

CHƯƠNG 3 – CONDITIONAL EXECUTION (Điều kiện)



1. Biểu thức điều kiện

Python sử dụng biểu thức logic để quyết định thực thi:

- `<`, `>`, `<=`, `>=`, `==`, `!=`
- Toán tử logic: `and`, `or`, `not`

2. Cấu trúc if

```
if x > 10:  
    print("Large")
```

3. Cấu trúc if-else

```
if x % 2 == 0:  
    print("Even")  
else:  
    print("Odd")
```

4. Cấu trúc if-elif-else

```
if score >= 0.9:  
    grade = 'A'  
elif score >= 0.8:  
    grade = 'B'  
elif score >= 0.7:  
    grade = 'C'  
elif score >= 0.6:  
    grade = 'D'  
else:  
    grade = 'F'
```

5. Lệnh try-except (bắt lỗi)

Dùng để xử lý input không hợp lệ:

```
try:  
    value = int(input("Enter  
number: "))  
except:  
    print("Invalid input")
```

6. Viết chương trình có thêm điều kiện nâng cao

Ví dụ: tính lương có tăng ca (overtime)

```
hours = float(input("Hours: "))  
rate = float(input("Rate: "))  
  
if hours > 40:  
    pay = 40 * rate + (hours - 40) *  
        rate * 1.5  
else:  
    pay = hours * rate  
print(pay)
```

Chương 4: Function

1. Định nghĩa hàm

```
def greet():  
    print("Hello")
```

Gọi hàm:

```
greet()
```

2. Hàm có tham số

```
def greet(name):  
    print("Hello", name)
```

```
greet("Ninh")
```

7. Ví dụ: Hàm tính điểm

```
def computegrade(score):  
    if score >= 0.9:  
        return 'A'  
    elif score >= 0.8:  
        return 'B'  
    elif score >= 0.7:  
        return 'C'  
    elif score >= 0.6:  
        return 'D'  
    else:  
        return 'F'
```

```
s = float(input("Enter score: "))  
print(computegrade(s))
```

CHƯƠNG 5: CẤU TRÚC VÒNG LẶP

While Loop

```
n = 5
while n > 0:
    print(n)
    n = n - 1
print("Blastoff!")
```

break

thoát khỏi vòng lặp

```
while True:
    line = input("> ")
    if line == "done":
        break
    print(line)
print("Done!")
```

For Loop

Vòng lặp chạy số lần xác định

```
for i in [5,4,3,2,1]:
    print(i)
print("Blastoff!")
```

```
for thing in [9, 41, 12, 3, 74, 15]:
    print(thing)
```

In từng phần tử

Tìm giá trị lớn nhất

```
largest = -1
for num in [9,41,12,3,74,15]:
    if num > largest:
        largest = num
print(largest)
```

```
count = 0
for x in [9,41,12]:
    count = count + 1
print(count)
```

Đếm số phần tử

Tính tổng

```
total = 0
for x in [9,41,12]:
    total = total + x
print(total)
```

```
count = 0
total = 0
for x in [9,41,12,3,74,15]:
    count += 1
    total += x
print(total/count)
```

Tính trung bình

Lọc giá trị

```
for value in [9,41,12,3,74,15]:
    if value > 20:
        print("Large number:", value)
```

```
found = False
for value in [9,41,12,3,74,15]:
    if value == 3:
        found = True
print(found)
```

Tìm giá trị trong danh sách

Tìm giá trị nhỏ nhất

```
smallest = None
for value in [9,41,12,3,74,15]:
    if smallest is None or value < smallest:
        smallest = value
print(smallest)
```

CHƯƠNG 6: STRING

Kiểu dữ liệu chuỗi (String) <pre>str1 = "Hello " str2 = "there" print(str1 + str2) # Hellothere x = int("123") + 1 # 124</pre>	Nhập dữ liệu và chuyển kiểu <pre>name = input("Enter: ") apple = input("Enter: 100") x = int(apple) - 10</pre>	Truy cập ký tự – Indexing Dùng [] để lấy ký tự. Chỉ số bắt đầu từ 0. <pre>fruit = "banana" print(fruit[1]) # a print(fruit[3-1]) # n</pre>
Độ dài chuỗi – len() <pre>fruit = "banana" print(len(fruit)) # 6</pre>	Lặp qua chuỗi <div><div>✓ While loop <pre>fruit = "banana" i = 0 while i < len(fruit): print(i, fruit[i]) i += 1</pre></div><div>✓ For loop <pre>for letter in "banana": print(letter)</pre></div></div>	Đếm ký tự trong chuỗi <pre>word = "banana" count = 0 for letter in word: if letter == 'a': count += 1 print(count)</pre>
Slicing – Cắt chuỗi s[start:end] (kết thúc tại end-1) <pre>s = "Monty Python" s[0:4] # Mont s[6:20] # Python s[:2] # Mo s[8:] # thon s[:] # Monty Python</pre>	Nối chuỗi – Concatenation <pre>a = "Hello" b = a + " There"</pre>	Toán tử in – kiểm tra chuỗi con <pre>'n' in 'banana' # True 'm' in 'banana' # False if 'a' in 'banana': print("Found it!")</pre>
So sánh chuỗi <pre>if word < "banana": print("comes before banana") elif word > "banana": print("comes after banana") else: print("equal")</pre>	Chữ thường và Chữ hoa <pre>greet = "Hello Bob" greet.lower() # hello bob greet.upper() # HELLO BOB</pre>	Tìm trong chuỗi – find() <pre>fruit = "banana" fruit.find("na") # 2 fruit.find("z") # -1</pre>
Thay thế chuỗi – replace() <pre>s = "Hello Bob" s.replace("Bob", "Jane") # Hello Jane</pre>	Xóa khoảng trắng – strip() <pre>line = " Hello Bob " line.strip() # Hello Bob</pre>	Tách và trích xuất chuỗi <pre>data = "From stephen@uct.ac.za Sat 5" at = data.find("@") sp = data.find(" ", at) host = data[at+1 : sp] print(host) # uct.ac.za</pre>

CHƯƠNG 2: VARIABLES, EXPRESSIONS, AND STATEMENTS (BIẾN, BIỂU THỨC VÀ CÂU LỆNH)

1. BIẾN (VARIABLES)

```
x = 10
Name = "THIEN"
```

Dùng = để **gán giá trị** cho biến. - Biến không cần khai báo kiểu trước

2. KIỂU DỮ LIỆU (TYPES)

```
type ( 10 )      # int
type ( 3.14 )    # float
type ( hi )      # string
```

4. TOÁN TỬ TOÁN HỌC

+	Cộng
-	Trừ
*	Nhân
/	Chia
//	Chia lấy nguyên
%	Chia lấy dư
**	Lũy Thừa

5**2	#25
5//2	#2
5%2	#1

3. INPUT & CHUYỂN KIỂU

```
x=input("nhập số: ") # luôn trả về chuỗi
x = int(x)             # chuyển sang số nguyên
x = float(x)           # chuyển sang số thực
```

5. BIỂU THỨC (EXPRESSIONS)

```
x = 5
y = x* 2 + 3
```

MỤC TIÊU

- Lưu trữ dữ liệu trong BIẾN (Variable)
- Thực hiện tính toán với biểu thức (expressions)
- viết câu lệnh (statements) để python thực thi
- Nhập dữ liệu từ người dùng
- viết chú thích (comments) và đặt tên biến hợp lý

LƯU Ý

- Gán giá trị = không phải là ‘bằng nhau’ mà là ‘đặt giá trị cho BIẾN’
- **input ()** luôn trả về chuỗi **str** - muốn tính toán phải đổi sang **int ()** hoặc **float()**

6. CÂU LỆNH (STATEMENTS)

```
print("Hello")
x = 10
```

CHƯƠNG 7 – FILES (LÀM VIỆC VỚI TỆP)

1. MỞ FILE

```
fhand = open('data.txt', 'r')
```

mở file ở chế độ đọc

- 'r' = read (đọc) - 'w' = write (ghi – xóa nội dung cũ)

2. ĐỌC FILE DÒNG-THEO-DÒNG

```
fhand = open('data.txt')  
for line in fhand:  
    print(line)
```

mỗi dòng có dấu xuống hàng

3. XÓA KHOẢNG TRẮNG CUỐI DÒNG

```
line = line.rstrip()
```

bỏ \n

4. ĐẾM SỐ DÒNG

```
count = 0  
for line in fhand:  
    count += 1  
print('Số dòng: ', count)
```

5. ĐỌC TOÀN BỘ FILE

```
text = fhand.read()  
print(len(text))
```

số ký tự

6. TÌM DÒNG BẮT ĐẦU BẰNG TỪ KHÓA

```
for line in fhand:  
    line = line.rstrip()  
    if line.startswith('From:'): print(line)
```

7. XỬ LÝ LỖI KHI MỞ FILE

```
fname = input('Enter file: ')  
try:  
    fhand = open(fname)  
except:  
    print('File không tồn tại!')  
    quit()
```

8. TÁCH DỮ LIỆU TRONG DÒNG

```
for line in fhand:  
    line = line.rstrip()  
    words = line.split()
```

tách theo khoảng trắng

9. VÍ DỤ TRÍCH EMAIL DOMAIN

```
line = 'From user@gmail.com Sat Jan 5'  
words = line.split()  
email = words[1]  
parts = email.split('@')  
domain = parts[1]  
print(domain)
```

gmail.com

MỤC TIÊU

- Biết mở file bằng `open()`.
- Biết đọc file dòng-theo-dòng bằng `for line in file:`.
- Dùng `.rstrip()` để xóa `\n`.
- Dùng `.read()` để đọc toàn bộ file.
- Dùng `.startswith()` và `.split()` để lọc và tách dữ liệu.
- Dùng `try/except` để xử lý file không tồn tại.

LƯU Ý

- `open()` mở file → phải có tên file đúng.
- Mỗi dòng đọc ra có ký tự xuống dòng `\n`.
- `.rstrip()` để bỏ khoảng trắng cuối dòng.
- `.split()` tách dòng thành danh sách từ.
- Luôn dùng `try/except` khi người dùng nhập tên file.
- File lớn → tránh dùng `.read()` vì chiếm nhiều RAM.