

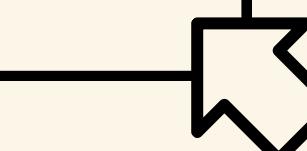
GIÁO TRÌNH LẬP TRÌNH PYTHON

Phần Cơ Bản

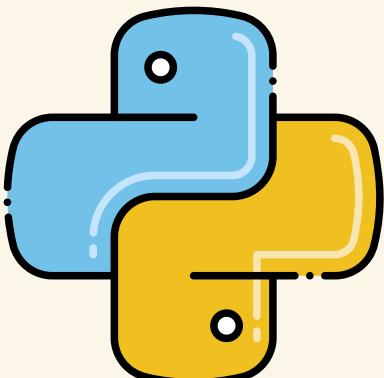
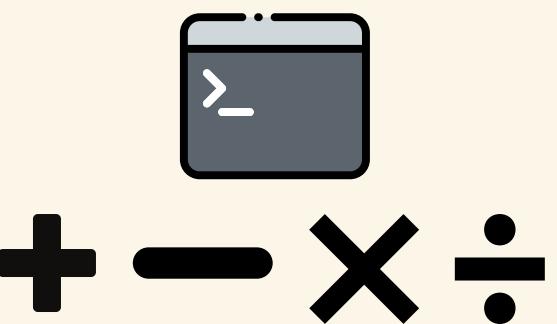
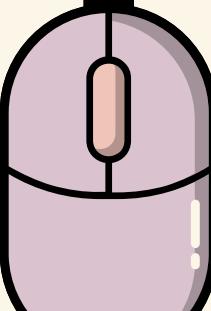
Khám phá thế giới lập trình với ngôn ngữ Python

>>>

Nguyễn Duy Thiên



...



</>

>>>

MỤC LỤC

~~~~  
• • • •

01 Làm Quen, Cài Đặt Và Chạy  
Chương Trình Đầu Tiên

02 Nhập Và Xuất Dữ Liệu

03 Biến, Kiểu Dữ Liệu Và Toán Tử

04 Câu Lệnh Điều Kiện if - else Và  
match - case

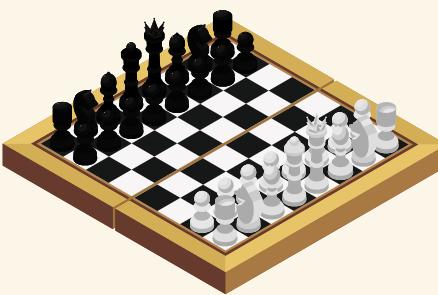
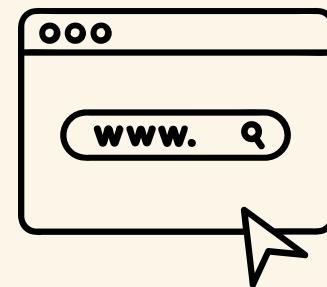
05 Vòng Lặp for, while

06 Bài Tập Ôn Tập Kết Thúc Môn

# CHƯƠNG 1: Làm Quen, Cài Đặt Và Chạy Chương Trình Đầu Tiên

## Bài 1. Lý thuyết cơ bản

- Python là ngôn ngữ lập trình bậc cao, thông dịch, cú pháp ngắn gọn, dễ đọc.
- Được dùng rộng rãi trong: phát triển web, AI, khoa học dữ liệu, tự động hóa, game, ứng dụng desktop.



### Ví dụ về câu lệnh Python

Giải thích:

- **a** và **b** là 2 biến dùng để **lưu trữ dữ liệu**
- **tong** là biến dùng để lưu trữ tổng của a và b
- câu lệnh **print** để in kết quả tổng ra màn hình

```
a = 5  
b = 3  
tong = a + b  
print("Tổng =", tong)
```

## Trắc nghiệm củng cố kiến thức bài 1

**Câu 1: Python là ngôn ngữ lập trình có đặc điểm nào sau đây?**

- A. Chỉ dùng để lập trình game
- B. Có cú pháp dài dòng, khó đọc
- C. Là ngôn ngữ bậc cao, thông dịch, cú pháp ngắn gọn, dễ đọc
- D. Không thể dùng trong AI hoặc khoa học dữ liệu

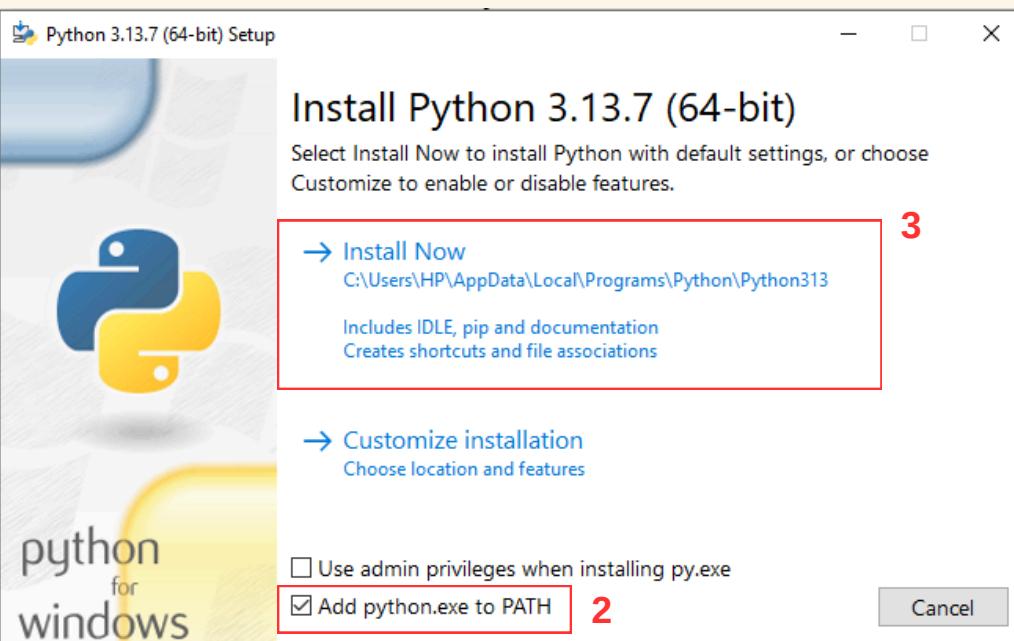
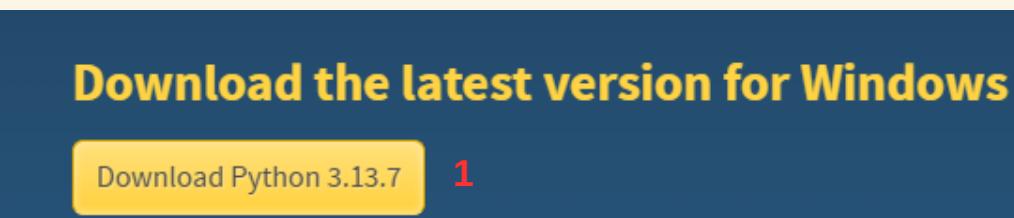
**Câu 2: Python được sử dụng rộng rãi trong lĩnh vực nào sau đây?**

- A. Chỉ phát triển ứng dụng di động
- B. Phát triển web, AI, khoa học dữ liệu, tự động hóa, game, ứng dụng desktop
- C. Chỉ dùng cho hệ điều hành Windows
- D. Chỉ dành cho người mới bắt đầu, không phù hợp chuyên gia

## Bài 2. Hướng dẫn cài đặt Python và trình soạn thảo Visual Studio Code

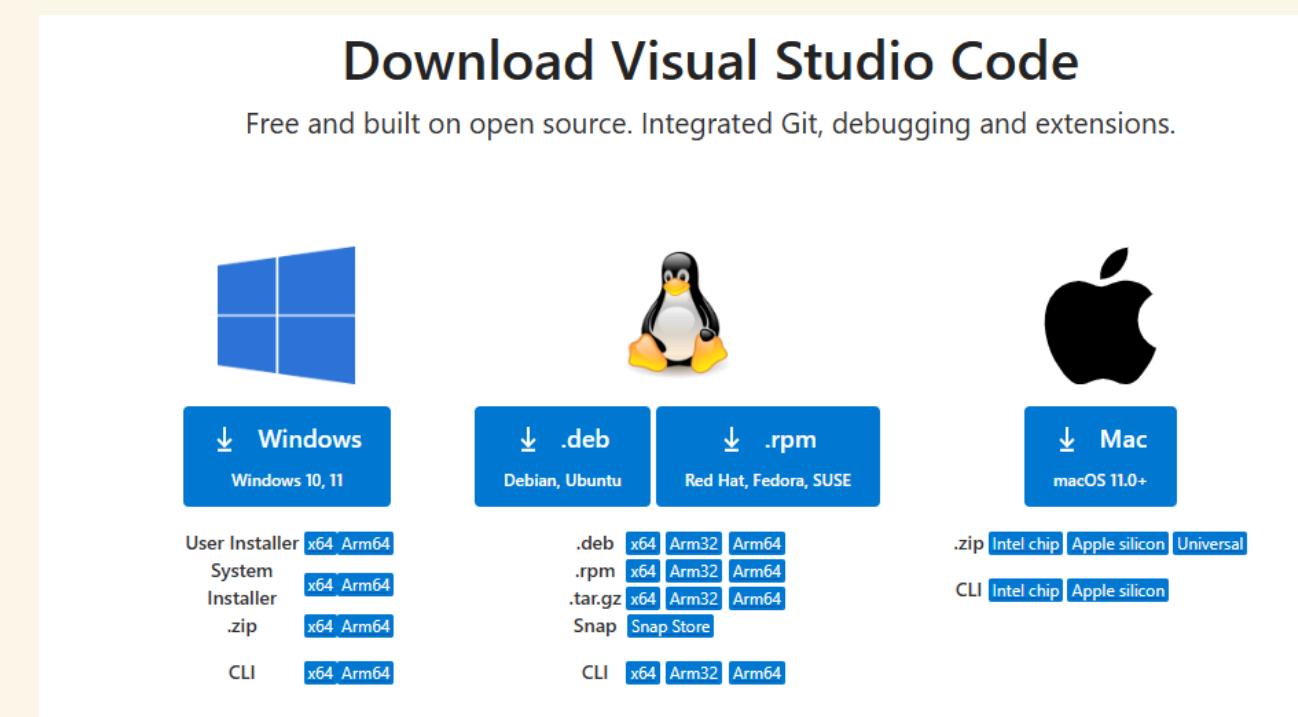
### a. Tải và cài đặt Python

- Vào trang tải: <https://www.python.org/downloads/>
- Bấm Download phiên bản mới nhất và đợi file .exe tải xong.
- Mở file cài đặt → Tích chọn ô “Add Python to PATH”
- Bấm “Install Now”



### b. Tải và cài đặt Visual Studio Code

- Vào trang tải: <https://code.visualstudio.com/Download>
- Tải bản phù hợp với hệ điều hành (Windows / macOS / Linux).
- Tích chọn “accept the agreement” → “Next” → “Install”



## Trắc nghiệm củng cố kiến thức bài 2

**Câu 1: Khi cài đặt Python, bước nào sau đây là *quan trọng nhất* để có thể chạy lệnh Python từ bất kỳ đâu trong hệ thống?**

- A. Chọn “Customize installation”
- B. Nhấn “Install Now” ngay lập tức
- C. Tích chọn ô “Add Python to PATH”
- D. Tải file .exe về rồi xóa đi

**Câu 2: Khi tải Visual Studio Code, người dùng cần làm gì đầu tiên?**

- A. Mở phần mềm và bắt đầu viết code luôn
- B. Vào trang <https://code.visualstudio.com/Download> và tải bản phù hợp với hệ điều hành
- C. Cài đặt Python trước khi tải VS Code
- D. Chỉ tải phiên bản dành cho Windows, không hỗ trợ macOS/Linux

## Bài 3. Mở phần mềm Visual Studio Code và cài đặt Extension cho Python trong VS Code

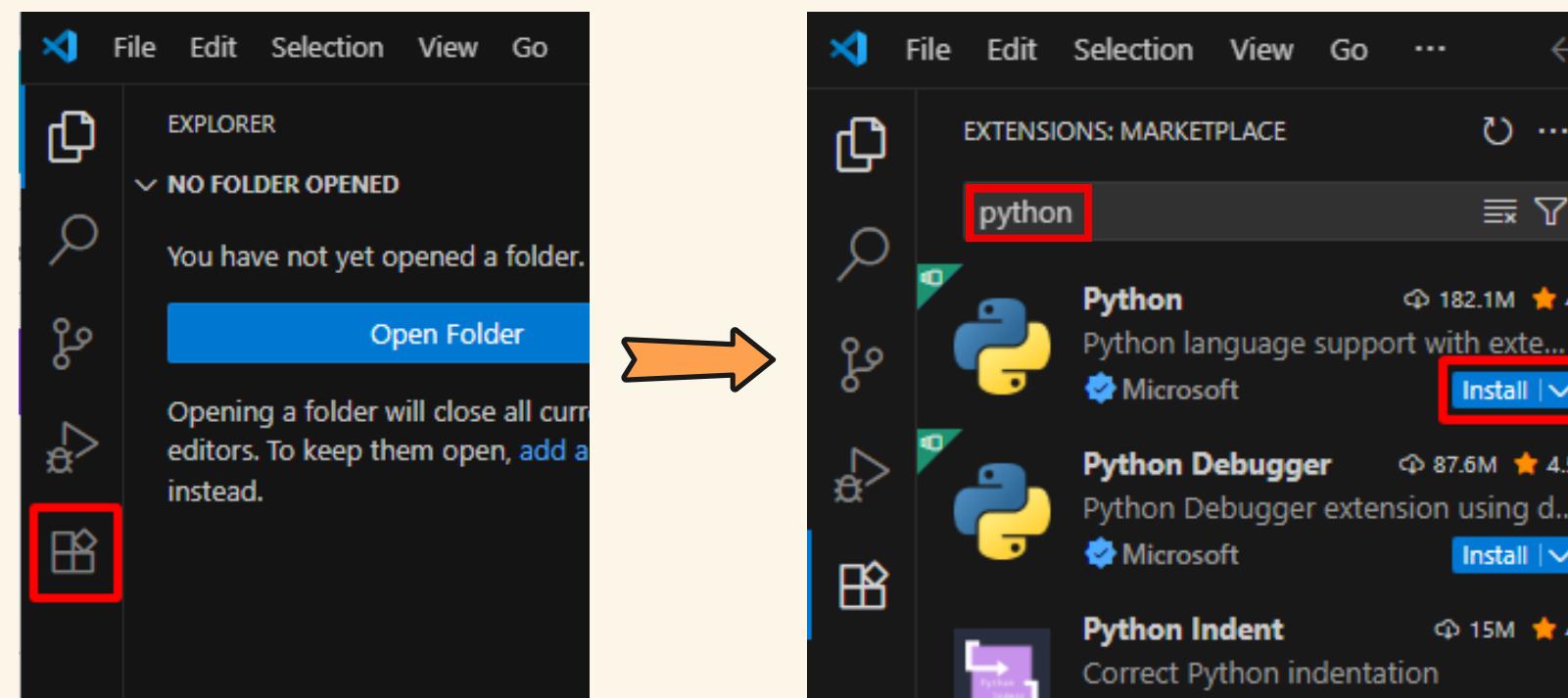
### a. Mở phần mềm

Nhấn chuột trái hai lần liên tiếp vào biểu tượng Visual Studio Code trên màn hình hoặc nhấn Window, gõ “Visual Studio Code” để khởi động

### b. Cài đặt Extension Python

Bước 1. Sau khi mở phần mềm nhấn Ctrl + Shift + X (Windows) hoặc chọn vào biểu tượng

Bước 2. Tìm Python (của Microsoft) → chọn Install.

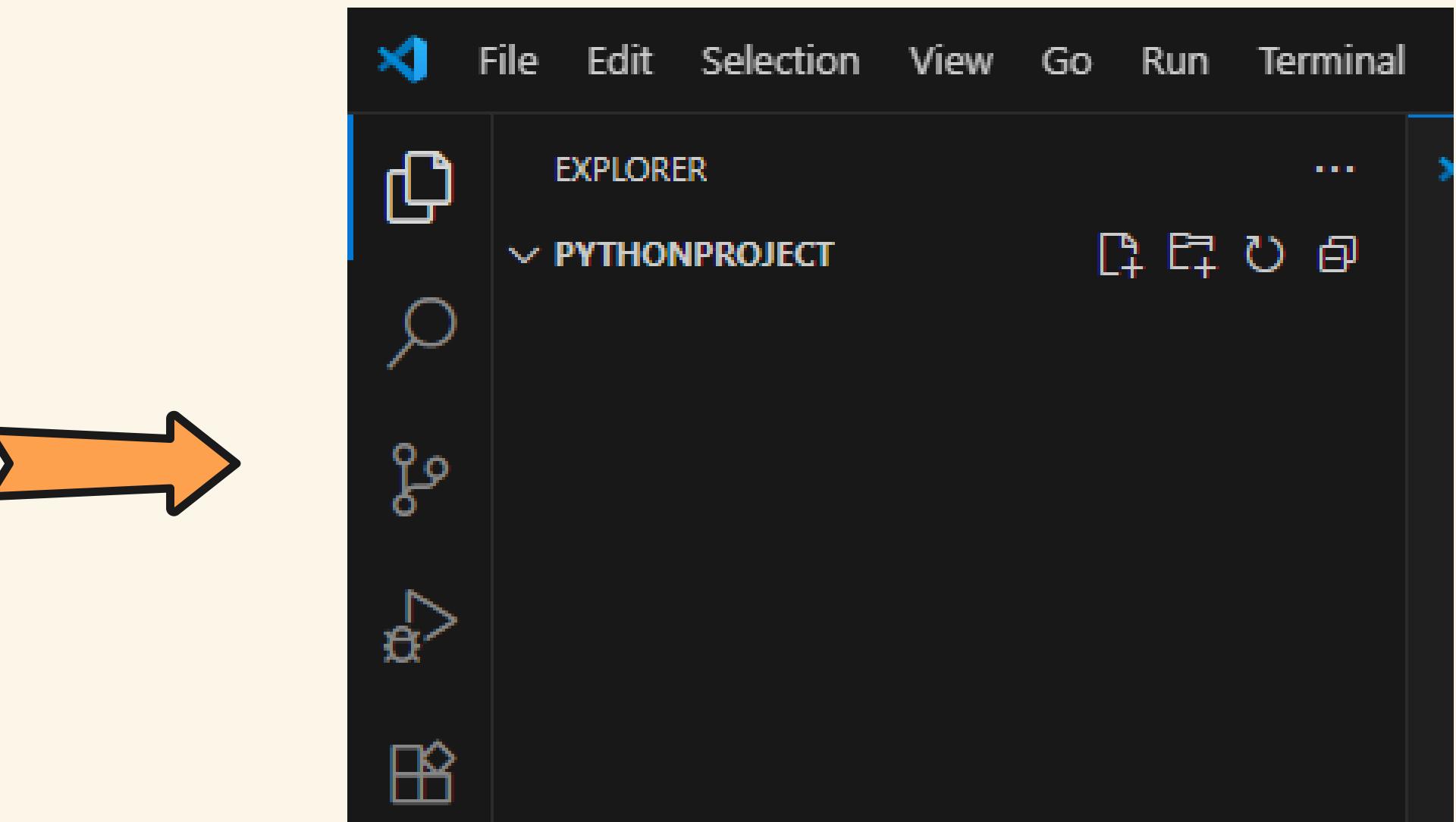
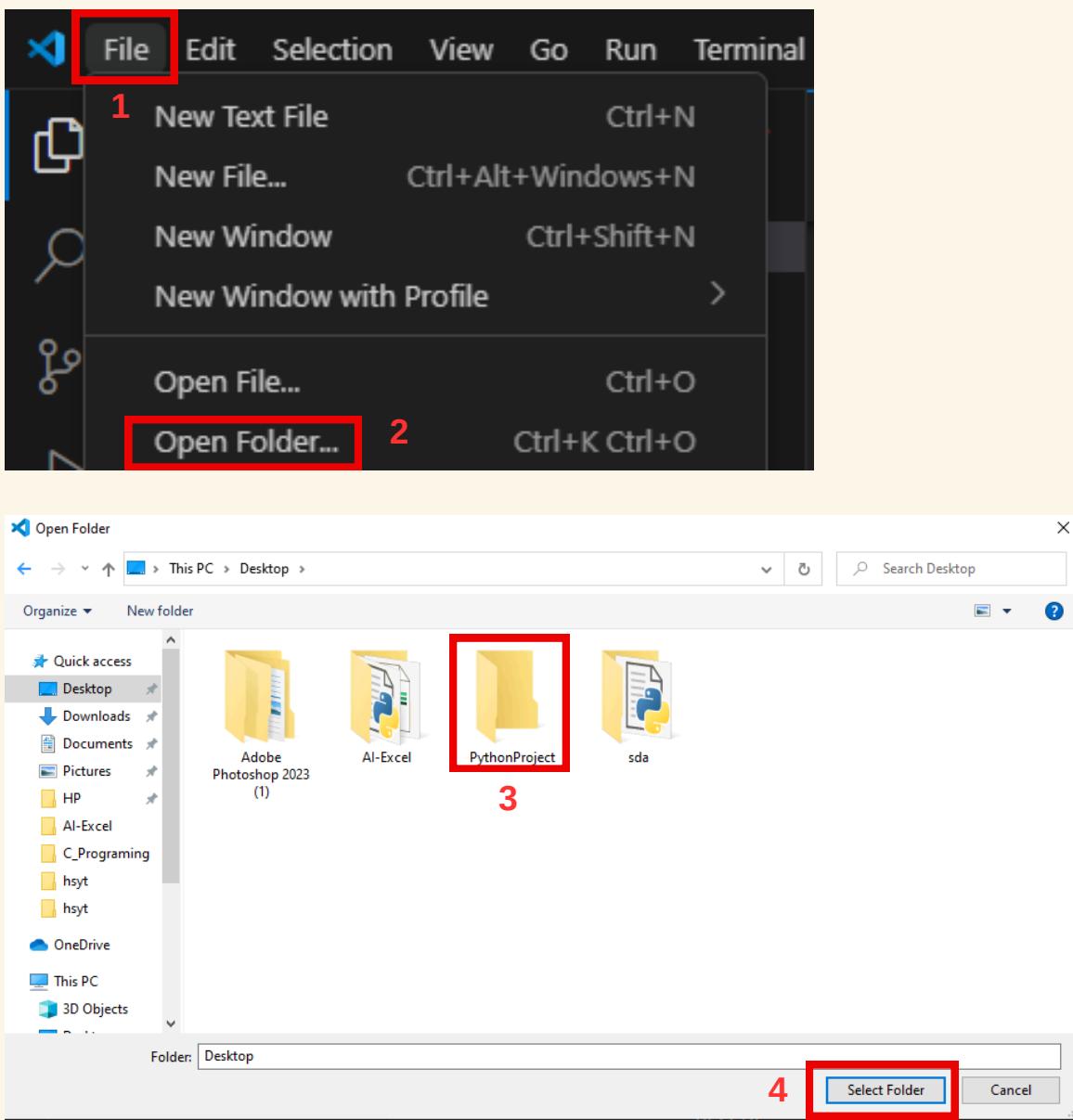


## Bài 4. Tạo file Python, viết và chạy chương trình Python đầu tiên

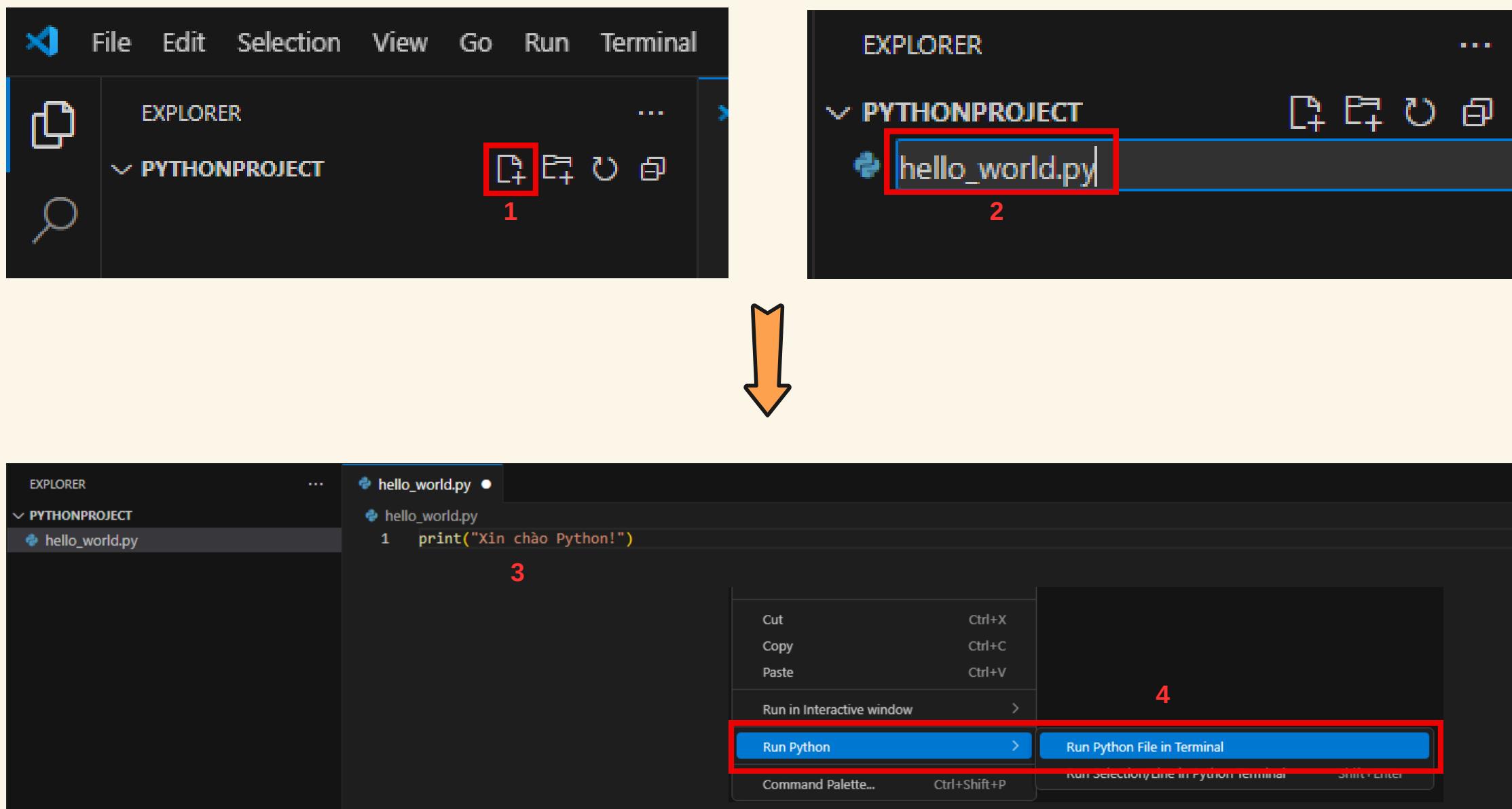
### a. Chuẩn bị thư mục

B1. Chọn File > Open Folder

B2. Chọn thư mục trống có sẵn hoặc tạo một thư mục mới (Ctrl+Shift+N) > Select Folder



## b. Tạo file Python mới



- B1. Chọn vào biểu tượng tạo file 
- B2. Đặt tên cho file
- B3. Viết đoạn code theo mẫu
- B4. Chạy chương trình: Nhấn chuột phải → Run Python → “Run Python File in Terminal”.

Kết quả

Xin chào python!

⚠ Sau khi chạy file Python thì file sẽ tự động được lưu hoặc bấm **Ctrl + S** để lưu

## Trắc nghiệm củng cố kiến thức bài 3 và bài 4

**Câu 1: Để cài đặt extension hỗ trợ Python trong VS Code, bạn cần thực hiện thao tác nào sau đây?**

- A. Nhấn Ctrl + Shift + X để mở Extensions Marketplace, tìm “Python” của Microsoft, rồi chọn Install
- B. Vào File → New File → chọn Python
- C. Tải extension từ Google Play Store
- D. Mở Terminal và gõ pip install python-extension

**Câu 2: Khi đã tạo file hello.py với nội dung print("Xin chào Python!"), để chạy chương trình:**

- A. Nhấp đúp vào file hello.py trong thư mục PythonProject
- B. Nhấn chuột phải vào mã → Run Python → chọn “Run Python File in Terminal”
- C. Gõ lệnh python hello.py trong Command Prompt mà không cần mở VS Code
- D. Lưu file là chương trình tự chạy

## CHƯƠNG 2: Nhập Và Xuất Dữ Liệu

### A. Mục đích nghiên cứu

- Biết cách dùng `print()` để hiển thị thông tin ra màn hình.
- Biết cách dùng `input()` để nhập dữ liệu từ người dùng.
- Hiểu vai trò của nhập và xuất: xuất để in và kiểm tra kết quả, nhập để cung cấp dữ liệu.
- Làm nền tảng cho các chương tiếp theo (biến, toán tử, điều kiện, vòng lặp).

### B. Nội dung nghiên cứu

**Bài 5.** Xuất dữ liệu với `print()`

**Bài 6.** Nhập dữ liệu với `input()`

### C. Bài tập tổng hợp

## Bài 5. Xuất dữ liệu với print()

- Dùng để hiển thị văn bản hoặc kết quả.
- Có thể in chữ, số, hoặc nhiều dữ liệu cùng lúc.

```
print("Xin chào các bạn")
print(123)
print("Hôm nay là", "Thứ Hai")
```

Kết quả

### Cú pháp

- print("chuỗi kí tự")
- print(số)
- printt(<dữ liệu 1>, <dữ liệu 2>, ...)

```
Xin chào các bạn
123
Hôm nay là Thứ Hai
```

Khi dùng print in ra nhiều dữ liệu, giữa các dữ liệu phải có dấu phẩy và Python sẽ tự động thêm khoảng trắng giữa các phần tử.

## Bài 6. Nhập dữ liệu với input()

- Dùng để lấy thông tin từ người dùng.
- Khi chạy, chương trình sẽ hiện câu hỏi và chờ gõ câu trả lời.

```
input("Bạn tên gì? ")
input("Bạn học lớp mấy? ")
```

Kết quả

### Cú pháp

- Input("<yêu cầu hoặc câu hỏi>")

```
Bạn tên gì? (nhập thông tin từ bàn phím → enter)
# Sau khi nhấn enter thì sẽ xuất hiện dòng bên dưới
Bạn học lớp mấy? (nhập thông tin từ bàn phím → enter)
```

Chương trình ví dụ kết hợp *input* và *print*

```
print("Xin chào !!")
print("Bạn tên là", input("Hãy nhập tên của bạn: "))
print("Bạn", input("Hãy nhập số tuổi của bạn: "), "tuổi")
print("Bạn đang học python")
```

Dòng lệnh này sẽ hiển thị “**Hãy nhập tên của bạn:**” để nhận thông tin từ bàn phím  
→ Sau đó mới In ra “Bạn tên là + thông tin vừa nhập”

## Kết quả

```
Xin chào !!
Hãy nhập tên của bạn: Nguyen Van A
Bạn tên là Nguyen Van A
Hãy nhập tuổi của bạn: 15
Bạn 15 tuổi
Bạn đang học python
```

Nhập  
từ  
bàn  
phím

## Trắc nghiệm củng cố kiến thức bài 5 và bài 6

### Câu 1: Hàm `print()` trong Python dùng để làm gì?

- A. Lưu trữ dữ liệu vào biến
- B. Nhập dữ liệu từ người dùng
- C. Hiển thị văn bản, số hoặc nhiều giá trị cùng lúc ra màn hình
- D. Tính toán giá trị toán học

### Câu 3: Khi chạy đoạn mã bên dưới thì chương trình sẽ:

```
input("Bạn tên gì? ")
```

- A. Tự động in ra tên người dùng
- B. In ra “Bạn tên gì?” và dừng lại và chờ người dùng nhập thông tin từ bàn phím
- C. In ra lỗi cú pháp
- D. Không làm gì cả

### Câu 2: Hàm `input()` trong Python dùng để làm gì?

- A. Hiển thị thông báo lỗi trên màn hình
- B. Lưu dữ liệu vào tệp tin
- C. Nhận dữ liệu do người dùng nhập từ bàn phím
- D. Tính toán giá trị của một biểu thức toán học

### Câu 4: Kết quả của đoạn mã sau sẽ là gì?

```
print("Hôm nay là", "Thứ Hai")
```

- A. Hôm nay là Thứ Hai
- B. Hôm nay là Thứ Hai
- C. Hôm nay là, Thứ Hai
- D. Lỗi cú pháp

### Bài tập tổng hợp

#### Bài 1:

Dùng print() viết chương trình in ra:

- Tên của bạn.
- Tuổi của bạn.
- Môn học bạn thích nhất.

\*gợi ý: mỗi dòng dùng một lệnh print()

#### Bài 2:

Dùng input() viết chương trình hỏi:

- "Bạn tên gì?"
- "Bạn thích ăn gì?"
- "Bạn đang học lớp mấy?"

#### Bài 3:

Kết hợp print() và input():

- Viết chương trình hỏi bây giờ là mấy giờ và in lại câu trả lời.
- Viết chương trình hỏi thời tiết hôm nay và in lại câu trả lời.

### Kết quả ví dụ

Tên của tôi là Nguyen Van A

Tôi 16 tuổi

Python là môn học tôi thích nhất

Bạn tên gì? Nguyễn Văn A

Bạn thích ăn gì? Phở

Bạn học lớp mấy? Lớp 10

Bây giờ là mấy giờ rồi? 23 giờ

Bây giờ là 23 giờ

Bạn thấy hôm nay trời thế nào? Nhiều mây

Hôm nay trời Nhiều mây

## CHƯƠNG 3: BIẾN, KIỂU DỮ LIỆU VÀ TOÁN TỬ

### A. Mục đích nghiên cứu

- Hiểu biến là nơi lưu trữ dữ liệu và biết cách đặt tên biến đúng quy tắc.
- Biết các kiểu dữ liệu cơ bản trong Python: int (số nguyên), float (số thực), str(chuỗi), bool (đúng/sai).
- Sử dụng được toán tử số học (+, -, ×, ÷, chia lấy dư, lũy thừa) để làm các phép tính đơn giản.
- Làm quen với việc nhập dữ liệu từ bàn phím → lưu vào biến → tính toán → in kết quả.

### B. Nội dung nghiên cứu

**Bài 7.** Biến - Hộp lưu trữ dữ liệu trong lập trình

**Bài 8.** Các loại toán tử

### C. Bài tập tổng hợp

## Bài 7. Biến - Hộp lưu trữ dữ liệu trong lập trình

### a. Khái niệm

- Giống như một chiếc hộp có tên để cất dữ liệu tạm thời trong quá trình chạy chương trình.
- Khi cần sử dụng giá trị, chỉ cần gọi tên biến.

### b. Khai báo biến

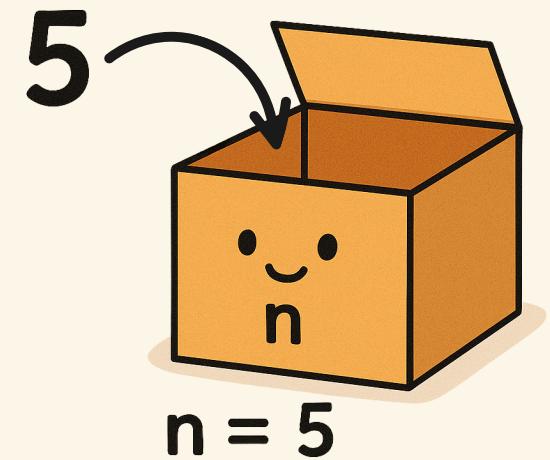
```
ten_bien = gia_tri
```

### c. Quy tắc đặt tên

- Chỉ dùng chữ cái, số và dấu gạch dưới \_.
- Không trùng từ khóa trong Python
- Không bắt đầu bằng số.
- Tên phải dễ hiểu và dễ nhớ.

### d. Kiểu dữ liệu cơ bản

- **int** (số nguyên): 5, 10, -3
- **float** (số thực): 3.14, 2.5
- **str** (chuỗi): "Xin chào", "Python"
- **bool** (đúng/sai): True, False



>>>

## Trắc nghiệm củng cố kiến thức bài 7

**Câu 1: Trong lập trình, biến được dùng để làm gì?**

- A. Lưu trữ dữ liệu tạm thời trong quá trình chạy chương trình
- B. In kết quả ra màn hình
- C. Tự động tính toán giá trị toán học
- D. Ghi lại lịch sử nhập liệu của người dùng

**Câu 3: Kiểu dữ liệu nào sau đây dùng để lưu trữ số nguyên?**

- A. float
- B. str
- C. int
- D. bool

**Câu 2: Câu lệnh nào sau đây là câu lệnh khai báo biến đúng trong Python?**

- A. bien = 10
- B. 10 = bien
- C. bien == 10
- D. int bien = 10

**Câu 4: Tên biến nào sau đây là không hợp lệ trong Python?**

- A. diem\_toan
- B. 1diem
- C. ten\_hoc\_sinh
- D. so\_luong

## Bài 8. Các loại toán tử

### 8a. Toán tử số học

Dùng để thực hiện các phép tính với số

| Toán tử | Ý nghĩa              | Ví dụ<br>(a = 10, b = 3) | Kết quả |
|---------|----------------------|--------------------------|---------|
| +       | Cộng                 | a + b                    | 13      |
| -       | Trừ                  | a - b                    | 7       |
| *       | Nhân                 | a * b                    | 30      |
| /       | Chia                 | a / b                    | 3.333.. |
| //      | Chia lấy phần nguyên | a // b                   | 3       |
| %       | Chia lấy dư          | a % b                    | 1       |
| **      | Lũy thừa             | a**3                     | 1000    |

### Ví dụ minh họa

```
#in phép tính    kết quả
print(10 + 3)   # 13
print(10 - 3)   # 7
print(10 * 3)   # 30
print(10 / 3)   # 3.333...
print(10 // 3)  # 3
print(3 % 3)    # 1
print(10 ** 3)  # 1000
```

## Trắc nghiệm củng cố kiến thức bài 8a

**Câu 1: Phép toán nào sau đây trong Python dùng để lấy phần dư của phép chia?**

- A. /
- B. //
- C. %
- D. \*\*

**Câu 2: Kết quả của biểu thức  $7 // 2$  trong Python là gì?**

- A. 3.5
- B. 3
- C. 4
- D. 1



## 8b. Toán tử gán

Dùng dấu `=` để gán giá trị cho biến. Ngoài ra, Python còn có các toán tử gán kết hợp với phép toán.

| Toán tử          | Ý nghĩa                    |
|------------------|----------------------------|
| <code>=</code>   | Gán giá trị                |
| <code>+=</code>  | Cộng rồi gán               |
| <code>-=</code>  | Trừ rồi gán                |
| <code>*=</code>  | Nhân rồi gán               |
| <code>/=</code>  | Chia rồi gán               |
| <code>//=</code> | Chia lấy số nguyên rồi gán |
| <code>%=</code>  | Chia lấy số dư rồi gán     |
| <code>**=</code> | Lũy thừa rồi gán           |

### Ví dụ minh họa

```
# Ví dụ toán tử gán trong Python
```

```
x = 10    # x = 10
x += 5    # x = 10 + 5 => x = 15
x -= 3    # x = 15 - 3 => x = 12
x *= 2    # x = 12 * 2 => x = 24
x /= 4    # x = 24 / 4 => x = 6.0
x //= 2   # x = 6.0 // 2 => x = 3.0
x %= 2    # x = 1.0 % 2 => x = 1.0
x **= 3   # x = 1.0 ** 3 => x = 1.0
```

#### ⚠ Lưu ý:

- Khi gán giá trị trực tiếp thì biến sẽ tự động có kiểu dữ liệu theo giá trị
- Nếu gán giá trị được nhập từ `input` thì biến sẽ có kiểu dữ liệu là `string`  
→ nếu muốn tính toán thì phải ép kiểu dữ liệu thành `int` hoặc `float`

```
chieu_cao = float(input("Nhập chiều cao (m): "))
```

## Trắc nghiệm củng cố kiến thức bài 8b

**Câu 1: Trong Python, toán tử `+=` có nghĩa là gì?**

- A. Gán giá trị mới cho biến
- B. Cộng rồi gán lại cho biến
- C. Trừ rồi gán lại cho biến
- D. Nhân rồi gán lại cho biến

**Câu 2: Kết quả in ra màn hình của đoạn mã bên dưới là gì?**

```
x = 6  
x *= 2  
print(x)
```

- A. 6
- B. 8
- C. 12
- D. 3

## 8c. Toán tử so sánh

Dùng để so sánh hai giá trị, kết quả trả về **True** (đúng) hoặc **False** (sai).

| Toán tử            | Ý nghĩa           |
|--------------------|-------------------|
| <code>==</code>    | Bằng              |
| <code>!=</code>    | Khác nhau         |
| <code>&gt;</code>  | Lớn hơn           |
| <code>&lt;</code>  | Nhỏ hơn           |
| <code>&gt;=</code> | Lớn hơn hoặc bằng |
| <code>&lt;=</code> | Nhỏ hơn hoặc bằng |

### Ví dụ minh họa

```
# Ví dụ toán tử so sánh trong Python  
a = 5  
b = 3  
print(a == b) # False  
print(a != b) # True  
print(a > b) # True  
print(a < b) # False  
print(a >= b) # True  
print(a <= b) # False
```

## Trắc nghiệm củng cố kiến thức bài 8c

**Câu 1: Trong Python, toán tử == dùng để làm gì?**

- A. Gán giá trị cho biến
- B. So sánh hai giá trị có bằng nhau hay không
- C. Kiểm tra xem một giá trị có lớn hơn giá trị kia không
- D. Tính tổng của hai số

**Câu 2: Kết quả in ra màn hình của đoạn mã bên dưới là gì?**

```
x = 8  
y = 5  
print(x >= y)
```

- A. True
- B. False
- C. 8
- D. 5

~~~~~  
.....

Bài tập tổng hợp

Bài tập 1

Nhập từ bàn phím 2 cạnh của hình chữ nhật. Tính diện tích, chu vi và in kết quả ra màn hình.

Bài tập 3

Nhập từ bàn phím số kg và chiều cao (m). Viết chương trình tính và in ra màn hình số BMI
Biết công thức là: $BMI = \text{kg}/(\text{chieucao} * \text{chieucao})$;

Bài tập 5

Nhập từ bàn phím số tiền VND muốn đổi sang USD. Biết rằng $1 \text{ USD} = 25000 \text{ VND}$. In ra màn hình số tiền đã được quy đổi.

Bài tập 2

Nhập từ bàn phím 3 số nguyên a, b, c. Tính trung bình cộng của 3 số và in kết quả ra màn hình.

Bài tập 4

Nhập từ bàn phím số km (s) và thời gian (t) sau đi hết 1 quãng đường. Tính vận tốc (v) đi là bao nhiêu km/h và in kết quả ra màn hình.
Biết công thức tính vận tốc là: $v = s/t$.



Kết quả ví dụ

Bài tập 1

```
Nhap chieu dai: 5  
Nhap chieu rong: 10  
dien tich hinh chu nhat la 50.0  
chu vi hinh chu nhat la 30.0
```

Bài tập 2

```
nhap a: 4  
nhap b: 5  
nhap c: 7  
trung binh cong: 5.333333333333333
```

Bài tập 3

```
CHUONG TRINH TINH BMI  
hay nhap so kg: 80  
hay nhap chieu cao(m): 1.6  
BMI cua ban la: 31.24999999999993
```

Bài tập 4

```
nhap so km: 120  
nhap so thoi gian(h): 2.5  
van toc = 48.0km/h
```

Bài tập 5

```
Hay nhap so tien VND muon quy doi thanh USD: 2350000  
2350000 VND duoc doi thanh 94.00 USD
```

CHƯƠNG 4: CÂU LỆNH ĐIỀU KIỆN IF - ELSE VÀ MATCH - CASE

A. Mục đích nghiên cứu

- Hiểu cách sử dụng câu lệnh điều kiện if-else để ra quyết định trong chương trình.
- Biết cách xây dựng các nhánh xử lý khác nhau dựa trên điều kiện đúng hoặc sai.
- Vận dụng if-else để giải quyết các bài toán thực tế như kiểm tra, so sánh, phân loại dữ liệu.
- Nắm vững cách dùng switch-case để xử lý nhiều tình huống lựa chọn thay thế cho nhiều lệnh if-elif-else liên tiếp.

B. Nội dung nghiên cứu

Bài 9. if - Kiểm tra điều kiện đơn

Bài 10. if - else - Kiểm tra hai điều kiện ngược nhau

Bài 11. if - elif - else - Kiểm tra nhiều điều kiện

Bài 12. match - case - Kiểm tra nhiều điều kiện

C. Bài tập tổng hợp

Bài 9. if – Kiểm tra điều kiện đơn

- Cú pháp:

```
if điều_kiện:  
    # khói lệnh chạy khi điều kiện đúng
```

- Chỉ có một nhánh **thực hiện nếu điều kiện đúng**, nếu **sai thì bỏ qua**.

Bài 10. if - else – Kiểm tra hai điều kiện ngược nhau

- Cú pháp:

```
if điều_kiện:  
    # khói lệnh khi điều kiện đúng  
else:  
    # khói lệnh điều kiện sai
```

- Nếu điều kiện **đúng** → chạy nhánh **if**.
- Nếu điều kiện **sai** → chạy nhánh **else**.

⚠ Lưu ý: Trong Python khói lệnh bắt buộc phải chính xác, dùng 1 tab khói lệnh để xác định khói lệnh.

Bài 11. if - elif - else – Kiểm tra nhiều điều kiện

- Cú pháp:

```
if điều_kiện_1:  
    # khi điều kiện 1 đúng  
  
elif điều_kiện_2:  
    # khi điều kiện 1 sai nhưng điều kiện 2 đúng  
  
elif điều_kiện_3:  
    # khi điều kiện 1 và 2 sai nhưng điều kiện 3 đúng  
  
else:  
    # khi tất cả điều kiện trước đều sai
```

- Cho phép kiểm tra nhiều khả năng theo thứ tự.
- Chỉ một nhánh được thực hiện (nhánh đầu tiên có điều kiện đúng).

Trắc nghiệm củng cố kiến thức bài 9, 10, 11

Câu 1: Khi sử dụng cấu trúc if trong Python, điều gì xảy ra nếu điều kiện là False?

- A. Chương trình sẽ báo lỗi
- B. Khối lệnh trong if vẫn được thực hiện
- C. Khối lệnh trong if bị bỏ qua
- D. Chương trình tự động chạy khối else

Câu 2: Trong cấu trúc if-else, khi điều kiện là False, chương trình sẽ:

- A. Không làm gì cả
- B. Chạy khối lệnh trong if
- C. Chạy khối lệnh trong else
- D. Tự động chuyển sang elif

Câu 3: Trong cấu trúc if-elif-else, câu nào sau đây là đúng?

- A. Tất cả các nhánh elif đều được kiểm tra và thực hiện nếu điều kiện đúng
- B. Chỉ nhánh đầu tiên có điều kiện True được thực hiện, sau đó sẽ bỏ qua các trường hợp còn lại
- C. Nếu không có else, chương trình sẽ luôn lỗi
- D. elif có thể đứng trước if

Bài 12. match - case - Kiểm tra nhiều điều kiện

- Cú pháp:

```
match biến:  
    case giá_trị_1:  
        # khối lệnh khi biến == giá_trị_1  
    case giá_trị_2:  
        # khối lệnh khi biến == giá_trị_2  
    # ... các case khác  
    case _:  
        # khối lệnh khi không khớp giá trị nào ở trên
```

Chú thích

- biến: giá trị cần so khớp (ví dụ như số, chuỗi, ...).
- case: mỗi mẫu để kiểm tra sự khớp với giá trị của biến.
- case _ (dấu gạch dưới): thực hiện khi không khớp bất kỳ case nào.

- Dùng khi một biến cần so khớp với nhiều giá trị cố định.
- Giúp mã gọn hơn khi thay thế nhiều câu lệnh if-elif liên tiếp.

Trắc nghiệm củng cố kiến thức bài 12

Câu 1: Cú pháp nào sau đây là đúng khi sử dụng match-case trong Python?

A.

```
match x:  
    case 1:  
        print("Một")
```

C.

```
match(x):  
    case 1:  
        print("Một")
```

B.

```
match x:  
    case 1:  
        print("Một")
```

D.

```
math x:  
    case 1:  
        print("Một")
```

Câu 2: Trong cấu trúc match-case, **case_ được dùng để làm gì?**

- A. Khởi tạo biến mới
- B. So sánh với giá trị bằng 0
- C. Thực hiện khi không khớp với bất kỳ case nào trước đó
- D. Dừng chương trình và thoát khỏi khối match

Bài tập tổng hợp

Bài tập 1

- Nhập điểm trung bình (0–10).
- Xếp loại theo tiêu chí:
 - 9–10 → "Xuat sac"
 - 8–8.9 → "Gioi"
 - 6.5–7.9 → "Kha"
 - 5–6.4 → "Trung binh"
 - < 5 → "Yeu"
- Nếu nhập ngoài khoảng 0–10 → in "Diem khong hop le".

Bài tập 2

- Nhập số điện tiêu thụ (kWh).
- Tính tiền điện theo bảng giá giả định:
 - 0–50 kWh: 2000đ/kWh
 - 51–100 kWh: 2500đ/kWh
 - Trên 100 kWh: 3000đ/kWh
- In ra số tiền phải trả.

Kết quả ví dụ

```
Nhap diem trung binh: 8  
Gioi
```

```
nhap so dien tieu thu (kwh): 65  
So tien phai tra la: 162500.0
```

Bài tập 3

Viết chương trình nhập vào:

1. Cấp học (nhập số):
 - 1: Cấp 1 (Tiểu học)
 - 2: Cấp 2 (THCS)
 - 3: Cấp 3 (THPT)

2. Điểm các môn: Toán, Văn, Anh Văn (float, thang 10)

Yêu cầu:

- Nếu cấp 1: Trung bình = $(Toán + Văn + Anh Văn) / 3$
- Nếu cấp 2: Trung bình = $(Toán * 2 + Văn * 2 + Anh Văn) / 5$
- Nếu cấp 3: Trung bình = $(Toán + Văn + Anh Văn * 2) / 4$

Nếu nhập sai cấp học thì báo “Cấp học không hợp lệ”.

Kết quả ví dụ

```
Nhap cap hoc (1: Cap 1, 2: Cap 2, 3: Cap 3): 2
Nhap diem Toan: 7
Nhap diem Van: 8
Nhap diem Anh Van: 9
Diem trung binh Cap 2 = 7.80
```

Bài tập 4

Máy ATM

Khởi tạo một biến số dư và gán giá trị bất kì (float)

Khi khởi động, chương trình có 4 nút chức năng bao gồm:

- | | |
|-----------------------|----------------------|
| 0 Thoát chương trình. | 1 Xem số dư hiện tại |
| 2 Nạp tiền. | 3 Rút tiền |

Người dùng nhập số đại diện cho chức năng muốn sử dụng, sau đó thực hiện chức năng theo yêu cầu

Kết quả ví dụ

```
===== MAY ATM =====
```

0. Thoat chuong trinh
 1. Xem so du hien tai
 2. Nap tien
 3. Rut tien
- ```
=====
```

```
Nhap lua chon: 2
```

```
Nhap so tien muon nap: 50000
```

```
Nap tien thanh cong. So du moi: 1050000.0 VND
```

# CHƯƠNG 5: VÒNG LẶP FOR, WHILE

## A. Mục đích nghiên cứu

- Hiểu cơ chế hoạt động của các vòng lặp while và for để lặp lại khối lệnh nhiều lần.
- Biết lựa chọn loại vòng lặp phù hợp với từng tình huống (lặp khi biết trước số lần, lặp khi chưa biết số lần).
- Vận dụng vòng lặp để giải quyết các bài toán thực tế như tính toán, duyệt mảng,...

## B. Nội dung nghiên cứu

**Bài 13.** Vòng lặp while

**Bài 14.** Vòng lặp for

## C. Bài tập tổng hợp

## Bài 13. Vòng lặp while

- Lặp lại một khối lệnh khi điều kiện còn đúng.
- Kiểm tra điều kiện trước khi thực hiện vòng lặp.
- Nếu điều kiện sai ngay từ đầu → vòng lặp không chạy lần nào.

```
while điều_kiện:
 # khối lệnh lặp
```

### Ví dụ minh họa lặp từ 1 đến 5

```
i = 1
while i <= 5:
 print("Lần lặp thứ", i)
 i += 1 # Đừng quên cập nhật biến!
Kết quả in ra Lần lặp thứ 1 cho đến 5
```

🔥 Lưu ý: Nếu quên `i += 1` -> vòng lặp sẽ lặp vô tận!

## Bài 14. Vòng lặp for

- Lặp lại một khối lệnh qua một dãy như danh sách, khoảng số,...
- Số lần lặp biết trước, bằng số phần tử trong dãy.
- Không cần kiểm tra điều kiện — tự động dừng khi hết phần tử.

```
for biến in dãy:
 # khối lệnh lặp
```

### Ví dụ lặp qua dãy số

```
for i in range(1, 11):
 print("Lần lặp thứ", i)
Kết quả in ra Lần lặp thứ 1 đến 10
```

## Trắc nghiệm củng cố kiến thức bài 13 và bài 14

### Câu 1: Vòng lặp while trong Python sẽ:

- A. Lặp lại khôi lệnh khi điều kiện còn đúng, và kiểm tra điều kiện trước mỗi lần lặp
- B. Không cần điều kiện để chạy
- C. Luôn chạy ít nhất 1 lần
- D. Tự động dừng sau 10 lần lặp

### Câu 3: Câu nào sau đây mô tả đúng về vòng lặp for trong Python?

- A. Cần phải kiểm tra điều kiện ở đầu mỗi lần lặp
- B. Dùng để lặp một số lần không xác định
- C. Lặp qua từng phần tử trong một dãy (danh sách, chuỗi, range...) — số lần lặp biết trước
- D. Không thể dùng với số nguyên

### Câu 2: Khi nào nên dùng vòng lặp while thay vì for?

- A. Khi muốn in các số từ 1 đến 10
- B. Khi cần lặp qua một danh sách có sẵn
- C. Khi chưa biết trước số lần lặp — ví dụ: yêu cầu người dùng nhập số dương
- D. Khi muốn lặp chính xác 5 lần

### Câu 4: Trong đoạn mã sau chương trình sẽ:

```
i = 1
while i <= 5:
 print(i)
```

- A. In ra 1, 2, 3, 4, 5 rồi dừng
- B. Chỉ in ra 1 lần rồi dừng
- C. Báo lỗi
- D. In vô hạn số 1 → lặp vô tận

### Bài tập tổng hợp

#### Bài tập 1

In các số lẻ từ 1 đến 20 ra màn hình

#### Bài tập 3

Nhập một số nguyên dương n, in ra giai thừa n!  
(ví dụ  $5! = 1*2*3*4*5 = 120$ ).

#### Bài tập 2

In ra các số chẵn giảm dần từ 20 về 0

#### Bài tập 4

Nhập một số nguyên dương n, in ra màn hình tổng các  
số chẵn từ 0 đến n



## Kết quả ví dụ

### Bài tập 1

Các số lẻ từ 1 đến 20:

1  
3  
5  
...  
19

### Bài tập 2

Các số chẵn giảm dần từ 20 về 0:

20  
18  
16  
...  
0

### Bài tập 3

Nhập một số nguyên dương n: 5

$5! = 120$

### Bài tập 4

Nhập một số nguyên dương n: 10

Tổng các số chẵn từ 0 đến 10 là: 30

## CHƯƠNG 6: BÀI TẬP ÔN TẬP KẾT THÚC MÔN

### Bài tập 1: Fibonacci

In ra dãy số Fibonacci 0, 1, 1, 2, 3, 5, 8, ...  
(gợi ý sử dụng vòng lặp for)

### Bài tập 2: Kiểm tra số hoàn hảo

Nhập số nguyên n, kiểm tra số vừa nhập có phải là  
số hoàn hảo hay không  
(Số hoàn hảo là số có tổng các ước bằng chính nó)

### Bài tập 3: Kiểm tra số chính phương

Nhập số nguyên n, kiểm tra số vừa nhập có phải là  
số chính phương không  
(Số chính phương là số bằng bình phương của một số tự nhiên)

### Bài tập 4: Tổng các số nguyên tố

Nhập số nguyên n, kiểm tra từ 1 đến n có bao nhiêu  
số nguyên tố, in ra tổng các số nguyên tố  
(Số nguyên tố là số có ước bằng 1 và chính nó)

### Bài tập 5: Tìm ước chung lớn nhất

Nhập 2 số nguyên a, b. Đếm a và b có bao nhiêu  
ước chung, tìm ước chung lớn nhất của 2 số a và b  
và in ra màn hình

### Bài tập 6: In bảng cửu chương

Nhập số nguyên n sau đó in bảng cửu chương n  
theo mẫu:  
 $n \times 1 = \dots$   
 $n \times 2 = \dots$

## CHƯƠNG 1

### Bài 1

Câu 1. C

Câu 2. B

### Bài 2

Câu 1. C

Câu 2. B

### Bài 3 và Bài 4

Câu 1. A

Câu 2. B

## CHƯƠNG 2

### Bài 5 và Bài 6

Câu 1. C

Câu 2. C

Câu 3. B

Câu 4. B

## CHƯƠNG 3

### Bài 7

Câu 1. A

Câu 2. A

Câu 3. C

Câu 4. B

### Bài 8a

Câu 1. C

Câu 2. B

### Bài 8b

Câu 1. B

Câu 2. C

### Bài 8c

Câu 1. B

Câu 2. A

## CHƯƠNG 4

### Bài 9, 10, 11

Câu 1. C

Câu 2. C

Câu 3. B

### Bài 12

Câu 1. B

Câu 2. C

## CHƯƠNG 5

### Bài 13 và bài 14

Câu 1. A

Câu 2. C

Câu 3. C

Câu 4: D

## CHƯƠNG 2

### Bài tập 1

```
print("Tên của tôi là Nguyen Van A")
print("Tôi 16 tuổi")
print("Python là môn học tôi thích nhất")
```

### Bài tập 2

```
input("Bạn tên gì? ")
input("Bạn thích ăn gì? ")
input("Bạn học lớp mấy? ")
```

### Bài tập 3

```
print("Bây giờ là", input("Bây giờ là mấy giờ rồi? "))
print("Hôm nay trời", input("Bạn thấy hôm nay trời thế nào? "))
```

~~~~~  
• • • •



## CHƯƠNG 3

### Bài tập 1

```
cd = float(input("Nhập chiều dài: "))
cr = float(input("Nhập chiều rộng: "))
print("diện tích hình chữ nhật là", cd * cr)
print("chu vi hình chữ nhật là", (cd + cr) * 2)
```

### Bài tập 3

```
print("CHƯƠNG TRÌNH TÍNH BMI")
kg = float(input("Hay nhập số kg: "))
chieucao = float(input("Hay nhập chiều cao(m): "))
bmi = kg / (chieucao ** 2) # hoặc kg / (chieucao * chieucao)
print("BMI của bạn là:", bmi)
```

### Bài tập 5

```
tienVND = int(input("Hay nhập số tiền VND muốn quy đổi thành USD: "))
tienUSD = tienVND / 25000
print(tienVND, "VND được đổi thành", tienUSD, "USD")
```

### Bài tập 2

```
a = int(input("nhập a: "))
b = int(input("nhập b: "))
c = int(input("nhập c: "))
tbc = (a + b + c) / 3
print("trung bình cộng: ", tbc)
```

### Bài tập 4

```
s = float(input("nhập số km: "))
t = float(input("nhập số thời gian(h): "))
v = s / t
print("vận tốc =", v, "km/h")
```

## CHƯƠNG 4

## Bài tập 1

```
dtb = float(input("Nhập điểm trung bình: "))

if dtb > 10 or dtb < 0:
 print("điểm trung bình không hợp lệ")
elif dtb >= 9:
 print("Xuất sắc")
elif dtb >= 8:
 print("Gioi")
elif dtb >= 6.5:
 print("Kha")
elif dtb >= 5:
 print("Trung binh")
else:
 print("Yếu")
```

## Bài tập 2

```
kwh = float(input("nhập số điện tiêu thụ (kwh): "))

if kwh < 0:
 print("số điện tiêu thụ không hợp lệ")
else:
 if kwh <= 50:
 soTienPhaiTra = 2000 * kwh
 elif kwh <= 100:
 soTienPhaiTra = 2500 * kwh
 else:
 soTienPhaiTra = 3000 * kwh

 print("Số tiền phải trả là: ", soTienPhaiTra)
```

### Bài tập 3

```
capHoc = int(input("Nhập cấp học (1: Cấp 1, 2: Cấp 2, 3: Cấp 3): "))

toan = float(input("Nhập điểm Toán: "))
van = float(input("Nhập điểm Văn: "))
anh = float(input("Nhập điểm Anh Văn: "))

match capHoc:
 case 1:
 trungBinh = (toan + van + anh) / 3
 print("Điểm trung bình Cấp 1 = ", trungBinh)
 case 2:
 trungBinh = (toan * 2 + van * 2 + anh) / 5
 print("Điểm trung bình Cấp 2 = ", trungBinh)
 case 3:
 trungBinh = (toan + van + anh * 2) / 4
 print("Điểm trung bình Cấp 3 = ", trungBinh)
 case _:
 print("Cấp học không hợp lệ.")
```

## Bài tập 4

```
soDu = 1000000.0

print("\n===== MAY ATM =====")
print("0. Thoat chuong trinh")
print("1. Xem so du hien tai")
print("2. Nap tien")
print("3. Rut tien")
print("=====")
luaChon = int(input("Nhap lua chon: "))
match luaChon:
 case 0:
 print("Cam on ban da su dung ATM.")
 case 1:
 print("So du hien tai:", soDu, "VND")
 case 2:
 soTien = float(input("Nhap so tien muon nap: "))
```

```
if soTien > 0:
 soDu += soTien
 print("Nap tien thanh cong. So du moi:", soDu, "VND")
else:
 print("So tien nap khong hop le.")

case 3:
 soTien = float(input("Nhap so tien muon rut: "))
 if soTien > 0 and soTien <= soDu:
 soDu -= soTien
 print("Rut tien thanh cong. So du con lai:", soDu, VND")
 else:
 print("Rut tien that bai. So tien khong hop le hoac
khong du so du.")

case _:
 print("Lua chon khong hop le. Vui long chon lai.")
```

## CHƯƠNG 5

## Bài tập 1

```
print("Các số lẻ từ 1 đến 20:")
for i in range(1, 21):
 if i % 2 == 1:
 print(i)
```

## Bài tập 3

```
n = int(input("Nhập một số nguyên dương n: "))
giai_thua = 1
for i in range(1, n + 1):
 giai_thua = giai_thua * i
print(f"{n}! = {giai_thua}")
```

## Bài tập 2

```
print("Các số chẵn giảm dần từ 20 về 0:")
for i in range(20, -1, -2):
 print(i)
```

## Bài tập 4

```
n = int(input("Nhập một số nguyên dương n: "))
tong = 0
for i in range(0, n + 1):
 if i % 2 == 0:
 tong = tong + i
print("Tổng các số chẵn từ 0 đến" , n , "là:", tong)
```