**PythonBasic - Collections**

List : It is only an ordered collection of valid Python values. One of the most useful collections in Python is a list. The list type is probably the most commonly used collection type in Python.

list() : A built-in function that creates or converts to a list type.

.append() :It is a method that adds an object to the end of a list.

.insert() : It is a method that adds a new object to a list at a specific index.

.sort() ;It is a method that lists the elements of a list in alphabetical or numerical order.

.remove() ; It is a method that removes the elements from the list.

len() ; A built-in function that calculates the length of an object. It means that it gives the number of all elements of an iterable object.

sequence[start:stop:step] ; This formula produces a slice of the sequence where start is an index of the first element required (the element is included in the slice) and stop is an index of the end element (the element is not included in the slice), step is an interval between elements to be chosen.

range(start, stop, step) ; Returns an object that produces a sequence of integers from start (including) to stop (excluding) by step.

my\_list[:] ; returns the full copy of the sequence.

my\_list[start:] ; returns elements from start to the end element.

my\_list[:stop] ; returns element from the 1st element to stop-1.

my\_list[::step] ; returns each element with a given step.

Negative indexing ; It is the best and shortest way to reach the elements at the end of the list. It works in reverse. We can reach the last element of a list as list\_name[-1].

Tuple ; It is another collection type that can hold multiple data very similar to the list. The most important difference from the list is that the tuple is immutable. Therefore, methods like .append() or .remove() do not exist in the operations of this type.

list or tuple? Which one is faster? ; If we have unchanged data, we should choose tuples because it is much faster than lists.

tuple() ; A built-in function that creates or converts to a tuple type.

item in dictionary ; A dictionary in Python is a collection of 'key-value' pairs called items of a dictionary.

dict() ; A built-in function that creates or converts to a dictionary type.

.items() ; It is a method that allows us to access all items in the dictionary.

.keys() ; It is a method that allows us to access all keys in the dictionary.

.values() ; It is a method that allows us to access all values in the dictionary.

.update() ; It is a method that allows us to add a new item into the dictionary.

Del ; Removes an item from the dictionary.

Set ; It is a collection of elements with no repeats and without insertion order but sorted order. Basic uses include membership testing and eliminating duplicate entries. Set objects also support mathematical operations like union, intersection, difference, and symmetric difference.

set() ; A built in function that creates or converts to a set type.

.add() : It is a method that adds a new item to the set.

...

.remove() : It is a method that allows us to delete an item.

...

.intersection() : It is a method that returns the intersection of two sets.

...

.union() : It is a method that returns the unification of two sets.

...

.difference() : It is a method that gets the difference of two sets.

...

a – b a.difference(b)

b – a b.difference(a)

a | b a.union(b)

b | a b.union(a)

a & b a.intersection(b)

b & a b.intersection(a)

**PythonBasic - Control Flow Statements**

**==** Equal to. It returns True if two values are equal or False if different.

**!=**  Not equal to. It returns True if two values are not equal or False if equal.

**>**Greater than. It returns True if the value on the left is greater than the value on the right otherwise returns False.

**<** Less than. It returns True if the value on the left is less than the value on the right otherwise returns False.

**>=** Greater than or equal to. It returns True if the value on the left is greater than or equal to the value on the right otherwise returns False.

**<=** Less than or equal to. It returns True if the value on the left is less than or equal to the value on the right otherwise returns False.

**input()** A built-in function that allows us to get a value from the user and assigns it to a variable that you choose.

**Loop** When writing programs in Python, in some cases it is not enough to execute our block of code only once. The loops are used to repeat (iterate) the execution of a block of code.

Loop types in Python 1.while loop, 2.for loop

**for loop** It is used to repeat the code block using iterable type objects such as string and collections.

**while loop** It is used to repeat the code block using conditional statements. As long as the condition is true, the loop/repetition continues.

Iterable object

It can be anything for which items are received one by one, forward only. In Python, the process of recurrent execution of a block of code is called an iteration.

Definite iteration

If the number of repetitions is predetermined, it is called definite iteration

Indefinite iteration

The repetition structure that makes the code block run as long as the predetermined condition generates True is called indefinite iteration.