

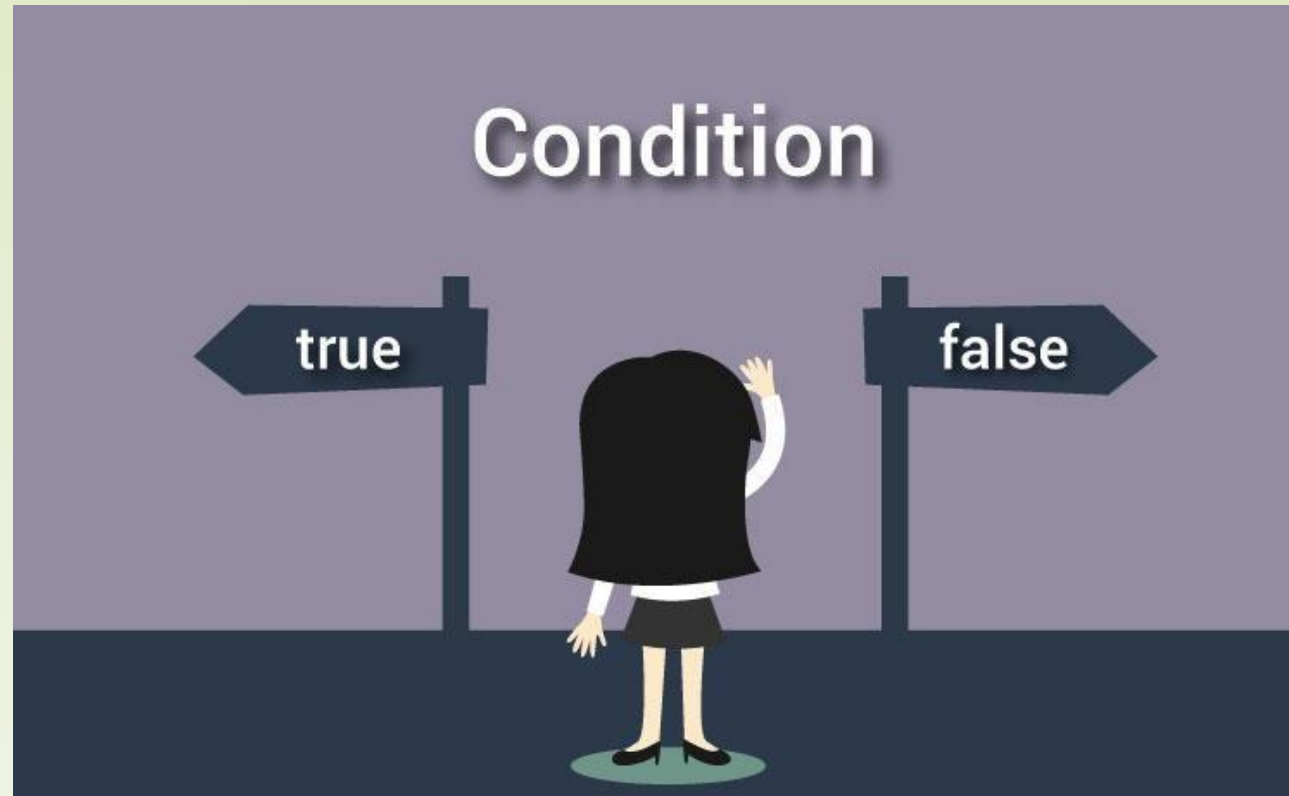
Python Programming



Conditional statements, Loop and Iterations



Conditions in PYTHON



Python if...else Statement:

- ▮ In Python, there are three forms of the if...else statement.
 - ▮ if statement
 - ▮ if...else statement
 - ▮ if...elif...else statement

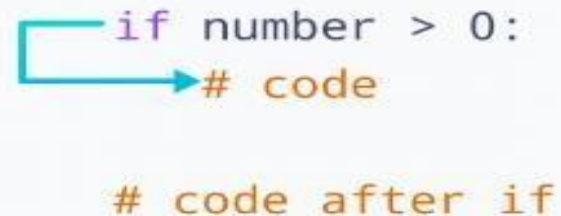
Python if statement:

- ▮ The if statement evaluates condition.
 - ▮ If **condition** is evaluated to **True**, the code inside the body of **if** is **executed**.
 - ▮ If **condition** is evaluated to **False**, the code inside the body of **if** is **skipped**.

```
if condition:  
    # body of if statement
```

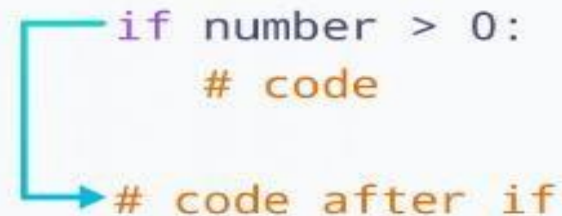
Condition is True

```
number = 10  
if number > 0:  
    # code  
  
# code after if
```



Condition is False

```
number = -5  
if number > 0:  
    # code  
  
# code after if
```



Python if statement:

- ▮ The if statement evaluates condition.
 - ▮ If **condition** is evaluated to **True**, the code inside the body of **if** is **executed**.
 - ▮ If **condition** is evaluated to **False**, the code inside the body of **if** is **skipped**.

```
number = 10

if number > 0:
    print('Number is positive.')

print('The if statement is easy')
```

```
a = 33
b = 200
if b > a:
    print("b is greater than a")
```

```
a = 33
b = 200
if b > a:
    print("b is greater than a")
```

If ... else statements for python

- the program evaluates the test expression and will execute statement(s) only if the test expression is True.

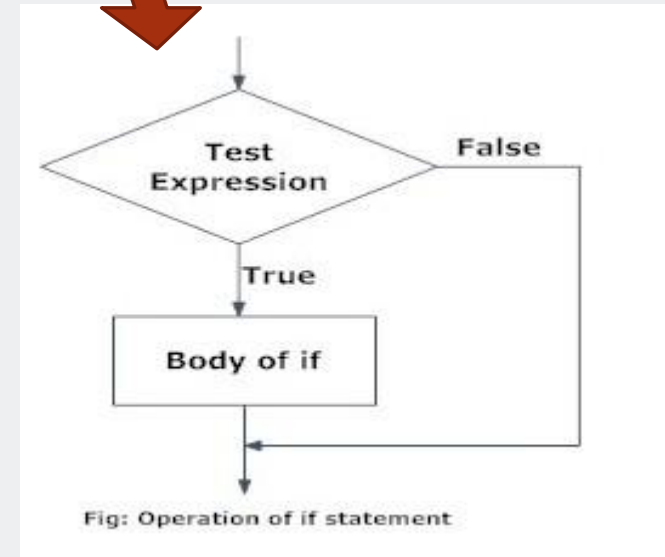
- Syntax :

**If test expression :
Statement(s)**

- Examples:

```
>>>a=6  
>>> if a>0:  
    print (a "is positive number")
```

```
>>> a=-1  
>>> if a>0:  
    print (a "is positive number")
```



8

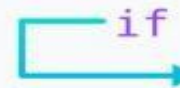
Python if...else Statement

- ▮ An if statement can have an optional else clause.

```
if condition:  
    # block of code if condition is True  
  
else:  
    # block of code if condition is False
```


Condition is True

```
number = 10  
if number > 0:  
    # code  
  
else:  
    # code  
  
# code after if
```



Condition is False

```
number = -5  
if number > 0:  
    # code  
  
else:  
    # code  
  
# code after if
```



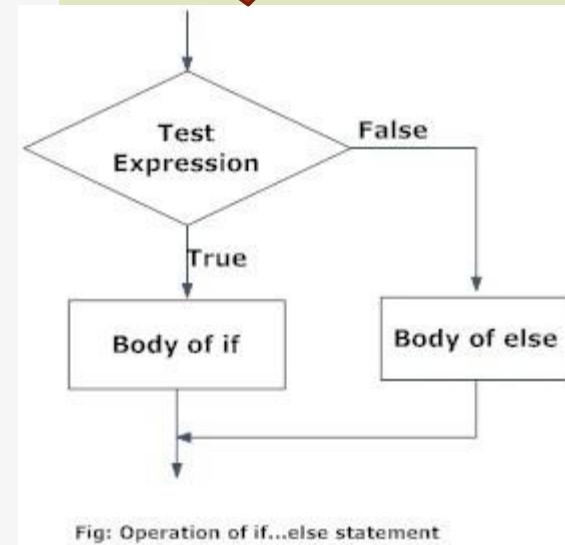
Python if...else Statement

▮ Syntax:

```
if test expression:  
    Body of if  
else:  
    Body of else
```

▮ Example :

```
>>> a = 9  
>>> if a > 5:  
    print (a "is greater than 5")  
else :  
    print (a, "is less than 5")
```



Python if...else Statement

- ▮ An if statement can have an optional else clause.

```
number = 10

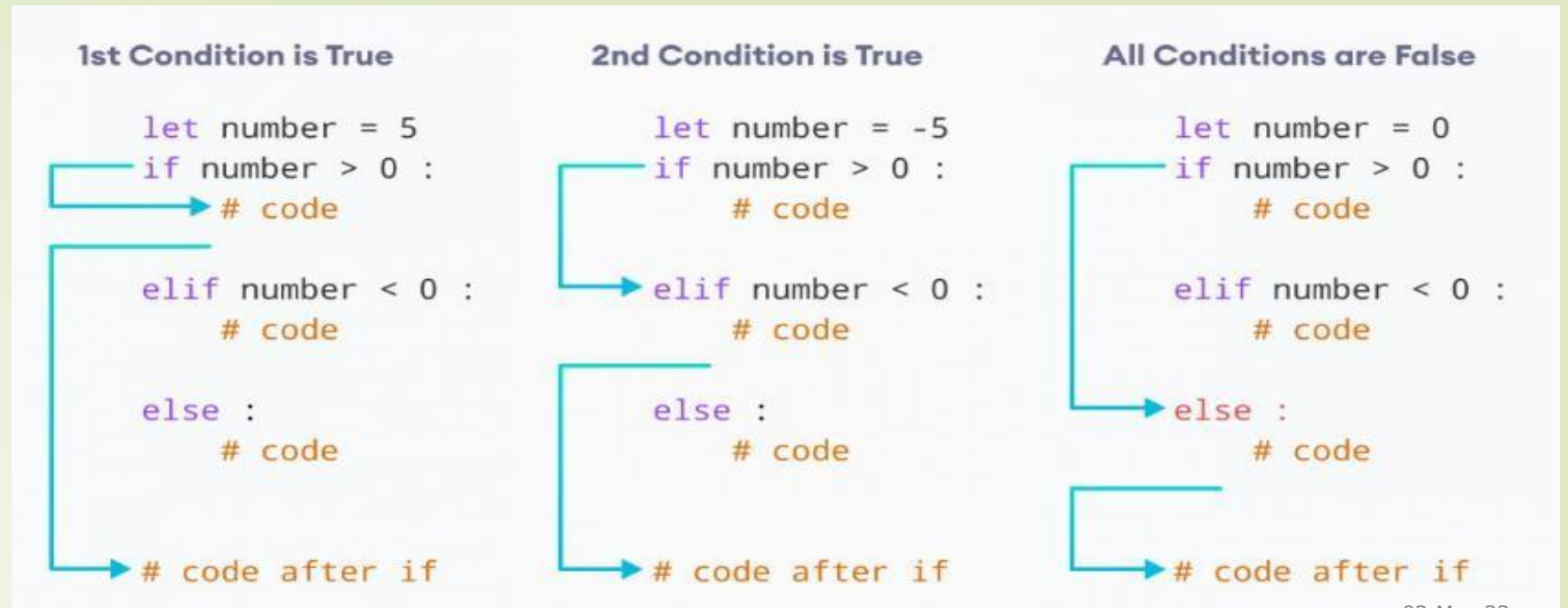
if number > 0:
    print('Positive number')
else:
    print('Negative number')

print('This statement is always executed')
```

Python if...elif...else Statement

- ▮ The **if...else** statement is used to execute a block of code **among two alternatives**.
- ▮ However, if we need to make a choice between more than two alternatives, then we use the if...elif...else statement.

```
if condition1:  
    # code block 1  
  
elif condition2:  
    # code block 2  
  
else:  
    # code block 3
```



Python if...elif...else statements

□ Syntax :

if test expression:

 Body of if

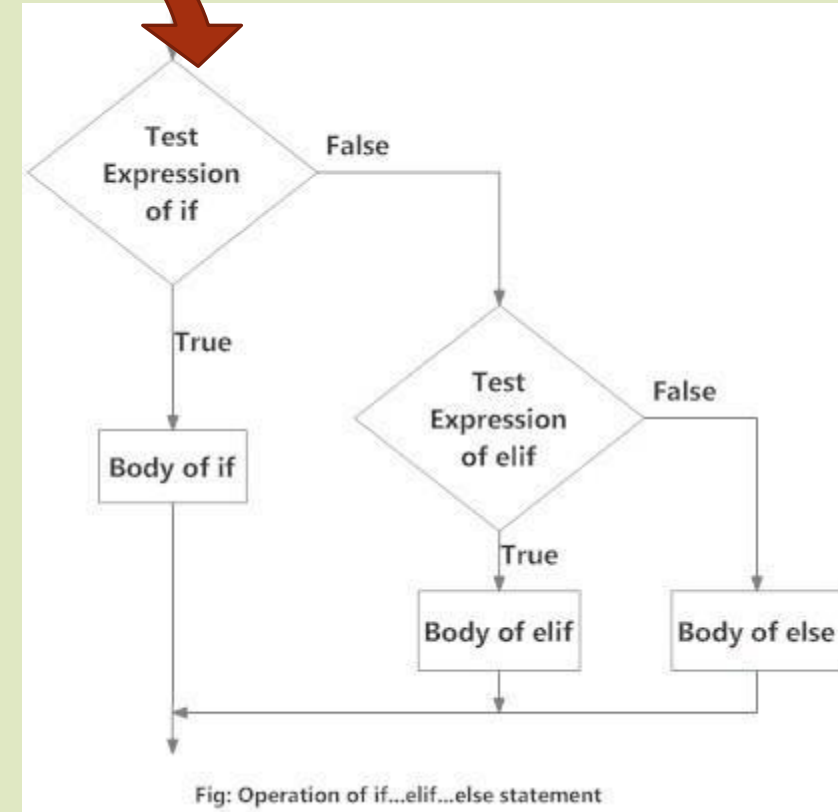
elif test expression:

 Body of elif

else:

 Body of else

Try by yourself



Python if...elif...else Statement

- ▮ The **if...else** statement is used to execute a block of code **among two alternatives**.
- ▮ However, if we need to make a choice between more than two alternatives, then we use the if...elif...else statement.

```
number = 0

if number > 0:
    print("Positive number")

elif number == 0:
    print('Zero')
else:
    print('Negative number')

print('This statement is always executed')
```

Python Nested if statements

- ▮ An if statement inside of an if statement. This is known as a nested if statement.

```
# outer if statement
if condition1:
    # statement(s)

    # inner if statement
    if condition2:
        # statement(s)
```

- ▮ We can add else and elif statements to the inner if statement as required.
- ▮ We can also insert inner if statement inside the outer else or elif statements(if they exist)
- ▮ We can nest multiple layers of if statements.

Python Nested if statements

```
number = 5

# outer if statement
if (number >= 0):
    # inner if statement
    if number == 0:
        print('Number is 0')

    # inner else statement
    else:
        print('Number is positive')

# outer else statement
else:
    print('Number is negative')
```

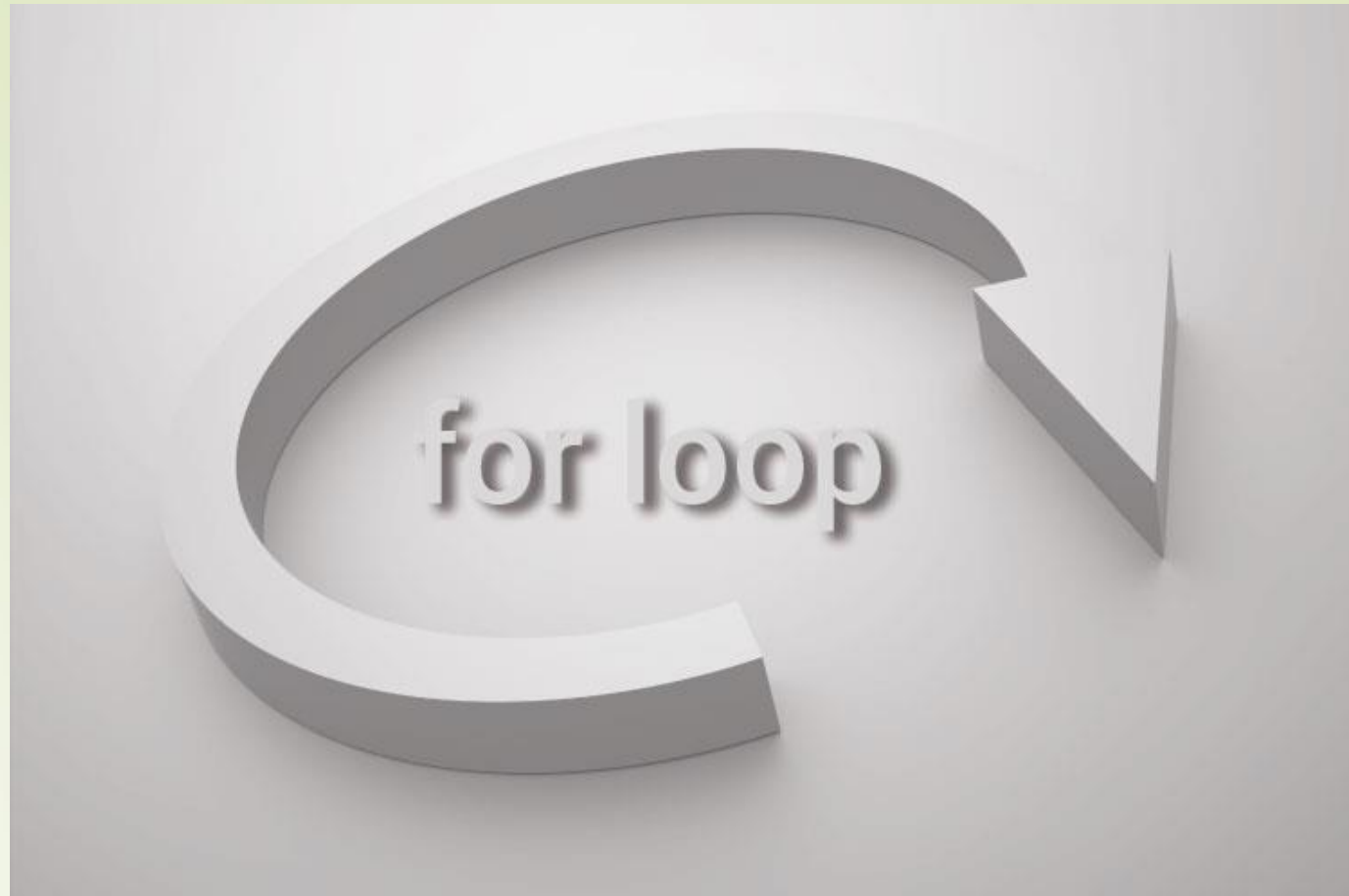
Short hand if, if..else

```
a=15  
b=10  
if a > b: print("a is greater than b")
```

```
a=15  
b=10  
print("A") if a > b else print("B")
```

```
a=15  
b=15  
print("A") if a > b else print("=") if a == b else print("B")
```


FOR LOOP for PYTHON

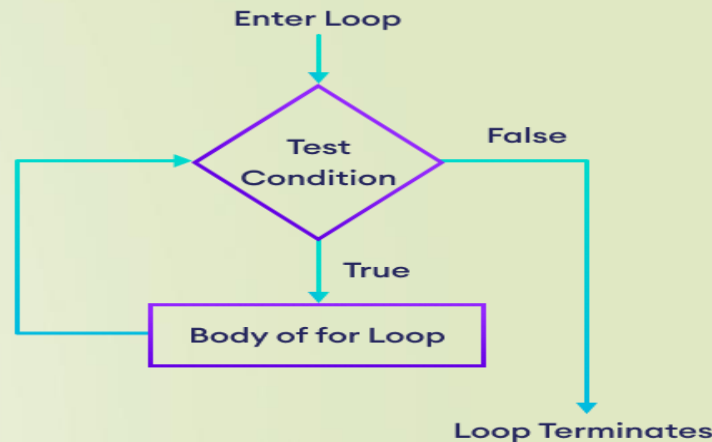


Python for Loop:

- ❑ In Python, the for loop is used to run a block of code for a certain number of times. It is used to iterate over any sequences such as list, tuple, string, etc.

```
for val in sequence:  
    # statement(s)
```

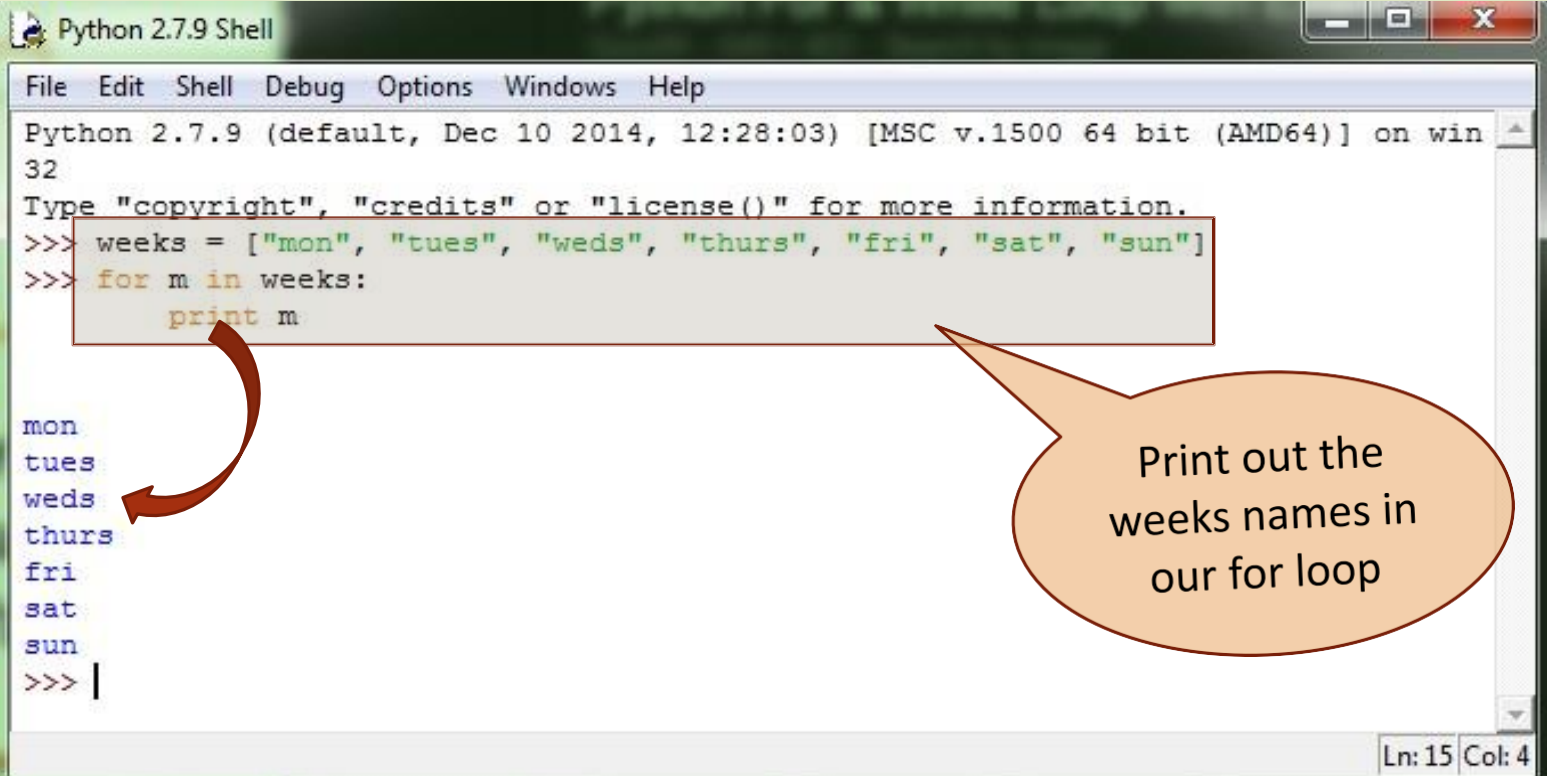
- ❑ **val** accesses each item of sequence on each iteration. Loop continues until we reach the last item in the sequence.



LOOPS for Python

For - Loop

- Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.



```
Python 2.7.9 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.9 (default, Dec 10 2014, 12:28:03) [MSC v.1500 64 bit (AMD64)] on win
32
Type "copyright", "credits" or "license()" for more information.
>>> weeks = ["mon", "tues", "weds", "thurs", "fri", "sat", "sun"]
>>> for m in weeks:
>>>     print m

mon
tues
weds
thurs
fri
sat
sun
>>> |
```

Print out the weeks names in our for loop

Ln: 15 Col: 4

Python for Loop:

```
languages = ['Swift', 'Python', 'Go', 'JavaScript']  
  
# access items of a list using for loop  
for language in languages:  
    print(language)
```

Python for Loop with Python range():

- ▮ A range is a series of values between two numeric intervals.
- ▮ We use Python's built-in function range() to define a range of values.

```
values = range(4)
```

- ▮ Here, 4 inside range() defines a range containing values 0, 1, 2, 3.

```
# use of range() to define a range of values
values = range(4)

# iterate from i = 0 to i = 3
for i in values:
    print(i)
```

Python for loop with else

- ▮ A for loop can have an optional else block as well. The else part is executed when the loop is finished.

```
digits = [0, 1, 5]

for i in digits:
    print(i)
else:
    print("No items left.")
```

- ▮ Here, the for loop prints all the items of the digits list. When the loop finishes, it executes the else block and prints No items left.
- ▮ **Note: The else block will not execute if the for loop is stopped by a break statement.**

Python range() Function

- ▮ Create a sequence of numbers from 0 to 5, and print each item in the sequence:

```
x = range(6)
for n in x:
    print(n)
```

- ▮ The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.
- ▮ Create a sequence of numbers from 3 to 5, and print each item in the sequence:

```
x = range(3, 6)
for n in x:
    print(n)
```

Python range() Function

```
range(start, stop, step)
```

- ▮ **start Optional.** An integer number specifying at which position to start. **Default is 0**
- ▮ **stop Required.** An integer number specifying at which position to stop (not included).
- ▮ **step Optional.** An integer number specifying the incrementation. **Default is 1**
- ▮ Create a sequence of numbers from 3 to 19, but increment by 2 instead of 1:

```
x = range(3, 20, 2)
for n in x:
    print(n)
```

- ▮ `>>> range (6, 18, 3) # returns: [6, 9, 12, 15]`
- ▮ `>>> range (10) # returns: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]`

While LOOP for PYTHON

▮ Syntax of while Loop in Python

While test_expression:

Body of while

▮ Example :

```
>>> n=10
```

```
>>> s=0
```

```
>>> i=1
```

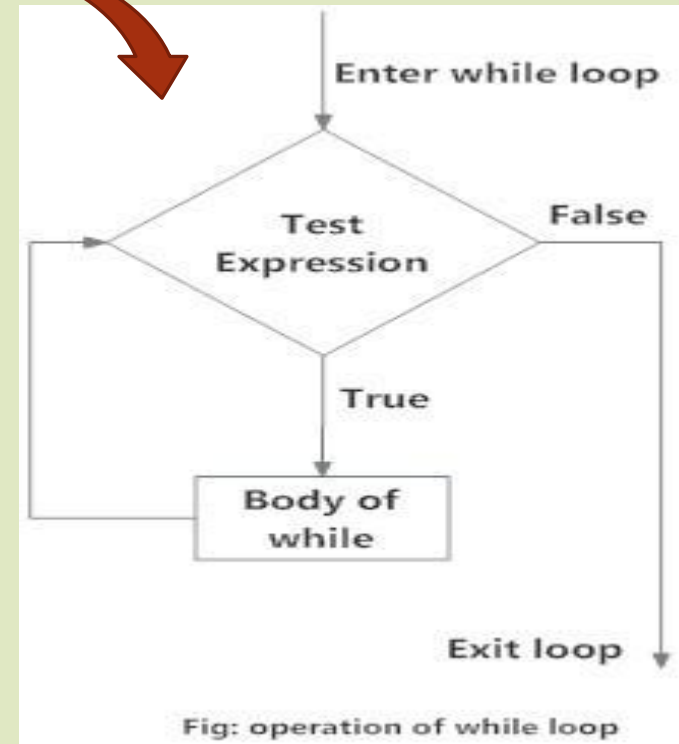
```
>>> while
```

```
    i<=n: s=s+i
```

```
    i=i+1
```

```
>>> print s
```

Print : 55



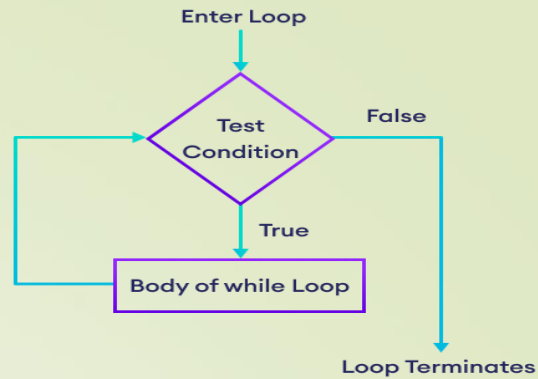
Python while Loop

- Python while loop is used to run a block code until a certain condition is met.

```
while condition:  
    # body of while loop
```

- A while loop evaluates the condition
- If the condition evaluates to True, the code inside the while loop is executed.
- condition is evaluated again.
- This process continues until the condition is False.
- When condition evaluates to False, the loop stops.

Python while Loop



```
# initialize the variable
i = 1
n = 5

# while loop from i = 1 to 5
while i <= n:
    print(i)
    i = i + 1
```

```
total = 0

number = int(input('Enter a number: '))

# add numbers until number is zero
while number != 0:
    total += number    # total = total + number

    # take integer input again
    number = int(input('Enter a number: '))

print('total =', total)
```

Infinite while Loop in Python

- ▮ If the condition of a loop is always True, the loop runs for infinite times (until the memory is full).

```
age = 32

# the test condition is always True
while age > 18:
    print('You can vote')
```

Python While loop with else

- ❑ In Python, a while loop may have an optional else block.
- ❑ Here, the else part is executed after the condition of the loop evaluates to False.

```
counter = 0

while counter < 3:
    print('Inside loop')
    counter = counter + 1
else:
    print('Inside else')
```

- ❑ **Note:** The else block will not execute if the while loop is terminated by a break statement.

Python While loop with else

- ❑ In Python, a while loop may have an optional else block.
- ❑ Here, the else part is executed after the condition of the loop evaluates to False.

```
counter = 0

while counter < 3:
    # loop ends because of break
    # the else part is not executed
    if counter == 1:
        break

    print('Inside loop')
    counter = counter + 1
else:
    print('Inside else')
```

while loop with else

- ▮ Same as that of [for loop](#)
- ▮ The else part is executed if the condition in the while loop evaluates to False.
- ▮ Examples:

```
>>> a = 0
>>> while a < 3:
    print "inside"
    a = a + 1
>>> else:
    print "outside"
```

Python for conditional operators

```
Module6MakingDecisionsWithCode.py - C:\Users\BRNS\Desktop\Python Programming\Introduction...
File Edit Format Run Options Windows Help

#if with numbers
freeToaster = None

deposit = int(input("how much do you want to deposit "))
if deposit > 100 :
    freeToaster = True

#complex code here...
if freeToaster :
    print("enjoy your toaster")
print("Have a nice day!")
```

```
Python 2.7.9 Shell
File Edit Shell Debug Options Windows Help

Python 2.7.9 (default, Dec 10 2014, 12:28:03) [MSC v.1500 64 bit (AMD64)] on win
32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
how much do you want to deposit 200
enjoy your toaster
Have a nice day!
>>> ===== RESTART =====
>>>
how much do you want to deposit 10
Have a nice day!
>>>
```


Python for Vs while loops:

- ▮ The for loop is usually used when the number of iterations is known. For example,

```
# this loop is iterated 4 times (0 to 3)
for i in range(4):
    print(i)
```

- ▮ The while loop is usually used when the number of iterations is unknown. For example,

```
while condition:
    # run code until the condition evaluates to False
```

Python break Statement:

- ▮ The break statement is used to terminate the loop immediately when it is encountered.

```
for val in sequence:  
    # code  
    if condition:  
        break
```

code

```
while condition:  
    # code  
    if condition:  
        break
```

code

Python break Statement:

- ▮ The break statement is used to terminate the loop immediately when it is encountered.

```
for i in range(5):  
    if i == 3:  
        break  
    print(i)
```

```
i = 1  
  
while i <= 10:  
    print('6 * ',(i), '=',6 * i)  
  
    if i >= 5:  
        break  
  
    i = i + 1
```

- ▮ Note: The break statement is almost always used with decision-making statements.

Python continue Statement:

- ▮ The continue statement is used to skip the current iteration of the loop and the control flow of the program goes to the next iteration.

```
→ for val in sequence:  
    # code  
    if condition:  
        continue
```

```
    # code
```

```
→ while condition:  
    # code  
    if condition:  
        continue
```

```
    # code
```

Python continue Statement:

- ▮ The continue statement is used to skip the current iteration of the loop and the control flow of the program goes to the next iteration.

```
for i in range(5):  
    if i == 3:  
        continue  
    print(i)
```

```
num = 0  
  
while num < 10:  
    num += 1  
  
    if (num % 2) == 0:  
        continue  
  
    print(num)
```

▮ For loop in range limit

```
>>> for a in range(3, 10):
    print a      result : 3, 4, 5 , 6, 7, 8, 9, 10
```

▮ For loop with BREAK point

```
>>> for i in range(3, 10):
    if(i==7): break
    print i      result: 3, 4, 5, 6
```

▮ For loop for continue statement

```
>>> for x in range(10, 20):
    if (x%5 ==0): continue
    print x      result : 11, 12, 13, 14,16, 17, 18, 19
```

- For loop with enumeration

```
>>> months= ["jan", "feb", "mar", "april",
              "May", "june"]
>>> for i, m in enumerate(months):
    print i, m
```

```
1  jan
2  Feb
3  Mar
3  April
4  May
5  June
```

For Practice:

- ▮ Python Program to Print Hello world!
- ▮ Python Program to Add Two Numbers.
- ▮ Python Program to Find the Square Root.
- ▮ Python Program to Calculate the Area of a Triangle.
- ▮ Python Program to Solve Quadratic Equation.
- ▮ Python Program to Swap Two Variables.
- ▮ Python Program to Generate a Random Number.
- ▮ Python Program to Convert Kilometers to Miles.
- ▮ Python Program to Convert Celsius To Fahrenheit.
- ▮ Python Program to Print Output Without a Newline.
- ▮ <https://www.programiz.com/python-programming/examples>

For Practice:

- ▮ Python Program to Check if a Number is Positive, Negative or 0.
- ▮ Python Program to Check if a Number is Odd or Even.
- ▮ Python Program to Check Leap Year.
- ▮ Python Program to Find the Largest Among Three Numbers.
- ▮ Python Program to Check Prime Number.
- ▮ Python Program to Find the Factorial of a Number.
- ▮ Python Program to Display the multiplication Table.
- ▮ Python Program to Print the Fibonacci sequence.
- ▮ Python Program to Check Armstrong Number.
- ▮ Python Program to Find the Sum of Natural Numbers.

For Practice:

- ▮ Python Program to Create Pyramid Patterns.
- ▮ Python Program to Iterate Over Dictionaries Using for Loop.
- ▮ Python Program to Reverse a Number.
- ▮ Python Program to Compute the Power of a Number.
- ▮ <https://www.programiz.com/python-programming/examples?select=23>

Blindly copy and paste will not help you so please make sure you understand the syntax and logic.

Thank You.