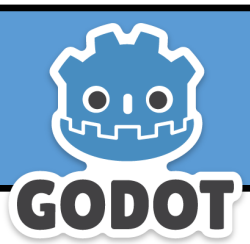
Godot Engine Reference Card v3.x



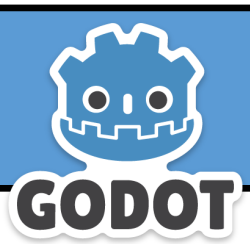
CONSTANTS

Const [var]=[value]: defines constant
Null: empty object
PI: 3.14159265358979

NODES

AnimatedSprite: sprite with animation
.animation = "[name]": changes anim
.flip_h|flip_v: flips sprite hor/ver
.frame = [number]: go to frame
.play(): plays animation
AnimationPlayer: multiple sprites, multiple animations of properties of sprites
animation_finished: when anim done
.play([name]): play animation 'name'
.playback_speed=[value]: speed
Area: a 3D area
Area2D: a 2D area with collision support
area_entered(area): another area object entered the Area2D
.is_in_group("[name]"): checks the group assignment of the node
.position=[Vec2]: sets the position
AudioStreamPlayer: plays audio streams
.play(): plays the stream
.stop(): stops the audio stream
Button: UI button
Camera: a 3D camera node
.current=[bool]: sets current camera
Camera2D: a 2D camera node
.limit_left|limit_right|limit_top|limit_bottom=[int]: limit in pixels
CanvasLayer: HUD on the screen
CircleShape2D: a circular shape object
.new(): instances new node
.radius=[value]: sets circle radius
CollisionPolygon2D: convex polygon
CollisionShape: base collision shape
CollisionShape2D: a 2D collision shape
.disabled=[bool]: disables detection
.shape=[shape]: sets shape to node
.extents.x|y: gets extents of shape
Container: arranges specialized containers in a specific way
Control: holds UI containers
DirectionalLight: rays of light (sunlight)
GridMap: 3D TileMap equivalent
HBoxContainer: row based UI container
KinematicBody2D: Collision based body
.is_on_ceiling(): true if body on ceiling
.is_on_floor(): true if body on floor
.is_on_wall(): true if body on wall
KinematicCollision2D: collision for Kinematic nodes
.collider: collision object
.name: gets object name
Label: a text label
.show(): shows the label
.text = "[text]": set label text
MarginContainer: padded UI container
MeshInstance: a 3D mesh instance
MultiMeshInstance: multiple meshes
Node|Node2D: the base node object
.add_child([inst_scene]): adds a child instance variable to the tree
.connect('[signal]',self,'[func]'): connects a signal to a function
.get_child_count: number of childs
.get_children(): used in for loop to get child nodes
.hide(): hides node visually
.instance(): instances a scene
.queuefree(): removes node from tree
.screensize: Vec2 with screensize
.show(): shows node visually
ParallaxBackground: holds different background layers with own slide speed
ParallaxLayer: a background with speed
Particles2D: particles node
.emitting=[bool]: controls emitting
Path2D: a polygon 2D path
PathFollow2D: child of a Path2D node
.loop=[bool]: loop around path
.position: gets position in real coords
.set_offset([int]): sets offset on path
.unit_offset: relative offset vs length
Position2D: a 2D position node
.position: Vec2 with position
Position3D: a 3D position node
RayCast2D: checks for nearby nodes
.castTo([x],[y]): report contact from current position to extended ray
.enabled=[bool]: enable raycast
.is_colliding(): contact found in cast
RigidBody: 3D physics simulated body
.apply_impulse([Vec3],[force]): apply impulse with power force direction Vec3
.hide(): hides the node
.rotation.x|y|z=[float]: sets the rotation in radians
.transform
.origin=[Vec3]: sets the position
RigidBody2D: simulated physics body
.angular_velocity=[float]: sets angular velocity of body
.linear_velocity=[float]: sets linear velocity of body
.mass=[value]: sets body mass
.position=[Vec2]: sets position
Shape2D: a 2D shape node
Spatial: 3D node with position, rotation
.rotate.x|y|z=[val]: rotates val radians
.transform
.origin=[Vec3]: sets the position
Sprite: a sprite with a texture
.hide(): hides the sprite
.modulate: modulates appearance
.a=[0.0...1.0]: change transparency
.scale=[Vec2]: sets sprite scale
.texture=load([file]): loads sprite
.get_size().x|y: gets pixel size
StaticBody2D: static body that is not moved by the physics engine
TextureButton: UI button with texture
pressed: when button pressed
.show(): shows the button
TextureProgress: UI progress bar
.texture_progress=[file]: texture file
.value=[value]: sets value
TextureRect: UI rectangle with texture
.visible=[bool]: sets visibility
TileMap: grid based tiles node
.cell_size: returns Vec2 with cell size
.get_cellv([cell]): returns id of cell
.get_used_cells(): returns used cells
.get_used_cells_by_id([id]): returns all cells with a specific id
.get_used_rect(): returns Vec2 extends
.map_to_world([cell]): gets cell pixel coordinates of upper-left corner
.set_cellv([cell],-1): sets empty tile
.tile_set
.find_tile_by_name('[name]'): returns the id of the tile
.tile_get_name([id]): gets cell name
Timer: a countdown timer
.start(): starts the countdown timer
.stop(): stops and resets the timer
timeout: signal when timer ends
.wait_time=[sec]: set wait time
Tween: interpolates properties node
.interpolate_property([node], '[property]', [from], [to], [duration], [function], [direction]): interpolates property of node
.start(): starts the tween

Godot Engine Reference Card v3.x



tween_completed: tween completed
VBoxContainer: column UI container
VisibilityNotifier2D: checks visibility
screen_exited: node is off screen
WorldEnvironment: ambient light 3D

STATEMENTS

$[node1]/[node11].[property]: accesses the node properties in the current scene
Clamp ([value],[min],[max]): limits the value between min and max
Emit_signal("[name]",[args…]): triggers the signal with optional arguments
Export ([type]) var [name]: make variable with type available in the Inspector
Extends [class]: extends a class / scene
File: instances file object
.new(): new file object
For [var] in range([int]): loops variable string through the range
Func [name] ([args, …]): creates a function with arguments
Get_tree(): get child nodes
.change_scene('[file]'): loads scene
.paused=[bool]: pauses the scene
.reload_current_scene(): reload scene
Get_viewport(): Current viewport
.get_visible_rect(): visible part
.size: returns Vec2 with size
Match [var]:
[valx]: code to execute when matches
Onready var [name]: node reference before node is ready (_ready() function)
Print([string]): print to debug window
Queue_free(): removes node from tree
Randomize(): Random seed setup
Return: exits the current function
Set_applied_force([Vec2]): applied force vector on RigidBody2D
Set_applied_torque([int]): applied torque on RigidBody2D
Set_physics_process([bool]): starts/stops processing _physics_process func
Set_process([bool]): starts or stops processing _process(delta) function
Signal [name]: makes a message to connect to other scripts / scenes
Var [name] = [value]: variable declaration
setget [func]: calls function if changed
Yield([node], "[message]"): waits for signal of node

TYPES

{'[entry]':[type], …}: dictionary entries
'…': string
'…%s…' % string: format string
[…,…]: array
.append([obj]): appends object
Color([r],[g],[b],[a]): rgba color value
Vector2([float], [float]): 2-dimensional vector (x,y)
.bounce([normal]): reflects using normal vector
.length(): returns length of vector
.normalized(): set length to 1
.rotated([dir]): rotation in radians
.angle(): gets absolute angle
.tangent(): sets the vector at a perpendicular clockwise direction
.x : returns float x position
.y: returns float y position
Vector3([float],[float],[float]): 3-dimensional vector (x,y,z)
.rotated([Vec3],[angle]): rotates the vector around Vec3 axis, angle radians

DEFAULT FUNCTIONS

_input(event): gets input events
event is InputEventMouseMotion: if there is mouse motion
event.relative.x|y: gets motion val
event.is_action_pressed('[map]'): if keyboard mapping is pressed
_integrate_forces(state): change physics/position of RigidBody
state.linear_velocity=[Vec3]: sets vel.
.length(): returns Vec3 length
_integrate_forces(physics_state): change physics/position of RigidBody2D
Physics_state.get_transform(): get transformation matrix
.origin.x|y: sets origin coordinates
Physics_state.set_transform ([matrix]): sets transformation matrix
_physics_process(delta): movements of physics object Rigid|KinematicBody2D
get_slide_collision([int]): returns a KinematicCollision2D object
get_slide_count(): #collisions occured
Move_and_collide([velocity]): returns KinematicCollision2D object upun collision, and null if no collision is detected
Move_and_slide([velocity],[normal]): returns velocity Vec2 after sliding
_process(delta): function called on every frame change with time elapse delta
_ready(): triggered when all nodes and childs of a scene are created
File: a file instance
.file_exists([file_name]): does exist?
.get_as_text(): gets file string content
.open([file_name], File.READ | File.WRITE): opens a file for access
.store_string([string]): stores string
Input: detects user input (see project settings, input map)
.is_action_just_pressed("[map]"): true if mapping has been pressed once
.is_action_pressed("[map]"): returns boolean if mapping is pressed
OS: returns OS information
.get_user_data_dir(): gets user path
Randi(): generates 32-bit random integer
Rand_range([min],[max]): random number between min and max
Sign([value]): returns +1 or –1
Str([number]): converts number to string
.pad_zeros([int]): pad int zeros