

# THE ULTIMATE PYTHON CRASH COURSE - FULL PYTHON 101 BEGINNER TUTORIAL (CODE WITH HARRY)

Video Link: <https://youtu.be/vxHUFFiT0OI> (<https://youtu.be/vxHUFFiT0OI>)

## Comment and Escape

```
In [1]: #
        ...
        multiline comment
        print('ctrl+d for multiple cursors (or alt ?)')
        print('ctrl+d for multiple cursors (or alt ?)')
        ...
        print('it\'s 18th may')
        print('google escape characters to know more')
```

```
it's 18th may
google escape characters to know more
```

## Variables and Datatypes

*python* can change the datatypes.

```
In [2]: a, b, c, d = 45, 1.25, 1.2j, '3rd variable'
        print('datatype', type(a), type(b), type(c))
        d, e = False, None
        print(type(d), type(e))
```

```
datatype <class 'int'> <class 'float'> <class 'complex'>
<class 'bool'> <class 'NoneType'>
```

## Strings

Elements of a string are immutable.

```
In [3]: a = 'abcdefgh'
        print(len(a), a[2:5])
        #a[1] = 'c' elements of a string are immutable
        print(a.replace('def', 'DEF'))
        print(a.capitalize(), a.upper())
```

```
8 cde
abcDEFgh
Abcdefgh ABCDEFGH
```

## if, elif and else

```
In [4]: age = int(input('input your age for driving: '))
# input function (str by default)
if age>18:
    print('you can drive')
elif age==18:
    print('make your liscence and drive')
else:
    print('you can\'t drive')

num = int(input('guess an integer between 0-50 (win for 5 numbers): '))
if num==9:
    print('you won')
elif num==14:
    print('you won')
elif num==26:
    print('you won')
elif num==30:
    print('you won')
elif num==44:
    print('you won')
else:
    print('you lose')
```

```
input your age for driving: 21
you can drive
guess an integer between 0-50 (win for 5 numbers): 54
you lose
```

## Loops

### while loop

```
In [5]: i = 3
while(i<10):
    print(i)
    i = i+1
```

```
3
4
5
6
7
8
9
```

### for loop

```
In [6]: for i in range(2,10):
    print(i)
    if i==4:
        continue
    if i==6:
        break
```

```
2
3
4
5
6
```

## Functions

```
In [7]: def plus1(number):  
        return number +1  
        print(plus1(10))  
        print(abs(-6.98)) # built in python
```

```
11  
6.98
```

## Modules

```
In [8]: import math  
        print(math.ceil(5.36)) # greatest integer function  
        print(math.factorial(4))  
        print('google python built in modules')  
        print('install the modules (django, sklearn) (44:00)')
```

```
6  
24  
google python built in modules  
install the modules (django, sklearn) (44:00)
```

## List

```
In [9]: list1 = [12,1,0,8,'skp',7,3]  
        print(type(list1))  
        list1.append('added1')  
        for l in list1:  
            print(l)  
        list1[2] = 5  
        print(list1[1:5])
```

```
<class 'list'>  
12  
1  
0  
8  
skp  
7  
3  
added1  
[1, 5, 8, 'skp']
```

## Tuple

Tuples are immutable unlike lists.

```
In [10]: tup1 = (1,3,5,'g',5.1)  
         print(tup1, type(tup1))  
         print(tup1[1:4])
```

```
(1, 3, 5, 'g', 5.1) <class 'tuple'>  
(3, 5, 'g')
```

## Set

```
In [11]: set1 = {2,3,1.5,2,0.5}
print(set1)
set1.add(10)
set1.add(0.5) # already present, not added
print(set1)
set2 = {2,7,6,2.5,0,1,3}
print(set2)
print('union', set1.union(set2))
print('intersection', set1.intersection(set2))
print('check out other operations on sets')
```

```
{0.5, 1.5, 2, 3}
{0.5, 1.5, 2, 3, 10}
{0, 1, 2, 2.5, 3, 6, 7}
union {0.5, 1.5, 2, 3, 0, 1, 2.5, 6, 7, 10}
intersection {2, 3}
check out other operations on sets
```

## Dictionary

```
In [12]: dict1 = {
    'v': 'violet', 'i': 'indigo', 'b': 'blue',
    'g': 'green', 'y': 'yellow', 'o': 'orange', 'r': 'red'
}
print(type(dict1))
print(dict1['b'])
dict1['w'] = 'white' # adding an element
print(dict1)
print(len(dict1))
```

```
<class 'dict'>
blue
{'v': 'violet', 'i': 'indigo', 'b': 'blue', 'g': 'green', 'y': 'yellow', 'o': 'orange',
'r': 'red', 'w': 'white'}
8
```

## Exception Handling

```
In [13]: try:
    a1 = t.upper()
except Exception as e:
    print(e)
print('the error is skipped and the code is kept running')
```

```
name 't' is not defined
the error is skipped and the code is kept running
```

## File Handling

```
In [14]: file1 = open('open file 1 (cwh).txt', 'w') # 'w' for write mode
file1.write('it\'s the first file created by python code.')
file1.close()
file1 = open("open file 1 (cwh).txt", 'r')
read1 = file1.read()
print(read1)
file1 = open('open file 1 (cwh).txt', 'a')
file1.write('try to do more...') # check
```

it's the first file created by python code.

Out[14]: 17

## Classes and Objects

```
In [15]: class data1:
# constructor which will get executed everytime an object is created
def __init__(self, topic, starting, name) -> None:
    self.topic = topic
    self.starting = starting
    self.name = name
    print(f'data created: {topic} in {starting} by {name}')

def data1display(self):
    print(f'{self.topic} was explained in the year {self.starting} by {self.name}')

bbr = data1('black body radiation', '1900', 'Max Planck')
qm = data1('quantum mechanics', '1925', 'Heisenberg and Schrodinger')
str = data1('special relativity', '1905', 'Einstein')
peeff = data1('photo-electric effect', '1905', 'Einstein')
qm.data1display()
```

data created: black body radiation in 1900 by Max Planck  
data created: quantum mechanics in 1925 by Heisenberg and Schrodinger  
data created: special relativity in 1905 by Einstein  
data created: photo-electric effect in 1905 by Einstein  
quantum mechanics was explained in the year 1925 by Heisenberg and Schrodinger

In [ ]: