# 5. Looping (For, While)

## ❖ Introduction to for and while loops.

In Python, loops are fundamental constructs that allow you to execute a block of code repeatedly. The two primary loop types are for loops and while loops, each serving different purposes.

### for Loop:

A for loop is used to iterate over a sequence and execute a block of code for each item in the sequence.

### Syntax:

```
for item in sequence:
    # Execute code for each item
```

### while Loop:

A while loop repeatedly executes a block of code as long as a specified condition evaluates to True.

### Syntax:

```
while condition:
```

```
    # Execute code as long as condition is True
```

## ❖ How loops work in Python

In Python, loops are fundamental constructs that allow you to execute a block of code multiple times. Python primarily offers two types of loops: for loops and while loops. Each serves distinct purposes and is suited for different scenarios.

### for Loop:

A for loop is used to iterate over a sequence and execute a block of code for each item in the sequence.

### Syntax:

```
for variable in sequence:
    # execute this block of code
```

### while Loop:

A while loop repeatedly executes a block of code as long as the given condition is True.

**Syntax:**

```
while condition:
    # execute this block of code
```

## ❖ Using loops with collections (lists, tuples, etc.).

In Python, loops are essential for iterating over collections like lists, tuples, sets, and dictionaries. They allow for efficient and readable code when performing repetitive tasks or processing multiple data elements.

### for Loop:

The for loop in Python is designed to iterate over items of a sequence (such as a list, tuple, or string) or other iterable objects. This loop is ideal for executing a block of code for each item in the collection.

**Syntax:**

```
for item in iterable:
```

```
    # Code to execute
```

## while Loop:

The while loop repeatedly executes a block of code as long as a specified condition evaluates to True. It's useful when the number of iterations is not known beforehand and depends on a condition.

## Syntax:

```
while condition:
    # Code to execute
```

## List Comprehensions:

List comprehensions provide a concise way to create lists by iterating over an iterable and optionally applying a condition.

## Syntax:

```
[expression for item in iterable if condition]
```

Etc…