

Python Control Flow: Branching & Loop Statements

Introduction to Computer Programming (Python)

Week 3

Vivatsathorn Thitasirivit

Rev. 1.0 (Course 1/2023)

<https://vtneil.com>

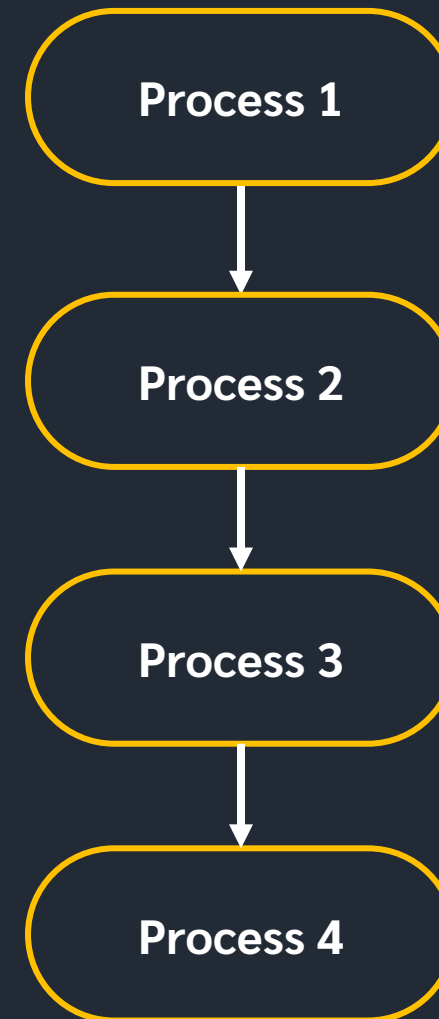
Computer Program Flow: Linear

1. Simple (Linear) Process

A linear process is the simplest form of combi-permutations of instructions.

A then B then C then ...

It has definite number of instructions executed and has definite ending point.



Computer Program Flow: Branching

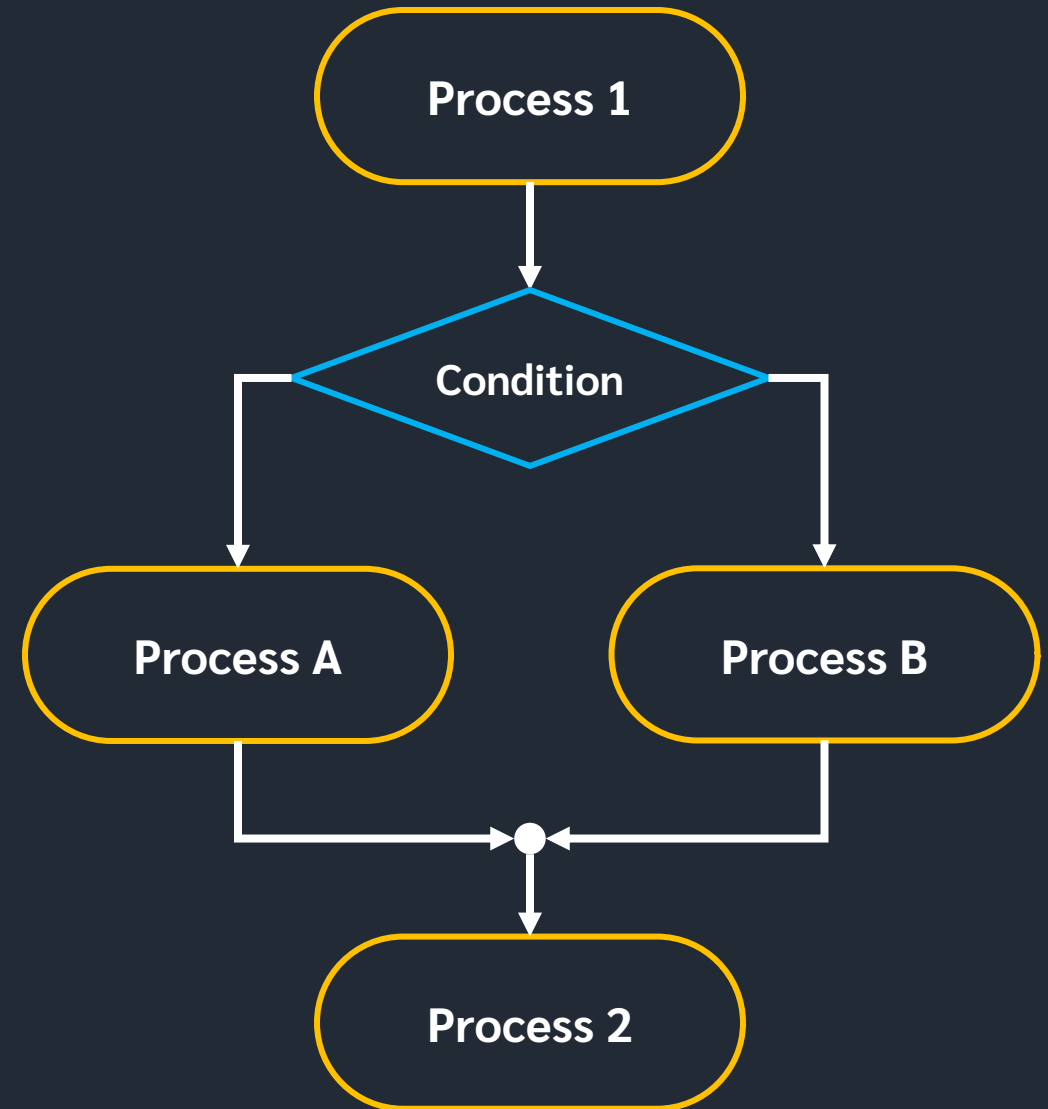
2. Branching

Branching is made by a control in a flow. When the program decides which operation should be done next based on conditions, there will be branches.

Conditional Branching is decided based on conditions, e.g., if-elif-else structure.

There is another type of branching: “Jump Table”, i.e., switch-case structure.

e.g., possible combinations are:
1-A-2 and *1-B-2*.

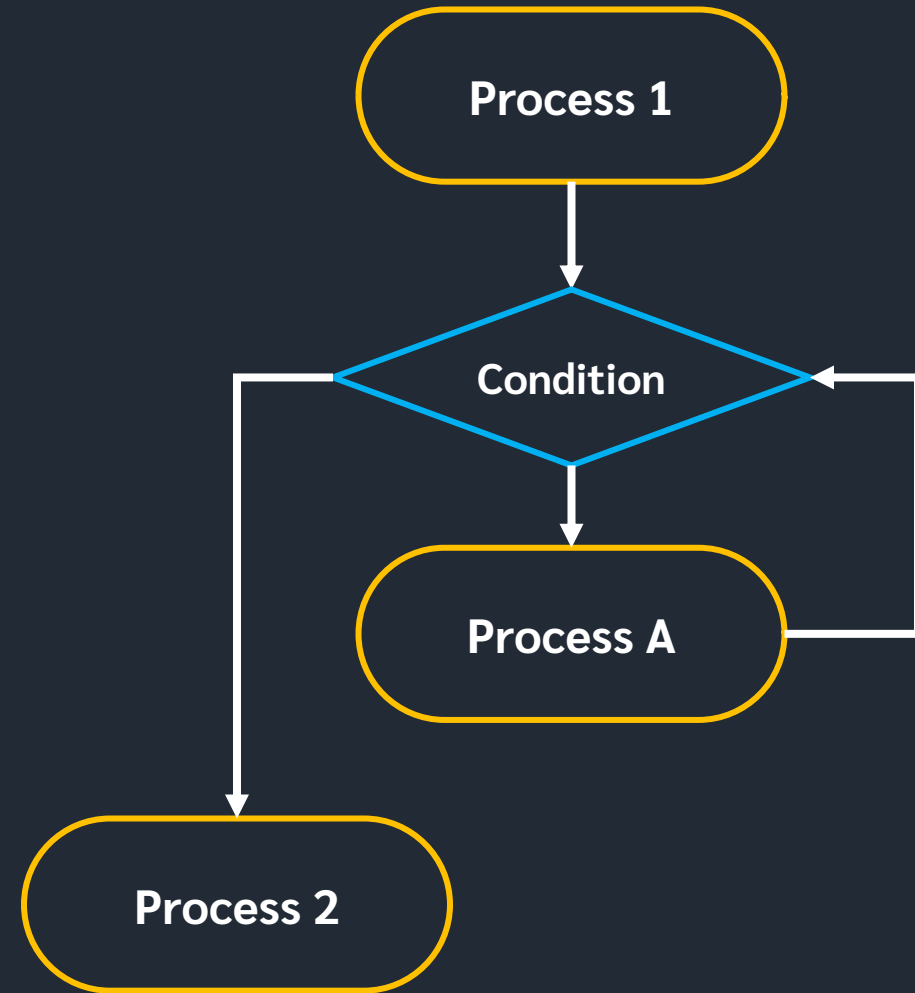


Computer Program Flow: Loop

3. Loop

A loop is an iterative statement where the program does some processes many times which may be the same or different processes.

A loop might be based on enumeration or decision.



Python

Boolean & Boolean Operators

We have learned about arithmetic operators: add, subtract, multiply, exponentiation, etc.

There is another type of operators: boolean. A boolean is a data type holding truth values. The truth value can be either true (1) or false (0)

In Python, literal True and False are used.

Common boolean operators are:

1. **AND** : Binary
2. **OR** : Binary
3. **NOT** : Unary
4. NAND
5. NOR
6. XOR
7. XNAND
8. XNOR

```
A = True
B = False

print(A and B)
print(A or B)
print(not A)
```

** Not to be confused with bitwise operators!*

Logic

Truth Table

A truth table is a relational table with every possible combination of different inputs under Boolean operations.

Example, AND, OR truth table

A	not A	B	A and B	A or B
False	True	False	False	False
False	True	True	False	True
True	False	False	False	True
True	False	True	True	True

Python

Comparison (Relational) Operators

Comparison operators are binary operators used to “compare” the value of 2 objects.

There are 6 comparison operators in Python:

1. Equal to (`a == b`): returns True if a equals b.
2. Not equal to (`a != b`)
3. Strictly less than (`a < b`)
4. Less than or equal to (`a <= b`)
5. Strictly greater than (`a > b`)
6. Greater than or equal to (`a >= b`)

```
a = 69
b = 24

print(a == b)
print(a != b)
print(a < b)
print(a <= b)
print(a > b)
print(a >= b)
```

Python

Conditional Statements

Conditional Statement is a part of control flow, which determines which part of source codes should be run or skipped.

In Python, and similar in many other languages, structure if – else if (elif) – else is used.

Any combination with if is allowed.

Example, if-elif-elif-else
if-else
if-elif
if-elif-elif
...



```
if condition1:
```

```
# do something
```

```
elif condition2:
```

```
# do something
```

```
elif condition3:
```

```
# do something
```

```
else:
```

```
# do something
```


Python

Conditional Statements

Conditional Statement syntax in Python:

```
if condition1:  
    ... # do something  
  
elif condition2:  
    ... # do something  
  
elif condition3:  
    ... # do something  
  
else:  
    ... # do something
```

if condition1:

do something

elif condition2:

do something

elif condition3:

do something

else:

do something

Python

Conditional Statements

Example 1: check if number is greater than some values.

What do you think the output is if:

1. a = 999
2. a = 81
3. a = 80
4. a = 60
5. a = 50
6. a = 40
7. a = 30
8. a = -999

```
a = int(input())

if a >= 80:
    print('Good')
elif a >= 50:
    print('Pass')
else:
    print('Failed')
```

Python

Conditional Statements

Example 2: check equality of 2 numbers

What do you think `__1__` and `__2__` is ?

```
a = int(input())
b = int(input())

__1__ = a == b
__2__ = a < b

if __1__:
    print("a equals b")
elif __2__:
    print("a is less than b")
else:
    print("a is greater than b")
```

Python

Conditional Statements

Example 3: check if a number is even or odd.

What do you think _____ is ?

```
a = int(input())  
  
if _____:  
    print("Odd")  
else:  
    print("Even")
```

Python: Exercise

Conditional Statements

Exercise 1

Write a program that do the following:

Input

1. Height in centimeters

Output

1. Categorized height

Example

Input: 35

Output: Dwarf

Height	Category
0 – 100	Dwarf
101 – 150	Short
151 – 200	Normal
> 200	Giant