Tutorial | WfWW basic programming workshop 3

November 30, 2018

1 Tutorial by djangogirls

Check out more of Python at tutorial.djangogirls.org

2 Comments

Comments begin with #. You can write whatever you want after the # and Python will ignore it. Comments can make your code easier for other people to understand.

3 Print a message

```
In [1]: print("Hello World")

        print("Hello, " + "How are you?")
```

```
Hello World
```

4 Perform calculations

```
In [2]: x = 5+2
        print(x)
```

```
7
```

4.1 You can also write the calculations into the same line as the print.

Try a few more with +, <, >, %, *, ==, and != . What do those operators do?

- x > y means: x is greater than y
- x < y means: x is less than y
- x <= y means: x is less than or equal to y
- x >= y means: x is greater than or equal to y

```
In [4]: print(2 + 4)
        print(7 > 2)
        print(9 < 1)
        print(7 % 2)
        print(2 * 4)
        print(3 == 3)
        print(5 != 2)
```

```
6
True
False
1
8
True
True
Hello, How are you?
```

5 Taking inputs

```
In [2]: usr_input = input('Enter anything')
        print(usr_input * 2)
```

```
Enter anythinghello
hello hello
```

6 Errors

6.1 Don't be afraid of errors!

What's wrong with the code below?

```
In [ ]: print("Hello " + 4) # How would you fix this?
```

7 Types

```
In [5]: print("Hello " + str(4))
```

```
Hello 4
```

8 If-Else Conditions

```
In [8]: # -------
        # Play around with the numbers!
        a=5
```

```
b=3
# -------

if a>b:
    print(str(a) + " is a larger number than " + str(b))
else:
    print(str(b) + " is a larger number than " + str(a))
```

```
5 is a larger number than 3
```

What if you have more than one if statement?
Use elif, and don't forget the final else statement

```
In [14]: # -------
         # Play around with the numbers!
         # How can you change these variables so that the else statement gets printed?
         a=5
         b=3
         # -------

         if a>b:
             print(str(a) + " is a larger number than " + str(b))
         elif a<b:
             print(str(a) + " is a smaller number than " + str(b))
         else:
             print(str(a) + " is equal to " + str(b))
```

```
5 is a larger number than 3
```

9 Indentation

Python is sensitive to indentations. You make an indentation with the 'tab' button on your keyboard.

This is ok:

```
In [9]: if 2>1:
            print('2 is bigger than 1')
            print('Yay!')
```

```
Yay!
```

This will lead to an error:

```
In [11]: if 2>1:
             print('2 is bigger than 1')
                 print('Yay!') # This has an extra indentation
```

```
  File "<ipython-input-11-f34ea251dde9>", line 3
    print('Yay!')
    ^
IndentationError: unexpected indent
```

10 Lists

Lists hold elements (numbers, strings, variables) where you can easily access them in order.

```
In [16]: this_is_a_list = ['hi','i','am','a','list','of','words']
         print(this_is_a_list)

         this_is_another_list = [1,2,3,4,5]
         print(this_is_another_list)
```

```
['hi', 'i', 'am', 'a', 'list', 'of', 'words']
[1, 2, 3, 4, 5]
```

11 Functions

Functions are commands you can use to do something.
print is a function to print something len is a function to find the length of a list

```
In [21]: this_is_a_list = ['hi','i','am','a','list','of','words']
         print(len(this_is_a_list)) # There are 7 elements (words) in that list
```

```
7
```

12 Getting elements in a list

You can get the elements from the list using brackets [] and the index (location number) of the element

13 Zero Index

Python uses zero-based numbering. So all counting (and indexing) starts as 0,1,2...

```
In [19]: a = [1,2,3,4,5]

         # print the FIRST element of the list
         print(a[0])
         print(a[2])
```

```
1
3
```

14 For Loops

What if you want to repeat some operation? You don't have to copy and paste the code over and over again – use loops!

```
In [22]: lower_bound = 0
         upper_bound = 4
         for idx in range(lower_bound,upper_bound):
             print(idx)
```

```
0
1
2
3
```