

Lesson 1 - Data and Types

May 2, 2019

1 Lesson 1: Data and Types

In this lesson we learn about the basic data types and data structures and play with them a little.

1. Defines the format by which you input data to a program, modify it and output it in the consol
2. Data types: integer, float, string and boolean
3. Data structures: Lists, dictionaries, tuples etc
4. Global vs. local variable

Lets start by initializing integer, string, list and dictionary

1.1 Data Types

Let us create an integer and string variable and see its output.

```
In [1]: integer = 99
        string = 'poonacha'

        print('This is an integer: ',integer )
        print('This is a string: ',string )
```

```
This is an integer:  99
This is a string:   poonacha
```

1.1.1 Some Simple Data Operations

Most integer/floating point operators that you use in stata or other languages work in python as well - eg (+,-,*,/).

Let us write a code to find the quotient and reminder of a number given a divisor. The operator '//' can be used to find the quotient and '%' can be used to find the reminder.

```
In [2]: number = 199
        divisor = 10
        rem = 199 % 10
        quotient = 199 // 10
        print('reminder = ', rem,'quotient = ', quotient)

reminder =  9 quotient =  19
```

1.2 Data Structures

Data structures combine the basic datatypes (integer, float, string, boolean) to create more complex types. There are several types of data structures but the most commonly used ones are "lists" and "dictionaries"

1.2.1 Lists

We start with a list. A list is a sequence of integer / character data types. Let us now create two list's one with only integer data "i_list" and the other with only character data "c_list".

```
In [3]: i_list = [1,2,3,4]
        c_list = ['day','month','year']

        print('This is an integer list: ',i_list )
        print('This is a char list: ',c_list )
```

```
This is an integer list:  [1, 2, 3, 4]
This is a char list:  ['day', 'month', 'year']
```

The elements in the lists are numbered upwards from 0. Hence, we can access each element in the list using its index number. The following code prints the first element of c_list.

```
In [4]: print('The first element is: ', c_list[0])

The first element is:  day
```

List Operations Lists are mutable i.e. we can append, insert and delete elements from the list.

Syntax - `append(data to be entered)`, `insert(index, data)`, `del(index)`

```
In [5]: i_list = [1,2,3,4]

        # Append the number 5 at the end
        i_list.append(5)
        print(i_list)

        # Insert the 1.5 between 1 and 2
        i_list.insert(1,1.5)
        print(i_list)

        #delete the number 1.5
        del i_list[1]
        print(i_list)

        #delete the number between position 0 and 2
        del i_list[0:2]
        print(i_list)
```

```
#delete the entire list
del i_list[:]
print(i_list)
```

```
[1, 2, 3, 4, 5]
[1, 1.5, 2, 3, 4, 5]
[1, 2, 3, 4, 5]
[3, 4, 5]
[]
```

1.2.2 String

Strings are used to input and read sentences and words. There are several operations that can be done on a string. 1. Concatinating - We can use either "+" operator combine two strings. Join() can be used to combine a list of strings into a single string.

```
In [6]: string = 'poonacha'
        string = string + ' medappa'
        print(string)
        print(string[0:3])
```

```
poonacha medappa
poo
```

1. split('separator') - We can use the split() command to split a string into a list of smaller strings depending on a specified separator
2. Conversely, join('separator') can be used to combine a list of strings into a single string using the separator
3. If no separator is specified, it assumes the separator is a space or a tab
4. The input('input message') function is used to enter data from the console

Let us write a code to enter a sentence and split it into a list of words.

```
In [7]: s_input = input('Enter the sentence to be split: ')
        words = s_input.split()
        print(words)
        print(words[0])
```

```
Enter the sentence to be split: My name is poonacha
['My', 'name', 'is', 'poonacha']
My
```

Exercise 1: Write a small code to enter a date in "dd/mm/yyyy" format. Use the split() command to split the entered date into day, month and year. Hint. you need to specify the separator as '/'.

```
In [0]: # Enter Code
```

```
Enter date in dd/mm/yyyy format:25/09/1984  
the entered date is; day 25 month: 09 year: 1984
```

1.2.3 Dictionaries

A dictionary is similar to a list but rather than an index which represents each element, the dictionary uses a unique user defined key. The following code creates a dictionary by the name "dict_date" with three elements representing the current day, month and year.

```
In [8]: dict_date = {'day' : 14 , 'month' : 3, 'year' : 2018}  
        print('This is a dictionary: ',dict_date)  
        print('Today in dictionary is: day - ',dict_date['day'],' month - ',dict_date['month']
```

```
This is a dictionary: {'day': 14, 'month': 3, 'year': 2018}  
Today in dictionary is: day - 14 month - 3 year - 2018
```