (Number: Boolean)
- **Custom Value** (String)

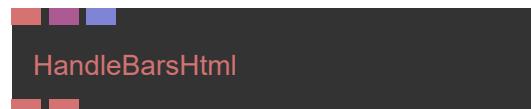
**< Output Ports:**

- **String Out** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.GateString>

### 91.1.15 HandleBarsHtml\_v2



**Full Name:** Ops.String.HandleBarsHtml\_v2

**Description:** string conversion using handlebars template engine

**> Input Ports:**

- **Template** (String)
- **Data** (Object)
- **Array** (Array)

**< Output Ports:**

- **Result** (String)
- **Errors** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.HandleBarsHtml\\_v2](https://cables.gl/op/Ops.String.HandleBarsHtml_v2)

### 91.1.16 HtmlDecode



**Full Name:** Ops.String.HtmlDecode

**Description:** convert a html encoded string to a normal UTF8 string

**> Input Ports:**

- **String** (String)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.HtmlDecode>

## 91.1.17 HtmlEncode



**Full Name:** Ops.String.HtmlEncode

**Description:** encode a string to html

**> Input Ports:**

- **String** (String)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.HtmlEncode>

## 91.1.18 LeftPad\_v2



**Full Name:** Ops.String.LeftPad\_v2

**Description:** create a fixed length string from a number 1 -> 0001

**> Input Ports:**

- **Value** (String)
- **Char** (String)
- **Num** (Number: Integer)

**< Output Ports:**

- **String** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.LeftPad\\_v2](https://cables.gl/op/Ops.String.LeftPad_v2)

## 91.1.19 LimitLineBreaks\_v2



**Full Name:** Ops.String.LimitLineBreaks\_v2

**Description:** Limit number of lines in a string

**> Input Ports:**

- **String** (String)
- **Num Lines** (Number: Integer)
- **Reverse** (Number: Boolean)
- **Force Num Lines** (Number: Boolean)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.LimitLineBreaks\\_v2](https://cables.gl/op/Ops.String.LimitLineBreaks_v2)

## 91.1.20 LineBreak



**Full Name:** Ops.String.LineBreak

**Description:** Outputs a linebreak, or adds a linebreak to a string

**> Input Ports:**

- **String** (String)

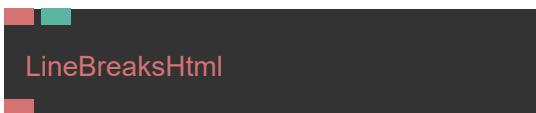
**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.LineBreak>

## 91.1.21 LineBreaksHtml



**Full Name:** Ops.String.LineBreaksHtml

**Description:** Convert linebreaks to html breaks

**> Input Ports:**

- **String** (String)
- **Add Num Breaks** (Number: Integer)

**< Output Ports:**

- **HTML** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.LineBreaksHtml>

## 91.1.22 LoremIpsum



**Full Name:** Ops.String.LoremIpsum

**Description:** Lorem ipsum dolor sit amet

**> Input Ports:**

- Visit *Ops.String.LoremIpsum documentation* for input port details

**< Output Ports:**

- **String** (String)
- **HTML String** (String)
- **Array** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.LoremIpsum>

## 91.1.23 Lowercase\_v2



**Full Name:** Ops.String.Lowercase\_v2

**Description:** convert all characters to small letters

**> Input Ports:**

- **String** (String)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.Lowercase\\_v2](https://cables.gl/op/Ops.String.Lowercase_v2)

## 91.1.24 Md5



**Full Name:** Ops.String.Md5

**Description:** Create a md5 hash of a string

**> Input Ports:**

- **String** (String)

**< Output Ports:**

- **MD5 Hash** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.Md5>

## 91.1.25 NumberFormatter



NumberFormatter

**Full Name:** Ops.String.NumberFormatter

**Description:** Format a number to a string in the given locale and format

**> Input Ports:**

- **Input Number** (Number)
- **Locale String** (String)
- **Minimum Integer Digits** (Number: Integer)
- **Minimum Fraction Digits** (Number: Integer)
- **Maximum Fraction Digits** (Number: Integer)
- **Minimum Significant Digits** (Number: Integer)
- **Maximum Significant Digits** (Number: Integer)
- **Use Grouping** (Number: Boolean)
- **Currency Name** (String)

**< Output Ports:**

- **Formatted Number** (String)
- **Has Error** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.NumberFormatter>

## 91.1.26 NumberSwitchByString



**Full Name:** Ops.String.NumberSwitchByString

**Description:** associate numbers by strings

**> Input Ports:**

- **String** (String)
- **String 1** (String)
- **Number 1** (Number)
- **String 2** (String)
- **Number 2** (Number)
- **String 3** (String)
- **Number 3** (Number)
- **String 4** (String)
- **Number 4** (Number)
- **String 5** (String)

• **Number 5** (Number)

- **String 6** (String)
- **Number 6** (Number)
- **String 7** (String)
- **Number 7** (Number)
- **String 8** (String)
- **Number 8** (Number)
- **String 9** (String)
- **Number 9** (Number)
- **String 10** (String)
- **Number 10** (Number)

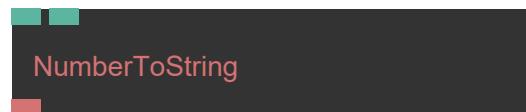
**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.NumberSwitchByString>

## 91.1.27 NumberToString\_v2



**Full Name:** Ops.String.NumberToString\_v2

**Description:** Convert a number to a string

**> Input Ports:**

- **Number** (Number)
- **Decimal Places** (Number: Integer)

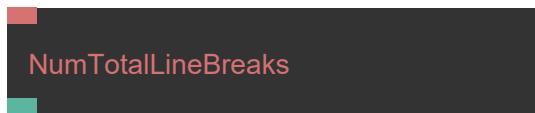
**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.NumberToString\\_v2](https://cables.gl/op/Ops.String.NumberToString_v2)

## 91.1.28 NumTotalLineBreaks



**Full Name:** Ops.String.NumTotalLineBreaks

**Description:** Count number of line breaks in a string

**> Input Ports:**

- **String** (String)

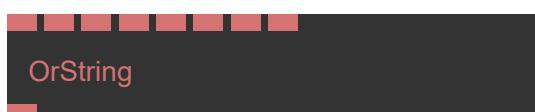
**< Output Ports:**

- **Total Lines** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.NumTotalLineBreaks>

## 91.1.29 OrString



**Full Name:** Ops.String.OrString

**Description:** outputs the first valid string

**> Input Ports:**

- **String 1** (String)
- **String 2** (String)
- **String 3** (String)
- **String 4** (String)
- **String 5** (String)
- **String 6** (String)
- **String 7** (String)
- **String 8** (String)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.OrString>

## 91.1.30 ParseInt\_v2



**Full Name:** Ops.String.ParseInt\_v2

**Description:** Parse a string to a integer number / string to number

**> Input Ports:**

- **String** (String)

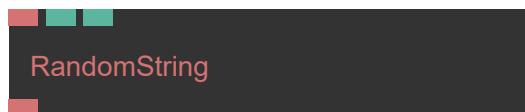
**< Output Ports:**

- **Number** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.ParseInt\\_v2](https://cables.gl/op/Ops.String.ParseInt_v2)

## 91.1.31 RandomString\_v3



**Full Name:** Ops.String.RandomString\_v3

**Description:** Generate a random string of given characters

**> Input Ports:**

- **Chars** (String)
- **Length** (Number: Integer)
- **Seed** (Number)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.RandomString\\_v3](https://cables.gl/op/Ops.String.RandomString_v3)

## 91.1.32 RightPad\_v2



RightPad

**Full Name:** Ops.String.RightPad\_v2

**Description:** create a string with a fixed length filling the space with a character

**> Input Ports:**

- **Value** (String)
- **Char** (String)
- **Num** (Number: Integer)

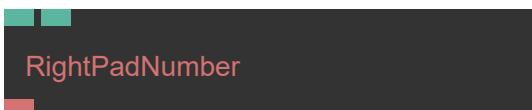
**< Output Ports:**

- **String** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.RightPad\\_v2](https://cables.gl/op/Ops.String.RightPad_v2)

## 91.1.33 RightPadNumber\_v2



RightPadNumber

**Full Name:** Ops.String.RightPadNumber\_v2

**Description:** Converts a number to a string with num decimal places, adds 0's

**> Input Ports:**

- **Value** (Number)
- **Num** (Number: Integer)

**< Output Ports:**

- **String** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.RightPadNumber\\_v2](https://cables.gl/op/Ops.String.RightPadNumber_v2)

## 91.1.34 RouteString



RouteString

**Full Name:** Ops.String.RouteString

**Description:** Route a string to an output port

**> Input Ports:**

- **Index** (Number: Integer)
- **String In** (String)
- **Default String** (String)
- **Set Inactive To Default** (Number: Boolean)

**< Output Ports:**

- **Index 0 String** (String)
- **Index 1 String** (String)
- **Index 2 String** (String)
- **Index 3 String** (String)
- **Index 4 String** (String)
- **Index 5 String** (String)
- **Index 6 String** (String)
- **Index 7 String** (String)
- **Index 8 String** (String)
- **Index 9 String** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.RouteString>

## 91.1.35 SaveTextFile



SaveTextFile

**Full Name:** Ops.String.SaveTextFile

**Description:** download a textfile containing the input string

**> Input Ports:**

- **Download** (Trigger)

- **Filename** (String)
- **Content String** (String)

**< Output Ports:**

- Visit *Ops.String.SaveTextFile* documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.SaveTextFile>

## 91.1.36 SequenceStrings



**Full Name:** Ops.String.SequenceStrings

**Description:** control order and flow of strings

**> Input Ports:**

- **String 0** (String)
- **String 1** (String)
- **String 2** (String)
- **String 3** (String)
- **String 4** (String)
- **String 5** (String)
- **String 6** (String)
- **String 7** (String)
- **String 8** (String)
- **String 9** (String)
- **String 10** (String)
- **String 11** (String)
- **String 12** (String)
- **String 13** (String)
- **String 14** (String)
- **String 15** (String)

**< Output Ports:**

- **Output 0** (String)
- **Output 1** (String)
- **Output 2** (String)
- **Output 3** (String)

- **Output 4** (String)
- **Output 5** (String)
- **Output 6** (String)
- **Output 7** (String)
- **Output 8** (String)
- **Output 9** (String)
- **Output 10** (String)
- **Output 11** (String)
- **Output 12** (String)
- **Output 13** (String)
- **Output 14** (String)
- **Output 15** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.SequenceStrings>

## 91.1.37 StartsWith



**Full Name:** Ops.String.StartsWith

**Description:** does a string starts with another string?

**> Input Ports:**

- **String** (String)
- **Search** (String)

**< Output Ports:**

- **Starts With** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.StartsWith>

## 91.1.38 String\_v3



**Full Name:** Ops.String.String\_v3

**Description:** String input/output

**> Input Ports:**

- **Value** (String)

**< Output Ports:**

- **String** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.String\\_v3](https://cables.gl/op/Ops.String.String_v3)

### 91.1.39 StringCompose\_v3



**Full Name:** Ops.String.StringCompose\_v3

**Description:** Combine multiple Values to a new String

**> Input Ports:**

- **Format** (String)
- **String A** (String)
- **String B** (String)
- **String C** (String)
- **String D** (String)
- **String E** (String)
- **String F** (String)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.StringCompose\\_v3](https://cables.gl/op/Ops.String.StringCompose_v3)

### 91.1.40 StringContains\_v2



**Full Name:** Ops.String.StringContains\_v2

**Description:** check if string contains another string (find,search,indexOf)

**> Input Ports:**

- **String** (String)
- **SearchValue** (String)

**< Output Ports:**

- **Found** (Number)
- **Index** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.StringContains\\_v2](https://cables.gl/op/Ops.String.StringContains_v2)

### 91.1.41 StringEditor



**Full Name:** Ops.String.StringEditor

**Description:** string text editor

**> Input Ports:**

- **Value** (String)
- **Syntax Index** (Number: Integer)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.StringEditor>

### 91.1.42 StringEquals\_v2



**Full Name:** Ops.String.StringEquals\_v2

**Description:** check if content of two strings is the same

**> Input Ports:**

- **String 1** (String)
- **String 2** (String)

**< Output Ports:**

- **Result** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.StringEquals\\_v2](https://cables.gl/op/Ops.String.StringEquals_v2)

### 91.1.43 StringGetLineNumAtIndex



StringGetLineNumAtIndex

**Full Name:** Ops.String.StringGetLineNumAtIndex

**Description:** output the line number at the character index

**> Input Ports:**

- **String** (String)
- **Index** (Number: Integer)

**< Output Ports:**

- **Line** (Number)
- **Found** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.StringGetLineNumAtIndex>

### 91.1.44 StringIterator\_v2



StringIterator

**Full Name:** Ops.String.StringIterator\_v2

**Description:** iterate over every character of a string

**> Input Ports:**

- **Exec** (Trigger)

- **String** (String)

**< Output Ports:**

- **Next** (Trigger)
- **Character** (String)
- **Index** (Number)
- **Length** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.StringIterator\\_v2](https://cables.gl/op/Ops.String.StringIterator_v2)

### 91.1.45 StringLength\_v2



StringLength

**Full Name:** Ops.String.StringLength\_v2

**Description:** number of characters in a string

**> Input Ports:**

- **String** (String)

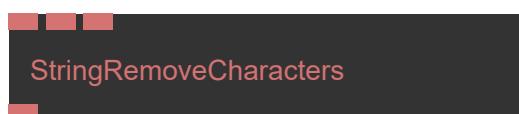
**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.StringLength\\_v2](https://cables.gl/op/Ops.String.StringLength_v2)

### 91.1.46 StringRemoveCharacters



StringRemoveCharacters

**Full Name:** Ops.String.StringRemoveCharacters

**Description:** Remove every occurrences of given characters from a string

**> Input Ports:**

- **String** (String)
- **Characters** (String)
- **Replace** (String)

#### < Output Ports:

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.StringRemoveCharacters>

## 91.1.47 StringReplace



**Full Name:** Ops.String.StringReplace

**Description:** replace occurrences of a string with another string

#### > Input Ports:

- **String** (String)
- **Search For** (String)
- **Replace** (String)
- **Replace What Index** (Number: Integer)

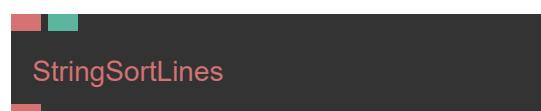
#### < Output Ports:

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.StringReplace>

## 91.1.48 StringSortLines



**Full Name:** Ops.String.StringSortLines

**Description:** sort each line of a string alphabetically

#### > Input Ports:

- **String** (String)
- **Reverse** (Number: Boolean)

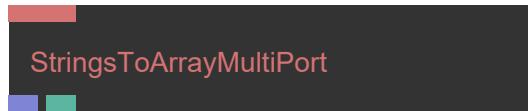
#### < Output Ports:

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.StringSortLines>

## 91.1.49 StringsToArrayMultiPort\_v2



**Full Name:** Ops.String.StringsToArrayMultiPort\_v2

**Description:** create an array from multiple string

#### > Input Ports:

- **Strings\_0** (String)
- **Add Port** (String)

#### < Output Ports:

- **Result** (Array)
- **Num Values** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.StringsToArrayMultiPort\\_v2](https://cables.gl/op/Ops.String.StringsToArrayMultiPort_v2)

## 91.1.50 StringSwitchByString



**Full Name:** Ops.String.StringSwitchByString

**Description:** Switch between multiple strings by a string index

#### > Input Ports:

- **String** (String)
- **Default** (String)
- **String 1** (String)
- **Result String 1** (String)
- **String 2** (String)
- **Result String 2** (String)

- **String 3** (String)
- **Result String 3** (String)
- **String 4** (String)
- **Result String 4** (String)
- **String 5** (String)
- **Result String 5** (String)
- **String 6** (String)
- **Result String 6** (String)
- **String 7** (String)
- **Result String 7** (String)
- **String 8** (String)
- **Result String 8** (String)
- **String 9** (String)
- **Result String 9** (String)
- **String 10** (String)
- **Result String 10** (String)

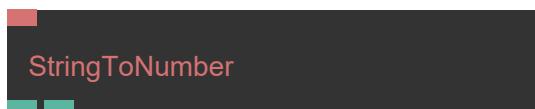
**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.StringSwitchByString>

## 91.1.51 StringToNumber



**Full Name:** Ops.String.StringToNumber

**Description:** Parses a string and returns a floating point number / string to number

**> Input Ports:**

- **String** (String)

**< Output Ports:**

- **Number** (Number)
- **Not A Number** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.StringToNumber>

## 91.1.52 StringTrim\_v2



**Full Name:** Ops.String.StringTrim\_v2

**Description:** Remove whitespace from both ends of a string

**> Input Ports:**

- **String** (String)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.StringTrim\\_v2](https://cables.gl/op/Ops.String.StringTrim_v2)

## 91.1.53 StripHtml



**Full Name:** Ops.String.StripHtml

**Description:** remove html tags from a string

**> Input Ports:**

- **String** (String)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.StripHtml>

## 91.1.54 SubString\_v2



**Full Name:** Ops.String.SubString\_v2

**Description:** Subset of a string between one index and another

**> Input Ports:**

- **String** (String)
- **Start** (Number: Integer)
- **End** (Number: Integer)
- **End Of String** (Number: Boolean)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.SubString\\_v2](https://cables.gl/op/Ops.String.SubString_v2)

## 91.1.55 SwitchString



**Full Name:** Ops.String.SwitchString

**Description:** Switch between multiple strings with an index

**> Input Ports:**

- **Index** (Number: Integer)
- **String 0** (String)
- **String 1** (String)
- **String 2** (String)
- **String 3** (String)
- **String 4** (String)
- **String 5** (String)
- **String 6** (String)
- **String 7** (String)
- **String 8** (String)
- **String 9** (String)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.SwitchString>

## 91.1.56 SwitchStringMultiPort\_v2



**Full Name:** Ops.String.SwitchStringMultiPort\_v2

**Description:** switch between multiple string inputs

**> Input Ports:**

- **Index** (Number: Integer)
- **Strings\_0** (String)
- **Add Port** (String)

**< Output Ports:**

- **String** (String)
- **Num Values** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.SwitchStringMultiPort\\_v2](https://cables.gl/op/Ops.String.SwitchStringMultiPort_v2)

## 91.1.57 Uppercase\_v2



**Full Name:** Ops.String.Uppercase\_v2

**Description:** Convert all characters in a string to uppercase

**> Input Ports:**

- **String** (String)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.Uppercase\\_v2](https://cables.gl/op/Ops.String.Uppercase_v2)

## 91.1.58 UUID



**Full Name:** Ops.String.UUID

**Description:** outputs a unique identifier string

**> Input Ports:**

- **Generate** (Trigger)

**< Output Ports:**

- **Id** (String)

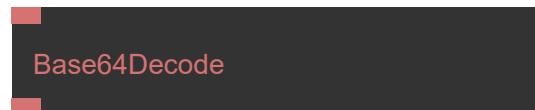
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.UUID>

## 92 Ops.String.Base64

### 92.1 Ops.String.Base64

#### 92.1.1 Base64Decode\_v2



**Full Name:** Ops.String.Base64.Base64Decode\_v2

**Description:** decode a string to base64

**> Input Ports:**

- **String** (String)

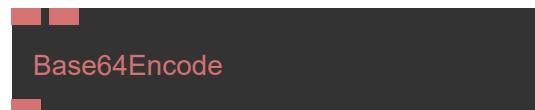
**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.Base64.Base64Decode\\_v2](https://cables.gl/op/Ops.String.Base64.Base64Decode_v2)

#### 92.1.2 Base64Encode\_v3



**Full Name:** Ops.String.Base64.Base64Encode\_v3

**Description:** encode a string to base64

**> Input Ports:**

- **String** (String)
- **MimeType** (String)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.Base64.Base64Encode\\_v3](https://cables.gl/op/Ops.String.Base64.Base64Encode_v3)

## 92.1.3 DownloadBase64File



**Full Name:** Ops.String.Base64.DownloadBase64File

**Description:** trigger a download of a base64 binary file

**> Input Ports:**

- **Data URL** (String)
- **Filename** (String)
- **Download** (Trigger)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.String.Base64.DownloadBase64File>

## 93 Ops.String.File

### 93.1 Ops.String.File

#### 93.1.1 FileInput\_v2



**Full Name:** Ops.String.File.FileInput\_v2

**Description:** get URL of a file

**> Input Ports:**

- **File** (String)

**< Output Ports:**

- **URL** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.File.FileInput\\_v2](https://cables.gl/op/Ops.String.File.FileInput_v2)

#### 93.1.2 SwitchFile\_v2



**Full Name:** Ops.String.File.SwitchFile\_v2

**Description:** switch between filenames

**> Input Ports:**

- **Index** (Number: Integer)
- **File 0** (String)
- **File 1** (String)
- **File 2** (String)
- **File 3** (String)
- **File 4** (String)
- **File 5** (String)

- **File 6** (String)
- **File 7** (String)
- **File 8** (String)
- **File 9** (String)
- **File 10** (String)
- **File 11** (String)
- **File 12** (String)
- **File 13** (String)
- **File 14** (String)
- **File 15** (String)

**< Output Ports:**

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.String.File.SwitchFile\\_v2](https://cables.gl/op/Ops.String.File.SwitchFile_v2)

## 94 Ops.Templates

---

### 94.1 Ops.Templates

#### 94.1.1 ExampleVizOp



ExampleVizOp

**Full Name:** Ops.Templates.ExampleVizOp

**Description:** example how to code a viz layer op

**> Input Ports:**

- **Number** (Number)

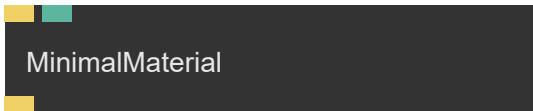
**< Output Ports:**

- Visit *Ops.Templates.ExampleVizOp documentation* for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Templates.ExampleVizOp>

#### 94.1.2 MinimalMaterial



MinimalMaterial

**Full Name:** Ops.Templates.MinimalMaterial

**Description:** Material Example Template

**> Input Ports:**

- **Render** (Trigger)
- **Red** (Number)

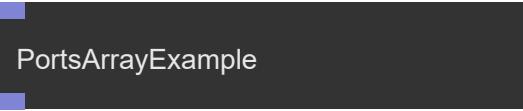
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Templates.MinimalMaterial>

### 94.1.3 PortsArrayExample



PortsArrayExample

**Full Name:** Ops.Templates.PortsArrayExample

**Description:** Is a template for creating Array ports

**> Input Ports:**

- **Array In** (Array)

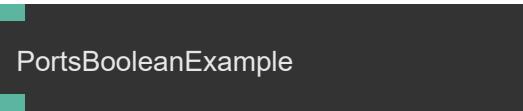
**< Output Ports:**

- **Array Out** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Templates.PortsArrayExample>

### 94.1.4 PortsBooleanExample



PortsBooleanExample

**Full Name:** Ops.Templates.PortsBooleanExample

**Description:** Is a template for creating Boolean ports

**> Input Ports:**

- **Boolean In** (Number: Boolean)

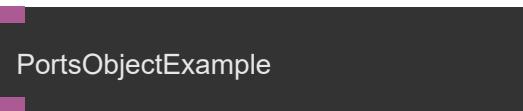
**< Output Ports:**

- **Boolean Out** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Templates.PortsBooleanExample>

### 94.1.5 PortsObjectExample



PortsObjectExample

**Full Name:** Ops.Templates.PortsObjectExample

**Description:** Is a template for creating Object ports

**> Input Ports:**

- **Object In** (Object)

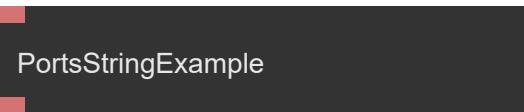
**< Output Ports:**

- **Object Out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Templates.PortsObjectExample>

### 94.1.6 PortsStringExample



PortsStringExample

**Full Name:** Ops.Templates.PortsStringExample

**Description:** Is a template for creating String ports

**> Input Ports:**

- **String In** (String)

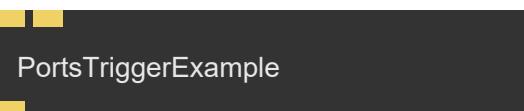
**< Output Ports:**

- **String Out** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Templates.PortsStringExample>

### 94.1.7 PortsTriggerExample



PortsTriggerExample

**Full Name:** Ops.Templates.PortsTriggerExample

**Description:** Is a template for creating Trigger ports

**> Input Ports:**

- **Trigger In** (Trigger)
- **Press Me** (Trigger)

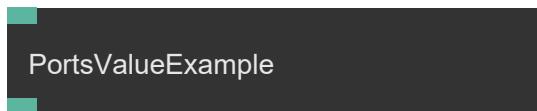
#### < Output Ports:

- **Trigger Out** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Templates.PortsTriggerExample>

### 94.1.10 UiTestOp



**Full Name:** Ops.Templates.PortsValueExample

**Description:** Is a template for creating Value ports

#### > Input Ports:

- **Number In** (Number)

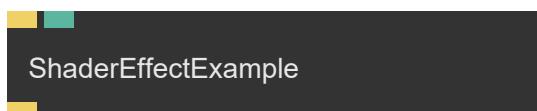
#### < Output Ports:

- **Value Out** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Templates.PortsValueExample>

### 94.1.9 ShaderEffectExample



**Full Name:** Ops.Templates.ShaderEffectExample

**Description:** shader effect example template

#### > Input Ports:

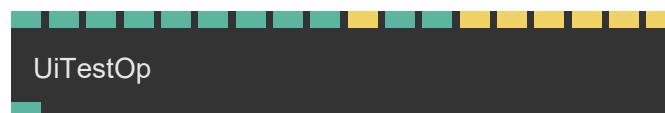
- **Render** (Trigger)
- **Width** (Number)

#### < Output Ports:

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Templates.ShaderEffectExample>



**Full Name:** Ops.Templates.UiTestOp

**Description:** UI indicators example op

#### > Input Ports:

- **Loading Task** (Number: Boolean)
- **Loading** (Number: Boolean)
- **Warning** (Number: Boolean)
- **Error** (Number: Boolean)
- **Hint** (Number: Boolean)
- **Not Working** (Number: Boolean)
- **Slider** (Number)
- **Gradient** (Number)
- **Resizable** (Number: Boolean)
- **Trigger** (Trigger)
- **Greyout** (Number: Boolean)
- **This Will Greyout** (Number)
- **Open Prompt** (Trigger)
- **Open Modal** (Trigger)
- **Open New Tab** (Trigger)

#### < Output Ports:

- **Something** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Templates.UiTestOp>

# 95 Ops.TimeLine

## 95.1 Ops.TimeLine

### 95.1.1 Anim



**Full Name:** Ops.TimeLine.Anim

**Description:** timeline keyframable animation object

#### > Input Ports:

- **Value** (Number)
- **Clip** (Number: Boolean)
- **Clip Name** (String)

#### < Output Ports:

- **Anim** (Object)
- **Loop Length** (Number)
- **Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.Anim>

### 95.1.2 AnimGetKey



**Full Name:** Ops.TimeLine.AnimGetKey

**Description:** Get data from a single key in an animation

#### > Input Ports:

- **Anim** (Object)
- **Time** (Number)

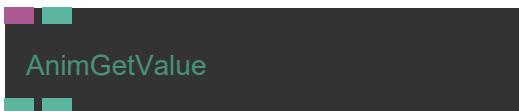
#### < Output Ports:

- **Index** (Number)
- **Key Value** (Number)
- **Key Time** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.AnimGetKey>

### 95.1.3 AnimGetValue



**Full Name:** Ops.TimeLine.AnimGetValue

**Description:** get the animated value at time x of an animation object

#### > Input Ports:

- **Anim** (Object)
- **Time** (Number)

#### < Output Ports:

- **Value** (Number)
- **Loop** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.AnimGetValue>

### 95.1.4 AnimInfo



**Full Name:** Ops.TimeLine.AnimInfo

**Description:** Get information about an anim object

#### > Input Ports:

- **Anim** (Object)

#### < Output Ports:

- **Total Keys** (Number)
- **Length Seconds** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.AnimInfo>

## 95.1.5 AutoPlay



**Full Name:** Ops.TimeLine.AutoPlay

**Description:** Automatically starts the timeline playback when opening patch

**> Input Ports:**

- Visit *Ops.TimeLine.AutoPlay* documentation for input port details

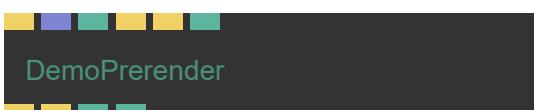
**< Output Ports:**

- Visit *Ops.TimeLine.AutoPlay* documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.AutoPlay>

## 95.1.6 DemoPrerender



**Full Name:** Ops.TimeLine.DemoPrerender

**Description:** Prerenderer based on timeline progress

**> Input Ports:**

- **Render** (Trigger)
- **Manual Timestamps** (Array)
- **Record Events** (Number: Boolean)
- **Reset** (Trigger)
- **Clear** (Trigger)
- **ReRender On Resize** (Number: Boolean)

**< Output Ports:**

- **Next** (Trigger)
- **Prerendered Frame** (Trigger)

- **Progress** (Number)

- **Num Events** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.DemoPrerender>

## 95.1.7 GotoFrame



**Full Name:** Ops.TimeLine.GotoFrame

**Description:** jump to a key in the timeline

**> Input Ports:**

- **Frame** (Number)

**< Output Ports:**

- Visit *Ops.TimeLine.GotoFrame* documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.GotoFrame>

## 95.1.8 PreRender



**Full Name:** Ops.TimeLine.PreRender

**Description:** Render the patch at certain times

**> Input Ports:**

- **Render** (Trigger)
- **Max Time** (Number: Integer)
- **Step** (Number: Integer)
- **Reset** (Trigger)

**< Output Ports:**

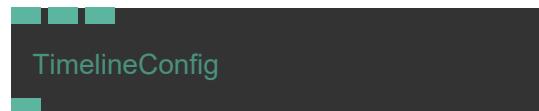
- **Next** (Trigger)
- **Render Progress** (Trigger)

- **Done** (Trigger)
- **Progress** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.PreRender>

## 95.1.9 TimelineConfig



**Full Name:** Ops.TimeLine.TimelineConfig

**Description:** configure the timeline for the current patch

**> Input Ports:**

- **FPS** (Number: Integer)
- **Restrict To Frames** (Number: Boolean)
- **Fade In Frames** (Number: Boolean)

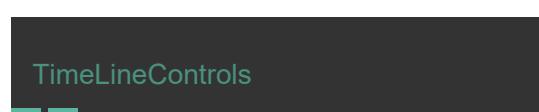
**< Output Ports:**

- **Duration Seconds** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.TimelineConfig>

## 95.1.10 TimeLineControls



**Full Name:** Ops.TimeLine.TimeLineControls

**Description:** use position and play pause state of cables timeline

**> Input Ports:**

- Visit *Ops.TimeLine.TimeLineControls* documentation for input port details

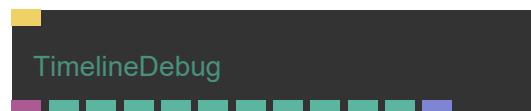
**< Output Ports:**

- **Time** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.TimeLineControls>

## 95.1.11 TimelineDebug



**Full Name:** Ops.TimeLine.TimelineDebug

**Description:** Visit documentation for details

**> Input Ports:**

- **Update** (Trigger)

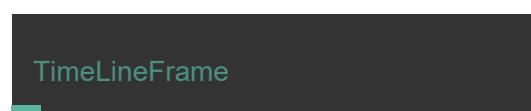
**< Output Ports:**

- **Data** (Object)
- **Time Cursor** (Number)
- **Visible Duration** (Number)
- **Visible Time Start** (Number)
- **Loop Start** (Number)
- **Loop End** (Number)
- **Num Selected Keys** (Number)
- **Selected Values Min** (Number)
- **Selected Values Max** (Number)
- **Selected Times Min** (Number)
- **Selected Times Max** (Number)
- **Selected Keys** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.TimelineDebug>

## 95.1.12 TimeLineFrame



**Full Name:** Ops.TimeLine.TimeLineFrame

**Description:** Returns the current frame number of the timeline

**> Input Ports:**

- Visit `Ops.TimeLine.TimeLineFrame` documentation for input port details

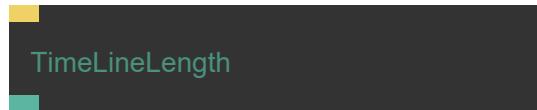
**< Output Ports:**

- **Time** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.TimeLineFrame>

## 95.1.13 TimeLineLength



**Full Name:** Ops.TimeLine.TimeLineLength

**Description:** current set length of the timeline

**> Input Ports:**

- **Update** (Trigger)

**< Output Ports:**

- **Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.TimeLineLength>

## 95.1.14 TimeLineLoop



**Full Name:** Ops.TimeLine.TimeLineLoop

**Description:** Automatic rewind of timeline at a certain time

**> Input Ports:**

- **Execute** (Trigger)
- **Duration** (Number)
- **How long the loop should be** (in seconds)

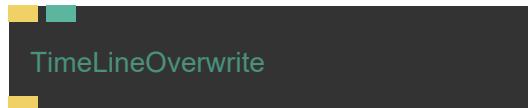
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.TimeLineLoop>

## 95.1.15 TimeLineOverwrite



**Full Name:** Ops.TimeLine.TimeLineOverwrite

**Description:** overwrite timeline time value

**> Input Ports:**

- **Exe** (Trigger)
- **New Time** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.TimeLineOverwrite>

## 95.1.16 TimeLinePlay



**Full Name:** Ops.TimeLine.TimeLinePlay

**Description:** Visit documentation for details

**> Input Ports:**

- **Play** (Trigger)
- **Pause** (Trigger)

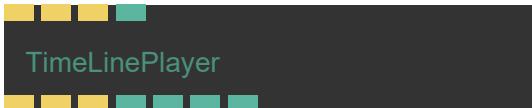
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.TimeLinePlay>

## 95.1.17 TimeLinePlayer



**Full Name:** Ops.TimeLine.TimeLinePlayer

**Description:** Player controls for the timeline

**> Input Ports:**

- **Play** (Trigger)
- **Pause** (Trigger)
- **Rewind** (Trigger)
- **Set Current Time** (Number)

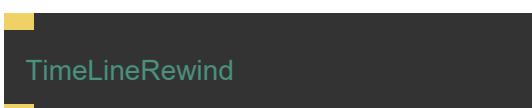
**< Output Ports:**

- **Play Trigger** (Trigger)
- **Pause Trigger** (Trigger)
- **Rewind Trigger** (Trigger)
- **Is Playing** (booleanNumber)
- **Current Time** (Number)
- **Current Frame** (Number)
- **Current time in frames** (30fps)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.TimeLinePlayer>

## 95.1.18 TimeLineRewind



**Full Name:** Ops.TimeLine.TimeLineRewind

**Description:** set time of timeline to 0 (rewind, restart)

**> Input Ports:**

- **Exe** (Trigger)

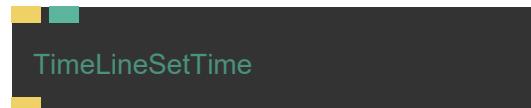
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.TimeLineRewind>

## 95.1.19 TimeLineSetTime



**Full Name:** Ops.TimeLine.TimeLineSetTime

**Description:** set current time of timeline

**> Input Ports:**

- **Update** (Trigger)
- **Time** (Number)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.TimeLineSetTime>

## 95.1.20 TimeLineTime



**Full Name:** Ops.TimeLine.TimeLineTime

**Description:** Returns the current time of the timeline

**> Input Ports:**

- Visit *Ops.TimeLine.TimeLineTime documentation for input port details*

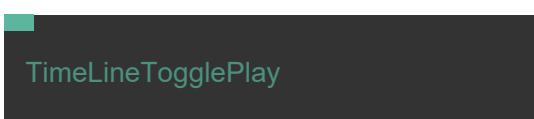
**< Output Ports:**

- **Time** (Number)
- **The current time of the timeline** (in seconds)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.TimeLineTime>

## 95.1.21 TimeLineTogglePlay



**Full Name:** Ops.TimeLine.TimeLineTogglePlay

**Description:** toggle between timeline playing and being paused

**> Input Ports:**

- **Play** (Number: Boolean)
- **Public** (20): MY IDENTITY PATTERN

**< Output Ports:**

- Visit *Ops.TimeLine.TimeLineTogglePlay* documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.TimeLineTogglePlay>

## 95.1.22 TimelineValue



**Full Name:** Ops.TimeLine.TimelineValue

**Description:** Animate and get value at “time” of timeline

**> Input Ports:**

- **Time** (Number)
- **Value** (Number)
- **Unit Index** (Number: Integer)

**< Output Ports:**

- **Result** (Number)
- **Anim Array** (Array)
- **Anim Finished** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.TimelineValue>

# 96 Ops.TimeLine.Viz

## 96.1 Ops.TimeLine.Viz

### 96.1.1 TimeLineBPM



**Full Name:** Ops.TimeLine.Viz.TimeLineBPM

**Description:** Display current Beat index and BPM timing information as beat rectangles on the timeline

**> Input Ports:**

- **BPM** (Number)
- **Offset** (Number)

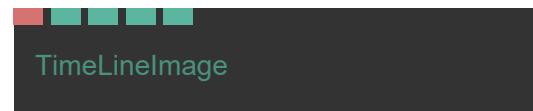
**< Output Ports:**

- Visit *Ops.TimeLine.Viz.TimeLineBPM* documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.TimeLine.Viz.TimeLineBPM>

### 96.1.2 TimeLineImage



**Full Name:** Ops.TimeLine.Viz.TimeLineImage

**Description:** Display an image on the timeline

**> Input Ports:**

- **File** (String)
- **Slot** (Number: Integer)
- **Opacity** (Number)
- **Start** (Number)
- **End** (Number)

#### < Output Ports:

- Visit *Ops.TimeLine.Viz.TimeLineImage* documentation for output port details

**Example Patch:** Open in Editor

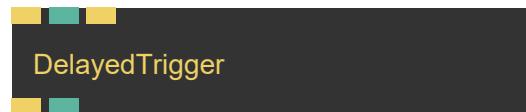
**Docs:** <https://cables.gl/op/Ops.TimeLine.Viz.TimeLineImage>

# 97 Ops.Trigger

---

## 97.1 Ops.Trigger

### 97.1.1 DelayedTrigger



**Full Name:** Ops.Trigger.DelayedTrigger

**Description:** delay triggering next port by x seconds

#### > Input Ports:

- **Exe** (Trigger)
- **Delay** (Number)
- **Cancel** (Trigger)

#### < Output Ports:

- **Next** (Trigger)
- **Delaying** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.DelayedTrigger>

### 97.1.2 GateTrigger



**Full Name:** Ops.Trigger.GateTrigger

**Description:** Allows a trigger to pass only if the gate is open

#### > Input Ports:

- **Execute** (Trigger)
- **Pass Through** (Number: Boolean)

#### < Output Ports:

- **Trigger Out** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.GateTrigger>

### 97.1.3 Interval



**Full Name:** Ops.Trigger.Interval

**Description:** Timed Trigger every x ms

**> Input Ports:**

- **Interval** (Number)
- **Active** (Number: Boolean)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.Interval>

### 97.1.4 IsTriggered



**Full Name:** Ops.Trigger.IsTriggered

**Description:** outputs true if being triggered last frame

**> Input Ports:**

- **Trigger** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **Was Triggered** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.IsTriggered>

### 97.1.5 NthTrigger\_v2



**Full Name:** Ops.Trigger.NthTrigger\_v2

**Description:** Lets a trigger through every nth time (trigger limiter)

**> Input Ports:**

- **Execute** (Trigger)
- **Nth** (Number)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Trigger.NthTrigger\\_v2](https://cables.gl/op/Ops.Trigger.NthTrigger_v2)

### 97.1.6 NumberByTrigger



**Full Name:** Ops.Trigger.NumberByTrigger

**Description:** Outputs the last number of the input port which was triggered

**> Input Ports:**

• Visit *Ops.Trigger.NumberByTrigger documentation* for input port details

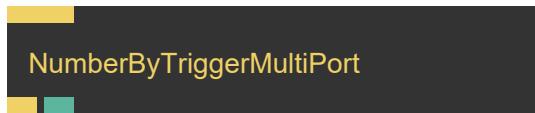
**< Output Ports:**

- **Number** (Number)
- **Triggered** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.NumberByTrigger>

## 97.1.7 NumberByTriggerMultiPort\_v2



**Full Name:** Ops.Trigger.NumberByTriggerMultiPort\_v2  
**Description:** output a number by triggering an index port

**> Input Ports:**

- **Trigger\_0** (Trigger)
- **Add Port** (Trigger)

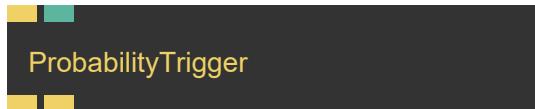
**< Output Ports:**

- **Next** (Trigger)
- **Number Triggered** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Trigger.NumberByTriggerMultiPort\\_v2](https://cables.gl/op/Ops.Trigger.NumberByTriggerMultiPort_v2)

## 97.1.8 ProbabilityTrigger



**Full Name:** Ops.Trigger.ProbabilityTrigger

**Description:** trigger by chance

**> Input Ports:**

- **Trigger In** (Trigger)
- **Probability** (Number)

**< Output Ports:**

- **Trigger Output** (Trigger)
- **Inverse Trigger Output** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.ProbabilityTrigger>

## 97.1.9 RandomTrigger



**Full Name:** Ops.Trigger.RandomTrigger

**Description:** randomly trigger

**> Input Ports:**

- **Render** (Trigger)
- **Num Times** (Number)
- **Seed** (Number)
- **Only Once** (Number: Boolean)

**< Output Ports:**

- **Render** (Trigger)
- **Num Times** (Number)
- **Seed** (Number)
- **Only Once** (Number: Boolean)
- **Index** (Number)
- **Trigger 0** (Trigger)
- **Trigger 1** (Trigger)
- **Trigger 2** (Trigger)
- **Trigger 3** (Trigger)
- **Trigger 4** (Trigger)
- **Trigger 5** (Trigger)
- **Trigger 6** (Trigger)
- **Trigger 7** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.RandomTrigger>

## 97.1.10 Repeat2d



**Full Name:** Ops.Trigger.Repeat2d

**Description:** Triggers all ops underneath Num X \* Num Y times

#### > Input Ports:

- **Exe** (Trigger)
- **Num X** (Number: Integer)
- **Num Y** (Number: Integer)
- **Mul** (Number)
- **Center** (Number: Boolean)
- **Centers X and Y around the origin (0/0)**

#### < Output Ports:

- **Trigger** (Trigger)
- **X** (Number)
- **Y** (Number)
- **Index** (Number)
- **Total Iterations** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.Repeat2d>

## 97.1.11 Repeat\_v2



**Full Name:** Ops.Trigger.Repeat\_v2

**Description:** Triggers all ops below x times (for loop / while)

#### > Input Ports:

- **Execute** (Trigger)
- **Repeats** (Number: Integer)

#### < Output Ports:

- **Next** (Trigger)
- **Index** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Trigger.Repeat\\_v2](https://cables.gl/op/Ops.Trigger.Repeat_v2)

## 97.1.12 RouteTrigger



**Full Name:** Ops.Trigger.RouteTrigger

**Description:** Triggers one of the out ports - value index switch case (was SwitchTrigger)

#### > Input Ports:

- **Execute** (Trigger)
- **Switch Value** (Number: Integer)

#### < Output Ports:

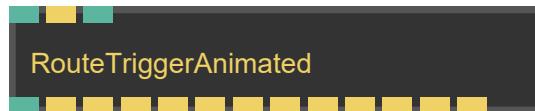
- **Next Trigger** (Trigger)
- **Switched Value** (Number)
- **Trigger 0** (Trigger)
- **Trigger 1** (Trigger)
- **Trigger 2** (Trigger)
- **Trigger 3** (Trigger)
- **Trigger 4** (Trigger)
- **Trigger 5** (Trigger)
- **Trigger 6** (Trigger)
- **Trigger 7** (Trigger)
- **Trigger 8** (Trigger)
- **Trigger 9** (Trigger)
- **Trigger 10** (Trigger)
- **Trigger 11** (Trigger)
- **Trigger 12** (Trigger)
- **Trigger 13** (Trigger)
- **Trigger 14** (Trigger)
- **Trigger 15** (Trigger)
- **Trigger 16** (Trigger)
- **Trigger 17** (Trigger)
- **Trigger 18** (Trigger)
- **Trigger 19** (Trigger)
- **Trigger 20** (Trigger)
- **Trigger 21** (Trigger)
- **Trigger 22** (Trigger)
- **Trigger 23** (Trigger)

- **Default Trigger** (Trigger)
- **Highest Index** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.RouteTrigger>

### 97.1.13 RouteTriggerAnimated



**Full Name:** Ops.Trigger.RouteTriggerAnimated

**Description:** animated switching between things

#### > Input Ports:

- **Index** (Number: Integer)
- **Exe** (Trigger)
- **Duration** (Number)

#### < Output Ports:

- **Qutsn94pc** (Trigger)
- **Hvylh9o8** (Trigger)
- **T8dvyjuoq** (Trigger)
- **A0w7orgi8** (Trigger)
- **R8h4qx4z8** (Trigger)
- **Cr80a86xi** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.RouteTriggerAnimated>

### 97.1.14 RouteTriggerMultiPort\_v2



**Full Name:** Ops.Trigger.RouteTriggerMultiPort\_v2

**Description:** Triggers one of the - value index switch case

#### > Input Ports:

- **Execute** (Trigger)
- **Switch Value** (Number: Integer)

#### < Output Ports:

- **Execute** (Trigger)
- **Switch Value** (Number: Integer)
- **Total Connections** (Number)
- **Connected Op Names** (Array)
- **Trigger\_0** (Trigger)
- **Trigger\_1** (Trigger)
- **Trigger\_2** (Trigger)
- **Trigger\_3** (Trigger)
- **Trigger\_4** (Trigger)
- **Trigger\_5** (Trigger)
- **Trigger\_6** (Trigger)
- **Trigger\_7** (Trigger)
- **Trigger\_8** (Trigger)
- **Trigger\_9** (Trigger)
- **Trigger\_10** (Trigger)
- **Trigger\_11** (Trigger)
- **Trigger\_12** (Trigger)
- **Trigger\_13** (Trigger)
- **Trigger\_14** (Trigger)
- **Trigger\_15** (Trigger)
- **Trigger\_16** (Trigger)
- **Trigger\_17** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Trigger.RouteTriggerMultiPort\\_v2](https://cables.gl/op/Ops.Trigger.RouteTriggerMultiPort_v2)

### 97.1.15 RouteTriggerString\_v2



**Full Name:** Ops.Trigger.RouteTriggerString\_v2

**Description:** route trigger output by string

#### > Input Ports:

- **Execute** (Trigger)

- **Switch Value** (String)
- **String 0** (String)
- **String 1** (String)
- **String 2** (String)
- **String 3** (String)
- **String 4** (String)
- **String 5** (String)
- **String 6** (String)
- **String 7** (String)
- **String 8** (String)
- **String 9** (String)
- **String 10** (String)
- **String 11** (String)
- **String 12** (String)
- **String 13** (String)
- **String 14** (String)
- **String 15** (String)
- **String 16** (String)
- **String 17** (String)
- **String 18** (String)
- **String 19** (String)
- **String 20** (String)
- **String 21** (String)
- **String 22** (String)
- **String 23** (String)

**< Output Ports:**

- **Next Trigger** (Trigger)
- **Switched Index** (Number)
- **Trigger 0** (Trigger)
- **Trigger 1** (Trigger)
- **Trigger 2** (Trigger)
- **Trigger 3** (Trigger)
- **Trigger 4** (Trigger)
- **Trigger 5** (Trigger)
- **Trigger 6** (Trigger)
- **Trigger 7** (Trigger)
- **Trigger 8** (Trigger)
- **Trigger 9** (Trigger)
- **Trigger 10** (Trigger)

- **Trigger 11** (Trigger)
- **Trigger 12** (Trigger)
- **Trigger 13** (Trigger)
- **Trigger 14** (Trigger)
- **Trigger 15** (Trigger)
- **Trigger 16** (Trigger)
- **Trigger 17** (Trigger)
- **Trigger 18** (Trigger)
- **Trigger 19** (Trigger)
- **Trigger 20** (Trigger)
- **Trigger 21** (Trigger)
- **Trigger 22** (Trigger)
- **Trigger 23** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Trigger.RouteTriggerString\\_v2](https://cables.gl/op/Ops.Trigger.RouteTriggerString_v2)

### 97.1.16 Sequence



**Full Name:** Ops.Trigger.Sequence

**Description:** control the order of execution/triggering

**> Input Ports:**

- **Exe** (Trigger)
- **Exe 0** (Trigger)
- **Exe 1** (Trigger)
- **Exe 2** (Trigger)
- **Exe 3** (Trigger)
- **Exe 4** (Trigger)
- **Exe 5** (Trigger)
- **Exe 6** (Trigger)
- **Exe 7** (Trigger)
- **Exe 8** (Trigger)
- **Exe 9** (Trigger)
- **Exe 10** (Trigger)
- **Exe 11** (Trigger)

- **Exe 12** (Trigger)
- **Exe 13** (Trigger)
- **Exe 14** (Trigger)

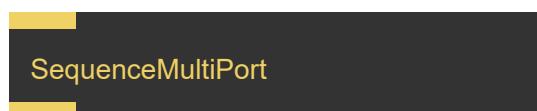
**< Output Ports:**

- **Trigger 0** (Trigger)
- **Trigger 1** (Trigger)
- **Trigger 2** (Trigger)
- **Trigger 3** (Trigger)
- **Trigger 4** (Trigger)
- **Trigger 5** (Trigger)
- **Trigger 6** (Trigger)
- **Trigger 7** (Trigger)
- **Trigger 8** (Trigger)
- **Trigger 9** (Trigger)
- **Trigger 10** (Trigger)
- **Trigger 11** (Trigger)
- **Trigger 12** (Trigger)
- **Trigger 13** (Trigger)
- **Trigger 14** (Trigger)
- **Trigger 15** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.Sequence>

## 97.1.17 SequenceMultiPort\_v2



**Full Name:** Ops.Trigger.SequenceMultiPort\_v2

**Description:** sequence trigger

**> Input Ports:**

- **Input\_0** (Trigger)
- **Add Port** (Trigger)

**< Output Ports:**

- **Output\_0** (Trigger)
- **Output\_1** (Trigger)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Trigger.SequenceMultiPort\\_v2](https://cables.gl/op/Ops.Trigger.SequenceMultiPort_v2)

## 97.1.18 SwitchTrigger



**Full Name:** Ops.Trigger.SwitchTrigger

**Description:** route input triggers by index to one output

**> Input Ports:**

- **Trigger Index** (Number: Integer)
- **Trigger In 0** (Trigger)
- **Trigger In 1** (Trigger)
- **Trigger In 2** (Trigger)
- **Trigger In 3** (Trigger)
- **Trigger In 4** (Trigger)
- **Trigger In 5** (Trigger)
- **Trigger In 6** (Trigger)
- **Trigger In 7** (Trigger)
- **Trigger In 8** (Trigger)
- **Trigger In 9** (Trigger)
- **Trigger In 10** (Trigger)
- **Trigger In 11** (Trigger)
- **Trigger In 12** (Trigger)
- **Trigger In 13** (Trigger)
- **Trigger In 14** (Trigger)
- **Trigger In 15** (Trigger)

**< Output Ports:**

- **Trigger Out** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.SwitchTrigger>

## 97.1.19 Threshold



**Full Name:** Ops.Trigger.Threshold

**Description:** Triggers only once when threshold is crossed

**> Input Ports:**

- **Threshold** (Number)

**< Output Ports:**

- Visit *Ops.Trigger.Threshold* documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.Threshold>

## 97.1.20 TimedSequence



**Full Name:** Ops.Trigger.TimedSequence

**Description:** timed switching of trigger

**> Input Ports:**

- **Exe** (Trigger)
- **Current** (Number: Integer)
- **OverwriteTime** (Number: Boolean)
- **IgnoreInSubPatch** (Number: Boolean)

**< Output Ports:**

- **TriggerAlways** (Trigger)
- **Names** (Array)
- **CurrentKeyTime** (Number)
- **Current** (Number)
- **Trigger 0** (Trigger)
- **Trigger 1** (Trigger)
- **Trigger 2** (Trigger)
- **Trigger 3** (Trigger)
- **Trigger 4** (Trigger)

- **Trigger 5** (Trigger)
- **Trigger 6** (Trigger)
- **Trigger 7** (Trigger)
- **Trigger 8** (Trigger)
- **Trigger 9** (Trigger)
- **Trigger 10** (Trigger)
- **Trigger 11** (Trigger)
- **Trigger 12** (Trigger)
- **Trigger 13** (Trigger)
- **Trigger 14** (Trigger)
- **Trigger 15** (Trigger)
- **Trigger 16** (Trigger)
- **Trigger 17** (Trigger)
- **Trigger 18** (Trigger)
- **Trigger 19** (Trigger)
- **Trigger 20** (Trigger)
- **Trigger 21** (Trigger)
- **Trigger 22** (Trigger)
- **Trigger 23** (Trigger)
- **Trigger 24** (Trigger)
- **Trigger 25** (Trigger)
- **Trigger 26** (Trigger)
- **Trigger 27** (Trigger)
- **Trigger 28** (Trigger)
- **Trigger 29** (Trigger)
- **Trigger 30** (Trigger)
- **Trigger 31** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TimedSequence>

## 97.1.21 TimeSinceTrigger



**Full Name:** Ops.Trigger.TimeSinceTrigger

**Description:** Get the time since last trigger

> **Input Ports:**

- **Exe** (Trigger)
- **Trigger** (Trigger)
- **Reset** (Trigger)

< **Output Ports:**

- **Next** (Trigger)
- **Time** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TimeSinceTrigger>

## 97.1.22 TriggerButton



**Full Name:** Ops.Trigger.TriggerButton

**Description:** simple button to trigger manually

> **Input Ports:**

- **Trigger** (Trigger)

< **Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TriggerButton>

## 97.1.23 TriggerCounter



**Full Name:** Ops.Trigger.TriggerCounter

**Description:** Counts how often the port was triggered

> **Input Ports:**

- **Exe** (Trigger)

- **Reset** (Trigger)

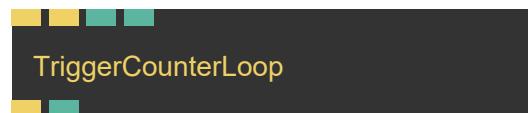
< **Output Ports:**

- **Trigger** (Trigger)
- **TimesTriggered** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TriggerCounter>

## 97.1.24 TriggerCounterLoop



**Full Name:** Ops.Trigger.TriggerCounterLoop

**Description:** Increments with each trigger and loops depending on min and max loop values.

> **Input Ports:**

- **Trigger In** (Trigger)
- **Reset** (Trigger)
- **Loop Min** (Number: Integer)
- **Loop Max** (Number: Integer)

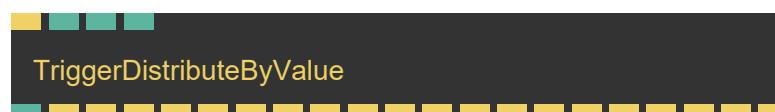
< **Output Ports:**

- **Trigger Out** (Trigger)
- **Current Count** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TriggerCounterLoop>

## 97.1.25 TriggerDistributeByValue



**Full Name:** Ops.Trigger.TriggerDistributeByValue

**Description:** triggers evenly distributed by value

> **Input Ports:**

- **Exe** (Trigger)
- **Number** (Number)
- **Max** (Number)
- **Num Outputs** (Number)

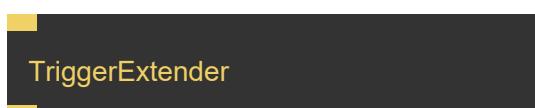
**< Output Ports:**

- **Num** (Number)
- **Trigger 0** (Trigger)
- **Trigger 1** (Trigger)
- **Trigger 2** (Trigger)
- **Trigger 3** (Trigger)
- **Trigger 4** (Trigger)
- **Trigger 5** (Trigger)
- **Trigger 6** (Trigger)
- **Trigger 7** (Trigger)
- **Trigger 8** (Trigger)
- **Trigger 9** (Trigger)
- **Trigger 10** (Trigger)
- **Trigger 11** (Trigger)
- **Trigger 12** (Trigger)
- **Trigger 13** (Trigger)
- **Trigger 14** (Trigger)
- **Trigger 15** (Trigger)
- **Trigger 16** (Trigger)
- **Trigger 17** (Trigger)
- **Trigger 18** (Trigger)
- **Trigger 19** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TriggerDistributeByValue>

## 97.1.26 TriggerExtender



**Full Name:** Ops.Trigger.TriggerExtender

**Description:** Extends a trigger (useful in big patches for better overview)

**> Input Ports:**

- **Execute** (Trigger)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TriggerExtender>

## 97.1.27 TriggerIfDecreased



**Full Name:** Ops.Trigger.TriggerIfDecreased

**Description:** trigger if a value decreases / gets smaller

**> Input Ports:**

- **Value** (Number)

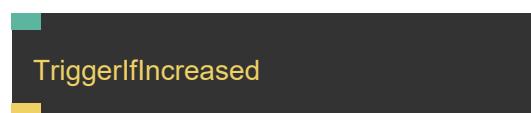
**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TriggerIfDecreased>

## 97.1.28 TriggerIfIncreased



**Full Name:** Ops.Trigger.TriggerIfIncreased

**Description:** Outputs a trigger if the value of a number increases

**> Input Ports:**

- **Value** (Number)

**< Output Ports:**

- **Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TriggerIfIncreased>

## 97.1.29 TriggerLimiter



**Full Name:** Ops.Trigger.TriggerLimiter

**Description:** Limits how often a trigger goes through to x ms

**> Input Ports:**

- **In Trigger** (Trigger)
- **Milliseconds** (Number)

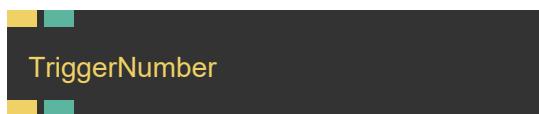
**< Output Ports:**

- **Out Trigger** (Trigger)
- **Progress** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TriggerLimiter>

## 97.1.30 TriggerNumber



**Full Name:** Ops.Trigger.TriggerNumber

**Description:** Outputs a number when triggered

**> Input Ports:**

- **Set** (Trigger)
- **Number** (Number)

**< Output Ports:**

- **Next** (Trigger)
- **Out Value** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TriggerNumber>

## 97.1.31 TriggerOnce



**Full Name:** Ops.Trigger.TriggerOnce

**Description:** Trigger the following children once

**> Input Ports:**

- **Exec** (Trigger)
- **Reset** (Trigger)

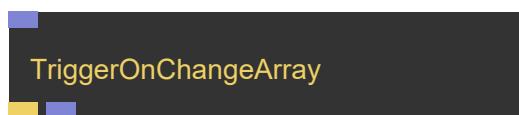
**< Output Ports:**

- **Next** (Trigger)
- **Was Triggered** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TriggerOnce>

## 97.1.32 TriggerOnChangeArray\_v2



**Full Name:** Ops.Trigger.TriggerOnChangeArray\_v2

**Description:** triggers when array has changed

**> Input Ports:**

- **Array** (Array)

**< Output Ports:**

- **Changed** (Trigger)
- **Result** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Trigger.TriggerOnChangeArray\\_v2](https://cables.gl/op/Ops.Trigger.TriggerOnChangeArray_v2)

### 97.1.33 TriggerOnChangeObject\_v2



**Full Name:** Ops.Trigger.TriggerOnChangeObject\_v2

**Description:** triggers when Object has changed

**> Input Ports:**

- **Object** (Object)

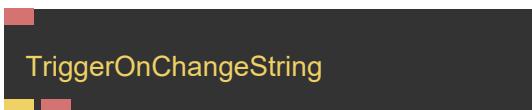
**< Output Ports:**

- **Changed** (Trigger)
- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Trigger.TriggerOnChangeObject\\_v2](https://cables.gl/op/Ops.Trigger.TriggerOnChangeObject_v2)

### 97.1.34 TriggerOnChangeString\_v2



**Full Name:** Ops.Trigger.TriggerOnChangeString\_v2

**Description:** triggers when string has changed

**> Input Ports:**

- **String** (String)

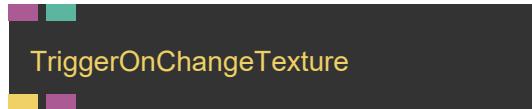
**< Output Ports:**

- **Changed** (Trigger)
- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Trigger.TriggerOnChangeString\\_v2](https://cables.gl/op/Ops.Trigger.TriggerOnChangeString_v2)

### 97.1.35 TriggerOnChangeTexture



**Full Name:** Ops.Trigger.TriggerOnChangeTexture

**Description:** triggers when texture has changed

**> Input Ports:**

- **Texture** (Object:Texture)

**< Output Ports:**

- **Changed** (Trigger)
- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TriggerOnChangeTexture>

### 97.1.36 TriggerReceive



**Full Name:** Ops.Trigger.TriggerReceive

**Description:** Receives triggers from a TriggerSend op with the same variable name

**> Input Ports:**

- Visit *Ops.Trigger.TriggerReceive documentation* for input port details

**< Output Ports:**

- **Triggered** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TriggerReceive>

## 97.1.37 TriggerReceiveFilter

TriggerReceiveFilter

**Full Name:** Ops.Trigger.TriggerReceiveFilter

**Description:** receives all named triggers and relays them, optionally using a filter-prefix on the name

**> Input Ports:**

- **Prefix** (String)

**< Output Ports:**

- **Trigger Out** (Trigger)
- **Trigger Name** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TriggerReceiveFilter>

## 97.1.38 TriggerSend

TriggerSend

**Full Name:** Ops.Trigger.TriggerSend

**Description:** Allows triggers to be sent to a TriggerReceive op with the same variable name

**> Input Ports:**

- **Trigger** (Trigger)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TriggerSend>

## 97.1.39 TriggerSendNamed

TriggerSendNamed

**Full Name:** Ops.Trigger.TriggerSendNamed

**Description:** Allows triggers to be sent to a TriggerReceive op with the same variable name

**> Input Ports:**

- **Trigger** (Trigger)
- **Named Trigger** (String)

**< Output Ports:**

• Visit *Ops.Trigger.TriggerSendNamed documentation* for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TriggerSendNamed>

## 97.1.40 TriggersPerSecond

TriggersPerSecond

**Full Name:** Ops.Trigger.TriggersPerSecond

**Description:** Counts how often the port is triggered per second

**> Input Ports:**

- **Exe** (Trigger)

**< Output Ports:**

- **Cps** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TriggersPerSecond>

## 97.1.41 TriggerString



**Full Name:** Ops.Trigger.TriggerString

**Description:** trigger a string

**> Input Ports:**

- **Trigger** (Trigger)
- **String** (String)

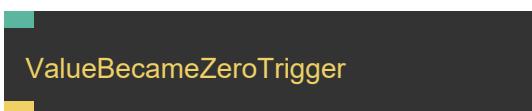
**< Output Ports:**

- **Next** (Trigger)
- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.TriggerString>

## 97.1.42 ValueBecameZeroTrigger



**Full Name:** Ops.Trigger.ValueBecameZeroTrigger

**Description:** Triggers when the input value became zero

**> Input Ports:**

- **Value** (Number)

**< Output Ports:**

- **Became Zero Trigger** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Trigger.ValueBecameZeroTrigger>

# 98 Ops.Ui

## 98.1 Ops.Ui

### 98.1.1 Area



**Full Name:** Ops.Ui.Area

**Description:** Organize and group your patch operators

**> Input Ports:**

- **Delete** (Trigger)

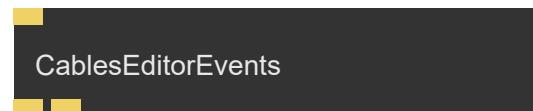
**< Output Ports:**

- Visit *Ops.Ui.Area documentation* for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.Area>

### 98.1.2 CablesEditorEvents



**Full Name:** Ops.Ui.CablesEditorEvents

**Description:** Cables UI Event Triggers

**> Input Ports:**

- **Set Changed Patch** (Trigger)

**< Output Ports:**

- **Saving Patch** (Trigger)
- **PortValueEdited** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.CablesEditorEvents>

### 98.1.3 Comment\_v2

Comment

**Full Name:** Ops.Ui.Comment\_v2

**Description:** Displays a comment in the patch area

**> Input Ports:**

- Visit *Ops.Ui.Comment\_v2 documentation for input port details*

**< Output Ports:**

- Visit *Ops.Ui.Comment\_v2 documentation for output port details*

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Ui.Comment\\_v2](https://cables.gl/op/Ops.Ui.Comment_v2)

### 98.1.4 GetCablesDefaultTheme

GetCablesDefaultTheme

**Full Name:** Ops.Ui.GetCablesDefaultTheme

**Description:** Get the default theme colors of the cables editor

**> Input Ports:**

- Visit *Ops.Ui.GetCablesDefaultTheme documentation for input port details*

**< Output Ports:**

- **Theme** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.GetCablesDefaultTheme>

### 98.1.5 MaximizeRenderer

MaximizeRenderer

**Full Name:** Ops.Ui.MaximizeRenderer

**Description:** maximize renderer to window size

**> Input Ports:**

- **Toggle Maximized** (Trigger)

**< Output Ports:**

- **Maximized** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.MaximizeRenderer>

### 98.1.6 PatchInput

PatchInput

**Full Name:** Ops.Ui.PatchInput

**Description:** Helper op for sub-patches

**> Input Ports:**

- Visit *Ops.Ui.PatchInput documentation for input port details*

**< Output Ports:**

- **Create Port** (Dynamic)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.PatchInput>

### 98.1.7 PatchOutput

PatchOutput

**Full Name:** Ops.Ui.PatchOutput

**Description:** Helper op for sub-patches

**> Input Ports:**

- **Create Port** (Dynamic)

**< Output Ports:**

- Visit *Ops.Ui.PatchOutput* documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.PatchOutput>

## 98.1.8 SetCablesTheme

SetCablesTheme

**Full Name:** Ops.Ui.SetCablesTheme

**Description:** Set cables editor colors

**> Input Ports:**

- **Theme** (Object)

**< Output Ports:**

- **Missing** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.SetCablesTheme>

## 98.1.9 SubPatch

SubPatch

**Full Name:** Ops.Ui.SubPatch

**Description:** Visit documentation for details

**> Input Ports:**

- **Create Port** (Dynamic)
- **DataStr** (Number)
- **PatchId** (Number)

**< Output Ports:**

- **Create Port Out** (Dynamic)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.SubPatch>

## 98.1.10 Subpatch2Template

Subpatch2Template

**Full Name:** Ops.Ui.Subpatch2Template

**Description:** Visit documentation for details

**> Input Ports:**

- **PatchId** (String)
- **Public** (3): 1

**< Output Ports:**

- Visit *Ops.Ui.Subpatch2Template* documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.Subpatch2Template>

## 98.1.11 SubPatchInput

SubPatchInput

**Full Name:** Ops.Ui.SubPatchInput

**Description:** Visit documentation for details

**> Input Ports:**

- Visit *Ops.Ui.SubPatchInput* documentation for input port details

**< Output Ports:**

- **A1jf8yr1w** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.SubPatchInput>

## 98.1.12 SubPatchOutput

SubPatchOutput

**Full Name:** Ops.Ui.SubPatchOutput

**Description:** Visit documentation for details

**> Input Ports:**

- Visit *Ops.Ui.SubPatchOutput* documentation for input port details

**< Output Ports:**

- Visit *Ops.Ui.SubPatchOutput* documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.SubPatchOutput>

## 98.1.13 VizArrayChart



**Full Name:** Ops.Ui.VizArrayChart

**Description:** Displays information of the distribution of numerical values in an array

**> Input Ports:**

- **Array Numbers** (Array)
- **Titles** (Array)

**< Output Ports:**

- Visit *Ops.Ui.VizArrayChart* documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.VizArrayChart>

## 98.1.14 VizArrayGraph



**Full Name:** Ops.Ui.VizArrayGraph

**Description:** Visualize Array as line graph

**> Input Ports:**

- **Array Numbers** (Array)

- **Curve** (Number: Boolean)

**< Output Ports:**

- **Passthrough Array** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.VizArrayGraph>

## 98.1.15 VizArrayTable\_v2



**Full Name:** Ops.Ui.VizArrayTable\_v2

**Description:** Show the contents of the input array in a table in the patch, useful for debugging

**> Input Ports:**

- **Array** (Array)
- **Stride** (Number: Integer)
- **Scroll** (Number)

**< Output Ports:**

- **Passthrough Array** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Ui.VizArrayTable\\_v2](https://cables.gl/op/Ops.Ui.VizArrayTable_v2)

## 98.1.16 VizBool



**Full Name:** Ops.Ui.VizBool

**Description:** Visualize the state of a boolean input in the patch, useful for debugging

**> Input Ports:**

- **Boolean** (Number: Boolean)

#### < Output Ports:

- **Bool** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.VizBool>

## 98.1.17 VizGraph



**Full Name:** Ops.Ui.VizGraph

**Description:** Displays graphs for the numbers on the input port in the patch-field

#### > Input Ports:

- **Number 1** (Number)
- **Number 2** (Number)
- **Number 3** (Number)
- **Number 4** (Number)
- **Number 5** (Number)
- **Number 6** (Number)
- **Number 7** (Number)
- **Number 8** (Number)
- **Fill Graph** (Number: Boolean)
- **Reset** (Trigger)

#### < Output Ports:

- Visit *Ops.Ui.VizGraph documentation* for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.VizGraph>

## 98.1.18 VizImageUrl



**Full Name:** Ops.Ui.VizImageUrl

**Description:** preview an image URL or a data/base64 URL

#### > Input Ports:

- **File** (String)

#### < Output Ports:

- **Width** (Number)
- **Height** (Number)
- **Loading** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.VizImageUrl>

## 98.1.19 VizLogger



**Full Name:** Ops.Ui.VizLogger

**Description:** Log changes of input values line by line, use like a logfile for debugging

#### > Input Ports:

- **Number** (Number)
- **String** (String)
- **Object** (Object)
- **Clear** (Trigger)

#### < Output Ports:

- Visit *Ops.Ui.VizLogger documentation* for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.VizLogger>

## 98.1.20 VizNumber



**Full Name:** Ops.Ui.VizNumber

**Description:** Displays input string on the patchfield

**> Input Ports:**

- **Number** (Number)

**< Output Ports:**

- **Result** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.VizNumber>

- **Line Numbers** (Number: Boolean)

- **Experimental Stringify** (Number: Boolean)

- **Sort Keys** (Number: Boolean)

- **Font Size** (Number)

- **Scroll** (Number)

**< Output Ports:**

- Visit *Ops.Ui.VizObject* documentation for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.VizObject>

## 98.1.21 VizNumberBar



**Full Name:** Ops.Ui.VizNumberBar

**Description:** Visualize numbers as a bar in patch, useful for debugging

**> Input Ports:**

- **Number** (Number)

**< Output Ports:**

- **Passthrough** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.VizNumberBar>

## 98.1.22 VizObject



**Full Name:** Ops.Ui.VizObject

**Description:** Show information about any object for patch debugging

**> Input Ports:**

- **Object** (Object)
- **ZoomText** (Number: Boolean)

## 98.1.23 VizString



**Full Name:** Ops.Ui.VizString

**Description:** Displays long input string on the patchfield

**> Input Ports:**

- **String** (String)
- **ZoomText** (Number: Boolean)
- **Line Numbers** (Number: Boolean)
- **Whitespace** (Number: Boolean)
- **Wrap Lines** (Number: Boolean)
- **Syntax Index** (Number: Integer)
- **Font Size** (Number)
- **Scroll** (Number)

**< Output Ports:**

- **Passthrough String** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.VizString>

## 98.1.24 VizTexture



**Full Name:** Ops.Ui.VizTexture

**Description:** Displays texture at input port

**> Input Ports:**

- **Texture In** (Object:Texture)
- **Show Info** (Number: Boolean)
- **Show Color** (Number: Boolean)
- **X** (Number)
- **Y** (Number)

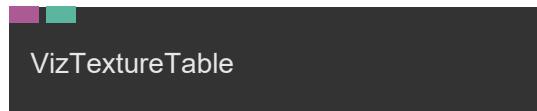
**< Output Ports:**

- **Texture Out** (Object)
- **Info** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.VizTexture>

## 98.1.25 VizTextureTable



**Full Name:** Ops.Ui.VizTextureTable

**Description:** Show pixel colors of connected texture as a table, useful for debugging

**> Input Ports:**

- **Texture** (Object:Texture)
- **Row Start** (Number: Integer)

**< Output Ports:**

- *Visit Ops.Ui.VizTextureTable documentation for output port details*

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.VizTextureTable>

## 98.1.26 VizTrigger



**Full Name:** Ops.Ui.VizTrigger

**Description:** Visualize triggering for debugging reasons

**> Input Ports:**

- **Trigger** (Trigger)
- **Reset** (Trigger)
- **Count Overlay** (Number: Boolean)

**< Output Ports:**

- **Count** (Number)
- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.VizTrigger>

# 99 Ops.Ui.Debug

---

## 99.1 Ops.Ui.Debug

### 99.1.1 PatchUiCoords



**Full Name:** Ops.Ui.Debug.PatchUiCoords

**Description:** Output the current patch coordinates

**> Input Ports:**

- Visit *Ops.Ui.Debug.PatchUiCoords documentation for input port details*

**< Output Ports:**

- X (Number)
- Y (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.Debug.PatchUiCoords>

# 100 Ops.Ui.Routing

---

## 100.1 Ops.Ui.Routing

### 100.1.1 RouteArray



**Full Name:** Ops.Ui.Routing.RouteArray

**Description:** Patchfield cable routing helper for array cables

**> Input Ports:**

- Array In (Array)

**< Output Ports:**

- Array Out (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.Routing.RouteArray>

### 100.1.2 RouteNumber



**Full Name:** Ops.Ui.Routing.RouteNumber

**Description:** Patchfield cable routing helper for number cables

**> Input Ports:**

- Value (Number)

**< Output Ports:**

- Result (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.Routing.RouteNumber>

## 100.1.3 RouteObject



RouteObject

**Full Name:** Ops.Ui.Routing.RouteObject

**Description:** Patchfield cable routing helper for object cables

**> Input Ports:**

- **Array In** (Object)

**< Output Ports:**

- **Array Out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.Routing.RouteObject>

**Full Name:** Ops.Ui.Routing.RouteTrigger

**Description:** Routing Helper for trigger cables

**> Input Ports:**

- **Trigger** (Trigger)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.Routing.RouteTrigger>

## 100.1.4 RouteString



RouteString

**Full Name:** Ops.Ui.Routing.RouteString

**Description:** Patchfield cable routing helper for string cables

**> Input Ports:**

- **Value** (String)

**< Output Ports:**

- **String** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Ui.Routing.RouteString>

## 100.1.5 RouteTrigger



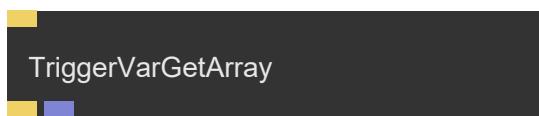
RouteTrigger

# 101 Ops.Vars

Docs: <https://cables.gl/op/Ops.Vars.TriggerVarGetNumber>

## 101.1 Ops.Vars

### 101.1.1 TriggerVarGetArray



**Full Name:** Ops.Vars.TriggerVarGetArray

**Description:** Get an array variable value at time of trigger

**> Input Ports:**

- **Update** (Trigger)

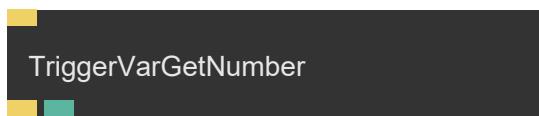
**< Output Ports:**

- **Next** (Trigger)
- **Value** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Vars.TriggerVarGetArray>

### 101.1.2 TriggerVarGetNumber



**Full Name:** Ops.Vars.TriggerVarGetNumber

**Description:** Get a number variable value at time of trigger

**> Input Ports:**

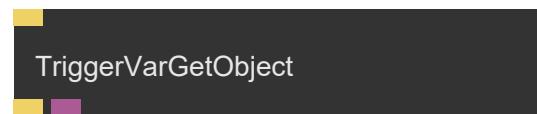
- **Update** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **Value** (Number)

**Example Patch:** Open in Editor

### 101.1.3 TriggerVarGetObject



**Full Name:** Ops.Vars.TriggerVarGetObject

**Description:** Get an object variable value at time of trigger

**> Input Ports:**

- **Update** (Trigger)

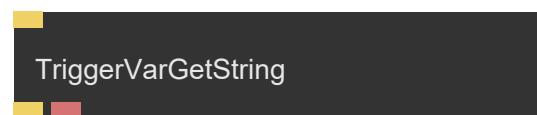
**< Output Ports:**

- **Next** (Trigger)
- **Value** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Vars.TriggerVarGetObject>

### 101.1.4 TriggerVarGetString



**Full Name:** Ops.Vars.TriggerVarGetString

**Description:** Get a string variable value at time of trigger

**> Input Ports:**

- **Update** (Trigger)

**< Output Ports:**

- **Next** (Trigger)
- **Value** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Vars.TriggerVarGetString>

## 101.1.5 VarGetArray\_v2



VarGetArray

**Full Name:** Ops.Vars.VarGetArray\_v2

**Description:** Get a variable array

**> Input Ports:**

- Visit *Ops.Vars.VarGetArray\_v2 documentation* for input port details

**< Output Ports:**

- **Value** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Vars.VarGetArray\\_v2](https://cables.gl/op/Ops.Vars.VarGetArray_v2)

## 101.1.6 VarGetNumber\_v2



VarGetNumber

**Full Name:** Ops.Vars.VarGetNumber\_v2

**Description:** read a variable number

**> Input Ports:**

- Visit *Ops.Vars.VarGetNumber\_v2 documentation* for input port details

**< Output Ports:**

- **Value** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Vars.VarGetNumber\\_v2](https://cables.gl/op/Ops.Vars.VarGetNumber_v2)

## 101.1.7 VarGetObject\_v2



VarGetObject

**Full Name:** Ops.Vars.VarGetObject\_v2

**Description:** Get a variable object

**> Input Ports:**

- **Variable** (Number: String)

**< Output Ports:**

- **Value** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Vars.VarGetObject\\_v2](https://cables.gl/op/Ops.Vars.VarGetObject_v2)

## 101.1.8 VarGetString



VarGetString

**Full Name:** Ops.Vars.VarGetString

**Description:** String variable getter

**> Input Ports:**

- Visit *Ops.Vars.VarGetString documentation* for input port details

**< Output Ports:**

- **Value** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Vars.VarGetString>

## 101.1.9 VarGetTexture\_v2



VarGetTexture

**Full Name:** Ops.Vars.VarGetTexture\_v2

**Description:** get a texture from a variable

**> Input Ports:**

- Visit *Ops.Vars.VarGetTexture\_v2 documentation* for input port details

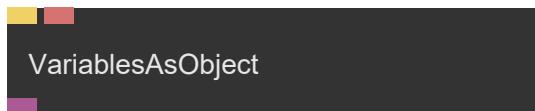
**< Output Ports:**

- **Value** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Vars.VarGetTexture\\_v2](https://cables.gl/op/Ops.Vars.VarGetTexture_v2)

## 101.1.10 VariablesAsObject



**Full Name:** Ops.Vars.VariablesAsObject

**Description:** outputs an object containing all variables

**> Input Ports:**

- **Execute** (Trigger)
- **Filter Prefix** (String)

**< Output Ports:**

- **Result** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Vars.VariablesAsObject>

## 101.1.11 VarSetArray\_v2



**Full Name:** Ops.Vars.VarSetArray\_v2

**Description:** Set a variable array

**> Input Ports:**

- **Value** (Array)

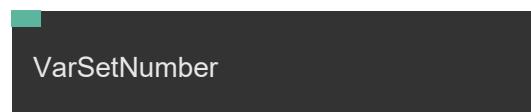
**< Output Ports:**

- Visit [Ops.Vars.VarSetArray\\_v2 documentation for output port details](#)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Vars.VarSetArray\\_v2](https://cables.gl/op/Ops.Vars.VarSetArray_v2)

## 101.1.12 VarSetNumber\_v2



**Full Name:** Ops.Vars.VarSetNumber\_v2

**Description:** set a variable number

**> Input Ports:**

- **Value** (Number)

**< Output Ports:**

- Visit [Ops.Vars.VarSetNumber\\_v2 documentation for output port details](#)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Vars.VarSetNumber\\_v2](https://cables.gl/op/Ops.Vars.VarSetNumber_v2)

## 101.1.13 VarSetObject\_v2



**Full Name:** Ops.Vars.VarSetObject\_v2

**Description:** Set a variable object

**> Input Ports:**

- **Value** (Object)

**< Output Ports:**

- Visit [Ops.Vars.VarSetObject\\_v2 documentation for output port details](#)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Vars.VarSetObject\\_v2](https://cables.gl/op/Ops.Vars.VarSetObject_v2)

## 101.1.14 VarSetString\_v2



**Full Name:** Ops.Vars.VarSetString\_v2

**Description:** Set string variable

**> Input Ports:**

- **Value** (String)

**< Output Ports:**

- Visit *Ops.Vars.VarSetString\_v2 documentation for output port details*

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Vars.VarSetString\\_v2](https://cables.gl/op/Ops.Vars.VarSetString_v2)

## 101.1.15 VarSetTexture\_v2



**Full Name:** Ops.Vars.VarSetTexture\_v2

**Description:** set a texture variable

**> Input Ports:**

- **Value** (Object:Texture)

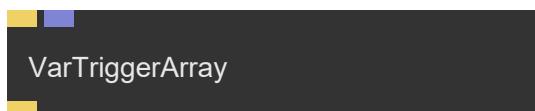
**< Output Ports:**

- Visit *Ops.Vars.VarSetTexture\_v2 documentation for output port details*

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Vars.VarSetTexture\\_v2](https://cables.gl/op/Ops.Vars.VarSetTexture_v2)

## 101.1.16 VarTriggerArray



**Full Name:** Ops.Vars.VarTriggerArray

**Description:** Set an array variable by a trigger

**> Input Ports:**

- **Trigger** (Trigger)
- **Value** (Array)

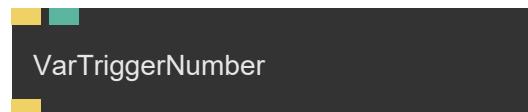
**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Vars.VarTriggerArray>

## 101.1.17 VarTriggerNumber



**Full Name:** Ops.Vars.VarTriggerNumber

**Description:** set number variable by trigger

**> Input Ports:**

- **Trigger** (Trigger)
- **Value** (Number)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Vars.VarTriggerNumber>

## 101.1.18 VarTriggerObject



**Full Name:** Ops.Vars.VarTriggerObject

**Description:** Set an object variable by trigger

**> Input Ports:**

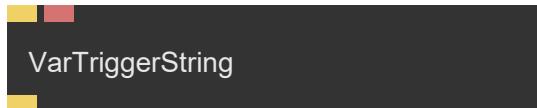
- **Trigger** (Trigger)
- **Value** (Object)

**< Output Ports:**

- **Next** (Trigger)

**Example Patch:** Open in Editor

## 101.1.19 VarTriggerString



**Full Name:** Ops.Vars.VarTriggerString

**Description:** set string variable by trigger

### > Input Ports:

- **Trigger** (Trigger)
- **Value** (String)

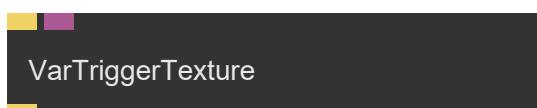
### < Output Ports:

- **Next** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Vars.VarTriggerString>

## 101.1.20 VarTriggerTexture



**Full Name:** Ops.Vars.VarTriggerTexture

**Description:** Set an object variable by trigger

### > Input Ports:

- **Trigger** (Trigger)
- **Value** (Object:Texture)

### < Output Ports:

- **Next** (Trigger)

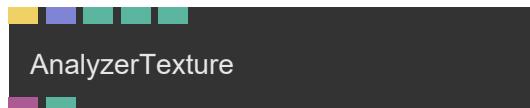
**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Vars.VarTriggerTexture>

# 102 Ops.WebAudio

## 102.1 Ops.WebAudio

### 102.1.1 AnalyzerTexture\_v2



**Full Name:** Ops.WebAudio.AnalyzerTexture\_v2

**Description:** Creates a spectrogram texture from an audio FFT array

### > Input Ports:

- **Refresh** (Trigger)
- **FFT Array** (Array)
- **Mirror Active** (Number: Boolean)
- **Mirror Width** (Number)
- **Texture Size Index** (Number: Integer)

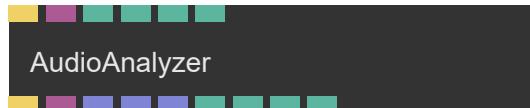
### < Output Ports:

- **Texture Out** (Object)
- **Position** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.WebAudio.AnalyzerTexture\\_v2](https://cables.gl/op/Ops.WebAudio.AnalyzerTexture_v2)

### 102.1.2 AudioAnalyzer\_v2



**Full Name:** Ops.WebAudio.AudioAnalyzer\_v2

**Description:** Extracts FFT, RMS & Waveform data from an incoming audio signal

### > Input Ports:

- **Trigger In** (Trigger)

- **Audio In** (Object:AudioNode)
- **FFT Size Index** (Number: Integer)
- **Smoothing** (Number)
- **Range** (in dBFS)
- **Min** (Number)
- **Max** (Number)

**< Output Ports:**

- **Trigger Out** (Trigger)
- **Audio Out** (Object)
- **FFT Array** (Array)
- **Waveform Array** (Array)
- **Frequencies By Index Array** (Array)
- **Array Length** (Number)
- **Average Volume** (Number)
- **Average Volume Time-Domain** (Number)
- **RMS Volume** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.WebAudio.AudioAnalyzer\\_v2](https://cables.gl/op/Ops.WebAudio.AudioAnalyzer_v2)

## 102.1.3 AudioBuffer\_v3



**Full Name:** Ops.WebAudio.AudioBuffer\_v3

**Description:** Holds an audio file / sample in a buffer

**> Input Ports:**

- **URL** (String)
- **Create Loading Task** (Number: Boolean)
- **Active** (Number: Boolean)

**< Output Ports:**

- **Audio Buffer** (Object)
- **Finished Loading** (booleanNumber)
- **Sample Rate** (Number)
- **Length** (Number)
- **Duration** (Number)

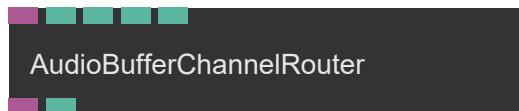
- **Number Of Channels** (Number)

- **IsLoading** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.WebAudio.AudioBuffer\\_v3](https://cables.gl/op/Ops.WebAudio.AudioBuffer_v3)

## 102.1.4 AudioBufferChannelRouter



**Full Name:** Ops.WebAudio.AudioBufferChannelRouter

**Description:** Route audio from one input channel to any output channel

**> Input Ports:**

- **Audio Buffer** (Object:AudioBuffer)
- **Channel In** (Number: Integer)
- **Channel Out** (Number: Integer)
- **Clear Others** (Number: Boolean)
- **Channel Offset** (Number: Boolean)

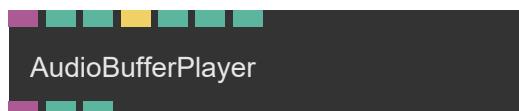
**< Output Ports:**

- **Audio Buffer Out** (Object)
- **Output Channels** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.WebAudio.AudioBufferChannelRouter>

## 102.1.5 AudioBufferPlayer\_v2



**Full Name:** Ops.WebAudio.AudioBufferPlayer\_v2

**Description:** Play back audio data stored in an AudioBuffer

**> Input Ports:**

- **Audio Buffer** (Object:AudioBuffer)
- **Loop** (Number: Boolean)

- **Restart** (Trigger)
- **Offset** (Number)
- **Playback Rate** (Number)
- **Detune** (Number)

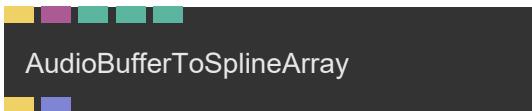
**< Output Ports:**

- **Audio Out** (Object)
- **Is Playing** (booleanNumber)
- **Loading** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.WebAudio.AudioBufferPlayer\\_v2](https://cables.gl/op/Ops.WebAudio.AudioBufferPlayer_v2)

## 102.1.6 AudioBufferToSplineArray



AudioBufferToSplineArray

**Full Name:** Ops.WebAudio.AudioBufferToSplineArray

**Description:** Outputs the waveform of an audio file as a spline array

**> Input Ports:**

- **Render** (Trigger)
- **Audio Buffer** (Object:AudioBuffer)
- **Width** (Number)
- **Height** (Number)
- **Samples Per Pixel** (Number: Integer)

**< Output Ports:**

- **Next** (Trigger)
- **Array Out** (Array)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.WebAudio.AudioBufferToSplineArray>

## 102.1.7 AudioPanner



AudioPanner

**Full Name:** Ops.WebAudio.AudioPanner

**Description:** stereo pan an audio signal from left to right

**> Input Ports:**

- **Audio In** (Object:AudioNode)
- **Pan** (Number)

**< Output Ports:**

- **Audio Out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.WebAudio.AudioPanner>

## 102.1.8 AudioRecorder



AudioRecorder

**Full Name:** Ops.WebAudio.AudioRecorder

**Description:** record, playback and download audio

**> Input Ports:**

- **Audio In** (Object:AudioNode)
- **Start Recording** (Trigger)
- **Stop Recording** (Trigger)
- **Input Gain** (Number)
- **Start Playback** (Trigger)
- **Stop Playback** (Trigger)
- **Clear Buffer** (Trigger)
- **Playback Gain** (Number)
- **Loop Playback** (Number: Boolean)

**< Output Ports:**

- **Audio Out** (Object)
- **Recorded Audio Out** (Object)
- **Is Recording** (booleanNumber)
- **Is Playing Back** (booleanNumber)
- **State** (String)
- **AudioBuffer Out** (Object)
- **Data URL** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.WebAudio.AudioRecorder>

## 102.1.9 BiquadFilter\_v2



**Full Name:** Ops.WebAudio.BiquadFilter\_v2

**Description:** Different kinds of audio filters

### > Input Ports:

- **Audio In** (Object:AudioNode)
- **Type Index** (Number: Integer)
- **Frequency** (Number)
- **Q** (Number)
- **Gain** (Number)
- **Detune** (in cents)
- **Frequency Array** (Array)

### < Output Ports:

- **Audio Out** (Object)
- **Magnitude Response Array** (Array)
- **Phase Response Array** (Array)
- **Response Arrays Length** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.WebAudio.BiquadFilter\\_v2](https://cables.gl/op/Ops.WebAudio.BiquadFilter_v2)

## 102.1.10 ClockSequencer



**Full Name:** Ops.WebAudio.ClockSequencer

**Description:** send bpm based triggers like a clocked trigger sequencer / clock divider

### > Input Ports:

- **BPM** (Number: Integer)
- **beats per minute** (tempo)
- **Start** (Trigger)
- **Stop** (Trigger)
- **Reset** (Trigger)

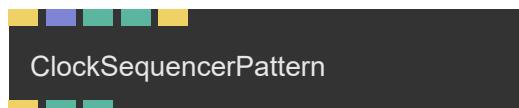
### < Output Ports:

- **Sequencer Running** (booleanNumber)
- **BPM Out** (Number)
- **Start Out** (Trigger)
- **Stop Out** (Trigger)
- **Reset Out** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.WebAudio.ClockSequencer>

## 102.1.11 ClockSequencerPattern



**Full Name:** Ops.WebAudio.ClockSequencerPattern

**Description:** sequence triggers by defining a pattern (like a drum machine)

### > Input Ports:

- **Clock Trigger Input** (Trigger)
- **Sequence Array** (Array)
- **Steps Index** (Number: Integer)
- **Steps** (Number: String)
- **Reset** (Trigger)

### < Output Ports:

- **Sequence Trigger Output** (Trigger)
- **Sequenced Value** (Number)
- **Current Step** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.WebAudio.ClockSequencerPattern>

## 102.1.12 Convolver\_v2



**Full Name:** Ops.WebAudio.Convolver\_v2

**Description:** Audio reverb using an impulse response (sample)

### > Input Ports:

- **Audio In** (Object:AudioNode)
- **Impulse Response** (String)
- **Normalize** (Number: Boolean)
- **IR Gain** (Number)
- **Output Gain** (Number)

### < Output Ports:

- **Audio Out** (Object)
- **Wet Out** (Object)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.WebAudio.Convolver\\_v2](https://cables.gl/op/Ops.WebAudio.Convolver_v2)

## 102.1.13 CutFilter



**Full Name:** Ops.WebAudio.CutFilter

**Description:** dj style filter (lowpass and highpass)

### > Input Ports:

- **Audio In** (Object:AudioNode)
- **Highpass Active** (Number: Boolean)
- **Low Frequency** (Number)
- **Low Q** (Number)
- **Lowpass Active** (Number: Boolean)
- **High Frequency** (Number)
- **High Q** (Number)

### < Output Ports:

- **Audio Out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.WebAudio.CutFilter>

## 102.1.14 Delay



**Full Name:** Ops.WebAudio.Delay

**Description:** add a delay effect to an audio stream

### > Input Ports:

- **Audio In** (Object:AudioNode)
- **Feedback** (Number)
- **BPM Based Delay Time** (Number: Boolean)
- **BPM** (Number)
- **Highpass Frequency** (Number)
- **Highpass Q** (Number)
- **Lowpass Frequency** (Number)
- **Lowpass Q** (Number)
- **LFO Intensity** (Number)
- **LFO Waveform Index** (Number: Integer)

### < Output Ports:

- **Mix Out** (Object)
- **Wet Out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.WebAudio.Delay>

## 102.1.15 FFTAreaAverage\_v3



**Full Name:** Ops.WebAudio.FFTAreaAverage\_v3

**Description:** get average value in an area of a fft audio analysis buffer

#### > Input Ports:

- **Refresh** (Trigger)
- **FFT Array** (Array)
- **X Position** (Number)
- **Y Position** (Number)
- **Width** (Number)
- **Height** (Number)
- **Create Texture** (Number: Boolean)

#### < Output Ports:

- **Texture Out** (Object)
- **Area Average Volume** (Number)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.WebAudio.FFTAreaAverage\\_v3](https://cables.gl/op/Ops.WebAudio.FFTAreaAverage_v3)

## 102.1.16 Gain



**Full Name:** Ops.WebAudio.Gain

**Description:** Changes the gain / volume

#### > Input Ports:

- **Audio In** (Object:AudioNode)
- **Gain** (Number)
- **Mute** (Number: Boolean)

#### < Output Ports:

- **Audio Out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.WebAudio.Gain>

## 102.1.17 KeyPiano



**Full Name:** Ops.WebAudio.KeyPiano

**Description:** Generates notes based on key presses

#### > Input Ports:

- **C Note On** (Trigger)
- **C Note Off** (Trigger)
- **Cis Note On** (Trigger)
- **Cis Note Off** (Trigger)
- **D Note On** (Trigger)
- **D Note Off** (Trigger)
- **Dis Note On** (Trigger)
- **Dis Note Off** (Trigger)
- **E Note On** (Trigger)
- **E Note Off** (Trigger)
- **F Note On** (Trigger)
- **F Note Off** (Trigger)
- **Fis Note On** (Trigger)
- **Fis Note Off** (Trigger)
- **G Note On** (Trigger)
- **G Note Off** (Trigger)
- **Gis Note Ons** (Trigger)
- **Gis Note Off** (Trigger)
- **A Note On** (Trigger)
- **A Note Off** (Trigger)
- **Ais Note On** (Trigger)
- **Ais Note Off** (Trigger)
- **B Note On** (Trigger)
- **B Note Off** (Trigger)
- **Octave** (Number)

#### < Output Ports:

- **Frequency** (Number)
- **Is Pressed** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.WebAudio.KeyPiano>

## 102.1.18 MicrophoneIn\_v2



**Full Name:** Ops.WebAudio.MicrophoneIn\_v2

**Description:** Access to the microphone and/or audio input devices

### > Input Ports:

- **Audio Input Index** (Number: Integer)
- **Volume** (Number)
- **Mute** (Number: Boolean)
- **Start** (Trigger)

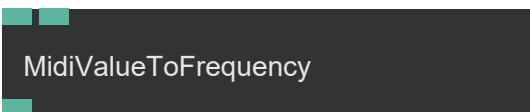
### < Output Ports:

- **Audio Out** (Object)
- **Listening** (booleanNumber)
- **List Of Input Devices** (Array)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.WebAudio.MicrophoneIn\\_v2](https://cables.gl/op/Ops.WebAudio.MicrophoneIn_v2)

## 102.1.19 MidiValueToFrequency



**Full Name:** Ops.WebAudio.MidiValueToFrequency

**Description:** Converts a midi value to a frequency

### > Input Ports:

- **MIDI Value** (Number)
- **Tuning** (Number)

### < Output Ports:

- **Frequency** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.WebAudio.MidiValueToFrequency>

## 102.1.20 Mixer



**Full Name:** Ops.WebAudio.Mixer

**Description:** Mix audio signals together

### > Input Ports:

- **Audio In 0** (Object:AudioNode)
- **Audio In 1** (Object:AudioNode)
- **Audio In 2** (Object:AudioNode)
- **Audio In 3** (Object:AudioNode)
- **Audio In 4** (Object:AudioNode)
- **Audio In 5** (Object:AudioNode)
- **Audio In 6** (Object:AudioNode)
- **Audio In 7** (Object:AudioNode)
- **In 0 Gain** (Number)
- **In 1 Gain** (Number)
- **In 2 Gain** (Number)
- **In 3 Gain** (Number)
- **In 4 Gain** (Number)
- **In 5 Gain** (Number)
- **In 6 Gain** (Number)
- **In 7 Gain** (Number)
- **In 0 Pan** (Number)
- **In 1 Pan** (Number)
- **In 2 Pan** (Number)
- **In 3 Pan** (Number)
- **In 4 Pan** (Number)
- **In 5 Pan** (Number)
- **In 6 Pan** (Number)
- **In 7 Pan** (Number)
- **Output Gain** (Number)

### < Output Ports:

- **Audio Out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.WebAudio.Mixer>

## 102.1.21 MusicalScales



**Full Name:** Ops.WebAudio.MusicalScales

**Description:** Outputs a musical scale array (major, minor, ...) as strings, steps and midi notes

### > Input Ports:

- **Root Note Index** (Number: Integer)
- **Root Note** (Number: String)
- **Scale Type Index** (Number: Integer)
- **Scale Type** (Number: String)
- **Include Upper Root Note** (Number: Boolean)
- **Octave** (Number: Integer)
- **the octave of the scale** (only for string & midi note outputs)
- **Append Octave To Names** (Number: Boolean)

### < Output Ports:

- **Note Names Array** (Array)
- **Note Step Number Array** (Array)
- **Midi Note Array** (Array)
- **Current Scale** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.WebAudio.MusicalScales>

## 102.1.22 Output\_v2



**Full Name:** Ops.WebAudio.Output\_v2

**Description:** Sends an audio signal to your speakers

### > Input Ports:

- **Audio In** (Object:AudioNode)
- **Volume** (Number)

- **Mute** (Number: Boolean)
- **Show Audio Suspended Button** (Number: Boolean)

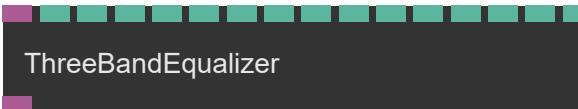
### < Output Ports:

- **Current Volume** (Number)
- **Number Of Channels** (Number)
- **Context State** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.WebAudio.Output\\_v2](https://cables.gl/op/Ops.WebAudio.Output_v2)

## 102.1.23 ThreeBandEqualizer



**Full Name:** Ops.WebAudio.ThreeBandEqualizer

**Description:** 3 filters in one - an eq to quickly process an audio signal

### > Input Ports:

- **Audio In** (Object:AudioNode)
- **Low Filter Type Index** (Number: Integer)
- **Low Filter Type** (Number: String)
- **Low Frequency** (Number)
- **Low Q** (Number)
- **Low Gain** (Number)
- **Mid Filter Type Index** (Number: Integer)
- **Mid Filter Type** (Number: String)
- **Mid Frequency** (Number)
- **Mid Q** (Number)
- **Mid Gain** (Number)
- **High Filter Type Index** (Number: Integer)
- **High Filter Type** (Number: String)
- **High Frequency** (Number)
- **High Q** (Number)
- **High Gain** (Number)

### < Output Ports:

- **Audio Out** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.WebAudio.ThreeBandEqualizer>

## 102.1.24 WaveformMesh



**Full Name:** Ops.WebAudio.WaveformMesh

**Description:** Outputs the waveform of an audio file as a geometry

### > Input Ports:

- **Render** (Trigger)
- **Audio Buffer** (Object:AudioBuffer)
- **Render Active** (Number: Boolean)
- **Show Bottom Half** (Number: Boolean)
- **Center Origin** (Number: Boolean)
- **Width** (Number)
- **Samples Per Pixel** (Number: Integer)
- **Calculate Tex Coords** (Number: Boolean)

### < Output Ports:

- **Spline Points** (Array)
- **Next** (Trigger)
- **Geometry** (Object)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.WebAudio.WaveformMesh>

## 102.1.25 Waveshaper



**Full Name:** Ops.WebAudio.Waveshaper

**Description:** add waveshaping (distortion, overdrive, fuzz) to an audio stream

### > Input Ports:

- **Audio In** (Object:AudioNode)

- **Oversampling Index** (Number: Integer)

- **Distortion Amount** (Number: Integer)

- **Waveshape Array In** (Array)

- **array input for the waveshaper** (custom distortion transfer function)

- **Output Gain** (Number)

### < Output Ports:

- **Audio Out** (Object)

- **Curve Out** (Array)

- **distortion curve array output** (one-dimensional)

- **Curve Length** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.WebAudio.Waveshaper>

# 103 Ops.Website

## 103.1 Ops.Website

### 103.1.1 Cookie



**Full Name:** Ops.Website.Cookie

**Description:** cookie of the current website as object

**> Input Ports:**

- Visit [Ops.Website.Cookie documentation](#) for input port details

**< Output Ports:**

- **Cookie** (Object)
- **Cookie String** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Website.Cookie>

### 103.1.2 FileInfo



**Full Name:** Ops.Website.FileInfo

**Description:** information about a filename, like url protocol, suffix etc

**> Input Ports:**

- **URL** (String)

**< Output Ports:**

- **Protocol** (String)
- **Host** (String)
- **FullPath** (String)

- **Filename** (String)
- **Basename** (String)
- **Suffix** (String)
- **Is URL** (String)
- **QueryParams** (String)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Website.FileInfo>

### 103.1.3 ForceHttps



**Full Name:** Ops.Website.ForceHttps

**Description:** will redirect to same URL using https protocol

**> Input Ports:**

- Visit [Ops.Website.ForceHttps documentation](#) for input port details

**< Output Ports:**

- Visit [Ops.Website.ForceHttps documentation](#) for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Website.ForceHttps>

### 103.1.4 InfoURL



**Full Name:** Ops.Website.InfoURL

**Description:** Information about the current URL

**> Input Ports:**

- Visit [Ops.Website.InfoURL documentation](#) for input port details

**< Output Ports:**

- **URL** (String)
- **Host** (String)

- **Hash** (String)
- **Pathname** (String)
- **Protocol** (String)
- **Port** (String)
- **Hash Changed** (Trigger)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Website.InfoURL>

## 103.1.5 InIframe



**Full Name:** Ops.Website.InIframe

**Description:** Outputs true if the patch is inside of an iframe

**> Input Ports:**

- Visit *Ops.Website.InIframe documentation for input port details*

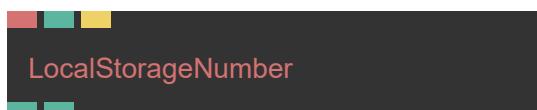
**< Output Ports:**

- **In Iframe** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Website.InIframe>

## 103.1.6 LocalStorageNumber



**Full Name:** Ops.Website.LocalStorageNumber

**Description:** Store and retreive a number in browser localstorage

**> Input Ports:**

- **Key** (String)
- **Number** (Number)
- **Store** (Trigger)

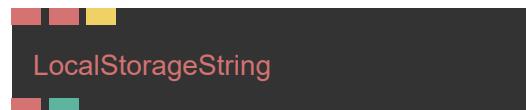
**< Output Ports:**

- **Stored Number** (Number)
- **Storage Support** (Number)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Website.LocalStorageNumber>

## 103.1.7 LocalStorageString



**Full Name:** Ops.Website.LocalStorageString

**Description:** Store and retreive a string in browser localstorage

**> Input Ports:**

- **Key** (String)
- **String** (String)
- **Store** (Trigger)

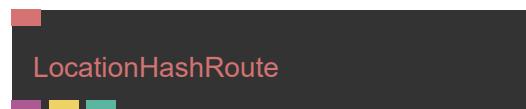
**< Output Ports:**

- **Stored String** (String)
- **Storage Support** (booleanNumber)

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Website.LocalStorageString>

## 103.1.8 LocationHashRoute



**Full Name:** Ops.Website.LocationHashRoute

**Description:** gives updated information about window.location.hash

**> Input Ports:**

- **Route** (String)
- **pattern for url and variables** (i.e. /scene/:id)

**< Output Ports:**

- **Values** (Object)

- **Changed** (Trigger)
- **Matching** (booleanNumber)

**Example Patch:** Open in Editor

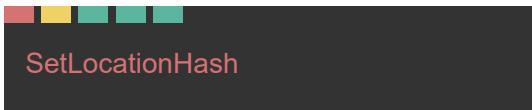
**Docs:** <https://cables.gl/op/Ops.Website.LocationHashRoute>

- **Result** (String)

**Example Patch:** Open in Editor

**Docs:** [https://cables.gl/op/Ops.Website.UrlqueryParams\\_v2](https://cables.gl/op/Ops.Website.UrlqueryParams_v2)

## 103.1.9 SetLocationHash



**Full Name:** Ops.Website.SetLocationHash

**Description:** sets window.location.hash to the specified value(s)

**> Input Ports:**

- **Hash** (String)
- **Update** (Trigger)
- **Active** (Number: Boolean)
- **Silent** (Number: Boolean)
- **Allow Empty** (Number: Boolean)

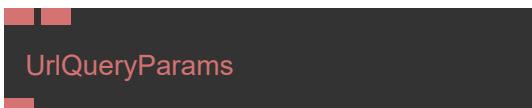
**< Output Ports:**

- Visit *Ops.Website.SetLocationHash documentation* for output port details

**Example Patch:** Open in Editor

**Docs:** <https://cables.gl/op/Ops.Website.SetLocationHash>

## 103.1.10 UrlqueryParams\_v2



**Full Name:** Ops.Website.UrlqueryParams\_v2

**Description:** Returns a URL query parameter

**> Input Ports:**

- **Parameter** (String)
- **Default** (String)

**< Output Ports:**