Python os Module - Notes

# 🔹 Purpose

The os module provides a way to interact with the operating system — including file system operations, environment variables, and process management.

# 📁 File and Directory Operations

* os.getcwd() – Returns current working directory
* os.chdir(path) – Changes the working directory
* os.listdir(path='.') – Lists files and directories in a path
* os.mkdir(path) – Creates a directory
* os.makedirs(path) – Creates directories recursively
* os.remove(path) – Removes a file
* os.rmdir(path) – Removes a directory (must be empty)
* os.removedirs(path) – Recursively removes directories
* os.rename(src, dst) – Renames a file or directory
* os.stat(path) – Returns metadata (size, mtime, etc.)
* os.path.exists(path) – Checks if a path exists
* os.path.isfile(path) – Checks if it's a file
* os.path.isdir(path) – Checks if it's a directory
* os.path.join(a, b) – Joins two path parts safely

# 🔄 Environment Variables

* os.environ – A dictionary of environment variables
* os.getenv(key) – Gets the value of an environment variable
* os.putenv(key, value) – Sets an environment variable (less common)

# ⚙️ Process Management

* os.system(command) – Executes a system command (returns exit code)
* os.startfile(path) – Opens a file with the associated application (Windows only)
* os.getpid() – Returns current process ID
* os.getppid() – Returns parent process ID

# 🕒 Timestamps & File Info

* os.path.getsize(path) – Size of file (in bytes)
* os.path.getmtime(path) – Last modification time (timestamp)
* os.path.getctime(path) – Creation time (or metadata change time on Unix)

# 📦 Platform Info

* os.name – 'posix', 'nt', etc.
* os.uname() – Detailed OS info (Unix only)
* os.sep – Path separator ('/' or '\')
* os.linesep – Line separator ('\n', '\r\n')

# ✅ Use Cases

* Clean up old files (os.remove, os.path.getmtime)
* Traverse directories (os.walk)
* Run system commands (os.system)
* Manage configuration with environment variables (os.environ)