

CỐT LÕI LẬP TRÌNH PYTHON

Nhóm 13

CÁC THÀNH VIÊN NHÓM

- Huỳnh Triệu Thanh Tâm-079206006498
- Nguyễn Mậu Thanh Toàn-062206001114
- Nguyễn Trọng Tính-052206015711
- Nguyễn Gia Quốc - 080206015120

Mục lục

I. Biến câu lệnh và biểu thức

II. Cấu trúc rẽ nhánh

III. Hàm(Functions)

IV. Cấu trúc vòng lặp

v. String, Files, List

VI. Các thư viện quan trọng

VII. Kết luận

I. Biến câu lệnh và biểu thức

Biến (Variables): Là tên gọi đại diện cho một giá trị trong bộ nhớ.

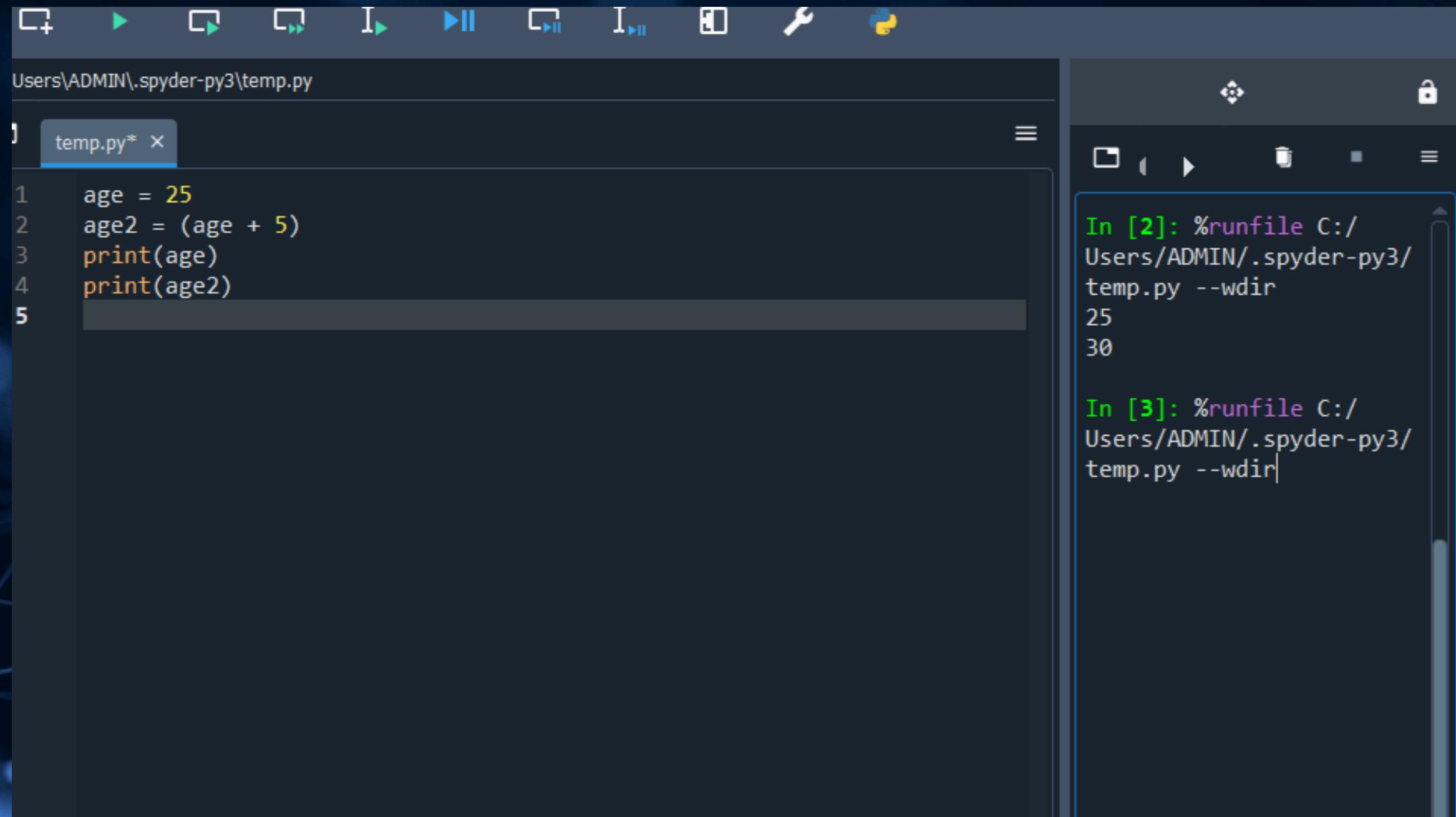
Ví dụ: age = 25

Biểu Thức (Expressions): Là sự kết hợp của biến, hằng số, toán tử, tạo ra một giá trị mới.

Ví dụ: age2 = (age + 5)

Câu Lệnh (Statements): Là một lệnh thực thi một hành động cụ thể.

Ví dụ:



The screenshot shows the Spyder Python IDE interface. On the left, the code editor displays a file named 'temp.py' with the following content:

```
1 age = 25
2 age2 = (age + 5)
3 print(age)
4 print(age2)
5
```

On the right, the IPython console shows the output of running the script:

```
In [2]: %runfile C:/Users/ADMIN/.spyder-py3/temp.py --wdir
25
30

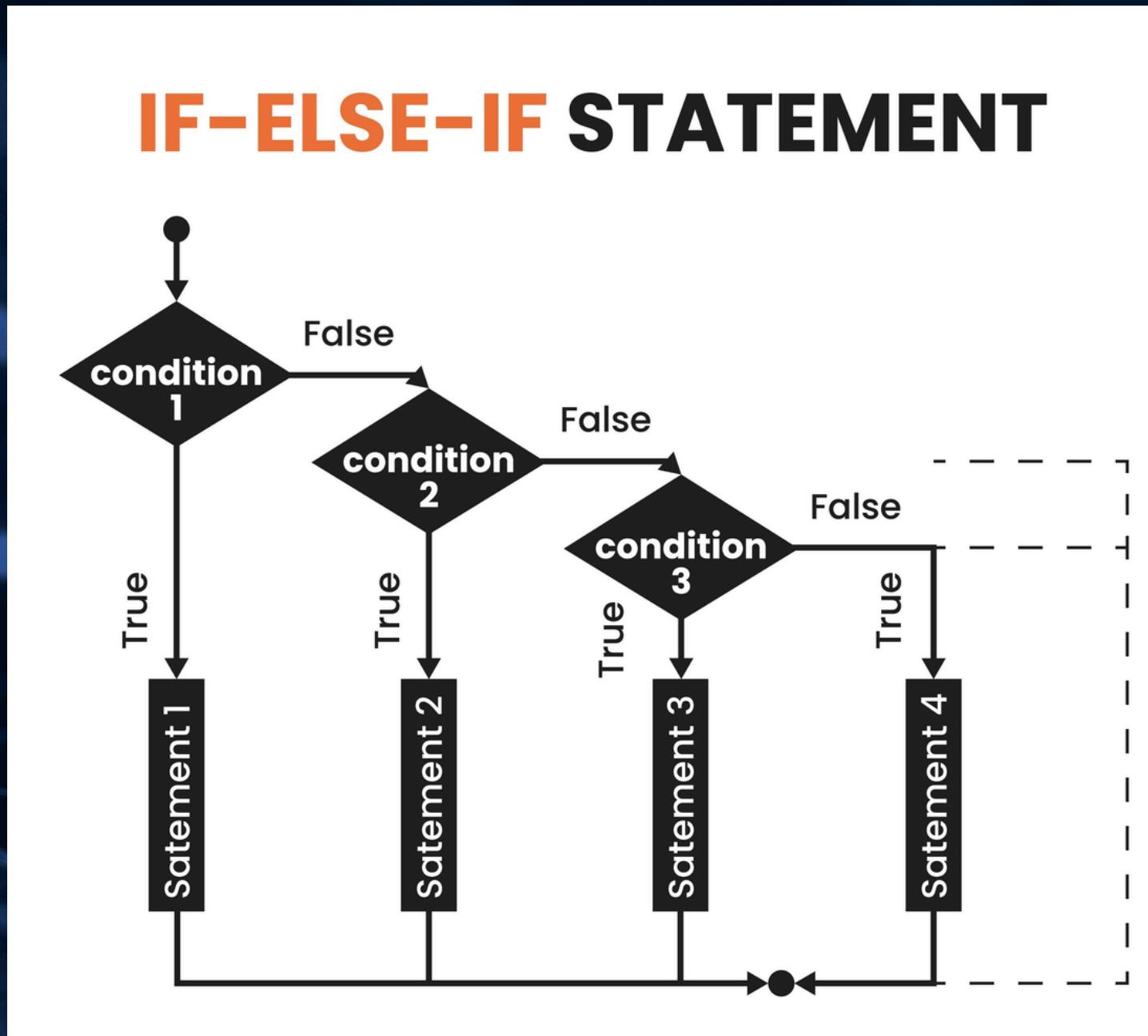
In [3]: %runfile C:/Users/ADMIN/.spyder-py3/temp.py --wdir
```

II. Cấu trúc rẽ nhánh

Mục đích: Thay đổi luồng thực thi của chương trình dựa trên kết quả kiểm tra điều kiện (True/False).

Cú pháp: Sử dụng các từ khóa if, elif (else if), và else.

Toán tử: Các toán tử so sánh (<, >, ==, !=, v.v.) và logic (and, or, not).



```
score = 99

if score >= 90:
    grade = "A"
elif score >= 80:
    grade = "B" # Khối lệnh này được thực thi
else:
    grade = "C"

print("Your grade is:", grade)
```

In [3]: %runfile C:/Users/ADMIN/.spyder-py3/temp.py --wdir
Your grade is: A

In [4]:

III. Hàm(Functions)

-Mục đích: Tái sử dụng mã, chia nhỏ chương trình.

Khái niệm: Đầu vào (Parameters/Arguments) → Thực thi Logic → Đầu ra (Return value).

-Cú pháp cơ bản: def Tên_Hàm(tham_số): ... return.

-Lợi ích: Tính mô-đun, dễ bảo trì.

-Ví dụ:

The screenshot shows the Spyder IDE interface. On the left, the code editor window displays a file named 'temp.py' with the following content:

```
1 def calculate_sum(a, b):
2     # a và b là tham số (parameters)
3     total = a + b
4     return total
5
6 # Gọi hàm và gán giá trị trả về
7 value = calculate_sum(105, 2350) # 10 và 20 là đối số (arguments)
8 print("Sum:", value)
9
```

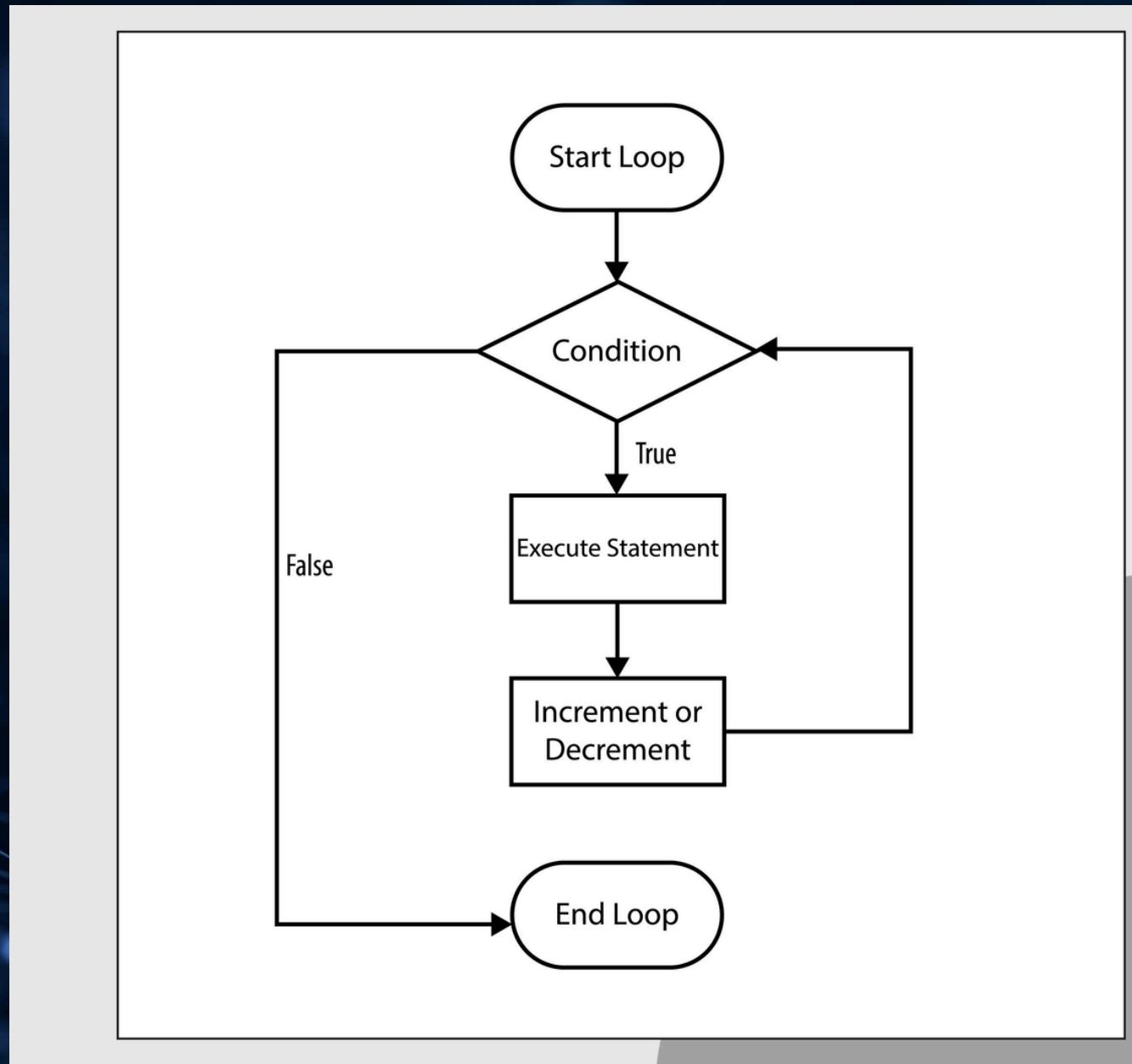
On the right, the IPython notebook interface shows the output of a run cell:

```
In [4]: %runfile C:/Users/ADMIN/.spyder-py3/temp.py --wdir
Sum: 2455

In [5]:
```

IV. Cấu trúc vòng lặp (loops)

- Mục đích: Lặp lại một khối lệnh nhiều lần.
- Vòng lặp while: Lặp khi điều kiện còn đúng (lặp vô hạn nếu không cẩn thận).
- Vòng lặp for: Lặp qua một chuỗi (ví dụ: range, list, string).
- Các lệnh bổ sung: break (thoát vòng lặp), continue (bỏ qua lần lặp hiện tại).
- Minh họa: Ví dụ đơn giản về for với range().



The screenshot shows a Python development environment with two code editors and two terminal panes. The top editor contains:

```
# Lặp qua các số từ 0 đến 2 (range(3))
for i in range(10):
    print("Loop iteration:", i)
# Output: 0, 1, 2
```

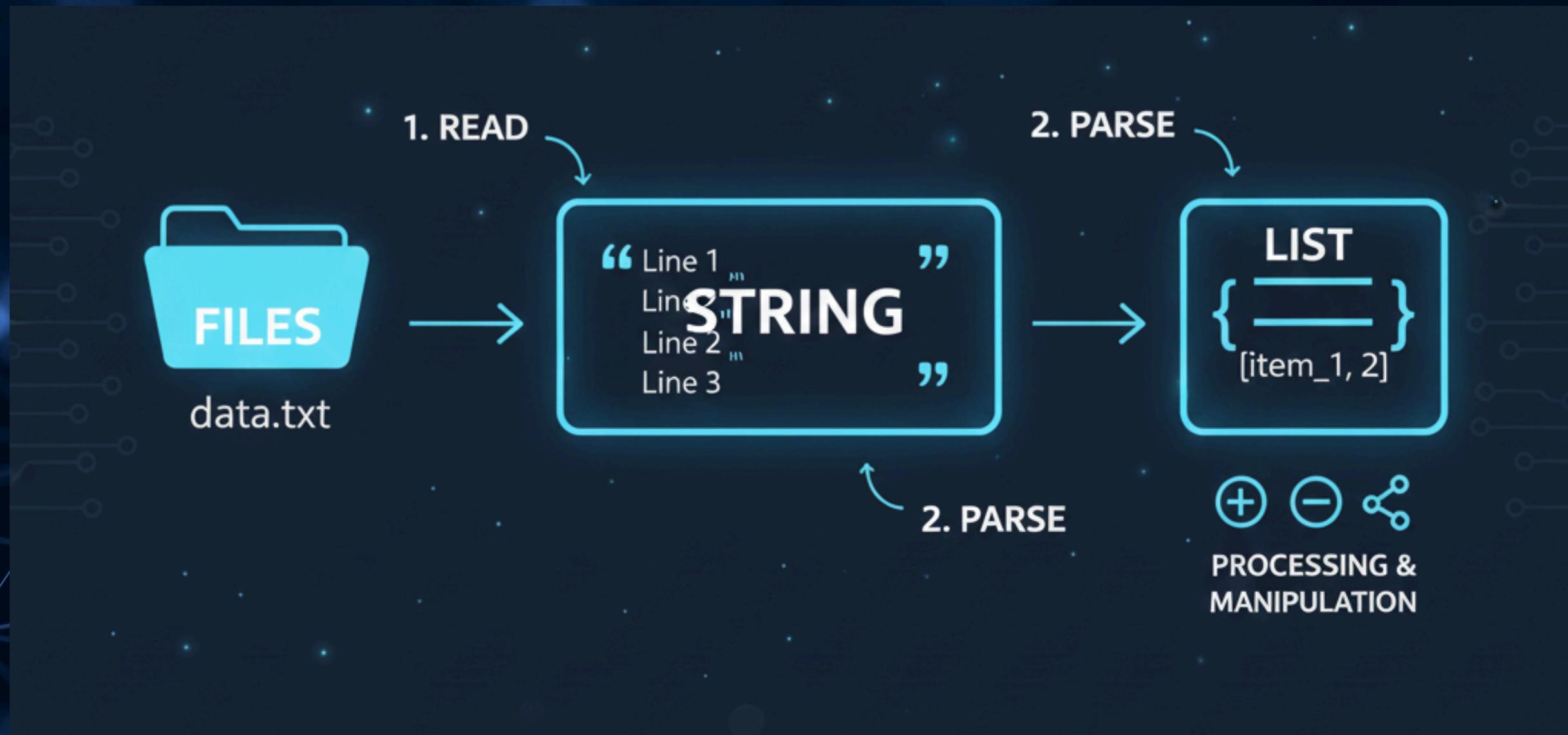
The top terminal pane (In [6]) shows the output of the first script: "Loop iteration: 0", "Loop iteration: 1", "Loop iteration: 2", ..., "Loop iteration: 9". The bottom editor contains:

```
count = 0
while count < 19:
    print("While count:", count)
    count = count + 1 # Phải có câu lệnh thay đổi điều kiện
# Output: 0, 1
```

The bottom terminal pane (In [7]) shows the output of the second script: "While count: 0", "While count: 1", "While count: 2", ..., "While count: 18".

V. String, Files, List (Dữ liệu cấu trúc)

- String: Là chuỗi bất biến (immutable) của các ký tự.
- Các thao tác cơ bản: truy cập index, cắt chuỗi (slicing).
- List: Là danh sách có thứ tự, có thể thay đổi (mutable).
- Các thao tác cơ bản: thêm (append), xóa (remove), truy cập index.
- Files (Tệp): Khái niệm cơ bản về I/O (Đọc/Ghi) để tương tác với bộ nhớ ngoài.



VI. Các thư viện quan trọng

Add a subheading

Mục đích: Mở rộng tính năng và cung cấp các hàm chuyên biệt.

Cú pháp: import Tên_Thư_Viện

Ví dụ Nổi Bật:

The diagram illustrates three examples of Python libraries:

- 1. MATH**: Shows the symbol π .
Code:

```
import math  
area = math.pi * (5**2)  
print.sarea)
```


Output: 78.539...
Icon: A circle with a dot and the text "r=5", a calculator icon.
- 2. RANDOM**: Shows two dice.
Code:

```
import random  
roll = random.randint * 6  
print.sroll
```


Output: 4
new random number each time
Icon: A hand holding a die.
- 3. OS**: Shows a folder icon with a cursor.
Code:

```
import os  
files = os.listdir("."),  
print.sroll
```


Output:
files: ["main.py,
data.txt, docs/"]
Icon: A folder icon with several files inside.

Kết Luận

Tóm lại, toàn bộ môn học Kỹ thuật Lập trình xoay quanh việc kết nối logic giữa các thành phần sau:

1. **Dữ liệu:** Bắt đầu với Biến (lưu giá trị), được tổ chức trong String và List, và được lưu trữ bởi Files.
2. **Luồng Thực thi:** Được kiểm soát bởi Rẽ nhánh (if-else) để đưa ra quyết định và Vòng lặp (for/while) để thực thi lặp lại.
3. **Tổ chức:** Hàm là công cụ cốt lõi để tái sử dụng và mô-đun hóa chương trình.
4. **Mở rộng:** Việc sử dụng các Thư viện (math, random, v.v.) giúp chúng ta mở rộng khả năng của Python một cách dễ dàng.

Cảm ơn thầy và các bạn đã lắng nghe