Section 1 – Introduction & setup

* 1. Introduction (Youtube link : [Install Python & setup](https://youtu.be/GLmWquMr9aE))

1. Pyhton is case sensitive
2. Anaconda distribution has many library like numpy, Panda
3. Jupyter is a IDE(integreted Delevopment Environment)
4. File from Jupyter is saved as .ipynb and spyder is saved as .py
5. kernel is the server that processes the codes, in case python hangs one can restart kerner
6. Jupyter and Spyder uses same code jst GUI is different
7. Setup & shortcuts

On command mode(press esc to get to command mode) sample short cuts:

* 1. m : markdown
  2. y : code
  3. a : add cell above
  4. b : add cell below
  5. x : delete cell
  6. h : help for shortcuts
  7. F: find and replace
  8. Shit+enter to run a cell of code

NOTE:

* Index in python starts from 0 not 1.

1. Variable Name convention
   1. Start with letter or underscore. DON’T start with a number
   2. Should contain alpha-numeric characters and underscores
   3. Case sentitive. Sales and sales is different.
   4. It should not have any space in between.
   5. Reserved key words not to be used as var names.
   6. Give meaning full names rather than xyz or abc.

Operators

Operators are used to perform operations on variables and values.

|  |  |  |
| --- | --- | --- |
| Arithmetic Operator | | |
| + , - | Addition |  |
| / | Division |  |
| \* | Multiplication |  |
| \*\* | Power |  |
| Assignment Operator | | |
| = | Assignment operators are used to assign values to variables |  |
| Comparision | | |
| == | equal |  |
| != | not equal |  |
| > | greater than |  |
| < | less than |  |
| Logical Operators | | |
| and | Returns True if both statements are true |  |
| or | Returns True if one of the statements is true |  |
| not | Reverse the result, returns False if the result is true |  |
| Some other Operators | | |
|  |  |  |
|  |  |  |
|  |  |  |

Some Common Functions

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| type( x) | returns the datatype of x | d.dtypes |  |
| d.shape |  | df.info() |  |
| d.values |  | df.describe | Only numeric |
| len( ) |  | df.describe(include=’all’) | Includes categorical variables too |
| d.index |  | df.count() | :: all var non NA count |
| min() max() |  | df.columns |  |
| print() |  |  |  |
| sum( ) |  |  |  |
|  |  |  |  |

Basic data type

Data types in python

1. Numeric
   1. Integer. Whole numbers like 10,15,200
   2. Float. Decimals, 10.25, 15.1 , 200.75
   3. Complex numbers. Not usefull in Data science.
2. String

Writen in single or double quotes.

“Hello friends”, ‘What a Lovely Day’

1. Boolean
   1. **T**rue & **F**alse. Researved key words True with a Capital T.
2. date

Compound Data Type

Lists, dictionaries, tuples are compound datatypes that are more complex datatypes that are made of one or more different datatypes.

1. List
   1. A list is an array of objects.List is an ordered object & it has index. An object can be an integer, a string, float, or  a combination or these.
   2. uses square bracket to create a list [ ]
   3. INDEX:
      1. Index in the position of element
      2. in Python counting Starts from 0 not 1
      3. Use index to extract elements from a list
   4. Operations on List
      1. **List sliceing with [ ]**

fruits[1:4]

* + 1. **Add new element to a list**

x.append(100)

* + 1. **Replace element in list**

x[6]="new\_name" #change ‘old\_name’ to new\_name

* + 1. **Delete element**

Del ( list\_name)

* + 1. **Insert**list.insert( 2, “d”) , 2 is position where to insert

1. Tuples
   1. Tuples are just like lists but they are immutable. Once you define a list you can add new items to them, remove existing items and so on, but you cannot do that with a tuple
   2. Uses round bracket to create a tuple ( )
2. Dictionaries
   1. Dictionaries are used to store data values in key:value pairs.
   2. A dictionary is a collection which is unordered, changeable and does not allow duplicates
   3. Uses curly bracket.
   4. Operations on a distionary:
      1. Accessing Elements of a Dictionary.

We pass name of key in [ ] to access the value of the key.

dictonary\_1= {'Name': 'Mary', 'Age': 20,

dictonary\_1[‘Name’]: Output: Mary

* + 1. Update.

Dictionaries are mutable, ie they can be updated.

We can add new items or change the value of existing items using an assignment operator.If the key is already present, then the existing value gets updated. In case the key is not present, a new (**key: value**) pair is added to the dictionary.

dictonary\_1[‘Name’]: ‘John’

* + 1. Drop value.

dictonary\_1.pop('Name'). Key ‘Name’ and its value will be dropped from the dict.

Function

 A **function** is a block of code that encapsulates a specific task or related group of tasks. There are two types of functions

1. Built in functions
2. Custom funtions