



Reference
Tutorials
Examples
Contribute
Community
About

</> Start Coding

♥ Donate

Jump to



Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

Search

🌐 English



♿ Accessibility



Reference

Find easy explanations for every piece of p5.js code.

Filter by keyword

Shape

2D Primitives

`arc()`

Draws an arc.

`ellipse()`

Draws an ellipse (oval).

`circle()`

Draws a circle.

`line()`

Draws a straight line between two points.

`</>` Start Coding

 Donate

`point()`
Draws a single point in space.

`rect()`
Draws a rectangle.

`triangle()`
Draws a triangle.

3D Models

`loadModel()`
Loads a 3D model to create a p5.Geometry object.

3D Primitives

`beginGeometry()`
Begins adding shapes to a new p5.Geometry object.

`buildGeometry()`
Creates a custom p5.Geometry object from simpler
3D shapes.

`quad()`
Draws a quadrilateral (four-sided shape).

`square()`
Draws a square.

`model()`
Draws a p5.Geometry object to the canvas.

`box()`
Draws a box (rectangular prism).

`cone()`
Draws a cone.

</> Start Coding

♥ Donate

`cylinder()`

Draws a cylinder.

`endGeometry()`

Stops adding shapes to a new p5.Geometry object and returns the object.

`plane()`

Draws a plane.

`torus()`

Draws a torus.

`ellipsoid()`

Draws an ellipsoid.

`freeGeometry()`

Clears a p5.Geometry object from the graphics processing unit (GPU) memory.

`sphere()`

Draws a sphere.

Attributes

`ellipseMode()`

Changes where ellipses, circles, and arcs are drawn.

`rectMode()`

Changes where rectangles and squares are drawn.

`strokeCap()`

Sets the style for rendering the ends of lines.

`noSmooth()`

Draws certain features with jagged (aliased) edges.

`smooth()`

Draws certain features with smooth (antialiased) edges.

`strokeJoin()`

Sets the style of the joints that connect line segments.

strokeWeight()
Sets the width of the stroke used for points, lines, and the outlines of shapes.

Curves

bezier()

Draws a Bézier curve.

bezierPoint()

Calculates coordinates along a Bézier curve using interpolation.

curve()

Draws a curve using a Catmull-Rom spline.

curvePoint()

Calculates coordinates along a spline curve using interpolation.

curveTightness()

Adjusts the way `curve()` and `curveVertex()` draw.

bezierDetail()

Sets the number of segments used to draw Bézier curves in WebGL mode.

bezierTangent()

Calculates coordinates along a line that's tangent to a Bézier curve.

curveDetail()

Sets the number of segments used to draw spline curves in WebGL mode.

curveTangent()

Calculates coordinates along a line that's tangent to a spline curve.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

</> Start Coding

Heart Donate

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

Vertex

`beginContour()`

Begins creating a hole within a flat shape.

`bezierVertex()`

Adds a Bézier curve segment to a custom shape.

`endContour()`

Stops creating a hole within a flat shape.

`normal()`

Sets the normal vector for vertices in a custom 3D shape.

`vertex()`

Adds a vertex to a custom shape.

p5.Geometry

`averageNormals()`

Averages the vertex normals.

`beginShape()`

Begins adding vertices to a custom shape.

`curveVertex()`

Adds a spline curve segment to a custom shape.

`endShape()`

Begins adding vertices to a custom shape.

`quadraticVertex()`

Adds a quadratic Bézier curve segment to a custom shape.

`averagePoleNormals()`

Averages pole normals.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

</> Start Coding

Heart Donate

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

calculateBoundingBox()
Calculates the position and size of the smallest box that contains the geometry.

computeFaces()
Computes the geometry's faces using its vertices.

faces
An array that lists which of the geometry's vertices form each of its faces.

flipV()
Flips the geometry's texture v-coordinates.

saveObj()
The saveObj() function exports p5.Geometry objects as 3D models in the Wavefront .obj file format.

uvS
An array that lists the texture coordinates for each of the geometry's vertices.

vertices
An array with the geometry's vertices.

clearColors()
Removes the geometry's internal colors.

computeNormals()
Calculates the normal vector for each vertex on the geometry.

flipU()
Flips the geometry's texture u-coordinates.

normalize()
Transforms the geometry's vertices to fit snugly within a 100×100×100 box centered at the origin.

saveStl()
The saveStl() function exports p5.Geometry objects as 3D models in the STL stereolithography file format.

vertexNormals
An array with the vectors that are normal to the geometry's vertices.

[Start Coding](#)

[Donate](#)

Color

Creating & Reading

`alpha()`

Gets the alpha (transparency) value of a color.

`blue()`

Gets the blue value of a color.

`brightness()`

Gets the brightness value of a color.

`color()`

Creates a p5.Color object.

`green()`

Gets the green value of a color.

`hue()`

Gets the hue value of a color.

`lerpColor()`

Blends two colors to find a third color between them.

`lightness()`

Gets the lightness value of a color.

`red()`

Gets the red value of a color.

`saturation()`

Gets the saturation value of a color.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

</> Start Coding

Heart Donate

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

Setting

`background()`

Sets the color used for the background of the canvas.

`clear()`

Clears the pixels on the canvas.

`colorMode()`

Changes the way color values are interpreted.

`erase()`

Starts using shapes to erase parts of the canvas.

`noErase()`

Ends erasing that was started with `erase()`.

`noStroke()`

Disables drawing points, lines, and the outlines of shapes.

`beginClip()`

Starts defining a shape that will mask any shapes drawn afterward.

`clip()`

Defines a shape that will mask any shapes drawn afterward.

`endClip()`

Ends defining a mask that was started with `beginClip()`.

`fill()`

Sets the color used to fill shapes.

`noFill()`

Disables setting the fill color for shapes.

`stroke()`

Sets the color used to draw points, lines, and the outlines of shapes.

p5.js

Reference

Tutorials

Examples

Contribute

Community

About

</> Start Coding

♥ Donate

Shape

Color

Typography

Image

Transform

Environment

3D

Rendering

Math

p5.Color

`setAlpha()`

Sets the alpha (transparency) value of a color.

`setGreen()`

Sets the green component of a color.

`toString()`

Returns the color formatted as a String.

`setBlue()`

Sets the blue component of a color.

`setRed()`

Sets the red component of a color.

Typography

Attributes

`textAlign()`

Sets the way text is aligned when `text()` is called.

`textAscent()`

Calculates the ascent of the current font at its current size.

[Start Coding](#)

[Donate](#)

textDescent()
Calculates the descent of the current font at its current size.

textSize()
Sets the font size when `text()` is called.

textWidth()
Calculates the maximum width of a string of text drawn when `text()` is called.

textLeading()
Sets the spacing between lines of text when `text()` is called.

textStyle()
Sets the style for system fonts when `text()` is called.

textWrap()
Sets the style for wrapping text when `text()` is called.

Loading & Displaying

loadFont()
Loads a font and creates a `p5.Font` object.

textFont()
Sets the font used by the `text()` function.

p5.Font

font
The font's underlying `opentype.js` font object.

text()
Draws text to the canvas.

textBounds()
Returns the bounding box for a string of text written using the font.

[Start Coding](#)

[Donate](#)

textToPoints()
Returns an array of points outlining a string of text written using the font.

Image

createImage()
Creates a new p5.Image object.

saveCanvas()
Saves the current canvas as an image.

saveFrames()
Captures a sequence of frames from the canvas that can be saved as images.

Loading & Displaying

image()
Draws an image to the canvas.

imageMode()
Changes the location from which images are drawn when image() is called.

loadImage()
Loads an image to create a p5.Image object.

noTint()
Removes the current tint set by tint().

[Start Coding](#)

[Donate](#)

`saveGif()`

Generates a gif from a sketch and saves it to a file.

`tint()`

Tints images using a color.

Pixels

`blend()`

Copies a region of pixels from one image to another.

`copy()`

Copies pixels from a source image to a region of the canvas.

Looking for the old p5.js site? Find it [here!](#)



`loadPixels()`

Loads the current value of each pixel on the canvas into the pixels array.

`pixels`

An array containing the color of each pixel on the canvas.

`set()`

Sets the color of a pixel or draws an image to the canvas.

`updatePixels()`

Updates the canvas with the RGBA values in the pixels array.

p5.Image

`blend()`

Copies a region of pixels from another image into this one.

`copy()`

Copies pixels from a source image to this image.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

[Start Coding](#)

[Donate](#)

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

<code>delay()</code>	Changes the delay between frames in an animated GIF.	<code>filter()</code>	Applies an image filter to the image.
<code>get()</code>	Gets a pixel or a region of pixels from the image.	<code>getCurrentFrame()</code>	Gets the index of the current frame in an animated GIF.
<code>height</code>	The image's height in pixels.	<code>loadPixels()</code>	Loads the current value of each pixel in the image into the <code>img.pixels</code> array.
<code>mask()</code>	Masks part of the image with another.	<code>numFrames()</code>	Returns the number of frames in an animated GIF.
<code>pause()</code>	Pauses an animated GIF.	<code>pixelDensity()</code>	Gets or sets the pixel density for high pixel density displays.
<code>pixels</code>	An array containing the color of each pixel in the image.	<code>play()</code>	Plays an animated GIF that was paused with <code>img.pause()</code> .
<code>reset()</code>	Restarts an animated GIF at its first frame.	<code>resize()</code>	Resizes the image to a given width and height.

[Start Coding](#)

[Donate](#)

save()

Saves the image to a file.

setFrame()

Sets the current frame in an animated GIF.

width

The image's width in pixels.

set()

Sets the color of one or more pixels within an image.

updatePixels()

Updates the canvas with the RGBA values in the img.pixels array.

Transform

applyMatrix()

Applies a transformation matrix to the coordinate system.

resetMatrix()

Clears all transformations applied to the coordinate system.

rotate()

Rotates the coordinate system.

rotateX()

Rotates the coordinate system about the x-axis in WebGL mode.

`rotateY()`

Rotates the coordinate system about the y-axis in WebGL mode.

`scale()`

Scales the coordinate system.

`shearY()`

Shears the y-axis so that shapes appear skewed.

`rotateZ()`

Rotates the coordinate system about the z-axis in WebGL mode.

`shearX()`

Shears the x-axis so that shapes appear skewed.

`translate()`

Translates the coordinate system.

Environment

`cursor()`

Changes the cursor's appearance.

`describe()`

Creates a screen reader-accessible description of the canvas.

`displayDensity()`

Returns the display's current pixel density.

`deltaTime`

A Number variable that tracks the number of milliseconds it took to draw the last frame.

`describeElement()`

Creates a screen reader-accessible description of elements in the canvas.

`displayHeight`

A Number variable that stores the height of the screen display.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

`</>` Start Coding

 Donate

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

displayWidth
A Number variable that stores the width of the screen display.

frameCount
A Number variable that tracks the number of frames drawn since the sketch started.

fullscreen()
Toggles full-screen mode or returns the current mode.

getURL()
Returns the sketch's current URL as a String.

getURLPath()
Returns the current URL path as an Array of Strings.

height
A Number variable that stores the height of the canvas in pixels.

pixelDensity()
Sets the pixel density or returns the current density.

focused
A Boolean variable that's true if the browser is focused and false if not.

frameRate()
Sets the number of frames to draw per second.

getTargetFrameRate()
Returns the target frame rate.

getURLParams()
Returns the current URL parameters in an Object.

gridOutput()
Creates a screen reader-accessible description of shapes on the canvas.

noCursor()
Hides the cursor from view.

print()
Displays text in the web browser's console.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

</> Start Coding

♥ Donate

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

`textOutput()`
Creates a screen reader-accessible description of shapes on the canvas.

`width`
A Number variable that stores the width of the canvas in pixels.

`windowResized()`
A function that's called when the browser window is resized.

`webglVersion`
A String variable with the WebGL version in use.

`windowHeight`
A Number variable that stores the height of the browser's viewport.

`windowWidth`
A Number variable that stores the width of the browser's viewport.

3D

Camera

`camera()`
Sets the position and orientation of the current camera in a 3D sketch.

`createCamera()`
Creates a new p5.Camera object and sets it as the current (active) camera.

 Start Coding

 Donate

<code>frustum()</code>	Sets the frustum of the current camera in a 3D sketch.	<code>linePerspective()</code>	Enables or disables perspective for lines in 3D sketches.
<code>ortho()</code>	Sets an orthographic projection for the current camera in a 3D sketch.	<code>perspective()</code>	Sets a perspective projection for the current camera in a 3D sketch.
<code>setCamera()</code>	Sets the current (active) camera of a 3D sketch.		

Interaction

<code>debugMode()</code>	Adds a grid and an axes icon to clarify orientation in 3D sketches.	<code>noDebugMode()</code>	Turns off <code>debugMode()</code> in a 3D sketch.
<code>orbitControl()</code>	Allows the user to orbit around a 3D sketch using a mouse, trackpad, or touchscreen.		

Lights

<code>ambientLight()</code>	Creates a light that shines from all directions.	<code>directionalLight()</code>	Creates a light that shines in one direction.
-----------------------------	--	---------------------------------	---

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

</> Start Coding

♥ Donate

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

`imageLight()`
Creates an ambient light from an image.

`lights()`
Places an ambient and directional light in the scene.

`panorama()`
Creates an immersive 3D background.

`specularColor()`
Sets the specular color for lights.

`lightFalloff()`
Sets the falloff rate for `pointLight()` and `spotLight()`.

`noLights()`
Removes all lights from the sketch.

`pointLight()`
Creates a light that shines from a point in all directions.

`spotLight()`
Creates a light that shines from a point in one direction.

Material

`ambientMaterial()`
Sets the ambient color of shapes' surface material.

`createShader()`
Creates a new p5.Shader object.

`createFilterShader()`
Creates a p5.Shader object to be used with the `filter()` function.

`emissiveMaterial()`
Sets the emissive color of shapes' surface material.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

 Start Coding

 Donate

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

loadShader()
Loads vertex and fragment shaders to create a p5.Shader object.

normalMaterial()
Sets the current material as a normal material.

shader()
Sets the p5.Shader object to apply while drawing.

specularMaterial()
Sets the specular color of shapes' surface material.

textureMode()
Changes the coordinate system used for textures when they're applied to custom shapes.

p5.Camera

camera()
Sets the position and orientation of the camera.

metalness()
Sets the amount of "metalness" of a specularMaterial().

resetShader()
Restores the default shaders.

shininess()
Sets the amount of gloss ("shininess") of a specularMaterial().

texture()
Sets the texture that will be used on shapes.

textureWrap()
Changes the way textures behave when a shape's uv coordinates go beyond the texture.

centerX
The x-coordinate of the place where the camera looks.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

`</> Start Coding`

 `Donate`

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

centerY

The y-coordinate of the place where the camera looks.

eyeX

The camera's y-coordinate.

eyeZ

The camera's z-coordinate.

lookAt()

Points the camera at a location.

ortho()

Sets an orthographic projection for the camera.

perspective()

Sets a perspective projection for the camera.

set()

Sets the camera's position, orientation, and projection by copying another camera.

centerZ

The y-coordinate of the place where the camera looks.

eyeY

The camera's y-coordinate.

frustum()

Sets the camera's frustum.

move()

Moves the camera along its "local" axes without changing its orientation.

pan()

Rotates the camera left and right.

roll()

Rotates the camera in a clockwise/counter-clockwise direction.

setPosition()

Sets the camera's position in "world" space without changing its orientation.



Reference
Tutorials
Examples
Contribute
Community
About

[Start Coding](#)

[Donate](#)

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

`slerp()`

Sets the camera's position and orientation to values that are in-between those of two other cameras.

`upX`

The x-component of the camera's "up" vector.

`upZ`

The z-component of the camera's "up" vector.

p5.Shader

`copyToContext()`

Copies the shader from one drawing context to another.

`tilt()`

Rotates the camera up and down.

`upY`

The y-component of the camera's "up" vector.

`setUniform()`

Sets the shader's uniform (global) variables.

Rendering

`blendMode()`

Sets the way colors blend when added to the canvas.

`clearDepth()`

Clears the depth buffer in WebGL mode.

</> Start Coding

♥ Donate

createCanvas ()

Creates a canvas element on the web page.

createGraphics ()

Creates a p5.Graphics object.

noCanvas ()

Removes the default canvas.

setAttributes ()

Set attributes for the WebGL Drawing context.

createFramebuffer ()

Creates and a new p5.Framebuffer object.

drawingContext

A system variable that provides direct access to the sketch's element.

resizeCanvas ()

Resizes the canvas to a given width and height.

p5.Framebuffer

autoSized ()

Toggles the framebuffer's autosizing mode or returns the current mode.

color

An object that stores the framebuffer's color data.

begin ()

Begins drawing shapes to the framebuffer.

createCamera ()

Creates a new p5.Camera object to use with the framebuffer.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

`</>` Start Coding

 Donate

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

`depth`

An object that stores the framebuffer's depth data.

`end()`

Stops drawing shapes to the framebuffer.

`loadPixels()`

Loads the current value of each pixel in the framebuffer into its pixels array.

`pixels`

An array containing the color of each pixel in the framebuffer.

`resize()`

Resizes the framebuffer to a given width and height.

p5.Graphics

`createFramebuffer()`

Creates a new p5.Framebuffer object with the same WebGL context as the graphics buffer.

`draw()`

Draws to the framebuffer by calling a function that contains drawing instructions.

`get()`

Gets a pixel or a region of pixels from the framebuffer.

`pixelDensity()`

Sets the framebuffer's pixel density or returns its current density.

`remove()`

Deletes the framebuffer from GPU memory.

`updatePixels()`

Updates the framebuffer with the RGBA values in the pixels array.

`remove()`

Removes the graphics buffer from the web page.

</> Start Coding

♥ Donate

`reset()`
Resets the graphics buffer's transformations and lighting.

p5.Renderer

Math

Calculation

`abs()`

Calculates the absolute value of a number.

`ceil()`

Calculates the closest integer value that is greater than or equal to a number.

`constrain()`

Constrains a number between a minimum and maximum value.

`dist()`

Calculates the distance between two points.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

 Start Coding

 Donate

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

exp ()	Calculates the value of Euler's number e (2.71828...) raised to the power of a number.	floor ()	Calculates the closest integer value that is less than or equal to the value of a number.
fract ()	Calculates the fractional part of a number.	lerp ()	Calculates a number between two numbers at a specific increment.
log ()	Calculates the natural logarithm (the base-e logarithm) of a number.	mag ()	Calculates the magnitude, or length, of a vector.
map ()	Re-maps a number from one range to another.	max ()	Returns the largest value in a sequence of numbers.
min ()	Returns the smallest value in a sequence of numbers.	norm ()	Maps a number from one range to a value between 0 and 1.
pow ()	Calculates exponential expressions such as 2 ³ .	round ()	Calculates the integer closest to a number.
sq ()	Calculates the square of a number.	sqrt ()	Calculates the square root of a number.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

[Start Coding](#)

[Donate](#)

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

Noise

`noise()`

Returns random numbers that can be tuned to feel organic.

`noiseDetail()`

Adjusts the character of the noise produced by the `noise()` function.

`noiseSeed()`

Sets the seed value for the `noise()` function.

Random

`random()`

Returns a random number or a random element from an array.

`randomGaussian()`

Returns a random number fitting a Gaussian, or normal, distribution.

`randomSeed()`

Sets the seed value for the `random()` and `randomGaussian()` functions.

Trigonometry

`acos()`

Calculates the arc cosine of a number.

`angleMode()`

Changes the unit system used to measure angles.

 Start Coding

 Donate

`asin()`
Calculates the arc sine of a number.

`atan2()`
Calculates the angle formed by a point, the origin, and the positive x-axis.

`degrees()`
Converts an angle measured in radians to its value in degrees.

`sin()`
Calculates the sine of an angle.

`atan()`
Calculates the arc tangent of a number.

`cos()`
Calculates the cosine of an angle.

`radians()`
Converts an angle measured in degrees to its value in radians.

`tan()`
Calculates the tangent of an angle.

Vector

`createVector()`
Creates a new p5.Vector object.

p5.Vector

`add()`
Adds to a vector's x, y, and z components.

`angleBetween()`
Calculates the angle between two vectors.

p5*js

Reference
Tutorials
Examples
Contribute
Community
About

`</>` Start Coding

 Donate

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

<code>array()</code>	Returns the vector's components as an array of numbers.	<code>clampToZero()</code>	Replaces the components of a p5.Vector that are very close to zero with zero.
<code>copy()</code>	Returns a copy of the p5.Vector object.	<code>cross()</code>	Calculates the cross product of two vectors.
<code>dist()</code>	Calculates the distance between two points represented by vectors.	<code>div()</code>	Divides a vector's x, y, and z components.
<code>dot()</code>	Calculates the dot product of two vectors.	<code>equals()</code>	Checks whether all the vector's components are equal to another vector's.
<code>fromAngle()</code>	Creates a new 2D vector from an angle.	<code>fromAngles()</code>	Creates a new 3D vector from a pair of ISO spherical angles.
<code>heading()</code>	Calculates the angle a 2D vector makes with the positive x-axis.	<code>lerp()</code>	Calculates new x, y, and z components that are proportionally the same distance between two vectors.
<code>limit()</code>	Limits a vector's magnitude to a maximum value.	<code>mag()</code>	Calculates the magnitude (length) of the vector.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

[Start Coding](#)

[Donate](#)

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

<code>magSq()</code>	Calculates the magnitude (length) of the vector squared.	<code>mult()</code>	Multiples a vector's x, y, and z components.
<code>normalize()</code>	Scales the components of a p5.Vector object so that its magnitude is 1.	<code>random2D()</code>	Creates a new 2D unit vector with a random heading.
<code>random3D()</code>	Creates a new 3D unit vector with a random heading.	<code>reflect()</code>	Reflects a vector about a line in 2D or a plane in 3D.
<code>rem()</code>	Performs modulo (remainder) division with a vector's x, y, and z components.	<code>rotate()</code>	Rotates a 2D vector by an angle without changing its magnitude.
<code>set()</code>	Sets the vector's x, y, and z components.	<code>setHeading()</code>	Rotates a 2D vector to a specific angle without changing its magnitude.
<code>setMag()</code>	Sets a vector's magnitude to a given value.	<code>slerp()</code>	Calculates a new heading and magnitude that are between two vectors.
<code>sub()</code>	Subtracts from a vector's x, y, and z components.	<code>toString()</code>	Returns a string representation of a vector.

[\(</> Start Coding\)](#)

[!\[\]\(909ff976bf47376a7a813dc1902dc014_img.jpg\) Donate](#)

x
The x component of the vector

z
The z component of the vector

y
The y component of the vector

IO

Input

httpDo()
Method for executing an HTTP request.

httpPost()
Method for executing an HTTP POST request.

loadJSON()
Loads a JSON file to create an Object.

httpGet()
Method for executing an HTTP GET request.

loadBytes()
This method is suitable for fetching files up to size of 64MB.

loadStrings()
Loads a text file to create an Array.

`loadTable()`

Reads the contents of a file or URL and creates a p5.Table object with its values.

`loadXML()`

Loads an XML file to create a p5.XML object.

Output

`createWriter()`

Creates a new p5.PrintWriter object.

`p5.PrintWriter`

A class to describe a print stream.

`save()`

Saves a given element(image, text, json, csv, wav, or html) to the client's computer.

`saveJSON()`

Saves an Object or Array to a JSON file.

`saveStrings()`

Saves an Array of Strings to a file, one per line.

`saveTable()`

Writes the contents of a Table object to a file.

Table

Time & Date

`day()`

Returns the current day as a number from 1–31.

`hour()`

Returns the current hour as a number from 0–23.

[Start Coding](#)

[Donate](#)

millis()

Returns the number of milliseconds since a sketch started running.

month()

Returns the current month as a number from 1–12.

year()

Returns the current year as a number such as 1999.

p5.Table

addColumn()

Use `addColumn()` to add a new column to a Table object.

clearRows()

Removes all rows from a Table.

findRow()

Finds the first row in the Table that contains the value provided, and returns a reference to that row.

minute()

Returns the current minute as a number from 0–59.

second()

Returns the current second as a number from 0–59.

addRow()

Use `addRow()` to add a new row of data to a p5.Table object.

columns

An array containing the names of the columns in the table, if the "header" the table is loaded with the "header" parameter.

findRows()

Finds the rows in the Table that contain the value provided, and returns references to those rows.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

</> Start Coding

Heart Donate

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

get ()	Retrieves a value from the Table's specified row and column.	getArray ()	Retrieves all table data and returns it as a multidimensional array.
getColumn ()	Retrieves all values in the specified column, and returns them as an array.	getColumnCount ()	Returns the total number of columns in a Table.
getNum ()	Retrieves a Float value from the Table's specified row and column.	getObject ()	Retrieves all table data and returns as an object.
getRow ()	Returns a reference to the specified p5.TableRow.	getRowCount ()	Returns the total number of rows in a Table.
getRows ()	Gets all rows from the table.	getString ()	Retrieves a String value from the Table's specified row and column.
matchRow ()	Finds the first row in the Table that matches the regular expression provided, and returns a reference to that row.	matchRows ()	Finds the rows in the Table that match the regular expression provided, and returns references to those rows.
removeColumn ()	Use <code>removeColumn()</code> to remove an existing column from a Table object.	removeRow ()	Removes a row from the table object.

[Start Coding](#)

[Donate](#)

removeTokens()

Removes any of the specified characters (or "tokens").

set()

Stores a value in the Table's specified row and column.

setString()

Stores a String value in the Table's specified row and column.

rows

An array containing the p5.TableRow objects that make up the rows of the table.

setNum()

Stores a Float value in the Table's specified row and column.

trim()

Trims leading and trailing whitespace, such as spaces and tabs, from String table values.

p5.TableRow

get()

Retrieves a value from the TableRow's specified column.

getString()

Retrieves an String value from the TableRow's specified column.

setNum()

Stores a Float value in the TableRow's specified column.

getNum()

Retrieves a Float value from the TableRow's specified column.

set()

Stores a value in the TableRow's specified column.

setString()

Stores a String value in the TableRow's specified column.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

</> Start Coding

Heart Donate

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

p5.XML

`addChild()`

Adds a new child element and returns a reference to it.

`getChild()`

Returns the first matching child element as a new p5.XML object.

`getContent()`

Returns the element's content as a String.

`getNum()`

Return an attribute's value as a Number.

`getString()`

Return an attribute's value as a string.

`hasChildren()`

Returns true if the element has child elements and false if not.

`getAttributeCount()`

Returns the number of attributes the element has.

`getChildren()`

Returns an array with the element's child elements as new p5.XML objects.

`getName()`

Returns the element's name as a String.

`getParent()`

Returns the element's parent element as a new p5.XML object.

`hasAttribute()`

Returns true if the element has a given attribute and false if not.

`listAttributes()`

Returns an Array with the names of the element's attributes.

</> Start Coding

♥ Donate

`listChildren()`
Returns an array with the names of the element's child elements as Strings.

`serialize()`
Returns the element as a String.

`setContent()`
Sets the element's content.

`removeChild()`
Removes the first matching child element.

`setAttribute()`
Sets an attribute to a given value.

`setName()`
Sets the element's tag name.

Events

Acceleration

`accelerationX`
The system variable `accelerationX` always contains the acceleration of the device along the x axis.

`accelerationY`
The system variable `accelerationY` always contains the acceleration of the device along the y axis.

[Start Coding](#)

[Donate](#)

accelerationZ

The system variable accelerationZ always contains the acceleration of the device along the z axis.

deviceOrientation

The system variable deviceOrientation always contains the orientation of the device.

deviceTurned()

The deviceTurned() function is called when the device rotates by more than 90 degrees continuously.

pAccelerationY

The system variable pAccelerationY always contains the acceleration of the device along the y axis in the frame previous to the current frame.

pRotationX

The system variable pRotationX always contains the rotation of the device along the x axis in the frame previous to the current frame.

pRotationZ

The system variable pRotationZ always contains the rotation of the device along the z axis in the frame previous to the current frame.

deviceMoved()

The deviceMoved() function is called when the device is moved by more than the threshold value along X, Y or Z axis.

deviceShaken()

The deviceShaken() function is called when the device total acceleration changes of accelerationX and accelerationY values is more than the threshold value.

pAccelerationX

The system variable pAccelerationX always contains the acceleration of the device along the x axis in the frame previous to the current frame.

pAccelerationZ

The system variable pAccelerationZ always contains the acceleration of the device along the z axis in the frame previous to the current frame.

pRotationY

The system variable pRotationY always contains the rotation of the device along the y axis in the frame previous to the current frame.

rotationX

The system variable rotationX always contains the rotation of the device along the x axis.

 Start Coding

 Donate

rotationY

The system variable rotationY always contains the rotation of the device along the y axis.

setMoveThreshold()

The setMoveThreshold() function is used to set the movement threshold for the deviceMoved() function.

turnAxis

When a device is rotated, the axis that triggers the deviceTurned() method is stored in the turnAxis variable.

Keyboard

key

A String system variable that contains the value of the last key typed.

keyIsDown()

Returns true if the key it's checking is pressed and false if not.

keyPressed()

A function that's called once when any key is pressed.

rotationZ

The system variable rotationZ always contains the rotation of the device along the z axis.

setShakeThreshold()

The setShakeThreshold() function is used to set the movement threshold for the deviceShaken() function.

keyCode

A Number system variable that contains the code of the last key typed.

keyIsPressed

A Boolean system variable that's true if any key is currently pressed and false if not.

keyReleased()

A function that's called once when any key is released.

keyTyped()
A function that's called once when keys with printable characters are pressed.

Mouse

doubleClicked()
A function that's called once when a mouse button is clicked twice quickly.

mouseButton
A String system variable that contains the value of the last mouse button pressed.

mouseDragged()
A function that's called when the mouse moves while a button is pressed.

mouseMoved()
A function that's called when the mouse moves.

mouseReleased()
A function that's called once when a mouse button is released.

exitPointerLock()
Exits a pointer lock started with `requestPointerLock`.

mouseClicked()
A function that's called once after a mouse button is pressed and released.

mouseIsPressed
A Boolean system variable that's true if the mouse is pressed and false if not.

mousePressed()
A function that's called once when a mouse button is pressed.

mouseWheel()
A function that's called once when the mouse wheel moves.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

[Start Coding](#)

[!\[\]\(b24c916560b9f61306da39645d78552c_img.jpg\) Donate](#)

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

`mouseX`

A Number system variable that tracks the mouse's horizontal position.

`movedX`

A Number system variable that tracks the mouse's horizontal movement.

`pmouseX`

A Number system variable that tracks the mouse's previous horizontal position.

`pwinMouseX`

A Number variable that tracks the mouse's previous horizontal position within the browser.

`requestPointerLock()`

Locks the mouse pointer to its current position and makes it invisible.

`winMouseY`

A Number variable that tracks the mouse's vertical position within the browser.

`mouseY`

A Number system variable that tracks the mouse's vertical position.

`movedY`

A Number system variable that tracks the mouse's vertical movement.

`pmouseY`

A Number system variable that tracks the mouse's previous vertical position.

`pwinMouseY`

A Number variable that tracks the mouse's previous vertical position within the browser.

`winMouseX`

A Number variable that tracks the mouse's horizontal position within the browser.

[\(</> Start Coding\)](#)

[Donate](#)

Touch

`touchEnded()`

A function that's called once each time a screen touch ends.

`touchStarted()`

A function that's called once each time the user touches the screen.

`touchMoved()`

A function that's called when the user touches the screen and moves.

`touches`

An Array of all the current touch points on a touchscreen device.

DOM

`changed()`

Calls a function when the element changes.

`createAudio()`

Creates a hidden element for simple audio playback.

`createCapture()`

Creates a element that "captures" the audio/video stream from the webcam and microphone.

`createA()`

Creates an element that links to another web page.

`createButton()`

Creates a element.

`createCheckbox()`

Creates a checkbox element.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

</> Start Coding

♥ Donate

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

`createColorPicker()`

Creates a color picker element.

`createElement()`

Creates a new p5.Element object.

`createImg()`

Creates an element that can appear outside of the canvas.

`createP()`

Creates a element.

`createSelect()`

Creates a dropdown menu element.

`createSpan()`

Creates a element.

`input()`

Calls a function when the element receives input.

`createDiv()`

Creates a element.

`createFileInput()`

Creates an element of type 'file'.

`createInput()`

Creates a text element.

`createRadio()`

Creates a radio button element.

`createSlider()`

Creates a slider element.

`createVideo()`

Creates a element for simple audio/video playback.

`removeElements()`

Removes all elements created by p5.js, including any event handlers.

`</>` Start Coding

 Donate

`select()`

Searches the page for the first element that matches the given CSS selector string.

`selectAll()`

Searches the page for all elements that matches the given CSS selector string.

p5.Element

`addClass()`

Adds a class to the element.

`attribute()`

Adds an attribute to the element.

`center()`

Centers the element either vertically, horizontally, or both.

`child()`

Attaches the element as a child of another element.

`class()`

Adds a class attribute to the element using a given string.

`doubleClicked()`

Calls a function when the mouse is pressed twice over the element.

`dragLeave()`

Calls a function when a file is dragged off the element.

`dragOver()`

Calls a function when a file is dragged over the element.

`draggable()`

Makes the element draggable.

`drop()`

Calls a function when the user drops a file on the element.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

 Start Coding

 Donate

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

<code>elt</code>	The element's underlying <code>HTMLElement</code> object.	<code>hasClass()</code>	Checks if a class is already applied to element.
<code>height</code>	A <code>Number</code> property that stores the element's height.	<code>hide()</code>	Hides the current element.
<code>html()</code>	Sets the inner HTML of the element, replacing any existing HTML.	<code>id()</code>	Sets the element's ID using a given string.
<code>mouseClicked()</code>	Calls a function when the mouse is pressed and released over the element.	<code>mouseMoved()</code>	Calls a function when the mouse moves over the element.
<code>mouseOut()</code>	Calls a function when the mouse moves off the element.	<code>mouseOver()</code>	Calls a function when the mouse moves onto the element.
<code>mousePressed()</code>	Calls a function when the mouse is pressed over the element.	<code>mouseReleased()</code>	Calls a function when the mouse is released over the element.
<code>mouseWheel()</code>	Calls a function when the mouse wheel scrolls over the element.	<code>parent()</code>	Attaches the element to a parent element.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

</> Start Coding

Heart Donate

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

position()

Sets the element's position.

removeAttribute()

Removes an attribute from the element.

show()

Shows the current element.

style()

Applies a style to the element by adding a CSS declaration.

touchEnded()

Calls a function when the user stops touching the element.

touchStarted()

Calls a function when the element is touched.

width

A Number property that stores the element's width.

remove()

Removes the element, stops all audio/video streams, and removes all callback functions.

removeClass()

Removes a class from the element.

size()

Sets the element's width and height.

toggleClass()

Toggles whether a class is applied to the element.

touchMoved()

Calls a function when the user touches the element and moves.

value()

Returns or sets the element's value.

p5.js

Reference

Tutorials

Examples

Contribute

Community

About

</> Start Coding

♥ Donate

Shape

Color

Typography

Image

Transform

Environment

3D

Rendering

Math

p5.File

`data`

A string containing the file's data.

`file`

Underlying File object.

`name`

The file name as a string.

`size`

The number of bytes in the file.

`subtype`

The file subtype as a string.

`type`

The file MIME type as a string.

p5.MediaElement

`addCue()`

Schedules a function to call when the audio/video reaches a specific time during its playback.

`autoplay()`

Sets the audio/video to play once it's loaded.

`clearCues()`

Removes all functions scheduled with `media.addCue()`.

`connect()`

Sends the element's audio to an output.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

</> Start Coding

Heart Donate

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

disconnect ()	duration ()
Disconnect all Web Audio routing, including to the main output.	Returns the audio/video's duration in seconds.
hideControls ()	loop ()
Hide the default HTMLMediaElement controls.	Plays the audio/video repeatedly in a loop.
noLoop ()	onended ()
Stops the audio/video from playing in a loop.	Calls a function when the audio/video reaches the end of its playback.
pause ()	play ()
Pauses a media element.	Plays audio or video from a media element.
removeCue ()	showControls ()
Removes a callback based on its ID.	Show the default HTMLMediaElement controls.
speed ()	src
Sets the audio/video playback speed.	Path to the media element's source as a string.
stop ()	time ()
Stops a media element and sets its current time to 0.	Sets the media element's playback time.
volume ()	
Sets the audio/video volume.	

[Start Coding](#)

[Donate](#)

Data

Array Functions

`append()`

Adds a value to the end of an array.

`arrayCopy()`

Copies an array (or part of an array) to another array.

`concat()`

Concatenates two arrays, maps to `Array.concat()`.

`reverse()`

Reverses the order of an array, maps to `Array.reverse()`

`shorten()`

Decreases an array by one element and returns the shortened array, maps to `Array.pop()`.

`shuffle()`

Shuffles the elements of an array.

`sort()`

Sorts an array of numbers from smallest to largest, or puts an array of words in alphabetical order.

`splice()`

Inserts a value or an array of values into an existing array.

`</>` Start Coding

 Donate

subset()

Extracts an array of elements from an existing array.

Conversion

boolean()

Converts a String or Number to a Boolean.

char()

Converts a Number or String to a single-character String.

hex()

Converts a Number to a String with its hexadecimal value.

str()

Converts a Boolean or Number to String.

unhex()

Converts a String with a hexadecimal value to a Number.

byte()

Converts a Boolean, String, or Number to its byte value.

float()

Converts a String to a floating point (decimal) Number.

int()

Converts a Boolean, String, or decimal Number to an integer.

uchar()

Converts a single-character String to a Number.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

[Start Coding](#)

[Donate](#)

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

Dictionary

`createNumberDict()`

Creates a new instance of p5.NumberDict using the key-value pair or object you provide.

`p5.StringDict`

A simple Dictionary class for Strings.

LocalStorage

`clearStorage()`

Removes all items in the web browser's local storage.

`removeItem()`

Removes an item from the web browser's local storage.

String Functions

`join()`

Combines an array of strings into one string.

`createStringDict()`

Creates a new instance of p5.StringDict using the key-value pair or the object you provide.

`getItem()`

Returns a value in the web browser's local storage.

`storeItem()`

Stores a value in the web browser's local storage.

`match()`

Applies a regular expression to a string and returns an array with the first match.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

[\(</> Start Coding\)](#)

[!\[\]\(c5fc890ebce2e61be5f14af6b38725d1_img.jpg\) Donate](#)

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

matchAll()
Applies a regular expression to a string and returns an array of matches.

nfc()
Converts a Number into a String with commas to mark units of 1,000.

nfs()
Converts a positive Number into a String with an extra space in front.

splitTokens()
Splits a String into pieces and returns an array containing the pieces.

p5.NumberDict

add()
Add the given number to the value currently stored at the given key.

maxKey()
Return the highest key currently used in the Dictionary.

nf()
Converts a Number into a String with a given number of digits.

nfp()
Converts a Number into a String with a plus or minus sign.

split()
Splits a String into pieces and returns an array containing the pieces.

trim()
Removes whitespace from the start and end of a String without changing the middle.

div()
Divide the given number with the value currently stored at the given key.

maxValue()
Return the highest number currently stored in the Dictionary.

`minKey()`

Return the lowest key currently used in the Dictionary.

`mult()`

Multiply the given number with the value currently stored at the given key.

p5.TypedDict

`clear()`

Removes all previously stored key-value pairs from the Dictionary.

`get()`

Returns the value stored at the given key.

`print()`

Logs the set of items currently stored in the Dictionary to the console.

`saveJSON()`

Converts the Dictionary into a JSON file for local download.

`minValue()`

Return the lowest number currently stored in the Dictionary.

`sub()`

Subtract the given number from the value currently stored at the given key.

`create()`

Creates a new key-value pair in the Dictionary.

`hasKey()`

Returns true if the given key exists in the Dictionary, otherwise returns false.

`remove()`

Removes the key-value pair stored at the given key from the Dictionary.

`saveTable()`

Converts the Dictionary into a CSV file for local download.

[Start Coding](#)

[Donate](#)

`set()`

Updates the value associated with the given key in case it already exists in the Dictionary.

`size()`

Returns the number of key-value pairs currently stored in the Dictionary.

Structure

`disableFriendlyErrors`

Turns off the parts of the Friendly Error System (FES) that impact performance.

`draw()`

A function that's called repeatedly while the sketch runs.

`isLooping()`

Returns true if the draw loop is running and false if not.

`loop()`

Resumes the draw loop after noLoop() has been called.

`noLoop()`

Stops the code in draw() from running repeatedly.

`pop()`

Ends a drawing group that contains its own styles and transformations.

`preload()`

A function that's called once to load assets before the sketch runs.

`push()`

Begins a drawing group that contains its own styles and transformations.

`redraw()`
Runs the code in `draw()` once.

`setup()`
A function that's called once when the sketch begins running.

Constants

`AUTO`
`AUTO` allows us to automatically set the width or height of an element (but not both), based on the current height and width of the element.

`HALF_PI`
A Number constant that's approximately 1.5708.

`P2D`
The default, two-dimensional renderer.

`QUARTER_PI`
A Number constant that's approximately 0.7854.

`remove()`
Removes the sketch from the web page.

`DEGREES`
A String constant that's used to set the `angleMode()`.

`HSB`
HSB (hue, saturation, brightness) is a type of color model.

`PI`
A Number constant that's approximately 3.1416.

`RADIANS`
A String constant that's used to set the `angleMode()`.

`</>` Start Coding

 Donate

TAU

A Number constant that's approximately 6.2382.

VERSION

Version of this p5.js.

WEBGL2

One of the two possible values of a WebGL canvas (either WEBGL or WEBGL2), which can be used to determine what capabilities the rendering environment has.

TWO_PI

A Number constant that's approximately 6.2382.

WEBGL

One of the two render modes in p5.js, used for computationally intensive tasks like 3D rendering and shaders.

Foundation

====

The strict equality operator === checks to see if two values are equal and of the same type.

class

A template for creating objects of a particular type.

Array

A list that keeps several pieces of data in order.

console

Prints a message to the web browser's console.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

`</>` Start Coding

 Donate

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

`const`
Creates and names a new constant.

`function`
A named group of statements.

`>=`
The greater than or equal to operator `=>=` evaluates to true if the left value is greater than or equal to the right value.

`if`
A way to choose whether to run a block of code.

`<`
The less than operator evaluates to true if the left value is less than the right value.

`Number`
A number that can be positive, negative, or zero.

`return`
Specifies the value to be returned by a function.

`for`
A way to repeat a block of code when the number of iterations is known.

`>`
The greater than operator `>>` evaluates to true if the left value is greater than the right value.

`if-else`
The if-else statement helps control the flow of your code.

`let`
Declares a new variable.

`<=`
The less than or equal to operator evaluates to true if the left value is less than or equal to the right value.

`Object`
A container for data that's stored as key-value pairs.

`Boolean`
A value that's either true or false.

p5.js

Reference
Tutorials
Examples
Contribute
Community
About

[\(</> Start Coding\)](#)

[!\[\]\(5a2cc4eb4ea5950e5b686a7bc2081e37_img.jpg\) Donate](#)

Shape
Color
Typography
Image
Transform
Environment
3D
Rendering
Math

Number
A number that can be positive, negative, or zero.

String
A sequence of text characters.

Object
A container for data that's stored as key-value pairs.

while
A way to repeat a block of code.

p5.js

Resources

[Reference](#)
[Tutorials](#)
[Examples](#)
[Contribute](#)
[Community](#)
[About](#)
[Start Coding](#)
[Donate](#)

Information

[Download](#)
[Contact](#)
[Copyright](#)
[Privacy Policy](#)
[Terms of Use](#)

Socials

[GitHub](#)
[Instagram](#)
[X](#)
[YouTube](#)
[Discord](#)
[Forum](#)

