**System Management and File IO in Python**

**System Management and File I/O in Python**

Python provides various modules for **system management** and **file handling**. Below is an overview of commonly used modules:

**1. System Management Modules**

**1.1 os Module (Interacting with the Operating System)**

The os module allows interaction with the operating system, such as managing files, directories, and environment variables.

**Examples:**

import os

# Get current working directory

print(os.getcwd())

# List files in a directory

print(os.listdir("."))

# Create a directory

os.mkdir("new\_folder")

# Remove a file

os.remove("example.txt")

**1.2 sys Module (System-Specific Parameters and Functions)**

The sys module provides access to system-specific parameters and functions.

**Examples:**

import sys

# Get Python version

print(sys.version)

# Get command-line arguments

print(sys.argv)

# Exit the script

sys.exit("Exiting script")

**1.3 subprocess Module (Executing System Commands)**

The subprocess module is used to run system commands and interact with external processes.

**Examples:**

import subprocess

# Run a shell command

result = subprocess.run(["ls", "-l"], capture\_output=True, text=True)

print(result.stdout)

# Run a command and get output

output = subprocess.check\_output("echo Hello", shell=True)

print(output.decode())

**2. File I/O and File Management**

**2.1 shutil Module (File and Directory Operations)**

The shutil module is used for high-level file operations such as copying, moving, and deleting files.

**Examples:**

import shutil

# Copy a file

shutil.copy("source.txt", "destination.txt")

# Move a file

shutil.move("source.txt", "new\_location/source.txt")

# Delete a directory

shutil.rmtree("folder\_to\_delete")

**2.2 tempfile Module (Handling Temporary Files)**

The tempfile module allows the creation of temporary files and directories.

**Examples:**

import tempfile

# Create a temporary file

with tempfile.TemporaryFile(mode="w+") as temp:

temp.write("Hello, temporary file!")

temp.seek(0)

print(temp.read())

# Create a temporary directory

temp\_dir = tempfile.TemporaryDirectory()

print("Temporary directory created:", temp\_dir.name)

**2.3 glob Module (Pattern Matching for Files)**

The glob module is used for pattern matching in file paths.

**Examples:**

import glob

# Get all `.txt` files in the current directory

txt\_files = glob.glob("\*.txt")

print(txt\_files)

# Get all files recursively in a directory

all\_files = glob.glob("\*\*/\*.py", recursive=True)

print(all\_files)

**Summary Table**

| **Module** | **Purpose** |
| --- | --- |
| os | File system operations, environment variables |
| sys | System parameters, command-line arguments |
| subprocess | Running system commands |
| shutil | High-level file operations (copy, move, delete) |
| tempfile | Create temporary files and directories |
| glob | Pattern matching for files |