



# Control Statements

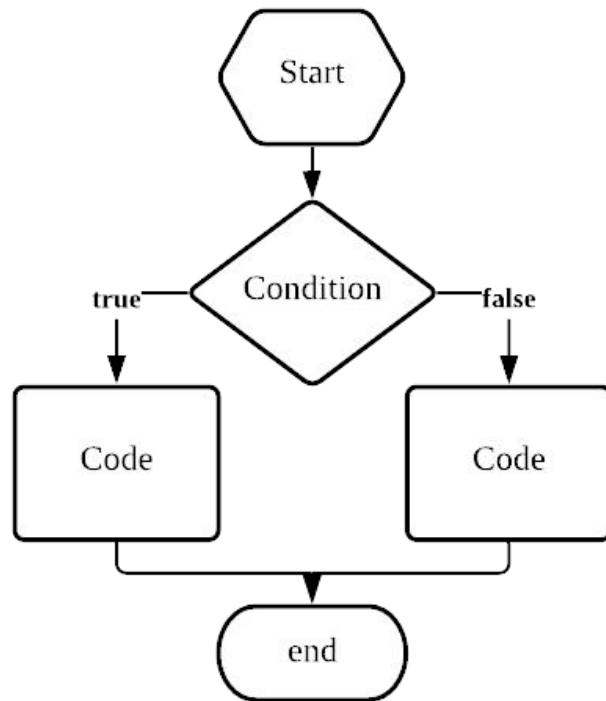
Ba Nguyễn

# Conditional Statements

Thực thi các câu lệnh dựa theo điều kiện

```
age = 28
if age ≥ 20:
    print("Adult")
elif age ≥ 13:
    print("Teenager")
else:
    print("Child")

print("Done")
```



# Code block, pass

💡 Python sử dụng indent (phần thụt lề) để xác định một khối code

💡 Sử dụng từ khóa pass để xác định một khối code rỗng

```
if age > 20:    ❌ error
else:
    # do some thing
```

```
if age > 20:
    pass        🔥 okey
else:
    pass
```

# Logical Operators

---

**not** - Phủ định kết quả của biểu thức

```
name = ""  
if not name:  
    print("Name is empty")
```

**or** - Kết hợp điều kiện, biểu thức đúng khi một trong các điều kiện đúng

```
handsome = False  
rich = True  
if handsome or rich:  
    print("I love you")
```

# Logical Operators

**and** - Kết hợp điều kiện, biểu thức đúng khi tất cả điều kiện đều đúng

```
age = 25
```

```
handsome = True
```

```
rich = True
```

```
if 18 ≤ age and age ≤ 25 and handsome and rich:  
    print("Perfect")
```



Python hỗ trợ chuỗi toán tử so sánh, cho phép viết biểu thức logic rõ ràng hơn

```
if 18 ≤ age ≤ 25 and handsome and rich:  
    print("Perfect")
```

# Ternary Operator

Ternary Operator (toán tử 3 ngôi) là cú pháp rút gọn của **if else**, thường được sử dụng để gán (hoặc trả về) một giá trị theo điều kiện

```
age = 25
```

```
# if age ≥ 18:  
#     message = "Eligible"  
# else:  
#     message = "Not eligible"
```

```
message = "Eligible" if age ≥ 18 else "Not eligible"
```

```
print(message)
```

# For loops

Lặp qua bất kỳ đối tượng nào có thể lặp (iterable), thực hiện các hành động (khối code) lặp lại với mỗi phần của đối tượng

```
for char in "Mr Ba":  
    print(char)
```

```
# M  
# r  
#  
# B  
# a
```

```
for i in ['B', 'a']:  
    print(i)
```

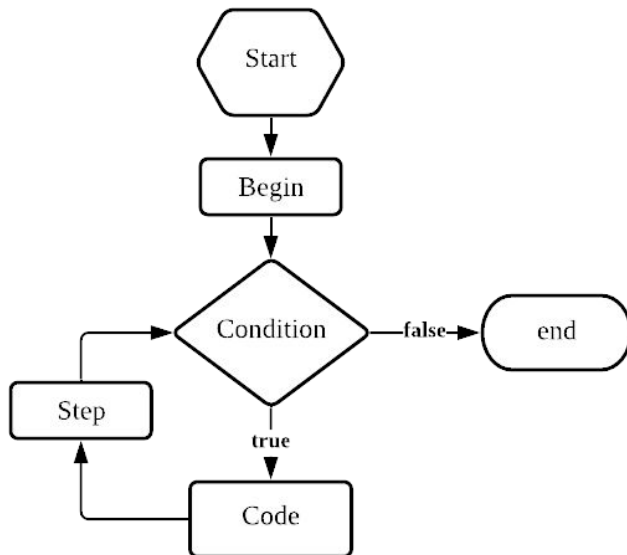
```
# B  
# a
```

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

```
# 0  
# 1  
# 2  
# 3  
# 4
```

# For loops

Lặp qua bất kỳ đối tượng nào có thể lặp (iterable), thực hiện các hành động (khối code) lặp lại với mỗi phần của đối tượng





# range()

---

💡 `range()` tạo ra một chuỗi (sequence) các số nguyên

💡 `range()` trả về một *range object*, là một đối tượng có thể lặp (iterable) giống như string, list, ...

💡 `range` không giống như list, nó chiếm rất ít bộ nhớ

- `range(stop)`
- `range(start, stop [,step])`

*# In ra các số lẻ trong khoảng 1 → 10*

```
for n in range(1, 10, 2):  
    print(n);
```

# break, continue

---

**continue** - dừng lần lặp hiện tại, chuyển tới lần lặp tiếp theo

```
for n in range(10):  
    if n % 2 == 0:  
        continue  
    else:  
        print(n)
```

**break** - dừng vòng lặp

```
for n in range(10):  
    if n > 5:  
        break  
    else:  
        print(n)
```

# For else

---

Vòng lặp **for** cũng hỗ trợ mệnh đề **else**, nếu thực hiện hết vòng lặp mà không gặp câu lệnh **break**, mệnh đề **else** sẽ được thực thi, ngược lại, nếu gặp câu lệnh **break** ngắt vòng lặp, thì **else** không sẽ không được thực thi

```
for name in ["Ba", "Béo", "Ú"]:  
    if name == "Ba":  
        print("Found")  
        break  
else:  
    print("Not found")
```

# While loops

Tương tự như **for**, tuy nhiên vòng lặp **while** không xác định trước số lần lặp

```
guess = 0
answer = 5
while guess  $\neq$  answer:
    guess = int(input("Guess a number\n> "))

    if guess == answer:
        print("Congratulations")
    else:
        print("Wrong, try again!")
```

 **while** cũng hỗ trợ mệnh đề **else** tương tự **for**