Week1_Code_[.Your first Program.]

December 14, 2018

0.1 Writing Your First Program

To run python in the termianl use:

```
$> python3
```

```
In [3]: """
        **Name:** Writing your first program
        **Author:** Alexander Bjørnsrud
        **Created:** 2018-09-21
        **ID:** xxxxxxxx

        The purpose of this Notebook is to show the students what they can do with their first
        """

        """The print statement writes output from the program."""

        print("Hello World")
```

```
Hello World
```

```
In [1]: """If you use quotes in the text they have to be escaped with the \ character"""

        text = "Editing"

        print("Hello \"You\"...")
        print("Testing Python!")
        print(text+" the notebook!")
```

```
Hello "You"...
Testing Python!
Editing the notebook!
```

```
In [8]: """We can use special characters to show a line break or a tab"""

        """The \n key will move the text to a new line"""
        print("Line 1\nLine 2")

        """\r Is the return character. Note how Line1 is not printed. The return character rem
        print("Line1\rLine2")

        """\t Is the tab character"""
        print("Start\tTabbed Text")
```

```
Line 1
Line 2
Line2
Start        Tabbed Text
```

```
In [11]: """We can use the + sign to add strings together"""

         print("String 1"+" | "+"String 2")
```

```
String 1 | String 2
```

```
In [14]: """
         A Variable is used to hold a value.
         Review the following prices:
         """

         bread = 25
         butter = 30
         bacon = 25
         cheese = 50

         print("A Bread costs:", bread)
         print("If i buy bread and cheese they cost:", cheese+bread)
```

```
A Bread costs: 25
If i buy bread and cheese they cost: 75
```

```
In [1]: """
        We can also assign variables on a single line.
        """
        bread,butter,bacon,cheese = 25,30,25,50

        print("A Bread costs:", bread)
        print("If i buy bread and cheese they cost:", cheese+bread)
```

```
A Bread costs: 25
If i buy bread and cheese they cost: 75
```

```
In [17]: """
         Variables are similar to how x and y work in mathematics.
         """
         x=0
         y=0
         b=0

         print(x+y-b==5) #0 is not equal to 5.
```

```
False
```

```
In [18]: x=7
         y=1
         b=3

         print(x+y-b==5)# 5 is equal to 5.
```

```
True
```

```
In [20]: """
         Variables can also hold strings.
         They operate in the same manner as strings.
         This means that we can combine them like earlier.
         """
         name = "Alexander"
         message = "Hello "+name+"!"

         print(message)
```

```
Hello Alexander!
```

```
In [43]: """Multi initalizing of variables"""

         one,two,tree = 1,2,3 #assigns 1,2 and 3 to the variables before the = sign
         print(one,two,tree)

         print("-"*5)#------------------------------------------

         five=one_pluss_four=two_point_five_times_two = 5# We can assing all of the variables
         print(five,one_pluss_four,two_point_five_times_two)
```

```
1 2 3
-----
5 5 5
```

```
In [44]: """Some names we can't use!"""
         import keyword

         #Dont Panic!
         [print(i,n,z)for i,n,z in zip(keyword.kwlist[::3],keyword.kwlist[1::3],keyword.kwlist
         #its just there to show the words.

         print("-"*20)#-----------------------------------------

         while_True = True #will work dough
         print(while_True)
```

```
False None True
and as assert
async await break
class continue def
del elif else
except finally for
from global if
import in is
lambda nonlocal not
or pass raise
return try while
--------------------
True
```

```
In [2]: """
        We can use the input function to get input from the users
        """

        inp = input("Tell me your name!:")
        print("Hello "+inp+"!")
```

```
Tell me your name!:Alexander
Hello Alexander!
```

```
In [3]: """
        Errors!
        =======
        .. note:: We can use Exceptions to raise errors in out code.
        .. warning:: Raising an error will halt the execution.
        """

        raise Exception("Epic Error!")


        ---------------------------------------------------------------------------
```

```
    Exception                                 Traceback (most recent call last)

    <ipython-input-3-bb33b18183bd> in <module>()
      3 """
      4
----> 5 raise Exception("Epic Error!")


    Exception: Epic Error!
```

```
In [1]: """
        Python will raise errors if we do something unexpected.
        """

        print("Hello!)
```

```
      File "<ipython-input-1-8a6c106e30ba>", line 5
    print("Hello!)
                  ^
SyntaxError: EOL while scanning string literal
```

```
In [5]: """
        Python will raise errors if we do something unexpected.
        """
        print("1"+1)
```

```
    ---------------------------------------------------------------------------

    TypeError                                 Traceback (most recent call last)

    <ipython-input-5-fe78f2a44679> in <module>()
----> 1 print("1"+1)


    TypeError: can only concatenate str (not "int") to str
```