Godot Cheat Sheet v4.2

Var Creation var variable_name

@onready var variable_name = variable @export var variable_name : NodeType @export var variable_name = start value var variable_name = initial_value

ar Types ex.

Array: [1,5.2], ['string'], [[],1,true]
Dictionary: {}, {'key', 'value'}, {1, 2}
Vector2: Vector2(1,2), Vector2('', true) bool: false(0, null, false), Float: 1.23400, 0.0, 0.09123041 String: "im a StRing, y010", "12", 'lo' int: 123, 1, 0 true(anything that's not false)

[f Creation

if variable == 12: #if statement

else: #else statement elif variable > 12: #else if statement print('nothing above was true') print('it aint = but its greater') print('varible is cooler than 12')

unc Creation

func function_name(): #no parameter

func function_with_param(param_name); return param_name #optional param_name += 1

#using the function

function_with_param(variable) function_name()

Example

elif variable > 12: if le_number == 12: function_name() le_number = 12 le_number += 1

print(variable) #should only print 13 le_number = 13

func function_name(): le_number += 1

For Loop Creation Vector2D

for i in for_loop_variable: print(i)

#the i, is a "new" variable created for every item in for_loop_variable

limit_length(length)

normalized()

distance_to(Vector2)

Funcs

Shortcuts

ctrl+S #SAVE GAME!!! UNIVERSAL

#Arrays are commonly used with for loops

Operators

a ** b #power a % b #modulus a / b #division a * b #multiplication a - b #subtraction a + b #addition

Assignment #var = new_var

% % **= #var = var ** new_var += #var = var + new_var /= #var = var / new_var #var = var % new_var #var = var * new_var #var = var - new_var

Comparison

a >= b #greater than or equal a > b #greater than a <= b #less than or equal a < b #less than a != b #not equal

max(var1, var2, var3, etc...) floor(variable) #rounds down randf() sign(variable) min(var1, var2, var3, etc...) atan2(cord_y, cord_x) #very useful asin(cord), acos(cord), atan(cord) sin(angle),cos(angle),tan(angle) #radian move_toward(from, to, amount) lerp(from, to, weight) abs(variable) clamp(variable, max, min) randf_range(from, to) round(variable)

Node2D Funcs

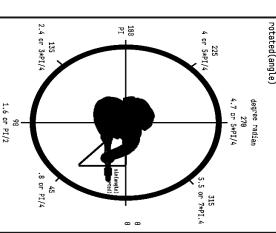
move_local_x(variable

refer to graphic #similar to position += transform.x * variable W #move mode move_local_y(variable) look_at(Vector2) ctrl+L #lock node



Rotation

angle() angle_to(Vector2 to) #same as Vector2(cos(angle), sin(angle)) from_angle(angle) deg_to_rad(degree angle) rad_to_deg(radian angle) rotate_toward(from, to, delta) angle_difference(from, to, delta) angle_to_point(Vector to) lerp_angle(from, to, weight)



move_toward(Vector2, delta) ctrl+shift+z #redo ctrl+z #undo f6 #run current scene f5 #run game

SCENE EDITOR ctrl+f1 #scene editor

Q #select mode

G #pan mode, or use mm button E #rotate mote #scale mode

CODE EDITOR

ctrl+A #create new node

ctrl+k #line/selection comment

tab #indent ctrl+left/right arrow key #jump word

ctrl+f #search shift+tab #un-indent

ctrl+click #open docs

tween.set_trans(trans type)|#optional tween.set_ease(ease type) #optional var tween = get_tree().create_tween()

Common Actions node.instantiate() CREATING A NODE

get_node('NodeName') OR \$NodeName add_child(node) get_parent() ACCESS NODES

get_children() PAUSING CODE EXAMPLES #pauses code for time await get_tree().create_timer(1).timeout

await animation_player.animation_finished RANDOM TIPS #waits for an animation to finish

nput

Engine.set_time_scale(.1) #slows game

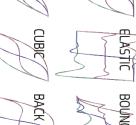
bool != bool #toggles boolean

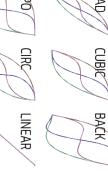
if Input.is_action_just_released('input'): if Input.is_anything_pressed(): if Input.is_action_just_pressed('input'): if Input.is_mouse_button_pressed(CONST): if Input.is_action_pressed('input'):

reation

tween.tween_property(Node, |property', wanted value, time)









1MU 1MU