Introduction to Machine Learning Applications

Spring 2021

Lecture-2

Lydia Manikonda

manikl@rpi.edu

Today’s agenda

- Python Basics
- Including class exercises
- Homework-1

Python fundamentals

Basics, loops, conditionals, functions, packages

Basics

Language introduction, setup, variables, data structures

Python Language Introduction

- General-purpose, high level programming language.
- Designed by Guido Van Rossum in 1991

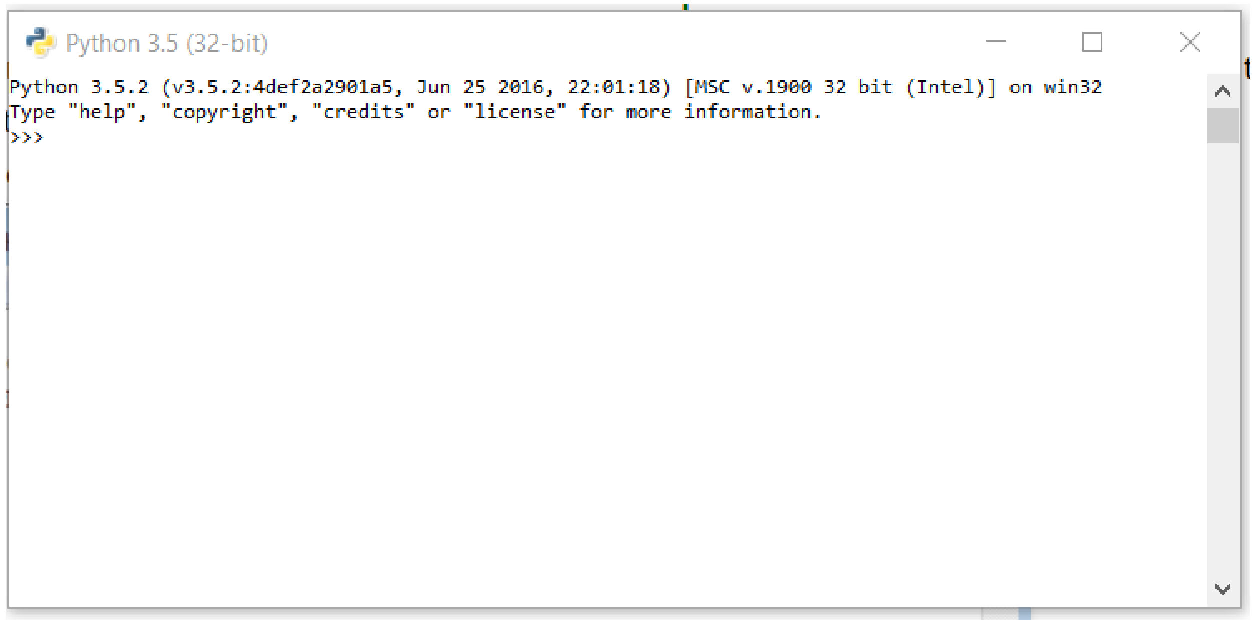
- Main emphasis on
  - Code readability
  - Simple syntax

- 2 major versions – **Python 2** and **Python 3**

*****Python 2 has officially reached End of Life status – no further updates or bugfixes

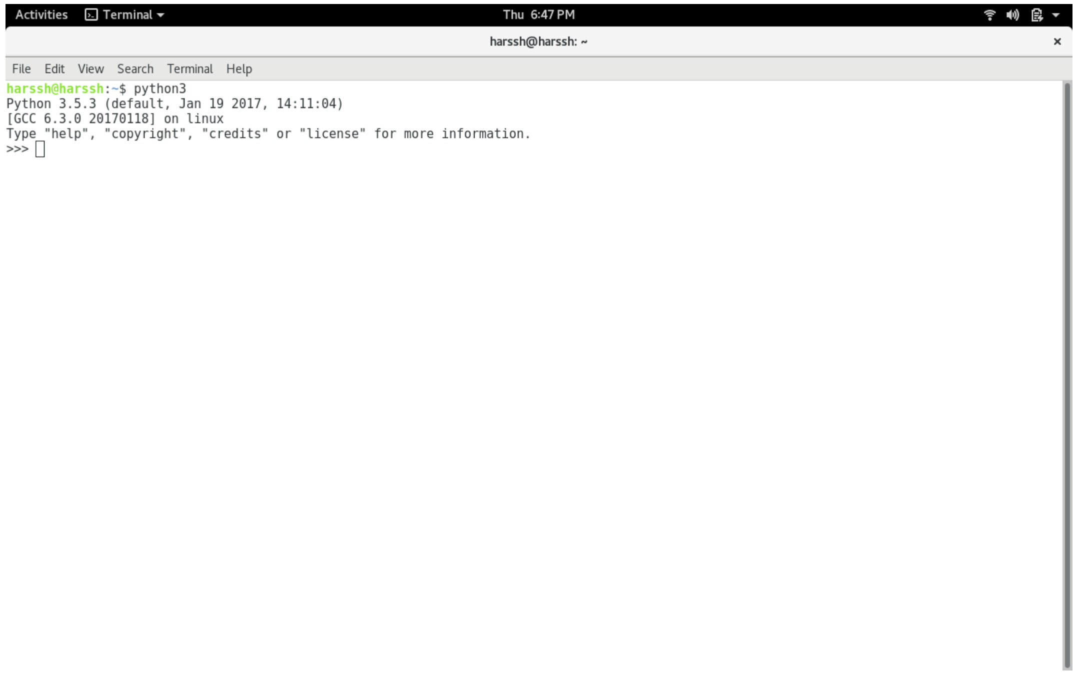
Finding an interpreter

- Windows



```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- Unix/Linux



```
harssh@harssh:~$ python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170118] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

First program in Python

```
>> #Begins -- Comments
>> print(“Hello World”)
>> #Ends – Comments
```

# is used for single line comment in Python

""" this is a comment """ is used for multi line comments

Variables and Data Structures

- In programming languages such as C, C++ or C#, you need to declare the **type of variables** exclusively.
  - Data types can be int, float, char, String, etc.
- Python – take a variable and the value assigned to it automatically tells the data type.

```
>> myVar = 2 #int
>> print(myVar)

>> myVar2 = 2.5 #float
>> print(myVar2)

>> myVar3 = “Hello World!” #string
>> print(myVar3)
```

Data Structures

- Create a variable and assign any value you want!

- Python has 4 types of inbuilt data structures
- **List**
- **Dictionary**
- **Tuple**
- **Set**

List

- Most basic data structure in Python programming language.
- Mutable data structure
  - Elements of this list can be altered after creating the data structure

1. append() – used to add elements in the list
2. insert() – used to add elements in the list at a certain index till the last element

List

append()

```
>> #Create an empty list
>> list1=[]

>> #Append elements to the list
>> list1.append(2)
>> list1.append(4.5)
>> list1.append(“four”)

>> print(list1)
```

insert()

```
>> list1 = [1, 2, 3, 4, 5]
>> list1.insert(5, 10)
>> print(list1)


>> list1.insert(1,10)
>> list1.insert(8,20)
>> print(list1)
```

Example – Mixing append(), insert() and remove()

```
>> list1=[1,2,3,4,5]
>> list1.insert(5,12)
>> list1.insert(1,14)
>> print(list1) # [1, 14, 2, 3, 4, 5, 12]

>> list1.insert(8,20)
>> print(list1) # [1, 14, 2, 3, 4, 5, 12, 20]

>> list1.append(11)
>> print(list1) # [1, 14, 2, 3, 4, 5, 12, 20, 11]

>> list1.pop(5) #removes the element at index 5; if only pop() – removes the last element
>> print(list1)
```

List – Exercise

1. Create a list of size 5 containing 10,20,30,40,50 – one at a time by using the method insert().
2. Print the list.
3. Remove element from index ‘3’ and print the list.
4. Remove the last element and print the list.

Dictionary

- An unordered collection of data values in Python.
- It is used to store data values like a map.
- Unlike other Data Types that hold only single value as an element, Dictionary holds <key:value> pair.
- Dictionary values can be of any datatype – can be duplicated no repeated keys.

Dictionary

```
>> diction1={}
>> print(diction1)

>> diction1 = {1: ‘First’, 2: ‘Python’, 3: ‘Dictionary’}
>> print(diction1)

>> diction1 = {1: ‘First’, 2: [1,2,3,4]}
>> print(diction1)
```

Dictionary

```
>> diction1={}
>> diction1[0]=2
>> diction1[1]=4
>> diction1[2]=“Hello”
>> diction1[“3”]=“It is possible”
```

Dictionary – Exercise

1. Create a dictionary (d1) of size 5 where the keys are from 1 to 5 and their associated values are twice the key value.

   For example, d1[3]=6 because the key is 3 and the value is twice the value of key which is 2*3.

Tuple

- Tuple is a collection of Python objects much like a list.
- The sequence of values stored in a tuple can be of any type, and they are indexed by integers.
- The important difference between a list and a tuple is that **tuples are immutable**.

Tuple

```
>> tuple1=()
>> print(tuple1)

>> tuple1=(1,2,3,4,5)
>> print(tuple1)

>> tuple1=(‘hello’, ‘world’)
>> print(tuple1)
```

Tuple

```
>> list1=[1,2,3,4,5]
>> list1[1]=3
>> print(list1)

>> list1=[7,6,5,4,3,2,1,0]
>> print(list1)

>> mytuple=(0,1,2,3,4,5,6,7)
>> print(mytuple)
>> mytuple[1]=3
```

Concatenate tuples

```
>> Tuple1 = (0, 1, 2, 3)
>> Tuple2 = (‘hello’, ‘world’)
>> Tuple3 = Tuple1 + Tuple2
>> print(Tuple3)
```

Tuple – Exercise

1. Create a tuple t1 that contains 1,2,3,4
2. Create a tuple t2 that contains ‘I’, ‘love’, ‘machine’, ‘learning’
3. Concatenate t1 and t2 to form t3 and print t3.

Set

- Set is an unordered collection of data type that is iterable, mutable and has no duplicate elements.
- Highly optimized method compared to list because it is very easy to check whether an element is present or not.

Set

```
>> set1 = set()
>> print(set1)

>> set1 = set(“Learning”)
>> print(set1)

>> s1=“Learning”
>> set1 = set(s1)
>> print(set1)

>> set1=set([“I”, “love”, “machine”, “learning”])
>> print(set1)
```

Set – Exercise

- S1=“Learning”
- Create a set that has only one element which is S1. In other words, create a set that is {“Learning”}.

Take input from the user

- input() function is used to take input from the user

```
>> # Python program to get input from user
>> name = input("Enter the course name: ")
>> # user entered the name ‘Machine Learning'
>> print("I registered for ", name)
```

User input – Exercise

1. Taking 2 integers as input from the user and print their product.

```
>> num1 = int(input("Enter num1: "))
>> num2 = int(input("Enter num2: "))

>> num3 = num1 * num2
>> print("Product is: ", num3)
```

In-Class Exercise

1. Write a python program to multiply two numbers.
2. Add two strings s1=‘machine’, s2=‘learning’ into one string.
3. Write a python program to find the sum of all the values in list l1 = [1,2,3,4,5].
4. Write a program to find the index of element ‘6’ in a list [1,2,3,4,5,6].
5. Create a dictionary with keys as ‘a’, ‘e’, ‘i’, ‘o’, ‘u’ and their corresponding values as 1.
6. Take the string s1 = “Machine Learning” as a user input and replace the character “a” with “e” in this given string.

Final exam poll

- Webex Poll