Python_N1_20160917_AaronYu

September 16, 2016

0.1 Python_N1_20160917

- 1.Data Type
  - Int
  - Float
  - String
  - Boolean
  - None
- 2.Data Structure
  - List
  - Tuple
  - Dictionary
- 3.Indexing & Slicing
- 4.Class & Object, Function & Method
  - Class & Object
  - Function & Method
- 5.Other
  - Getting help in Python

0.2 Data Type

```
In [3]: d_int = 10
        d_float = 2.5
        d_str = 'abc'
        d_bool = True
        d_none = None

In [5]: d_int

Out[5]: 10

In [6]: d_float
```

```
Out[6]: 2.5

In [7]: d_str

Out[7]: 'abc'

In [8]: d_bool

Out[8]: True

In [9]: d_none
```

- Use type to check the data type of objects

```
In [12]: type(d_int)

Out[12]: int

In [15]: type(d_str)

Out[15]: str

In [14]: type(d_none)

Out[14]: NoneType

In [34]: ## No need to declare the data type beforehand
```

0.3 Data Structures

- The most commonly used Native data structures are List, Tuple and Dict

```
In [18]: lst1 = [1,2,3]
         lst1

Out[18]: [1, 2, 3]

In [49]: lst2 = ['a', False, 2.5, 10]
         lst2

Out[49]: ['a', False, 2.5, 10]

In [24]: tup1 = (3,5)
         tup1

Out[24]: (3, 5)

In [25]: tup2=('a',2)
         tup2

Out[25]: ('a', 2)
```

```
In [26]: dict1 = {'XL':1, 'L':2, 'M':3}
         dict1

Out[26]: {'L': 2, 'M': 3, 'XL': 1}
```

- function len() can get the length of List, Tuple and Dict

```
In [29]: len(lst1)

Out[29]: 3

In [31]: len(tup2)

Out[31]: 2

In [32]: len(dict1)

Out[32]: 3
```

- List manipulation

```
In [44]: ## Accessing values in List

In [43]: lst2

Out[43]: ['a', False, 2.5, 10]

In [38]: lst2[0]

Out[38]: 'a'

In [39]: lst2[-1]

Out[39]: 10

In [40]: lst2[1:3]

Out[40]: [False, 2.5]

In [42]: lst2[-2:]

Out[42]: [2.5, 10]

In [45]: ## Add and Delete values in List

In [50]: del lst2[2]
         lst2

Out[50]: ['a', False, 10]

In [51]: lst2.append('z')
         lst2

Out[51]: ['a', False, 10, 'z']

In [53]: lst2[0] = 2
         lst2

Out[53]: [2, False, 10, 'z']
```

0.4 Indexing and Slicing

- Iterables

```
In [76]: ## List, Tuple and String are Iterables

In [57]: for element in lst1:
             print(element)

1
2
3


In [58]: for x in tup1:
             print(x)

3
5


In [59]: for z in 'abc':
             print(z)

a
b
c


In [60]: ## Values in iterables can be accessed through Indexing

In [61]: tup1

Out[61]: (3, 5)

In [70]: tup1[0]

Out[70]: 3

In [79]: d_str

Out[79]: 'abc'

In [80]: d_str[0]

Out[80]: 'a'

In [81]: d_str[:2]

Out[81]: 'ab'

In [82]: d_str[-1]

Out[82]: 'c'
```

0.5 Class & Object, Function & Method

- An example of function

```
In [83]: # range() returns an object that produces a sequence of integers

In [87]: ex1 = range(5)

In [89]: for a in ex1:
             print(a)

0
1
2
3
4


In [90]: # .upper return a string changed to upper class

In [91]: d_str

Out[91]: 'abc'

In [92]: d_str.upper()

Out[92]: 'ABC'
```

0.6 Getting help in Python

- The magic '?'

```
In [93]: range?

In [96]: d_str.upper?
```

- Google and Use the official document

```
In [95]: ## https://docs.python.org/3/index.html
         ## Numpy: https://docs.scipy.org/doc/numpy-dev/user/quickstart.html
         ## Pandas: http://pandas.pydata.org/pandas-docs/stable/10min.html
```