

Python Crash Course Review (Part I)

1. Using Python as a calculator

There are different ways to import a library, which one will work here?

```
from math import sqrt, pi, exp, cos
#from math import *
import math
import math as m
```

```
#print(sqrt(4), pi, exp(1), 3**2)
#print(math.sqrt(4))
print(m.exp(1))
```

```
#help(math)
#help(sqrt)
help(math.sqrt)
```

Print 'help' on the function 'det' of the 'numpy' library 'linalg'.

```
x = 0
print(sqrt(x)+pi*cos(x)/exp(x))
# note that some python will interpret 1/5 as zero
print(1/5,1/5.,-1/5)
```

Calculate something.

2. Data types: integers, floats, complex numbers, strings, Boolean values

```
x = 1e-3
s = 'eh'
y = range(10)
z = True
```

```
print(x, str(x))
print(type(x), type(str(x)))
print(s, type(s), len(s))
print(y, type(y), len(y))
print(z, type(z))
```

Write a string 'mystring' "Hello World", a tuple t=(1,2), and a list with 3 elements, a float (e.g. 0.1), an integer (e.g. 10), and the string. Find the type of 'mystring' and of 'mystring[0]' .

3. Control Statements ("for", "if", "else", "while", "pass") control the flow of a code block.

Python uses indentation to mark blocks of code: 4 spaces not a tab

<https://docs.python.org/3/tutorial/controlflow.html>

https://docs.python.org/3/reference/compound_stmts.html

3.a) for-loop: iterate over the elements of a sequence.

```
for item in [x,s,y]:
    print(item, type(item))          # 4 spaces
```

```
for i in range(10):
    print(i)
```

```
for i in ['x','s','y']:
    #print(i, eval(i))
    print(eval(i), type(eval(i)))
```

3.b) if-else: conditional execution of code blocks

```
a = 10
b = 20
if a < b:
    print(a)
else:
    pass    # do nothing
```

3.c) Comparison expressions

Try other comparison expressions, replace 'a < b' by 'a == b', 'a != b', '(a==10) and (b==10)'

<https://docs.python.org/3/reference/expressions.html#comparisons>

3.d) while loop: repeated execution as long as an expression is true (input condition)

```
n = 3; s = 0; counter = 1
while counter <= n:
    s = s + counter
    counter += 1
print(s)
```

4. Formatted output

```
print(f'Sum of integer numbers from 1 until {n:d} is {s:1.2f}.')
print("Sum of integer numbers from 1 until %d is %1.2f." % (n,s)) # old way
```

<https://docs.python.org/3/tutorial/inputoutput.html>

5. List comprehension (makes for-loops shorter)

```
for i in range(3):
    print((i**2))
```

```
abc = [ (i**2) for i in range(3) ]
print(abc)
```

Create a list with numbers $j=[1,2,\dots,10]$, compute the square root of these numbers using control structures (for/while), and print the result (using formatted output).

5. Check out the following python documentation.

The official Python webpage has everything you need to know. How to install python, documentation (a python tutorial, the library reference, and more [15min]):

<https://python.org/>

<https://docs.python.org>

Check out these two highly recommended python tutorials [40min]:

<https://www.tutorialspoint.com/python/>

https://en.wikibooks.org/wiki/Python_Programming

Useful tips (dir, type, help, how to import libraries) [15min]:

<https://realpython.com/python-beginner-tips/>

<https://learnpythonthehardway.org/book/intro.html>