CODE OS

A comprehensive cheat sheet for the Godot Engine's OS class, covering processes, memory, environment, display, input, and system interaction.

Processes & Memory

Processes & Threads

`OS.get_processor_count()`	Returns the number of CPU cores available. **Example:** `print("CPU Cores: ", OS.get_processor_count())`
`OS.get_processor_name()`	Returns the name of the CPU. **Example:** `print("CPU Name: ", OS.get_processor_name())`
`OS.is_process_running(pid)`	Checks if a process with the given `pid` is running. **Example:** `print("Process Running: ", OS.is_process_running(1234))`
`OS.get_process_id()`	Returns the ID of the current process. **Example:** `print("Process ID: ", OS.get_process_id())`
`OS.kill(pid)`	Kills the process with the specified `pid`. **Example:** `OS.kill(1234)`
`OS.get_main_thread_id()`	Returns the ID of the main thread. **Example:** `print("Main Thread ID: ", OS.get_main_thread_id())`
`OS.get_thread_caller_id()`	Returns the ID of the current thread. **Example:** `print("Current Thread ID: ", OS.get_thread_caller_id())`
`OS.can_use_threads()`	Checks if multi-threading is supported. **Example:** `print("Multi-threading: ", OS.can_use_threads())`

Memory Usage

`OS.get_dynamic_memory_usage()`	Returns the amount of dynamic memory used (for debugging). **Example:** `print("Dynamic Memory: ", OS.get_dynamic_memory_usage())`
`OS.get_static_memory_peak_usage()`	Returns the peak static memory usage (for debugging). **Example:** `print("Peak Static Memory: ", OS.get_static_memory_peak_usage())`
`OS.get_static_memory_usage()`	Returns the current static memory usage (for debugging). **Example:** `print("Static Memory: ", OS.get_static_memory_usage())`

Resource Debugging

`OS.print_all_resources(path)`	Prints all resources to the specified file path. **Example:** `OS.print_all_resources("resources.txt")`
`OS.print_all_textures_by_size()`	Prints all textures sorted by size. **Example:** `OS.print_all_textures_by_size()`
`OS.print_resources_by_type(types)`	Prints resources of the specified types. **Example:** `OS.print_resources_by_type(["Texture"])`
`OS.print_resources_in_use()`	Prints the resources that are currently in use. **Example:** `OS.print_resources_in_use()`

`OS.set_thread_name(name)`	Sets the name of the current thread. **Example:** `OS.set_thread_name("Main Thread")`

Environment & System

Directories

`OS.get_cache_dir()`	Returns the global cache directory. **Example:** `print("Cache Dir: ", OS.get_cache_dir())`
`OS.get_config_dir()`	Returns the user configuration directory. **Example:** `print("Config Dir: ", OS.get_config_dir())`
`OS.get_data_dir()`	Returns the global data directory. **Example:** `print("Data Dir: ", OS.get_data_dir())`
`OS.get_user_data_dir()`	Returns the user data directory. **Example:** `print("User Data Dir: ", OS.get_user_data_dir())`
`OS.get_system_dir(dir_type)`	Returns a system directory based on the `dir_type` (e.g., `OS.SYSTEM_DIR_DOCUMENTS`). **Example:** `print("Documents Dir: ", OS.get_system_dir(OS.SYSTEM_DIR_DOCUMENTS))`

Environment Variables

`OS.get_executable_path()`	Returns the path to the executable file. **Example:** `print("Executable Path: ", OS.get_executable_path())`
`OS.has_environment(env)`	Checks if an environment variable exists. **Example:** `print("HOME exists: ", OS.has_environment("HOME"))`
`OS.get_environment(env)`	Returns the value of an environment variable. **Example:** `print("HOME: ", OS.get_environment("HOME"))`
`OS.set_environment(env, value)`	Sets an environment variable. **Example:** `OS.set_environment("TEST", "VALUE")`

System Interaction

`OS.is_userfs_persistent()`	Checks if the `user://` path is persistent. **Example:** `print("user:// persistent: ", OS.is_userfs_persistent())`
`OS.dump_memory_to_file(path)`	Dumps memory information to a file (debugging). **Example:** `OS.dump_memory_to_file("user://memory.log")`
`OS.dump_resources_to_file(path)`	Dumps resource information to a file. **Example:** `OS.dump_resources_to_file("user://resources.log")`
`OS.move_to_trash(path)`	Moves a file to the system trash. **Example:** `OS.move_to_trash(ProjectSettings.globalize_path("user://test.txt"))`

Display & Input

Display Settings

`OS.can_draw()`	Checks if the engine can draw to the screen. **Example:** `print("Can Draw: ", OS.can_draw())`
`OS.get_screen_position()`	Returns the screen position. **Example:** `print("Screen Position: ", OS.get_screen_position())`
`OS.get_screen_size()`	Returns the screen size. **Example:** `print("Screen Size: ", OS.get_screen_size())`
`OS.get_screen_dpi()`	Returns the screen DPI. **Example:** `print("Screen DPI: ", OS.get_screen_dpi())`
`OS.get_screen_refresh_rate()`	Returns the screen refresh rate. **Example:** `print("Refresh Rate: ", OS.get_screen_refresh_rate())`
`OS.get_screen_max_scale()`	Returns the maximum screen scale. **Example:** `print("Max Scale: ", OS.get_screen_max_scale())`
`OS.get_screen_scale()`	Returns the current screen scale. **Example:** `print("Screen Scale: ", OS.get_screen_scale())`

`OS.current_screen`	Sets or gets the current screen index.
	Example:
	`OS.current_screen = 0`
	`print("Current Screen: ", OS.get_current_screen())`

`OS.get_screen_count()`	Returns the number of screens. **Example:** `print("Screen Count: ", OS.get_screen_count())`

Input Management

`OS.keyboard_get_layout_count()`	Returns the number of keyboard layouts. **Example:** `print("Layout Count: ", OS.keyboard_get_layout_count())`
`OS.keyboard_get_layout_name(index)`	Returns the name of a keyboard layout by index. **Example:** `print("Layout Name: ", OS.keyboard_get_layout_name(0))`
`OS.keyboard_get_current_layout()`	Returns the index of the current keyboard layout. **Example:** `print("Current Layout: ", OS.keyboard_get_current_layout())`
`OS.keyboard_get_layout_language(index)`	Returns the language of a keyboard layout by index. **Example:** `print("Layout Lang: ", OS.keyboard_get_layout_language(0))`
`OS.keyboard_set_current_layout(index)`	Sets the current keyboard layout. **Example:** `OS.keyboard_set_current_layout(0)`
`OS.get_scancode_string(scancode)`	Returns the string representation of a scancode. **Example:** `print("Scancode String: ", OS.get_scancode_string(KEY_ESCAPE))`

Virtual Keyboard & Touchscreen

`OS.has_touchscreen_ui_hint()`	Checks if a touchscreen is available. **Example:** `print("Has Touchscreen: ", OS.has_touchscreen_ui_hint())`
`OS.has_virtual_keyboard()`	Checks if a virtual keyboard is available. **Example:** `print("Has Virtual Keyboard: ", OS.has_virtual_keyboard())`
`OS.show_virtual_keyboard()`	Shows the virtual keyboard. **Example:** `OS.show_virtual_keyboard()`
`OS.hide_virtual_keyboard()`	Hides the virtual keyboard. **Example:** `OS.hide_virtual_keyboard()`
`OS.get_virtual_keyboard_height()`	Returns the height of the virtual keyboard. **Example:** `print("Keyboard Height: ", OS.get_virtual_keyboard_height())`

System Information

OS Information

`OS.get_name()`	Returns the name of the operating system. **Example:** `print("OS Name: ", OS.get_name())`
`OS.get_locale()`	Returns the system locale. **Example:** `print("Locale: ", OS.get_locale())`
`OS.get_locale_language()`	Returns the system language code. **Example:** `print("Language: ", OS.get_locale_language())`
`OS.get_model_name()`	Returns the device model name. **Example:** `print("Model Name: ", OS.get_model_name())`
`OS.get_unique_id()`	Returns a unique device ID. **Example:** `print("Unique ID: ", OS.get_unique_id())`

System Features & Build Info

`OS.has_feature(feature)`	Checks if the engine has a specific feature. **Example:** `print("Has Standalone: ", OS.has_feature("standalone"))`
`OS.is_stdout_verbose()`	Checks if the engine is running in verbose mode. **Example:** `print("Verbose: ", OS.is_stdout_verbose())`
`OS.is_debug_build()`	Checks if the engine is a debug build. **Example:** `print("Debug Build: ", OS.is_debug_build())`

System Interaction

`OS.request_attention()`	Requests the window's attention. **Example:** `OS.request_attention()`
`OS.crash(message)`	Crashes the engine (for testing purposes only). **Example:** `OS.crash("Simulated Crash!")`
`OS.alert(message, title)`	Displays an alert dialog. **Example:** `OS.alert("This is an alert!", "Alert")`
`OS.shell_open(uri)`	Opens a URI in the default browser. **Example:** `OS.shell_open("https://godotengine.org")`
`OS.delay_msec(msec)`	Delays execution for a specified number of milliseconds. **Example:** `OS.delay_msec(1000)`

`OS.clipboard`	Gets or sets the system clipboard content.
	Example:
	`OS.clipboard = "Copied to clipboard"`
	`print(OS.get_clipboard())`

Multimedia & Platform Features

Sound & Text-to-Speech

`OS.tts_get_voices()`	Returns a list of available TTS voices. **Example:** `print("TTS Voices: ", OS.tts_get_voices())`
`OS.tts_get_voices_for_language(lang)`	Returns TTS voices for a specific language. **Example:** `print("English Voices: ", OS.tts_get_voices_for_language("en"))`
`OS.tts_speak(text, voice)`	Speaks the given text using the specified voice. **Example:** `OS.tts_speak("Hello from TTS", "voice01")`
`OS.tts_stop()`	Stops the current TTS utterance. **Example:** `OS.tts_stop()`
`OS.tts_pause()`	Pauses the current TTS utterance. **Example:** `OS.tts_pause()`
`OS.tts_resume()`	Resumes a paused TTS utterance. **Example:** `OS.tts_resume()`

Native Video Playback

`OS.native_video_play(path, volume, audio_track, subtitle_track)`	Plays a native video file. **Example:** `OS.native_video_play("res://video.mp4", 1.0, "", "")`
`OS.native_video_stop()`	Stops the currently playing native video. **Example:** `OS.native_video_stop()`
`OS.native_video_pause()`	Pauses the currently playing native video. **Example:** `OS.native_video_pause()`
`OS.native_video_unpause()`	Unpauses the currently playing native video. **Example:** `OS.native_video_unpause()`
`OS.native_video_is_playing()`	Checks if a native video is currently playing. **Example:** `print("Video Playing: ", OS.native_video_is_playing())`

Android Permissions

`OS.request_permission(name)`	Requests a specific Android permission. **Example:** `OS.request_permission("RECORD_AUDIO")`
`OS.request_permissions()`	Requests all required Android permissions. **Example:** `OS.request_permissions()`
`OS.get_granted_permissions()`	Returns a list of granted Android permissions. **Example:** `print("Granted Permissions: ", OS.get_granted_permissions())`
`OS.keep_screen_on`	Keeps the screen on (for mobile devices). **Example:** `OS.keep_screen_on = true`