

Đặt tên file trong python

Trong python tên file.py Ví dụ: lab1_bai1.py

Sử dụng Jupyter Notebook

Đặt tên file tên file.ipynb Ví Dụ: lab1.ipynb

Chạy chương trình Python

từ CMD hay Power Shell chạy lệnh : python lab1.py

✓ Chương trình đầu tiên

```
print("Hello, World!")  
print('This is a frst program')
```

```
Hello, World!  
This is a frst program
```

✓ Nhập dữ liệu

Nhập dữ liệu (Input)

1. Nhập từ bàn phím
2. Đọc từ file
3. Đọc từ cơ sở dữ liệu
4. Đọc từ Website
5. Đọc từ dòng lệnh (Command line arguments)
6. Đọc từ biến môi trường (Environment variables)
7. Đọc từ API (Web service)
8. Đọc từ Cảm biến / Thiết bị ngoại vi (IoT, Serial port)
9. Đọc từ Stream / Socket / Network
10. Đọc từ Clipboard (Bảng nhớ tạm)

✓ 1. Nhập từ bàn phím

Hàm để nhập dữ liệu từ bàn phím là: input()

```
# nhập dữ liệu  
name = input("Enter your name: ")  
age = input("Enter your age: ")  
height = input("Enter your height (in cm): ")  
weight = input("Enter your weight (in kg): ")  
print(name, age, height, weight)  
print(type(name), type(age), type(height), type(weight))
```

```
#print(f"Hello {name}! You are {age} years old {height} cm tall and weigh {weight} kg ")
```

```
print("Hello, {name}. You are {age} years old, {height} cm tall and weigh {weight} kg.")
```

```
<class 'str'> <class 'str'> <class 'str'> <class 'str'>
```

▼ Hàm input() nhận dạng dữ liệu nhập từ bàn phím là string

Muốn nhập đúng kiểu dữ liệu mình cần cần ép kiểu

```
# Nhập dữ liệu
name = input("Enter your name: ")
age = int(input("Enter your age: "))
height = float(input("Enter your height (in cm): "))
weight = float(input("Enter your weight (in kg): "))
print(type(age), type(height), type(weight))
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[36], line 3
      1 # Nhập dữ Liệu
      2 name = input("Enter your name: ")
----> 3 age = int(input("Enter your age: "))
      4 height = float(input("Enter your height (in cm): "))
      5 weight = float(input("Enter your weight (in kg): "))

ValueError: invalid literal for int() with base 10: ''
```

```
x = input("nhập x: ")
print(type(x))
```

```
<class 'str'>
```

```
x = int(input("nhập x: "))
print(type(x))
```

```
x = str(input("nhập x: "))
print(type(x))
```

```
x = float(input("nhập x: "))
print(type(x))
```

```
# mọi chuỗi khác rỗng đều là True
x = bool(input("nhập x: "))
print(x)
print(type(x))
```

```
True
<class 'bool'>
```

▼ Xuất dữ liệu dùng hàm print()

```
# nhập dữ liệu
name = input("Enter your name: ")
age = int(input("Enter your age: "))
height = float(input("Enter your height (in cm): "))
weight = float(input("Enter your weight (in kg): "))

print("Hello, %s! You are %d years old, %f cm tall and weigh %f kg." % (name, age, height, weight))
print("Hello, %s! You are %d years old, %0.2f cm tall and weigh %0.2f kg." % (name, age, height, weight))
print("Hello, %r! You are %r years old, %r cm tall and weigh %r kg." % (name, age, height, weight))
print(f"Hello, {name}! You are {age} years old, {height} cm tall and weigh {weight} kg.")
```

```
Hello, Huy! You are 45 years old, 175.000000 cm tall and weigh 65.000000 kg.
Hello, 'Huy'! You are 45 years old, 175.0 cm tall and weigh 65.0 kg.
Hello, Huy! You are 45 years old, 175.0 cm tall and weigh 65.0 kg.
```

```
print('...')
tusong = "Sông"
print(f"{tusong} không giận, không hờn không oán trách")
print()
```

```
"..."
Sông không giận, không hờn không oán trách
```

```
print(f"{'-'*85}")

print(f"|{'MSSV':<10}|{'Tên':<20}|{'Email':<25}|{'Số điện thoại':<15}|{'Lớp':<10}|")
print(f"{'-'*85}")
```

```
-----
|MSSV      |Tên          |Email          |Số điện thoại |Lớp          |
-----
```

✓ Comment trong python

```
# Nhập dữ liệu
# 1. Nhập từ bàn phím
# Hàm để nhập dữ liệu từ bàn phím là: input()
# 2. Xuất dữ liệu dùng hàm print()
print(f"|{'MSSV':<10}|{'Tên':<20}|{'Email':<25}|{'Số điện thoại':<15}|{'Lớp':<10}|")
# 3. Comment trong python
# Dùng dấu # để comment 1 dòng
```

```
MSSV      Tên          Email          Số điện thoại  Lớp
```

Quy tắc cơ bản

✓ 1. Chỉ được chứa:

```
name = "John Doe"
```

✓ 2. Không được bắt đầu bằng số

```
lname = "John Doe" # Sai: biến không được bắt đầu bằng số
my-name = "John Doe" # Sai: dấu gạch ngang không được phép trong tên biến
```

```
Cell In[47], line 2
    my-name = "John Doe" # Sai: dấu gạch ngang không được phép trong tên biến
    ^
SyntaxError: cannot assign to expression here. Maybe you meant '==' instead of '='?
```

✓ 3. Phân biệt chữ hoa, chữ thường

```
age = 20
Age = 30
_age = 24

print(age, Age) # 20 30
```

```
20 30
```

✓ 4. Không trùng với từ khóa (keywords)

```
class = "10A1" # Sai: 'class' là từ khóa trong Python

Cell In[40], line 1
    class = "10A1" # Sai: 'class' là từ khóa trong Python
      ^
SyntaxError: invalid syntax
```

✓ Dùng snake_case cho biến và hàm:

```
student_name = "Huy"
ten_sinhvien = "An"
TuoiSinhvien = 21
StudentName = "Trần"
total_score = 95
Totalscore = 85
print(student_name)
print(total_score)
print(StudentName)
print(Totalscore)
```

```
Huy
95
Trần
85
```

✓ Dùng UPPER_CASE cho hằng số:

```
PI = 3.14159
MAX_STUDENTS = 100
```

✓ Dùng PascalCase cho tên lớp:

```
class StudentRecord:
    pass
StudentName
student_name
PI = 3.14
```

✓ Tên biến nên ngắn gọn nhưng gợi ý nghĩa:

```
a = 20 # không rõ nghĩa
student_age = 20 # rõ nghĩa
```

✓ Dưới đây là các phép toán số học cơ bản và nâng cao trong Python

✓ 1. Phép toán cơ bản

```
# Phép cộng
a = 10 + 5 # = 15
print(a)
# Phép trừ
b = 10 - 3 # = 7
print(b)
# Phép nhân
c = 4 * 6 # = 24
print(c)
# Phép chia
d = 15 / 4 # = 3.75 (chia thật)
print(d)
# Phép chia lấy phần nguyên
```

```
e = 15 // 4 # = 3
print(e)
# Phép chia lấy dư
f = 15 % 4 # = 3
print(f)
# Phép lũy thừa
g = 2 ** 3 # = 8 (2^3)
print(g)
```

✓ 2. Toán tử gán kết hợp

```
x = 10

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

✓ 3. Thư viện math

```
import math

# Hàm căn bậc hai
math.sqrt(16) # = 4.0

# Hàm lũy thừa
math.pow(2, 3) # = 8.0

# Hàm trị tuyệt đối
abs(-5) # = 5
math.fabs(-3.5) # = 3.5

# Làm tròn
round(3.7) # = 4
round(3.14159, 2) # = 3.14

# Hàm sàn và trần
math.floor(3.8) # = 3
math.ceil(3.2) # = 4

# Hàm logarit
math.log(10) # ln(10)
math.log10(100) # log10(100) = 2
math.log(8, 2) # log2(8) = 3

# Hàm mũ
math.exp(1) # e^1 = 2.718...

# Hằng số toán học
math.pi # 3.14159...
math.e # 2.718...
```

✓ 4. Hàm lượng giác

```
import math

# Chuyển đổi độ - radian
degrees = 45
radians = math.radians(degrees) # = π/4
degrees_back = math.degrees(radians) # = 45
```

```
# Hàm lượng giác cơ bản
math.sin(math.pi/2)    # = 1.0
math.cos(0)            # = 1.0
math.tan(math.pi/4)    # = 1.0

# Hàm lượng giác ngược
math.asin(1)           # =  $\pi/2$ 
math.acos(0)           # =  $\pi/2$ 
math.atan(1)           # =  $\pi/4$ 
math.atan2(1, 1)       # =  $\pi/4$ 

# Hàm hyperbolic
math.sinh(0)           # = 0.0
math.cosh(0)           # = 1.0
math.tanh(0)           # = 0.0
```

✓ 5. Số phức (Complex Numbers)

```
# Tạo số phức
z1 = 3 + 4j
z2 = complex(2, -1) # 2 - 1j

# Phép toán với số phức
z3 = z1 + z2         # (5+3j)
z4 = z1 * z2         # (10-5j)

# Thuộc tính số phức
z1.real              # 3.0 (phần thực)
z1.imag              # 4.0 (phần ảo)
abs(z1)              # 5.0 (môđun)
z1.conjugate()       # (3-4j) (liên hợp)
```

(3-4j)

✓ 6. Hàm min, max, sum

```
numbers = [1, 5, 3, 9, 2]

min(numbers)         # = 1
max(numbers)         # = 9
sum(numbers)         # = 20

# Với nhiều tham số
min(3, 7, 1, 9)      # = 1
max(3, 7, 1, 9)      # = 9
```

9

✓ 7. Hàm divmod

```
# Trả về cả thương và dư
quotient, remainder = divmod(17, 5)
print(f"17 ÷ 5 = {quotient} dư {remainder}") # 17 ÷ 5 = 3 dư 2
```

17 ÷ 5 = 3 dư 2

✓ 8. Thư viện random cho số ngẫu nhiên

```
import random

# Số ngẫu nhiên từ 0 đến 1
random.random()      # 0.123456...
print(random.random())

# Số nguyên ngẫu nhiên
random.randint(1, 10) # Từ 1 đến 10
print(random.randint(1, 10))
```

```
# Số thực ngẫu nhiên trong khoảng
random.uniform(1.5, 10.5) # Từ 1.5 đến 10.5
print(random.uniform(1.5, 10.5))
# Chọn ngẫu nhiên từ danh sách
colors = ['đỏ', 'xanh', 'vàng']
random.choice(colors) # 'đỏ' hoặc 'xanh' hoặc 'vàng'
print(random.choice(colors))
# Xáo trộn danh sách
numbers = [1, 2, 3, 4, 5]
random.shuffle(numbers) # [3, 1, 5, 2, 4]
print(numbers)
```

```
0.37541096255277784
7
7.3817716762277135
xanh
[3, 1, 2, 5, 4]
```

✓ các ký tự đặc biệt (escape characters) trong Python:

✓ 1. Ký tự thoát cơ bản

```
# Xuống dòng
print("Dòng 1\nDòng 2")
# Output:
# Dòng 1
# Dòng 2

# Tab (thụt lề)
print("Tên:\tNguyễn Văn A")
# Output: Tên:      Nguyễn Văn A

# Dấu nhảy đơn trong chuỗi nhảy đơn
print('I\'m learning Python')
print("I'm learning Python")
# Output: Tôi'm learning Python

# Dấu nhảy kép trong chuỗi nhảy kép
print("Anh ấy nói: \"Hello World\"")
print('Anh ấy nói: "Hello World"')
# Output: Anh ấy nói: "Hello World"

# Dấu gạch chéo ngược
print("Đường dẫn: C:\\Users\\Desktop")
# Output: Đường dẫn: C:\Users\Desktop
```

```
Dòng 1
Dòng 2
Tên:      Nguyễn Văn A
I'm learning Python
I'm learning Python
Anh ấy nói: "Hello World"
Anh ấy nói: "Hello World"
Đường dẫn: C:\Users\Desktop
```

✓ Ví dụ nâng cao

```
# Xuống dòng
text = "Dòng 1\nDòng 2\nDòng 3"
print(text)
# Output:
# Dòng 1
# Dòng 2
# Dòng 3

# Tab
menu = "Món ăn:\tGiá tiền\nPhở:\t50,000đ\nCơm:\t30,000đ"
print(menu)
# Output:
# Món ăn:      Giá tiền
# Phở:         50,000đ
```

```
# Com:          30,000đ

# Backspace
print("Hello\bWorld") # HelloWorld (xóa ký tự 'o')

# Carriage return (về đầu dòng)
print("Hello\rWorld") # World (ghi đè từ đầu dòng)

# Kết hợp nhiều ký tự thoát
address = "Địa chỉ:\n\tSố nhà: 123\n\tĐường: \"Nguyễn Huệ\"\n\tThành phố: Hồ Chí Minh"
print(address)
# Output:
# Địa chỉ:
#     Số nhà: 123
#     Đường: "Nguyễn Huệ"
#     Thành phố: Hồ Chí Minh
```

```
Dòng 1
Dòng 2
Dòng 3
Món ăn: Giá tiền
Phở:    50,000đ
Cơm:    30,000đ
HelloWorld
World
Địa chỉ:
    Số nhà: 123
    Đường: "Nguyễn Huệ"
    Thành phố: Hồ Chí Minh
```

✓ Raw strings (chuỗi thô)

```
# Sử dụng r"" để bỏ qua escape characters
normal_string = "C:\new\folder\test.txt"
print(normal_string) # C:
                        # ew\folder  est.txt (lỗi do \n, \t)

raw_string = r"C:\new\folder\test.txt"
print(raw_string)    # C:\new\folder\test.txt (đúng)

# Raw string hữu ích cho regex
import re
pattern = r"\d+\.\d+" # Tìm số thập phân
text = "Giá: 15.99 và 20.5"
matches = re.findall(pattern, text)
print(matches) # ['15.99', '20.5']
```

```
C:
ewolder est.txt
C:\new\folder\test.txt
['15.99', '20.5']
```

✓ trong python hàm end mặc định xuống dòng end="\n"

```
print("Hello")
print("World")
print("Hello", end=" ")
print("World")
```

```
Hello
World
Hello World
```

```
# In liên tiếp trên 1 dòng, cách nhau bởi dấu phẩy
print("A", end=", ")
print("B", end=", ")
print("C")

# In không có khoảng cách
print("Python", end="")
print("3.10")

# In với dấu gạch ngang
```



```
for i in range(5):  
    print(i, end="-")  
print() # để xuống dòng sau vòng lặp
```

A, B, C
Python3.10
0-1-2-3-4-