



# Python Crash Course

Please note, this is not meant to be a comprehensive overview of Python or programming in general, if you have no programming experience, you should probably take my other course: [Complete Python Bootcamp](#) instead.

**This notebook is just a code reference for the videos, no written explanations here**

This notebook will just go through the basic topics in order:

- Data types
  - Numbers
  - Strings
  - Printing
  - Lists
  - Dictionaries
  - Booleans
  - Tuples
  - Sets
- Comparison Operators
- if, elif, else Statements
- for Loops
- while Loops
- range()
- list comprehension
- functions
- lambda expressions
- map and filter
- methods

## Data types

### Numbers

```
In [1]: 1 + 1
Out[1]: 2

In [2]: 1 + 9
Out[2]: 3

In [3]: 1 / 2
Out[3]: 0.5

In [4]: 2 ** 4
Out[4]: 16

In [5]: 4 % 2
Out[5]: 0

In [6]: 5 % 2
Out[6]: 1

In [7]: (2 + 3) * (5 + 5)
Out[7]: 50
```

### Variable Assignment

```
In [8]: # Can not start with number or special characters
name_of_var = 2

In [9]: x = 2
        y = 3

In [10]: z = x + y

In [11]: z
Out[11]: 5
```

### Strings

```
In [12]: 'single quotes'
Out[12]: 'single quotes'

In [13]: "double quotes"
Out[13]: 'double quotes'

In [14]: " wrap lot's of other quotes"
Out[14]: " wrap lot's of other quotes"

In [15]: x = 'hello'

In [16]: x
Out[16]: 'hello'

In [17]: print(x)
hello

In [18]: num = 12
        name = 'Sam'

In [19]: print('My number is: {one}, and my name is: {two}'.format(one=num, two=name))
My number is: 12, and my name is: Sam

In [20]: print('My number is: {}, and my name is: {}'.format(num, name))
My number is: 12, and my name is: Sam
```

### Lists

```
In [21]: [1, 2, 3]
Out[21]: [1, 2, 3]

In [22]: ['hi', 1, [1, 2]]
Out[22]: ['hi', 1, [1, 2]]

In [23]: my_list = ['a', 'b', 'c']

In [24]: my_list.append('d')

In [25]: my_list
Out[25]: ['a', 'b', 'c', 'd']

In [26]: my_list[0]
Out[26]: 'a'

In [27]: my_list[1]
Out[27]: 'b'

In [28]: my_list[1:]
Out[28]: ['b', 'c', 'd']

In [29]: my_list[:1]
Out[29]: ['a']

In [30]: my_list[0] = 'NEW'

In [31]: my_list
Out[31]: ['NEW', 'b', 'c', 'd']

In [32]: nest = [1, 2, 3, [4, 5, ['target']]]

In [33]: nest[3]
Out[33]: [4, 5, ['target']]

In [34]: nest[3][2]
Out[34]: ['target']

In [35]: nest[3][2][0]
Out[35]: 'target'

In [36]: d = {'key1': 'item1', 'key2': 'item2'}

In [37]: d
Out[37]: {'key1': 'item1', 'key2': 'item2'}

In [38]: d['key1']
Out[38]: 'item1'

In [39]: True
Out[39]: True

In [40]: False
Out[40]: False

In [41]: t = (1, 2, 3)

In [42]: t[0]
Out[42]: 1

In [43]: # t[0] = 'NEW'
        # 'tuple' object does not support item assignment

In [44]: {1, 2, 3}
Out[44]: {1, 2, 3}

In [45]: {1, 2, 3, 1, 2, 1, 2, 3, 3, 3, 2, 2, 2, 1, 1, 2}
Out[45]: {1, 2, 3}

In [46]: 1 > 2
Out[46]: False

In [47]: 1 < 2
Out[47]: True

In [48]: 1 >= 1
Out[48]: True

In [49]: 1 <= 4
Out[49]: True

In [50]: 1 == 1
Out[50]: True

In [51]: 'hi' == 'bye'
Out[51]: False

In [52]: (1 > 2) and (2 < 3)
Out[52]: False

In [53]: (1 > 2) or (2 < 3)
Out[53]: True

In [54]: (1 == 2) or (2 == 3) or (4 == 4)
Out[54]: True

In [55]: if 1 < 2:
        print('Yep!')
        Yep!

In [56]: if 1 < 2:
        print('Yep!')
        Yep!

In [57]: if 1 < 2:
        print('First')
        else:
            print('Last')
        First

In [58]: if 1 > 2:
        print('First')
        else:
            print('Last')
        Last

In [59]: if 1 == 2:
        print('First')
        elif 3 == 3:
            print('middle')
        else:
            print('Last')
        middle

In [60]: seq = [1, 2, 3, 4, 5]

In [61]: for item in seq:
        print(item)
        1
        2
        3
        4
        5

In [62]: for item in seq:
        print('Yep')
        Yep
        Yep
        Yep
        Yep

In [63]: for jelly in seq:
        print(jelly*jelly)
        2
        4
        6
        8
        10

In [64]: i = 1
        while i < 5:
            print('i is: {}'.format(i))
            i = i+1
        i is: 1
        i is: 2
        i is: 3
        i is: 4

In [65]: range(5)
Out[65]: range(0, 5)

In [66]: for i in range(5):
        print(i)
        0
        1
        2
        3
        4

In [67]: list(range(5))
Out[67]: [0, 1, 2, 3, 4]

In [68]: x = [1, 2, 3, 4]

In [69]: out = []
        for item in x:
            out.append(item**2)
        print(out)
        [1, 4, 9, 16]

In [70]: [item**2 for item in x]
Out[70]: [1, 4, 9, 16]

In [71]: def my_func(param1='default'):
        """
        Docstring goes here.
        """
        print(param1)

In [72]: my_func
Out[72]: <function __main__.my_func(param1='default')>

In [73]: my_func()
default

In [74]: my_func('new param')
new param

In [75]: my_func(param='new param')
new param

In [76]: def square(x):
        return x**2

In [77]: out = square(2)

In [78]: print(out)
4

In [79]: def times2(var):
        return var*2

In [80]: times2(2)
Out[80]: 4

In [81]: lambda var: var*2
Out[81]: <function __main__.<lambda>(var)>

In [82]: seq = [1, 2, 3, 4, 5]

In [83]: map(times2, seq)
Out[83]: <map at 0x256f76a85b0>

In [84]: list(map(times2, seq))
Out[84]: [2, 4, 6, 8, 10]

In [85]: list(map(lambda var: var*2, seq))
Out[85]: [2, 4, 6, 8, 10]

In [86]: filter(lambda item: item%2 == 0, seq)
Out[86]: <filter at 0x256f76ac7c0>

In [87]: list(filter(lambda item: item%2 == 0, seq))
Out[87]: [2, 4]

In [88]: st = 'hello my name is Sam'

In [89]: st.lower()
Out[89]: 'hello my name is sam'

In [90]: st.upper()
Out[90]: 'HELLO MY NAME IS SAM'

In [91]: st.split()
Out[91]: ['hello', 'my', 'name', 'is', 'Sam']

In [92]: tweet = 'Go Sports! #Sports'

In [93]: tweet.split('#')
Out[93]: ['Go Sports!', '']

In [94]: tweet.split('#')[1]
Out[94]: 'Sports'

In [95]: d

In [96]: {'key1': 'item1', 'key2': 'item2'}

In [97]: d.keys()
Out[97]: dict_keys(['key1', 'key2'])

In [98]: dict_keys(['key1', 'key2'])

In [99]: d.items()
Out[99]: dict_items([('key1', 'item1'), ('key2', 'item2')])

In [100]: lst = [1, 2, 3]

In [101]: lst.pop()
Out[101]: 3

In [102]: lst
Out[102]: [1, 2]

In [103]: 'x' in [1, 2, 3]
Out[103]: False

In [104]: 'x' in ['x', 'y', 'z']
Out[104]: True
```

## Great Job!



# Python Crash Course Exercises

This is an optional exercise to test your understanding of Python Basics. If you find this extremely challenging, then you probably are not ready for the rest of this course yet and don't have enough programming experience to continue. I would suggest you take another course more geared towards complete beginners, such as [Complete Python Bootcamp](#)

## Exercises

Answer the questions or complete the tasks outlined in bold below, use the specific method described if applicable.

**What is 7 to the power of 4?**

```
In [1]: 7**4
```

```
Out[1]: 2401
```

**Split this string:**

```
s = "Hi there Sam!"
```

**into a list.**

```
In [2]: s = "Hi there Sam!"
```

```
In [3]: s.split()
```

```
Out[3]: ['Hi', 'there', 'Sam!']
```

**Given the variables:**

```
planet = "Earth"  
diameter = 12742
```

**Use .format() to print the following string:**

```
The diameter of Earth is 12742 kilometers.
```

```
In [4]: planet = "Earth"  
diameter = 12742
```

```
In [5]: print("The diameter of {} is {} kilometers.".format(planet,diameter))
```

The diameter of Earth is 12742 kilometers.

**Given this nested list, use indexing to grab the word "hello"**

```
In [6]: lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]
```

```
In [7]: lst[3][1][2][0]
```

```
Out[7]: 'hello'
```

**Given this nested dictionary grab the word "hello". Be prepared, this will be annoying/tricky**

```
In [8]: d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]]]}
```

```
In [9]: d['k1'][3]['tricky'][3]['target'][3]
```

```
Out[9]: 'hello'
```

**What is the main difference between a tuple and a list?**

```
In [10]: # Tuple is immutable
```

**Create a function that grabs the email website domain from a string in the form:**

```
user@domain.com
```

**So for example, passing "user@domain.com" would return: domain.com**

```
In [11]: def domainGet(domain_name):  
         return domain_name.split("@")[1]
```

```
In [12]: domainGet('user@domain.com')
```

```
Out[12]: 'domain.com'
```

**Create a basic function that returns True if the word 'dog' is contained in the input string. Don't worry about edge cases like a punctuation being attached to the word dog, but do account for capitalization.**

```
In [13]: def findDog(name):  
         x = name.lower().split()  
         if "dog" in x:  
             return True  
         else:  
             return False
```

```
In [14]: findDog('Is there a dog here?')
```

```
Out[14]: True
```

**Create a function that counts the number of times the word "dog" occurs in a string. Again ignore edge cases.**

```
In [15]: def countDog(st):  
         x = st.lower().split()  
         count = 0  
         for name in x:  
             if name == "dog":  
                 count = count +1  
         print(count)
```

```
In [16]: countDog('This dog runs faster than the other dog dude!')
```

**Use lambda expressions and the filter() function to filter out words from a list that don't start with the letter 's'. For example:**

```
seq = ['soup','dog','salad','cat','great']
```

**should be filtered down to:**

```
['soup','salad']
```

```
In [17]: seq = ['soup','dog','salad','cat','great']
```

```
In [18]: list(filter(lambda word: word[0] != "s", seq))
```

```
Out[18]: ['soup', 'salad']
```

## Final Problem

You are driving a little too fast, and a police officer stops you. Write a function to return one of 3 possible results: "No ticket", "Small ticket", or "Big Ticket". If your speed is 60 or less, the result is "No Ticket". If speed is between 61 and 80 inclusive, the result is "Small Ticket". If speed is 81 or more, the result is "Big Ticket". Unless it is your birthday (encoded as a boolean value in the parameters of the function) -- on your birthday, your speed can be 5 higher in all cases.

```
In [19]: def caught_speeding(speed, is_birthday):  
  
         if is_birthday:  
             speeding = speed - 5  
         else:  
             speeding = speed  
  
         if speeding > 80:  
             return 'Big Ticket'  
         elif speeding > 60:  
             return 'Small Ticket'  
         else:  
             return 'No Ticket'
```

```
In [20]: caught_speeding(81,True)
```

```
Out[20]: 'Small Ticket'
```

```
In [21]: caught_speeding(81,False)
```

```
Out[21]: 'Big Ticket'
```

Great job!