Intermediate Python for Automation — Exercises

Easy (10 Questions)

1. Use the `os` module to print the current working directory.
2. Write Python code to list all files in the current directory.
3. Use `os.environ` to print the value of the `PATH` environment variable.
4. Create a new directory named `"automation_test"` using `os.mkdir()`.
5. Use the `sys` module to print the version of Python you are using.
6. With `sys.argv`, write a script that prints all command-line arguments passed to it.
7. Use `pathlib` to get the file name and extension from the path: `"/home/user/logs/error.log"`
8. Print today's date using `datetime.date.today()`.
9. Format the current date and time in the format: `YYYY-MM-DD HH:MM:SS`.
10. Use `re.search()` to check if the string `"Server started"` contains the word `"Server"`.

Medium (10 Questions)

11. Write Python code with `os` to rename a file `old.txt` → `new.txt`.
12. Use `os.walk()` to recursively list all files under the current directory.
13. Write a program that accepts a filename from `sys.argv` and prints the number of lines in that file.
14. Use `pathlib.Path` to:
    - Check if a file exists
    - Create a new empty file if it doesn't
15. Run the shell command `"echo Hello"` using `subprocess.run()` and capture the output.
16. Use `subprocess.run()` to check the return code of `"ls nonexistent_folder"`.
17. Write a script that prints the date 7 days from now using `datetime.timedelta`.
18. Parse the string `"2025-09-16 14:30:00"` into a datetime object and add 2 hours to it.

19. Use `re.findall()` to extract all email addresses from this string: `"Contact us at admin@example.com or support@company.org"`.
20. Use `re.sub()` to replace all digits in `"Order1234ID5678"` with `"#"`.

◈ Hard / Tricky (5 Questions)

21. Write a program using `os.environ` that sets a temporary environment variable and verifies it is available inside the Python process.
22. Use `sys.exit()` to gracefully stop a script if the user provides no command-line arguments.
23. Use `subprocess.Popen` to run `"ping -c 2 127.0.0.1"` and capture both stdout and stderr.
24. Write a regex to validate an IPv4 address (e.g., `"192.168.0.1"`).
25. Use `datetime` to calculate how many days remain until New Year's Day of the next year.

◈ Use-Case / Practical (5 Questions)

26. Write a Python script that:
    - Reads a JSON file `config.json`
    - Prints all keys and values
27. Create an INI config file named `settings.ini` with:

```
[database]
host=localhost
port=5432
```

Then write a script with `configparser` to read and print the `host` value.

28. Write a program that:
    - Reads a log file `server.log`
    - Uses regex to extract all lines containing the word `"ERROR"`
29. Write a script that runs `"df -h"` using `subprocess`, then filters and prints only the lines containing `"Filesystem"`.
30. Imagine you are parsing Apache access logs. Given this log line:

```
127.0.0.1 - - [16/Sep/2025:14:32:10 +0000] "GET /index.html HTTP/1.1" 200 2326
```

Write a regex to extract:

- The IP address
- The timestamp
- The HTTP method (`GET`)
- The requested resource (`/index.html`)
