14-Day Python Scripting Plan for VLSI/EDA Automation

Day 1: Python Basics

Topic: Variables, data types, loops, functions

Resource: W3Schools Python Basics

Task: Write a script to print even numbers from 1 to 100.

Day 2: File Handling

Topic: Reading and writing .txt, .log, .csv

Resource: RealPython - File Handling

Task: Parse a sample log file and extract lines containing the word "ERROR".

Day 3: Command-line Arguments

Topic: Using argparse to build command-line scripts

Resource: Corey Schafer - argparse

Task: Write a script that takes a filename as input and counts number of lines and words.

Day 4: OS and File System

Topic: Using os, shutil to create, copy, move, rename files/folders

Resource: GeeksforGeeks - OS Module

Task: Create a script that backs up .v or .tcl files to a backup/ folder.

Day 5: Regex (Regular Expressions)

Topic: Pattern matching using re module

Resource: Automate the Boring Stuff - Regex

Task: Extract all timing violations (e.g., "VIOLATION: Setup") from a report file.

Day 6: Shell Commands via Python

Topic: Using subprocess to run Linux/EDA commands

Resource: Python Docs - subprocess

Task: Run ls or dc_shell using Python and capture the output.

Day 7: Review + Mini Project 1

Task: Combine everything so far:

- A script that accepts a log file name

- Extracts all violation lines using regex

- Saves them to a new file

- Moves the result to a folder called violations/

Day 8: Working with CSV

Topic: Reading/writing .csv with csv module

Resource: RealPython - CSV Files

Task: Create a script that converts a .log report into a structured CSV summary.

Day 9: Using JSON/YAML for Configs

Topic: Read/write settings or flow parameters

Resource: GeeksforGeeks - JSON Module

Task: Create a JSON config for tool options (e.g., synthesis flags) and load it in your script.

Day 10: Automating STA Report Extraction

Topic: Simulate extracting timing info

Task: Given a fake report, extract:

- Worst negative slack

- Path name

- Clock domain

Day 11: Creating Reusable Python Modules

Topic: Write and import your own .py modules

Resource: Python Modules

Task: Refactor an existing script into reusable functions in a separate file.

Day 12: Logging and Error Handling

Topic: Use logging, try/except

Resource: RealPython - Logging

Task: Add logs and error handling to one of your scripts.

Day 13: Mini Project 2 Report Summarizer

Task: Write a full script that:

- Parses a sample timing report

- Extracts setup/hold violations

- Writes a summary CSV

- Sends notification (print or log)

Day 14: Final Review & Extras

Revise:

- Regex

- argparse

- File handling

Explore next topics:

- Jinja2 (for templated file generation)

- Flask (for web dashboards)

- Cocotb (Python testbench framework)