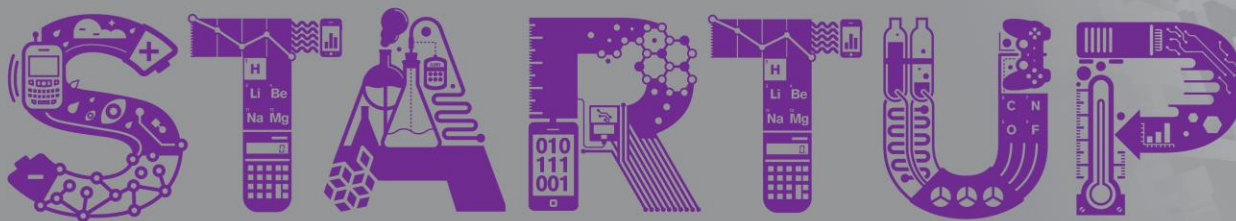


Bloomberg



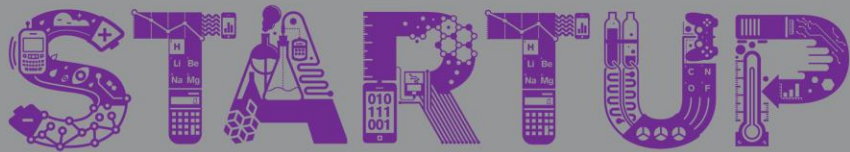
Copyright 2017 Bloomberg L.P.

Licensed under Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0)

Non-commercial use only. Modifications must not be distributed.

Please see <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Bloomberg



Python Lesson 5

Boolean Logic



Review: While Loops

```
counter = 0
while counter < 10:
    counter = counter + 1
    print(counter)
```

1
2
3
4
5
6
7
8
9
10



Logical Operators: Introduction

- To make a peanut butter and jelly sandwich, you need both peanut butter **and** jelly.
- You can use either whole wheat bread **or** white bread.
- A peanut butter and jelly sandwich is **not** a chicken salad sandwich.



Logic Operators: Truth Table

a	b	a and b
True	True	True
True	False	False
False	True	False
False	False	False

a	b	a or b
True	True	True
True	False	True
False	True	True
False	False	False

a	not a
True	False
False	True



Logic Operators: Activity

- What does the following code output?

```
x = 5

print(x == 5 and x == 2)
print(x == 5 and x != 2)
print(x == 5 or x == 2)
print(x == 5 or x != 2)
print(not x == 5)
print(not x == 2)
```

```
False
True
True
True
False
True
```



Conditionals: Elif Statements

if *boolean expression*:

statements

elif *boolean expression*:

statements

else:

statements

Example:

```
x = int(input("Enter a number: "))
if x > 0:
    print(x, "is positive")
elif x < 0:
    print(x, "is negative")
else:
    print(x, "is 0")
```

```
Enter a number: 5
5 is positive
```

```
Enter a number: -4
-4 is negative
```

```
Enter a number: 0
0 is 0
```



Anatomy of Elif Statements

elif tells the program you want to evaluate the 2nd condition if the 1st condition is **False**.

The statements must all be indented the same amount.

A condition that evaluates to either **True** or **False**.

```
if x > 0:
    print(x, "is positive")
elif x < 0:
    print(x, "is negative")
else:
    print(x, "is 0")
```

: tells the program you are starting the statements block.

The statements get executed if the 2nd condition is **True**.

else tells the program you want to execute the statements if both the 1st and the 2nd conditions are **False**.



Conditionals: Practice

What does the following code output?

```
x = 10
y = 20
z = 30

print(x < y or y > z)
print(x > 5 or y > 15)
print(x > 5 or x == 10)
print(x < y and y > z)
print(x < y and y <= z)
print(x > 5 and y > 15)
print(not z == 30)
print(not y > 25)
```



Conditionals: Practice

Modify the high low guessing game from Lesson 3 to use if-elif-else.

Write a rock paper scissor game.



Python Explorer Game

```
def handle_turn(direction):  
    if direction == 'N':  
        print("Walking towards north")  
    elif direction == 'S':  
        print("Walking towards south")  
    elif direction == 'W':  
        print("Walking towards west")  
    elif direction == 'E':  
        print("Walking towards east")  
    else:  
        print("Looking around")  
  
user_input = input("Please enter a direction: ").upper()  
while user_input != "QUIT":  
    handle_turn(user_input)  
    user_input = input("Please enter a direction: ").upper()
```



Python Explorer Game

```
Please enter a direction: n
Walking towards north
Please enter a direction: w
Walking towards west
Please enter a direction: look
Looking around
Please enter a direction: s
Walking towards south
Please enter a direction: e
Walking towards east
Please enter a direction: quit
```



Recap

- Logical operators: **and**, **or**, **not**
- To write conditional statements,
 - if boolean expression:*
 - statements*
 - elif boolean expression:*
 - statements*
 - else:*
 - statements*