

Giới Thiệu Về Python

1. Lịch Sử Hình Thành Và Phát Triển Của Python

Python được Guido van Rossum giới thiệu lần đầu vào năm 1991. Ban đầu, Python được thiết kế để dễ học và thân thiện với người dùng, tập trung vào tính dễ đọc. Qua thời gian, Python đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất, được sử dụng trong nhiều lĩnh vực như khoa học dữ liệu, trí tuệ nhân tạo, và phát triển web.

2. Các Kiểu Dữ Liệu Trong Python

Python cung cấp nhiều kiểu dữ liệu khác nhau bao gồm:

- **Kiểu số:** `int`, `float`, `complex`

Ví dụ:

```
x = 10      # int
y = 3.14    # float
z = 1 + 2j   # complex
print(type(x), type(y), type(z))
```

- **Kiểu chuỗi:** `str`

Ví dụ:

```
name = "Python"
print(name, type(name))
```

- **Kiểu danh sách:** `list`

Ví dụ:

```
fruits = ["apple", "banana", "cherry"]
print(fruits, type(fruits))
```

- **Kiểu tuple:** `tuple`

Ví dụ:

```
coordinates = (10, 20, 30)
print(coordinates, type(coordinates))
```

- **Kiểu tập hợp:** `set`, `frozenset`

Ví dụ:

```
unique_numbers = {1, 2, 3, 4}
print(unique_numbers, type(unique_numbers))
```

- **Kiểu từ điển:** dict

Ví dụ:

```
student = {"name": "John", "age": 20}
print(student, type(student))
```

- **Kiểu logic:** bool

Ví dụ:

```
is_active = True
print(is_active, type(is_active))
```

- **Kiểu rỗng:** NoneType

Ví dụ:

```
data = None
print(data, type(data))
```

3. Cách Khai Báo Biến

Trong Python, biến được khai báo bằng cách gán giá trị trực tiếp:

```
x = 10 # Biến nguyên
name = "Python" # Biến chuỗi
is_active = True # Biến logic
```

4. Các Phép Tính Và Toán Tử Cơ Bản

Python cung cấp nhiều toán tử cơ bản:

- Phép cộng: +

Ví dụ:

```
a = 5
b = 3
print(a + b) # Kết quả: 8
```

- Phép trừ: -

Ví dụ:

```
a = 5
b = 3
print(a - b) # Kết quả: 2
```

- Phép nhân: *

Ví dụ:

```
a = 5
b = 3
print(a * b) # Kết quả: 15
```

- Phép chia: /

Ví dụ:

```
a = 5
b = 2
print(a / b) # Kết quả: 2.5
```

- Phép chia nguyên: //

Ví dụ:

```
a = 5
b = 2
print(a // b) # Kết quả: 2
```

- Phép chia lấy dư: %

Ví dụ:

```
a = 5
b = 2
print(a % b) # Kết quả: 1
```

- Phép lũy thừa: **

Ví dụ:

```
a = 2
b = 3
print(a ** b) # Kết quả: 8
```

5. Các Câu Điều Kiện Và Vòng Lặp

Câu Điều Kiện

Sử dụng `if`, `elif`, `else` :

```
x = 10
if x > 5:
    print("x lớn hơn 5")
elif x == 5:
    print("x bằng 5")
else:
    print("x nhỏ hơn 5")
```

Vòng Lặp

- **Vòng lặp** `for` :

```
for i in range(5):
    print(i)
```

- **Vòng lặp** `while` :

```
x = 0
while x < 5:
    print(x)
    x += 1
```

6. Ví Dụ: Giải Phương Trình Bậc Hai

Phương trình bậc hai có dạng:

$$ax^2 + bx + c = 0$$

Code Python:

```
import math

def solve_quadratic(a, b, c):
    # Tính delta
    delta = b**2 - 4*a*c
    if delta < 0:
        return "Phương trình vô nghiệm"
    elif delta == 0:
        x = -b / (2*a)
        return f"Phương trình có nghiệm kép: x = {x}"
    else:
        x1 = (-b + math.sqrt(delta)) / (2*a)
        x2 = (-b - math.sqrt(delta)) / (2*a)
        return f"Phương trình có hai nghiệm: x1 = {x1}, x2 = {x2}"
```

```
# Ví dụ
```

```
print(solve_quadratic(1, -3, 2))
```

Khi chạy code trên, kết quả sẽ là:

```
Phương trình có hai nghiệm: x1 = 2.0, x2 = 1.0
```