Python Control Flow: Loop & Iteration

Introduction to Computer Programming (Python)

Week 4

Vivatsathorn Thitasirivit

Rev. 1.0 (Course 1/2023)

https://vtneil.com

Learning Strategy
Learning and Practicing

You want to learn new things and practice using them to build up muscle memory.

Even you studied a lot, but if you don’t practice a lot, you are just consuming weigh protein without working out.

“Bloating with knowledge but doesn’t know how to use it.”

A5 Wagyu	Low grade meat: Fat and protein separated
Practice	**Practice**
Learn	**Practice**
Practice	**Practice**
Learn	**Learn**
Practice	**Learn**
Learn	**Learn**

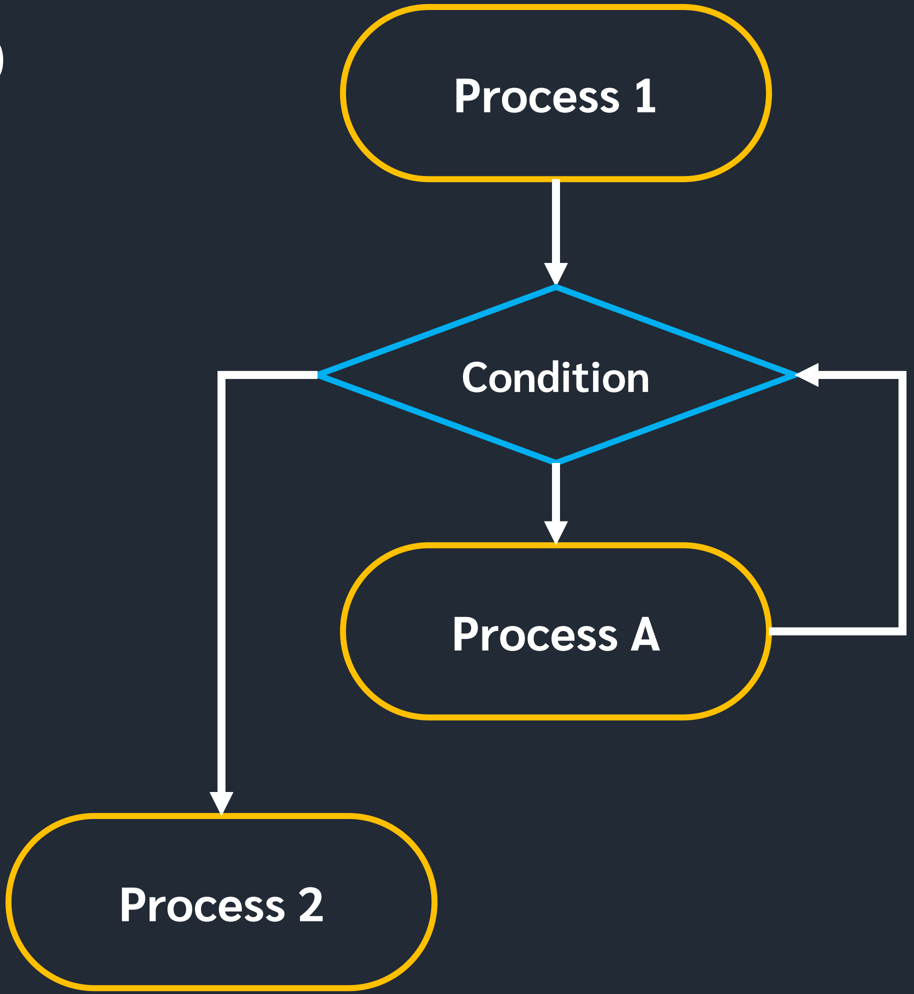
Computer

Review Program Flow: Loop

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

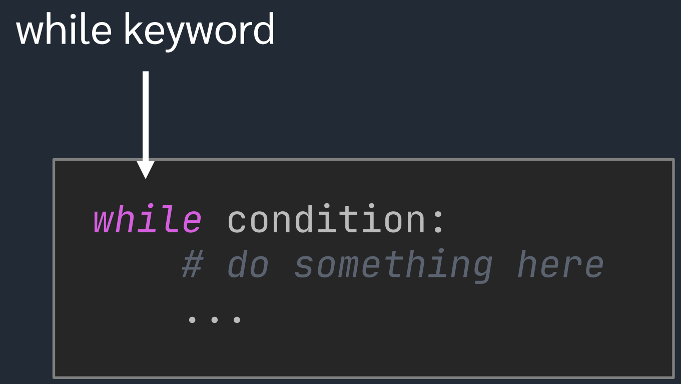
Conditional Loop

A conditional loop is a loop that checks for conditions every cycle of operations.

If the condition is true, then the loop continues until the condition is false.

In Python, and many other languages, a while loop is used in this context.

It is read as “While something, do something”



```python
while condition:
    # do something here
    ...
```

Python

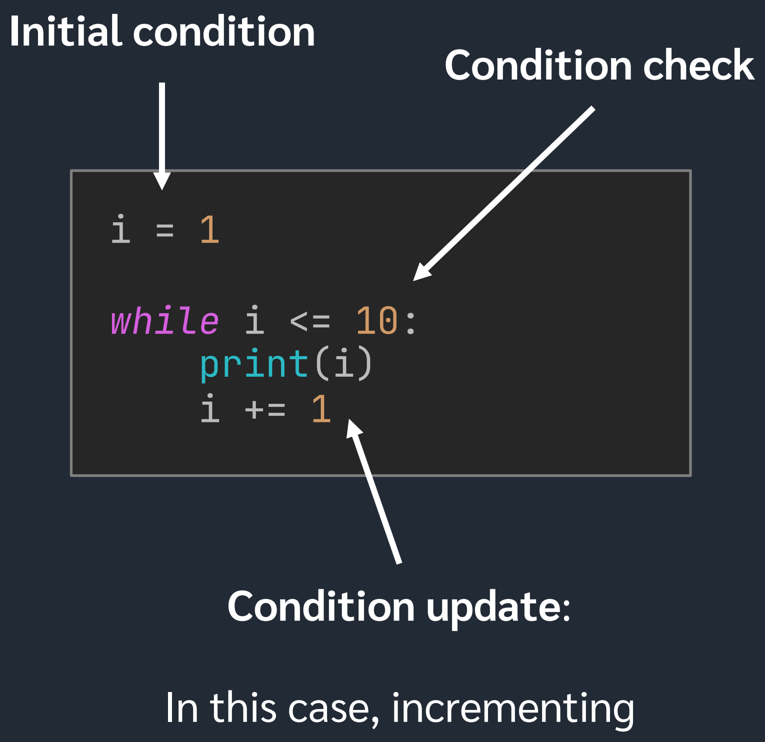
Conditional Loop

Usually, to make the while loop effective, it must not run forever.

To start, you set the initial condition, then check, then update the condition in the loop. If you do it correctly, the loop should not be a forever loop.

For example, a program that up-counts from 1 to 10.

The program should print integer from 1 to 10.



```
i = 1

while i <= 10:
    print(i)
    i += 1
```

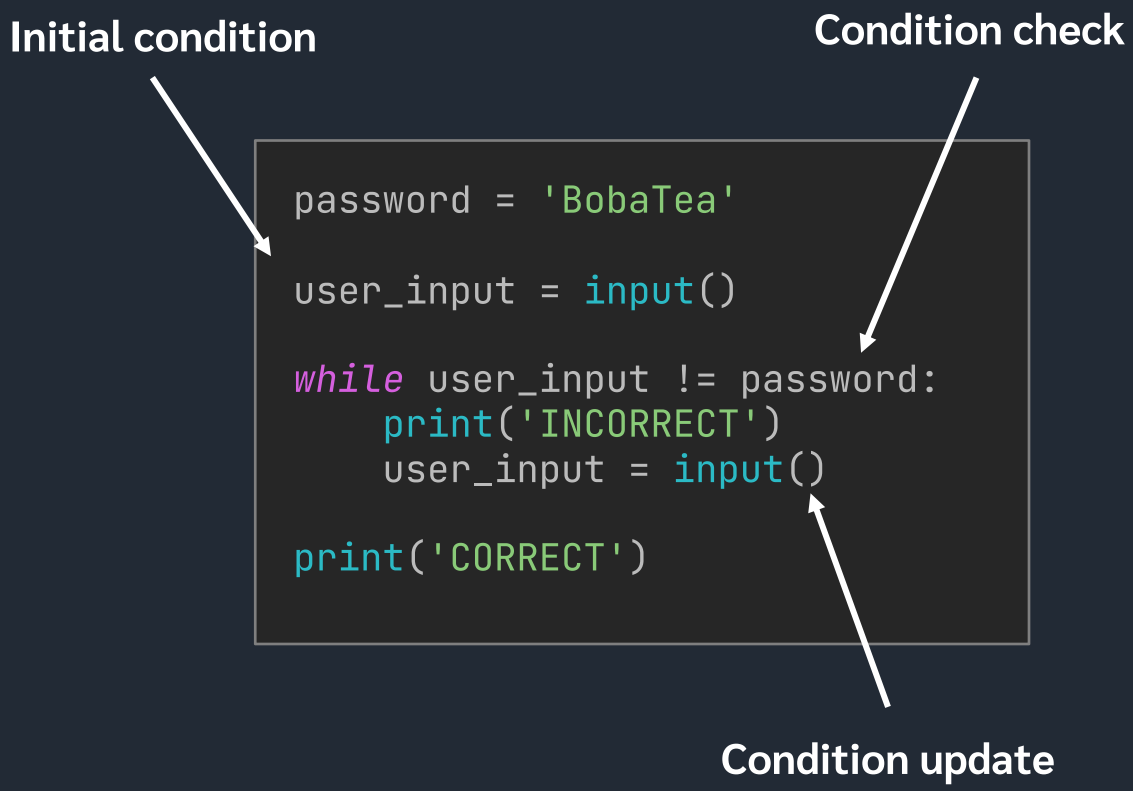
Python

Conditional Loop

Example,

A program that asks user the password. The program should print ”CORRECT” if user inputs the correct password and “INCORRECT” if not.

You can use any statements you have learned in the past chapters, for example, if-elif-else, print, etc...



```python
password = 'BobaTea'

user_input = input()

while user_input != password:
    print('INCORRECT')
    user_input = input()

print('CORRECT')
```

Python

Conditional Loop

Example,

A program that sums integer from 1 to *n* while *n* is an integer from user input.

Solution 1

```python
result = 0
i = 1

n = int(input('Enter an integer: '))

while i <= n:
    result += i
    i += 1

print(result)
```

Solution 2

```python
result = 0

n = int(input('Enter an integer: '))

while n > 0:
    result += n
    n -= 1

print(result)
```

Python

Conditional Loop

Exercise 1

Write a program that asks user for 5 integers.

Then, add those integers to an empty list.

The list should have those 5 integers.

Print the list.

The output should look like this:

```
[5, 80, 3, 2, 4]
```

Python

Conditional Loop

Exercise 2

Write a function that takes a list of numbers as a parameter.

The function should square every element, then return the new list.

Example:

Input	inp = [1, 3, 4, 5]
Output	f(inp) = [1, 9, 16, 25]

Use the given template.
Knowledge: list indexing and list length.

```python
# This asks the user how many numbers should be input
# For example, 5, then the next 5 lines you input the numbers
# Then, the list x should contain those numbers.
x = [int(input()) for i in range(int(input()))]


def f(inp):
    ret = []

    # do something

    return ret


print('First:', x)
print('Then: ', f(x))
```

Python
For Loop

Python’s “for loop” is very versatile. It can be used to iterate through a collection of elements, e.g., a list, set, dictionary, etc.

Sometimes a range(...) function is used instead of original while loop setups.

The “for loop” will iterate through every element in the specified collection.

for keyword

```
for something in collection:
    # do something here
    ...
```

Python
For Loop

For example, you want to print every element in a list beautifully.

One way, you can use string joining to do it, which this method is more Pythonic.

One way, you can loop through each element and print them one by one. This is a more standard method than the above one.

```
lst = ['Word1', 'Pepsi', 'Fish', 'Chess’]


for x in lst:
    # x now represents each element in lst
    print(x)
```

Python
For Loop

Sometimes, you want to iterate through a list of integers (counting): 0, 1, 2, 3, ... , n

In Python, there is an easy way for that.

You can use range(...) generator function.

```
for i in range(10):
    print(i)
```

But if you want to start from 1? Or from any number? The research work is on you!

You can also use range for iterating a list with index, but this practice is NOT recommended in Python because there is a better way to do it.

```
lst = ['Word1', 'Pepsi', 'Fish', 'Chess']

for i in range(len(lst)):
    print(i, lst[i])
```

Python

For Loop

Exercise 3

Write a program that calculate the factorial of *n* using “for loop.”

A program should prompt user for integer *n*.

A program should also check for negative integers. Print “INVALID” if input is negative.

Recall that	0! = 1
	1! = 1
	2! = 2
	3! = 6
	4! = 24
	5! = 120
	…

Python

For Loop

Exercise 4

Write a function that takes a list of strings as a parameter.

The function should return a new list of strings without empty element

Example:

Input	x = ["Joe", "Sarah", "Mike", "Jess", "", "Matt", "", "Greg"]
Output	f(x) = ["Joe", "Sarah", "Mike", "Jess", "Matt", "Greg"]

You should be able to begin writing functions on your own right now.

Python

Control Statement in a Loop

There are 3 types of control statements in Python, which are used to control the flow of the program, especially in a loop.

1. Break	breaks from current loop now
2. Continue	skips current loop and go to next iteration
3. Pass	means do nothing (just a placeholder statement)

while keyword

```
while condition:
    # do something here
    ...
```

Python

Control Statement: Break

Break Examples

A function that searches the list for an element *k*.

If the element exists, it should return True, else False.

Key Idea: we search the list from the beginning to the end. If the element is found, you can stop the loop there and not wasting time on other elements because you reached the objective.

```python
def search(lst, k):
    found = False

    for elem in lst:
        if elem == k:
            found = True
            break

    return found
```

Python

Control Statement: Continue

Continue Examples

A function that sums all integers that is not divisible by 5.

You can think it in 2 ways:

1. If the integer is not divisible by 5, then adds the number.

2. If the integer is divisible by 5, skips this iteration and go next.

The first one is straightforward, but if you have many conditions, it will be so many nested statements when you have a lot of codes.

The second one is more readable and understandable.

```python
def add5(lst):
    sums = 0

    for elem in lst:
        if elem % 5 == 0:
            continue
        sums += elem

    return sums
```

Python

Nested Loops

A nested loop is a loop that is within a loop. There can be as many levels of loop as you want.

You can put any types of loop within a loop. For example, while within for loop and vice versa.

You can try this program and see what it does.

Try writing each case:
When i = 0:   j = 0, 1, 2
When i = 1:   j = 0, 1, 2

So, the print statement run 6 times.

```python
n = 2
m = 3

for i in range(n):
    for j in range(m):
        print(i, j)
```

Nested Loops

Exercise 5

Write a program that prints a square of size *n* x *n* on the terminal screen with asterisk (star).

For example, when *n* = 6, it should show

```
******
******
******
******
******
******
```

Try using print's optional arguments:

```
print('*', end='')
```

Python

Nested Loops

DO THE PRACTICE PROBLEMS!