Bootcamp 134 | Python

Course 17 | Linux – Part 2

Amir Hossein Chegouniyan

Head of the Technical Team at Dariche Tejarat

Lecturer of Python – Django at Maktab Sharif

Amirhossein-chegounian

Content

System and User Commands

- ping: Test network connectivity.
- man: Display manual pages for commands (how to read and navigate man pages).
- cal and jcal: Display calendar and Julian calendar.
- who and whoami: Identify logged-in users and current user.

Basic Filters

- Purpose: Process text from files or input streams.
- Key Commands:
  - grep: Search for patterns in files.
  - tee: Redirect output to multiple locations (file and console).
  - tr: Translate or delete characters.
  - uniq: Remove duplicate lines from sorted data.
  - sort: Sort file contents alphabetically or numerically.
  - wc: Count lines, words, and characters in files.

File Compression

- gzip and gunzip: Compress and decompress files.
- tar: Archive multiple files into one (tar -cvf, tar -xvf).
- zip and unzip: Create and extract .zip files.

File Security

- Concepts: Importance of file permissions and encryption.
- Key Commands:
  - chmod: Change file permissions (rwx and numeric mode).
  - chown: Change file ownership.
  - umask: Default permissions for new files.

What is Bash Scripting?

- Automating repetitive tasks by executing a sequence of commands in a script file (.sh).
- Explain the shebang line (#!/bin/bash).

Basic Syntax

- Variables: Define and use variables in a script.
- Control Structures:
  - if, elif, else: Conditional statements.
  - for, while: Loops for iteration.
- Input and Output:
  - Reading user input with read.
  - Writing output with echo.

Key Commands for Scripting

- #!/bin/bash: Specify interpreter.
- Variables: Assign and reference (VAR="value"; echo $VAR).
- if, then, else: Control flow based on conditions.
- Loops: Simple for and while examples.
- read: Get input from the user.
- case: Pattern matching for multiple conditions.

Key Commands for Scripting | Example 1

```
# Define a list and a variable
numbers = []
count = int(input("How many numbers do you want to enter? "))

# Get data from the user and store it in the list
for i in range(count):
    num = int(input(f"Enter number {i+1}: "))
    numbers.append(num)
```

Key Commands for Scripting | Example 2

```
# If/else check
if len(numbers) > 0:
    print("Your list is not empty!")
else:
    print("Your list is empty!")

# Using match case (Python 3.10+)
choice = input("Choose an operation (sum / max / min): ")
```

Key Commands for Scripting | Example 3

```
match choice:
    case "sum":
        print("Sum of numbers:", sum(numbers))
    case "max":
        print("Maximum number:", max(numbers))
    case "min":
        print("Minimum number:", min(numbers))
    case _:
        print("Invalid option!")
```

Any question?

Next course

- Introduction to Regex
- Python Regex Library (re)
- Hands-On Activity:
- Understanding Python Packages
- Working with Third-Party Libraries
- Introduction to Virtual Environments
- The Zen of Python (PEP 20)
- PEP 8 - Python Style Guide
- Advanced Function Concepts