while Loop

A **while Loop** executes a block of code a certain number of times as long as the **condition is True**

for Loop

A **for Loop** executes a block of code a **certain number of times**.
- You know **when to start**
- and **when to stop**
- and what to **increase** or **decrease** by.

- **For Loops** are used to **iterate** through a sequence
  - (like a "string", numbers, [lists or arrays], etc.)
- **Iteration** – is a **repeat of an action** a predefined number of times.

for Loop SYNTAX:

for iteration_variable in range(start, stop, step):

  #body of for loop to be repeated

range( ) method – takes up to 3 arguments

range(start, stop, step)

- **start** - starting integer
- **stop** - stopping integer – 1
- **step (Optional)** - increase or decrease integer value(can be a positive or negative number)

range( ) method –   if only 1 argument

for num in range(5):

    #body of for loop to be repeated

    print(num)

#num is iteration_variable
STARTS at 0,
STOPS at num -1
increments by 1

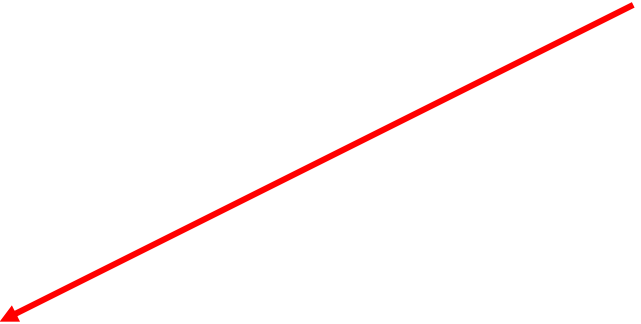# range( ) method — if 2 arguments

START at first number, STOP is at num -1, increments by 1.

for num in range(2, 5):

    #body of for loop to be repeated

    print(num)

range( ) method – if 3 arguments

#range(start, stop [,step])

```
for num in range(2, 10, 2):
        #body of for loop repeated
        print(num)
```

for Loops and range method Syntax

```
for num in range(5):
        print(num, end = " ")
```

What is OUTPUT?

for Loops Syntax

```
for num in 5:
        print(num)
```

#Error 'int' object is not iterable.

for Loops Syntax (start and stop values)

```python
for num in range(12, 16):
        print(num)
```

What is OUTPUT?

for Loops Syntax (start, stop and step values)

```python
for num in range(3, 8, 2):
    print(num)
```

What is OUTPUT?

for Loops Syntax (range negative values)

```python
for num in range(5, -1, -1):
        print(num)
```

What is OUTPUT?

break and continue

- Can also be used in for Loops.