PYTHON OS MODULE: SYSTEM & FILE OPERATIONS

1. IMPORT THE MODULE

```
import os
```

Note : Always the first step. Part of Python Standard Library — no installation needed.

2. GET CURRENT WORKING DIRECTORY

```
# Get current directory path
current_dir = os.getcwd()
print("Current Directory:", current_dir)
```

3. CHANGE WORKING DIRECTORY

```
# Navigate to a new directory
os.chdir("/Users/phanirajendra/data")
print("New Directory:", os.getcwd())
```

4. LIST DIRECTORY CONTENTS

```
# List all files/folders in current directory
files = os.listdir(".")
print("Files:", files)

# List specific directory
logs = os.listdir("logs/")
print("Log Files:", logs)
```

5. CREATE & REMOVE DIRECTORIES

```
# Create a single directory
os.mkdir("output")

# Create nested directories (like `mkdir -p`)
os.makedirs("data/raw/logs", exist_ok=True)

# Remove empty directory
os.rmdir("temp")

# Remove directory tree (dangerous!)
import shutil
shutil.rmtree("old_project")          # Not in os, but often used together
```

6. FILE & PATH OPERATIONS

```
# Check if path exists
if os.path.exists("model.pkl"):
    print("Model found!")

# Check if it's a file or directory
print(os.path.isfile("data.csv"))                                    # True
print(os.path.isdir("logs/"))                                        # True

# Get file size (in bytes)
size = os.path.getsize("data.csv")
print(f"File size: {size / 1024:.2f} KB")

# Join paths (OS-agnostic)
full_path = os.path.join("data", "raw", "input.csv")
print("Full path:", full_path)
```

7. ENVIRONMENT VARIABLES (CRITICAL FOR CLOUD/AI)

```
# Get environment variable
aws_key = os.getenv("AWS_ACCESS_KEY_ID")
if aws_key:
    print("AWS Key configured")
else:
    print(" AWS Key missing!")

# Set environment variable (process-only)
os.environ["MODEL_PATH"] = "/opt/models/bert/"

# Get all environment variables
all_vars = os.environ
print("Home:", all_vars["HOME"])
```

8. FILE PERMISSIONS & METADATA

```
# Get file stats
stats = os.stat("train.py")
print("Size:", stats.st_size, "bytes")
print("Modified:", stats.st_mtime)  # Unix timestamp

# Change file permissions (Unix/Mac only)
os.chmod("script.sh", 0o755)  # rwxr-xr-x
```

9. PROCESS MANAGEMENT (ADVANCED)

```
# Get current process ID
print("PID:", os.getpid())

# Get login name
print("User:", os.getlogin())

# Execute shell command (use subprocess for better control)
os.system("ls -l")
```

10. REAL-WORLD EXAMPLE: AI PROJECT SETUP

```
import os

# Ensure project structure exists
for folder in ["data/raw", "data/processed", "models", "logs"]:
    os.makedirs(folder, exist_ok=True)

# Set environment for cloud
os.environ["TF_CPP_MIN_LOG_LEVEL"] = "2"  # Reduce TensorFlow noise

# Load config from env
data_path = os.getenv("DATA_PATH", "data/raw/")  # Default fallback

#Verify data exists
if not os.path.exists(os.path.join(data_path, "train.csv")):
    raise FileNotFoundError("Training data missing!")
```

KEY BEST PRACTICES

- Always use os.path.join() — never hardcode / or \
- Use environment variables for configuration (not hardcoded secrets)
- Check os.path.exists() before reading files
- Prefer pathlib (Python 3.4+) for complex path logic — but os is still essential
- Avoid os.system() — use subprocess for external commands