Loops & Built-in Data Structures

- Abdullah Zayed

contact@zayedabdullah.com

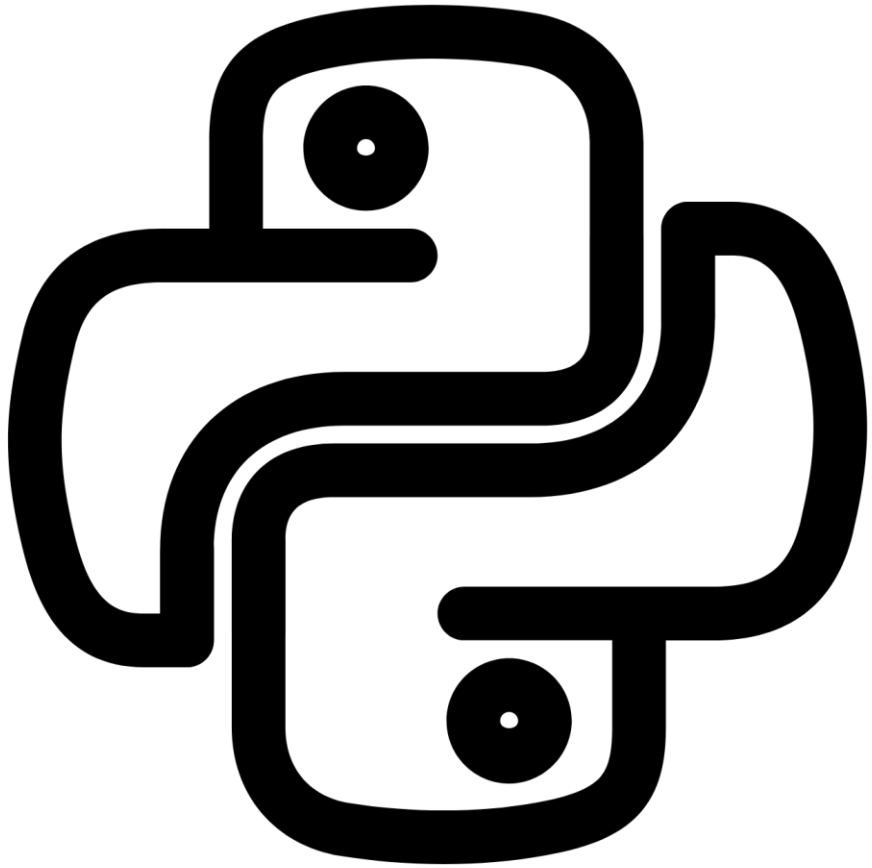
Agenda

Looping???

Break & Continue

List, Tuple, Set

Dictionary



contact@zayedabdullah.com

Looping

WHAT IS LOOPING? HOW CAN WE CONTROL IT?

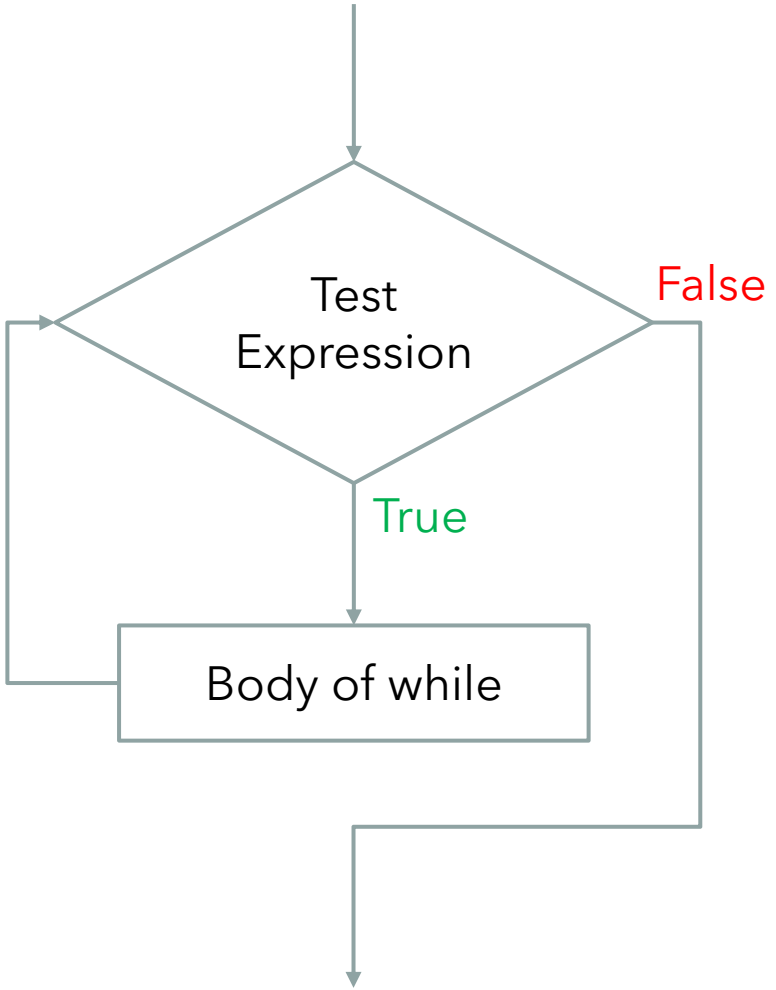
contact@zayedabdullah.com

Looping

contact@zayedabdullah.com

“ while ”
in Python

LOOPING WITH WHILE



contact@zayedabdullah.com

“ while ”
in Python

LOOPING WITH WHILE

main.py

```
1 count = 0
2
3 while count < =10:
4     print("Count:", count)
5     count = count + 1
6
7  print("Done Looping!")
```

contact@zayedabdullah.com

Break & Continue in Python

main.py

```python
i = 1

while i < 6:
  print(i)

  if i == 3:
    break

  i += 1
```

main.py

```python
i = 0

while i < 6:
  i += 1

  if i == 3:
    continue

  print(i)
```

List & Tuple in Python

Lists and **Tuples** are just collections of multiple items that are stored sequentially. **List** is **mutable**, but **Tuple isn't**.

list_example.py

```
1 fruits = ["apple", "banana"]
2
3 items = [True, 5.2, "Yo"]
4
5 print(fruits[0])
6 print(fruits[1][2])
```

tuple_example.py

```
1 fruits = ("apple", "banana")
2
3 items = (True, 5.2, "Yo")
4
5 print(fruits[0])
6 print(fruits[1][2])
```

contact@zayedabdullah.com

Update Value of List in Python

value_update_in_list.py

```
1 fruits = ["Apple", "Banana", "Cherry", "Mango", "Guava"]
2 print(fruits)
3
4 fruits[0] = "Pineapple"
5 fruits[1] = "Berry"
6 print(fruits)
```

List & Tuple Slicing in Python

list_slicing.py

```python
list1 = ["Apple", "Banana", "Cherry", "Mango", "Guava"]

print(list1[0:2])
print(list1[2:])
print(list1[-1])
print(list1[-3:-1])
print(list1[0:5])
print(list1[0:5:2])
print(list1[-1:-2])
print(list1[-1:-2:-1])
```

Operators on List & Tuple in Python

operator_in_list_or_tuple.py

```
1 list1 = ["apple", "banana", "cherry", "mango", "orange"]
2
3 print(list1 + ["tomato", 50])
4 print(list1 * 3)
5
6 print(list1)
```

Unpacking List & Tuple in Python

list_or_tuple_unpack.py

```python
fruits = ["Apple", "Cherry", "Banana", "Strawberry"]
watches = ("Cellox", "Montex")

fruit1, *rest_of_the_fruits = fruits
watch1, watch2 = watches

print(fruit1)
print(rest_of_the_fruits)

print(watch1)
```

List & Tuple Compare in Python

list_or_tuple_compare.py

```
1 a = [5, 6, 25, 51]
2 b = [1, 9, 18, 46, 25]
3
4 if a > b:
5     print("a is bigger")
6 else:
7     print("b is bigger")
```

List Methods in Python

These functions **modify** the original list.

Method	Description	Returns
append()	Adds an element at the end of the list	None
extend()	Appends all items from another list at the end of the current list	None
insert()	Inserts an item into specified index	None
reverse()	Reverses all items in the list	None

List Methods in Python

These functions **modify** the original list.

Method	Description	Returns
pop()	Returns an item and removes it	Removed Item
remove()	Removes an item from the list	None
clear()	Removes all element from the list	None
sort()	Sorts the list	None

List Methods in Python

These functions **don't modify** the original list.

Method	Description	Returns
index()	Finds the index of an item in the list	Index of the Item
count()	Finds the number of elements in the list	Number of Item

2D List (Matrix) in Python

list_2d.py

```python
my_matrix = [
    [ 1, 4,  5, 12],
    [-5, 8,  9,  0],
    [-6, 7, 11, 19]
]

print("A =", A)
print("A[1] =", A[1])
print("A[1][2] =", A[1][2])
print("A[0][-1] =", A[0][-1])
```

list_2d.py

```python
my_matrix = [
    [ 1, 4,  5, 12],
    [-5, 8,  9,  0],
    [-6, 7, 11, 19]
]

spec_col = []
for row in my_matrix:
  spec_col.append(row[2])

print(spec_col)
```

Tuples in Python

Tuples are just lists that are ***unchangeable***.

tuple_example.py

```
1 fruits = ("Banana", "Apple", "Cherry", "Strawberry")
2 watches = ("Cellox",)
3
4 print(fruits)
5 print(watches)
```

Sets in Python

Sets are just collections of items that are ***unordered***, ***partially unchangeable***, ***unindexed***, and ***unique***.

Set items can only be removed and added in a specific way.

set_example.py

```
1 fruits = {"apple", "banana", "cherry"}
2 numbers = {1, 5, 7, 9, 3}
3 mixeds = {"abc", 34, True, 40, "male"}
```

Set Operations in Python

set_example.py

```python
num1 ={1, 2, 3, 4, 5}
num2 ={4, 5, 6, 7}

print(num1 | num2)  #union
print(num1.union(num2))  #union

print(num1 & num2)  #intersection
print(num1.intersection(num2))  #intersection

print(num1 - num2)  #difference
print(num1.difference(num2))  #difference
```

Dictionary in Python

A dictionary is just a collection of two items in packed.

It’s simply key-value mapping.

dictionary_example.py

```python
student = {
  "name": "Kashem",
  "platform": "Ostad",
  "course": "Data Science",
  "batch": 38,
  "is_valid": True
}
```

dictionary_example.py

```python
print(type(student))

print(student["name"])

print(type(student["batch"]))

print(type(student["is_valid"]))
```

Operations in Dictionary

dictionary_example.py

```python
student = {
  "name": "Kashem",
  "course": "Ostad - Data Science",
  "batch": 38,
  "is_valid": True
}

dict_items = student.items()
print(dict_items)
print(type(list(dict_items)[0]))
print(student.keys())
print(student.values())
```

dictionary_example.py

```python
student = {
  "name": "Kashem",
  "course": "Ostad - Data Science",
  "batch": 38,
  "is_valid": True
}

student.update({
  "platform": "Ostad",
  "course": "Data Science"
})
print(student)
```

For Loop in Python

ITERATING THROUGH ITEMS?

contact@zayedabdullah.com

Iterating Items with For Loop

List, Sets, Tuples are **iterative**. So, a for loop can be used to iterate over every item.

forloop_example.py

```
1 fruits = ["Apple", "Banana", "Cherry", "Strawberry"]
2
3 for fruit in fruits:
4     print(fruit)
```

Looping in Dictionary

forloop_example.py

```python
student = {
  "name": "Kashem",
  "course": "Ostad - Data Science",
  "batch": 38,
  "is_valid": True
}

for info in student.items():
    print(info[0], "-", info[1])
```

Range in For Loop

forloop_example.py

```
1 for i in range(3):
2     print(i)
3
4 for i in range(2, 7):
5     print(i)
6
7 for i in range(2, 13, 2):
8     print(i)
```

Some Practice Problems in Python

- Make a program that receives an integer “n”, calculates the sum of 1 to n, then shows the sum.

- Make a program that takes numbers until it finds zero. Shows the result of multiplication of all the numbers taken.

contact@zayedabdullah.com

Practice Problem

Shopping List Manager

You are asked to create a shopping list manager.

The user can add items to the shopping list when “done” is typed, the input system closes, and you need to display the list along with the total number of items.

contact@zayedabdullah.com

Practice Problem

Score Tracker

You are asked to create a Score Tracker. You'll take the name and the score of a student as input. Store those by name and print all the student's names with their scores. The input ends when the student's name is "stop".

contact@zayedabdullah.com

Thank you

Abdullah Zayed

contact@zayedabdullah.com

www.zayedabdullah.com

linkedin.com/in/abdullahzayed01/