

PYTHON

Các quy tắc của Python

- Thụt lề đầu dòng: thụt lề đúng 1 khoảng trắng cố định ở tất cả các khối mã, nếu thụt k đều hoặc k có thì sẽ báo error (mã không được thực thi)

Ex:

```
if 10 > 2:  
    print("True")
```

```
if 5 > 2:  
    print("Five is greater than two!")  
    print("Five is greater than two!")
```

- Comment:
 - Comment từng dòng: #text
 - Comment nhiều dòng: """ texts """ hoặc ctrl + /
- Toán tử số học: +, -, +=, *=, /, %(lấy dư), ** (số mũ), //(chia làm tròn)
- Toán tử khai thác: in, not in
- Toán tử xác thức: is, not is (so sánh 2 số có = -> boolean)
- Toán tử logic: and or not

1. Data Types

- Chuỗi (văn bản): str
- Số: int, float, complex...
- boolean
- list, range...

2. Biến

Không cần khai báo kiểu trước khi khai báo biến, viết luôn giá trị, VD: x = 5, x = "str"...

Các quy tắc đặt tên biến same same code C: không bắt đầu bằng số, không có khoảng trống

- Hàm print() dùng để xuất biến:

```
x = 'Peter'
y = "koala"
print(x, y)
#Peter koala (tự có dấu cách giữa)
```

- Biến toàn cục: sử dụng cả trong và ngoài hàm
- Biến cục bộ chỉ sử dụng ở trong hàm def, ngoài hàm đó coi như out

3. String

- Dạng: 'str' or "str"
 - len(str): độ dài
 - kiểm tra 1/nhiều từ có/không có trong chuỗi không, kết quả trả về true/false
#ex: print("pandora" in/not in str)
 - Chuỗi có dạng mảng, xuất dữ liệu được: a[0], a[1]

```
str = "Hello World"
str[2:5] = "llo" lấy ký tự [2,5)
str[:n]: lấy từ đầu đến n : [0,n)
str[n:]: lấy từ n đến cuối
str[-4:-1] = orl (vị trí cuối là -1)
```

- In hoa: upper()

```
a = "Hello, World!"
print(a.upper())
#HELLO, WORLD!
```

- In thường: lower()

```
a = "Hello, World!"
print(a.lower())
#hello, world!
```

- Xóa space đầu & cuối: strip()

```
a = " Hello, world! "  
print(a.strip())  
# "Hello, World!"
```

- Thay thế từ/cụm từ: `replace()`
- `split()`: chuyển string -> list,
- Hàm `join()`: list -> string
- Ghép chuỗi: chỉ đúng khi cả 2 đều là dạng str, nếu k thuộc dạng str thì ép kiểu

```
a = "Hi"  
b = "mary"  
a+b = "Himary"
```

4. Boolean

- Trả về giá trị True or False
- False: (), {}, [], "", None, 0

Mọi strings đều True trừ chuỗi rỗng

Mọi số đều True trừ số 0

5. List

```
a = [4, 5.01, "Ba", "2i"]
```

- List tạo bởi [] lưu trữ nhiều loại type khác nhau
- Các hàm cơ bản same same strings

`len()`: số mục trong list

Truy cập list dạng array: `list[0]`, `list[2:5]`, `list[-5:-1]`

Thay đổi giá trị : `list[1] = "back"`

Sắp xếp, Chèn, xóa, thêm vào cuối DS

6. Loop : while / for

example

```
for x in range(100):  
    print(x)  
while x < 100:  
    print("Hello")
```

- Các câu lệnh Break, Continue same C

7. if else: bản chất giống trong C, khác cách viết

example

```
for x in list:  
    if x%2 == 0:  
        print("true")  
    else:  
        print("false")  
if a is not b:  
    a+=b  
elif a is b:  
    a-=b
```

8. Function : def name(...)

```
def func(x):  
    x*= 2  
    return x
```