



COLLECTION OF BASIC PYTHON CODING COMMANDS

Here you can find a collection of the basic Python coding commands. May be a useful reference, especially if you are starting to code.

NOTE: This is not a tutorial; this contains only a list of some basic code commands that can be useful to have close by when you start coding. Before, you may need to know the basics of Python .i.e types of objects, along with basic coding rules. There are lots of very good tutorials and books out there. Besides, the Python web is an excellent help (<https://www.python.org/about/help/>)

Colour code:

Sections headers

Comments

What I write in python (either in the console or in an IDE)

Build-in functions in Python

What appears in the python console after I run my code

Contents

#DECLARE A VARIABLE	2
#CONDITIONS.....	2
#ITERATIONS (for)	3
#WHILE LOOP	3
#DEFINE A FUNCTION	4
### USEFUL BUILD-IN COMMANDS AND FUNCTIONS ###	4
### THINGS TO REMEMBER ###	5
ABOUT	5



#DECLARE A VARIABLE

```
num=3 #I want to call the variable num and assign the number 3
print(num)
3
```

```
oper=3+7 # I want to call my variable oper
print(oper)
10
print("the total is", oper) #print string and number together
the total is 10
```

delete variables

```
del(num)
num
NameError: name 'num' is not defined #now it yields an error because num is not defined
```

#CONDITIONS

```
mnum=25
if mnum<45:
    bb=8 #this is the action I want to execute when the if condition is met.
    I want my new variable bb to take the value 8 when mnum < 45
else:
    bb=333333 #this is the action I want to execute when the if condition is met.
    I want my new variable bb to take the value 333333 when mnum >= 45
print(bb)
8
```

```
unon=1
dosn=2
if unon<dosn:
    say= "unon is less than dosn"
elif unon == dosn:
    say="unon is the same than dosn"
else:
    say= "unon is greater than dosn"
print(say)
unon is less than dosn
```

#combitations



```
if unon < 3 and dosn < 25:  
    print("both ok")  
both ok
```

```
if unon < 3 or dosn > 25:  
    print("some ok")  
some ok
```

#ITERATIONS (for)

```
items=[1,3,5,7]  
totmas= 0  
unos=0  
for unos in items:  
    totmas=totmas+unos  
print(totmas)  
16
```

```
print(sum(items)) #easier way, using a defined function in item  
16
```

```
print(len(items)) #with this function we count the number of elements  
4
```

```
for numb in range(1,6):  
    print(numb)  
1  
2  
3  
4  
5
```

```
totmas= 0  
for numb in range (1,10):  
    if (numb ==7): break #stops the loop when this condition is met  
    totmas = totmas + numb
```

```
print(numb)  
7  
print(totmas)  
21
```

#WHILE LOOP

```
x=1
```



```
while (x<5):
    print(x)
    x=x+1
1
2
3
4
```

#DEFINE A FUNCTION

```
def fun1 ():
    print("Hello, I am fun1")
```

```
def powerDEF (numb, n=1): #this is a definition of the power function
    result=1
    for i in range(n):
        result=result*numb
    return result
```

```
powerDEF (2,4) #this is 2^4 :)
16
```

USEFUL BUILD-IN COMMANDS AND FUNCTIONS

- Function **enumerate** # count elements in a list, loop, etc
mylist=[1,2,3,4,55,66,777,8888]
for i,d in enumerate(mylist):
 print(i,d)

```
0 1
1 2
2 3
3 4
4 55
5 66
6 777
7 8888
```



THINGS TO REMEMBER

- In a list, the first element is 0. Then you'll have the 1, and so on
- The function `input` will input in python a file with a code to run

ABOUT

This document was created on 19 April 2020

I will update the document and share it in my blog: www.ekonowsys.net

Any bug, problem, etc, please email me: alicialedo@gmail.com

Thank you very much, and happy coding! ;)