Everything is done through the "LogManager" class, use "LogManager.Instance" to get the current instance of the debug console in game, the "LogManager" class contains the following:

Methods:

    ToggleConsole() - Toggle the debug console on/off by reversing whichever one it's currently set to.
    SetConsoleActive(Bool) - Set the console on or off based off the inputted value.
    ClearConsole() - clears the console of all logs (Does not effect log files).

Events:

    OnDebugConsoleEnabled - Called when the debug console is enabled.
    OnDebugConsoleDisabled - Called when the debug console is disabled.
    OnDebugConsoleLogsCleared - Called when the consoles logs are cleared from the console.

Adjust Settings:

All settings found in the settings scriptable object can also be edited through script, to access them use "LogManager.Instance.Settings", everything inside of there can be edited, this includes the following:

    int LogCap - value between 1-1000, defaults to 250, "The max amount of logs that can be displayed at a given time, once the limit has been reached logs will begin getting replaced, the higher the number of logs the laggier the game"
    float fontSize - minimum value 1, defaults to 25, "The font size of each log in the console"
    int logFileCap - minimum value 0, defaults to 5, "The max amount of log files that can be generated, once this limit has been reached older logs will start being overwritten"
    Color backgroundColor - defaults to 0, 0, 0, 0.55, "The color of the debug consoles background"
    bool clearConsoleOnSceneChange - defaults to true, "Clears all logs when the current scene changes"
    ActiveInputSystem { None, OldInputSystem } activeInputSystem - defaults to ActiveInputSystem.OldInputSystem, "The input system to use to toggle the debug console on/off"
    KeyCode toggleConsoleKey - defaults to KeyCode.F2, "The keybind used to open/close the debug console"