Introduction

April 27, 2014

```
In [1]: from __future__ import print_function  # Skip this for now
```

1 Hello World

```
In [2]: print('Hello World!')
```

```
Hello World!
```

2 Variables

```
In [3]: my_int = 10  # An integer
        my_float = 12.45  # A floating point
        my_string = 'Hello world!'  # A string
        my_list = [1, 2, 3, 4]  # A list
        my_dict = {1: 'foo', 'bar': 2}  # A dictionary
```

```
In [4]: print(type(my_int))
        print(type(my_float))
        print(type(my_string))
        print(type(my_list))
        print(type(my_dict))
```

```
<type 'int'>
<type 'float'>
<type 'str'>
<type 'list'>
<type 'dict'>
```

3 Basic Operations

- Addition
- Subtraction
- Multiplication
- Division
- Squaring
- String Concatenation
- Type conversion

```
In [5]: first_num = 2
        sec_num = 4
        sum = first_num + sec_num
        print("Sum:", sum)
```

```
        diff = first_num - sec_num
        print("Difference:", diff)
```

```
Sum: 6
Difference: -2
```

```
In [6]: mul = first_num * sec_num
        div = sec_num / first_num
        square = first_num ** sec_num
        print("Product: ", mul)
        print("Quotient: ", div)
        print("Square: ", square)
```

```
Product:  8
Quotient:  2
Square:  16
```

```
In [7]: str1 = 'foo'
        str2 = 'bar'
        str3 = str1 + str2
        str4 = str1 + " " + str2 + "!"
        print(str3)
        print(str4)
```

```
foobar
foo bar!
```

This won't work

```
>>> str5 = str1 + 5
TypeError: cannot concatenate 'str' and 'int' objects
```

3.0.1 Type Conversion

```
In [8]: my_float = 4.3
        my_int = int(my_float)
        print(my_int)
        print(float(my_int))
```

```
4
4.0
```

4 Lists

List is equivalent to array that we use in C. Lists can have different datatypes which isn't possible with arrays. In case of an array you need to define the datatype while you declare the array, lists do not require that.

4.0.2 Initialization

```
In [9]: my_list = []
        print(my_list)
```

```
[]
```

```
In [10]: my_another_list = [7, 8, 9]
         print(my_another_list)
```

```
[7, 8, 9]
```

4.0.3 Operations

- **append** *Used to add an element at the end of a list*
- **extend** *Used to add all the elements of another list at the end of a list*

```
In [11]: my_list.append(4)
         my_list.append(6)
         my_list.append(8)

         my_another_list.append(12)
         print(my_list, my_another_list)

         my_list.extend(my_another_list)
         print(my_list)
```

```
[4, 6, 8] [7, 8, 9, 12]
[4, 6, 8, 7, 8, 9, 12]
```