Mastering Python

الدرس #2

Python’s Language Basics

اساسيات لغة بايثون

By:

Hussam Hourani

V1.0 - NOV 2019

Agenda

- If Statement
- Loops
- Exception Handling

- الجمل الشرطية IF
- جمل التكرار
- معالجة الاستثناءات

If Conditional

Python supports the following logical conditions from mathematics:

- Equals: `a == b`
- Not Equals: `a != b`
- Less than: `a < b`
- Less than or equal to: `a <= b`
- Greater than: `a > b`
- Greater than or equal to: `a >= b`

If Condition

```
if test expression:
    statement(s)
```

the program evaluates the test expression and will execute statement(s) only if the text expression is True. If the text expression is False, the statement(s) is not executed.

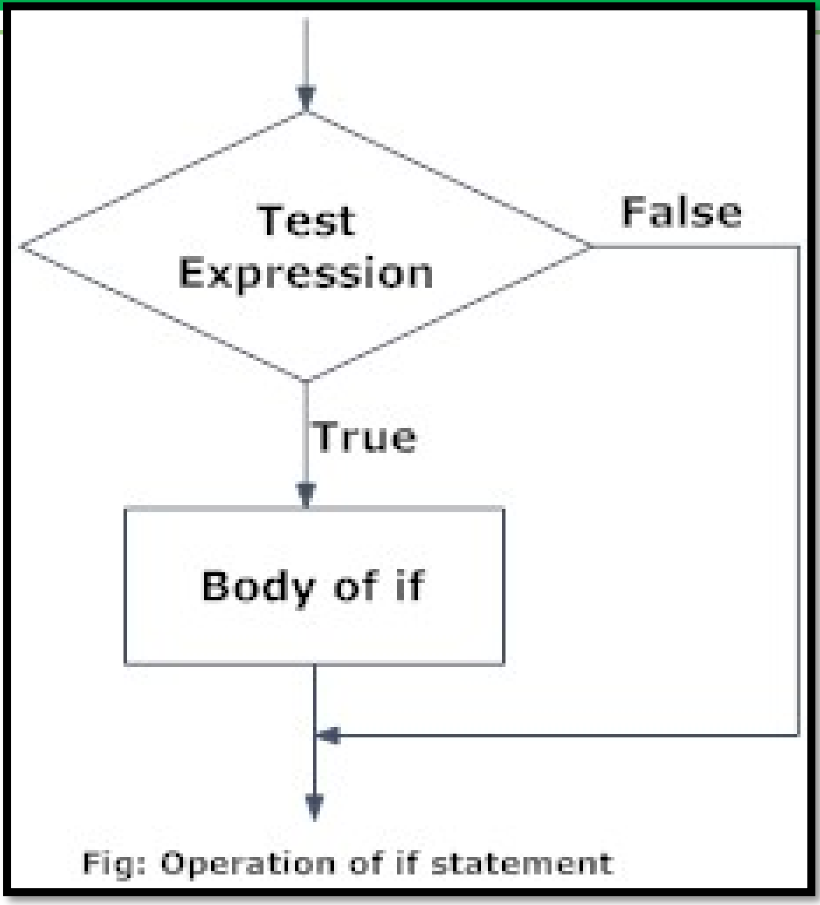


Fig: Operation of if statement

```
num = 3
if num > 0:
    print(num, "is a positive number.")
print("This is always printed.")
```

Output

```
3 is a positive number.
```

https://www.programiz.com/python-programming/if-elif-else

If Condition

```
if test expression:
    statement(s)
else
   statement(s)
```

the program evaluates the test expression and will execute statement(s) only if the text expression is True. If the text expression is False, the else statement(s) is executed.

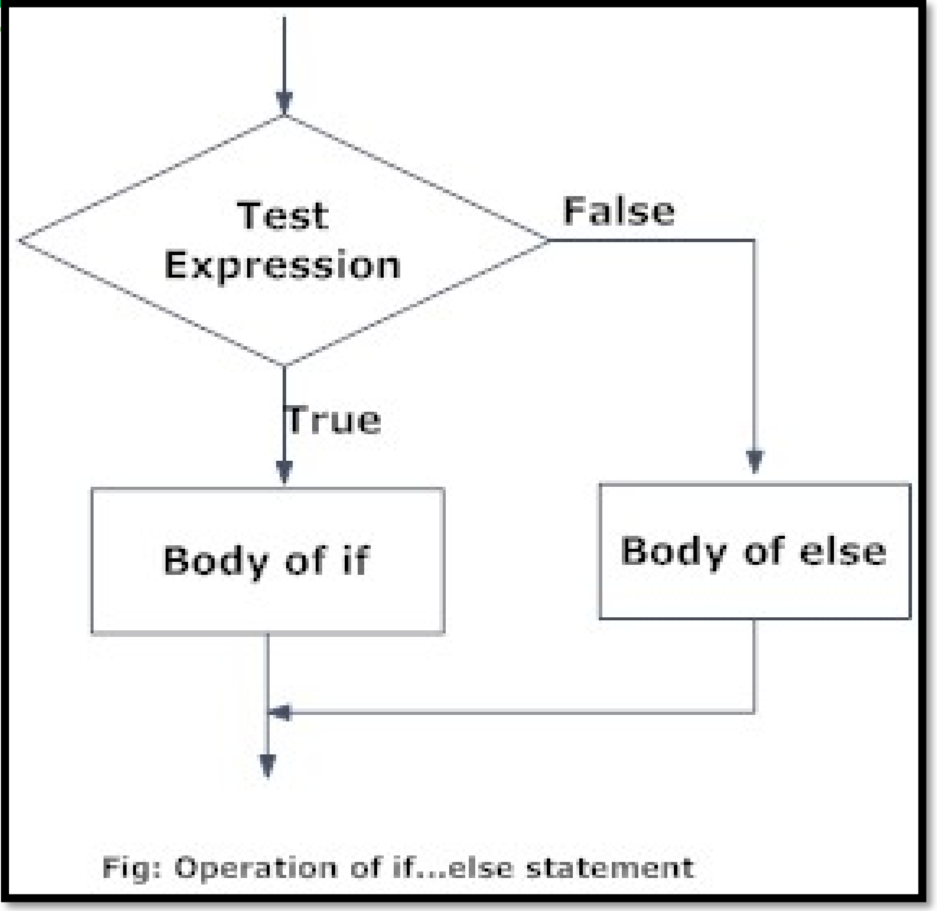


Fig: Operation of if...else statement

```
num = -5
if num >= 0:
     print("Positive or Zero")
else:
     print("Negative number")
```

Output

```
Negative number
```

https://www.programiz.com/python-programming/if-elif-else

If Condition

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

The elif is short for else if. It allows us to check for multiple expressions. If the condition for if is False, it checks the condition of the next elif block and so on. If all the conditions are False, body of else is executed. Only one block among the several if...elif...else blocks is executed according to the condition.

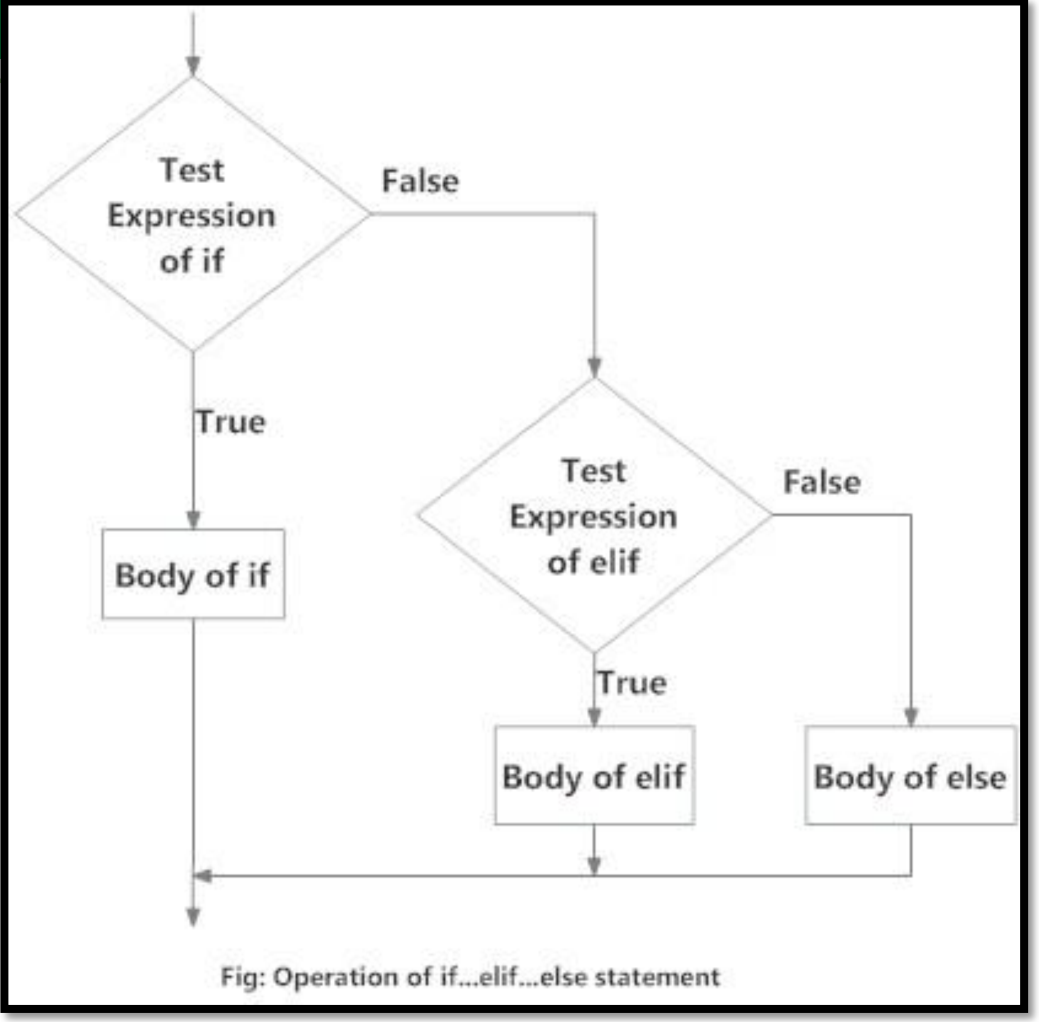


Fig: Operation of if...elif...else statement

```python
num = 3.4
if num > 0:
    print("Positive number")
elif num == 0:
    print("Zero")
else:
    print("Negative number")
```

Output

```
Positive number
```

https://www.programiz.com/python-programming/if-elif-else

If Condition

```
a, b = 1,10
if a > b:
    print("a > b")
elif a < b:
    print("a < b")
else:
    print("a = b")
```

Output

```
a < b
```

```
a, b = 1,10
max = a if (a > b) else b
print(max)
```

Output

```
10
```

```
if 'a' in ['b','c','a']:
    print("a in the list")
else: print("a not in the list")
```

Output

```
a in the list
```

If Condition

```
a=10
b=5
if a > b: print("a is greater than b")
```

Output

```
a is greater than b
```

```
a = 2
b = 330
print("A") if a > b else print("B")
```

Output

```
B
```

```
a = 1000
b = 330
print("A") if a > b else print("=") if a == b else print("B")
```

Output

```
1000
```

```
a = 200
b = 33
c = 500
if a > b and c > a:
  print("Both conditions are True")
```

Output

```
Both conditions are True
```

If Condition – Practice

1. Write a program to accept 2 numbers from the user and print them in order (ascending) using “if else” condition

2. Write a program to accept Name, Age from the user and :
   - If Age less than 18 to print “Under Age”
     - Ask the user to enter the School Average
       - If Average greater or equal 90 to print “Excellent Average”
       - Else if average greater or equal to 50 and less than 90 to print ” Passed”
       - Else print “Failed”
   - If Age Greater or equal to 18 to print “Adult”
     - Ask the user to enter his job title
     - print Age, Name and Job Tilt

Loops

A for loop is used for iterating over a sequence (that is either a rage, list, a tuple, a dictionary, a set, or a string).

```
for a in range(3):
    print(a)
```

Output

```
0
1
2
```

```
for a in range(1,6,2):
    print(a)
```

Output

```
1
3
5
```

```
for item in ['Jordan','US','UK']:
    print(item)
```

Output

```
Jordan
US
UK
```

```
for x in “name":
  print(x)
```

Output

```
n
a
N
e
```

Loops

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
  if x == "banana":
    break
```

Output

```
Apple
banana
```

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  if x == "banana":
    continue
  print(x)
```

Output

```
apple
cherry
```

```
for x in range(6):
  print(x)
else:
  print("Finally finished!")
```

Output

```
0
1
2
3
4
5
Finally finished!
```

Loops

```python
color = ["red", "green", "blue"]
fruits = ["apple", "banana", "cherry"]

for x in color :
  for y in fruits:
    print(x, y)
```

Output

```
red apple
red banana
red cherry
green apple
green banana
green cherry
blue apple
blue banana
blue cherry
```

```python
words = ['cat', 'window', 'defenestrate']
for w in words:
     print(w, len(w))
```

Output

```
cat 3
window 6
defenestrate 12
```

```python
l = ["eat","sleep","repeat"]
for count,e in enumerate(l):
    print (count,e)
```

Output

```
0 eat
1 sleep
2 repeat
```

Loops

```
i=0
while i<4:
    print(i)
    i+=1
```

Output

```
0
1
2
3
```

```
while True:
    a=input('>')
    if a=='exit':
        break
    print (a)
```

Output

```
>1
1

>exit
```

```
colors = ["red", "green", "blue", "purple"]
i = 0
while i < len(colors):
    print(colors[i])
    i += 1
```

Output

```
red
green
blue
purple
```

Loops– Practice

1. Write a program to print the following using for loop :

```
*
*
*
*
*
*
*
*
*
*
```

2. Write a program print the following using while loops :

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Loops– Practice

3. Write a program to print the following using nested loops :

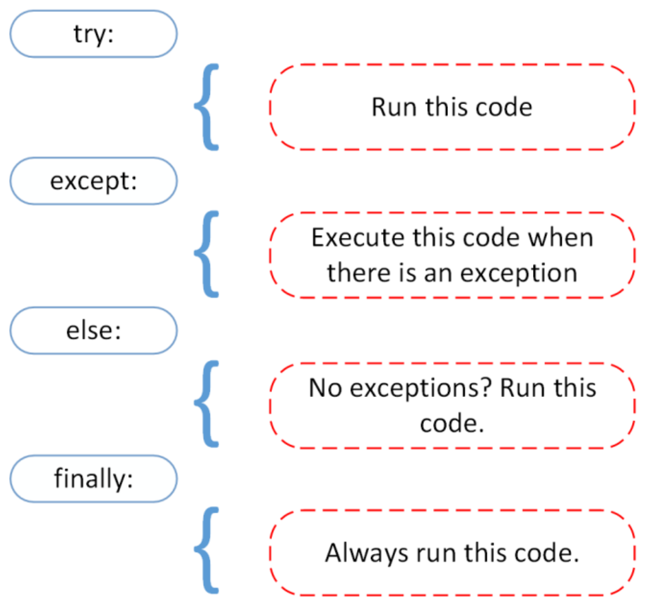
```
*
**
***
****
*****
******
*******
********
```

4. Write a program print the following using nested loops

```
       *
      **
     ***
    ****
   *****
  ******
 *******
********
```

Exceptions Handling

- An exception is an error that happens during execution of a program
- The try block lets you test a block of code for errors.
- The except block lets you handle the error.
- The finally block lets you execute code, regardless of the result of the try- and except blocks.


try:
Run this code
except:
Execute this code when there is an exception
else:
No exceptions? Run this code.
finally:
Always run this code.

https://realpython.com/python-exceptions/

Exceptions Handling

```python
while True:
    try:
        n = input("Please enter an integer: ")
        n = int(n)
        break
    except ValueError:
        print("No valid integer! Please try again ...")
print("Great, you successfully entered an integer!")
```

Output

```
Please enter an integer: e
No valid integer! Please try again ...

Please enter an integer: 1.2
No valid integer! Please try again ...

Please enter an integer: 3
Great, you successfully entered an integer!
```

```python
try:
    x = float(input("Your number: "))
    inverse = 1.0 / x
except ValueError:
    print("You should have given either an int or a float")
except ZeroDivisionError:
    print("Infinity")
finally:
    print("There may or may not have been an exception.")
```

Output

```
Your number: 0
Infinity
There may or may not have been an exception.
```

Master in Software Engineering

Hussam Hourani has over 25 years of Organizations Transformation, VROs, PMO, Large Scale and Enterprise Programs Global Delivery, Leadership, Business Development and Management Consulting. His client experience is wide ranging across many sectors but focuses on Performance Enhancement, Transformation, Enterprise Program Management, Artificial Intelligence and Data Science.