Python for Data Science - IBM Data Sci… 1 Jun 2019

Here, I will write down any good information that I reviewed upon. All the knowledges were review for me, but there were some concepts that I forgot or had to look into further.

Week 1 -

Typecasting: changing type of the expression, ex) float to int, int to string.

```
int(True) = 1, int(False) = 0
bool(1) = True, bool(0) = False
```

Indexing String and operations: Just like in lists, can use sample_name[n] to access letters.

[n:m] access multiple letters, including n and excluding m

[::2] –> Every second value

[0:5:2] –> Every second value up to index 4

"\n" –> New line, "\t" –> Tab within a sentence

.upper() –> changes to uppercase letters.

.replace(old string segment, new string segment)

.find(string) –> returns the starting index of the intended string that should be found.

Week 2

List and Tuples: list is mutable, while tuples are not.

- .extend(list) –> adds new items to the original list at the end, not creating a new one.
- .append(item) –> adds **one** item
- del() –> deletes an item
- .split(delimeter) –> Separates the string by a delimeter and makes a list.
- B = A[:] –> cloning the list A without referencing to the same list.

Set: List of items but unordered. use {} to make a set. Venn Diagram can be used to illustrate sets.

- When they are duplicates, the actual set is created with just one, as there are no duplicates in a set. A List can have duplicates, while set does not.
- set(item) –> converts a list or tuple into a set. Duplicates are deleted along the way.

- .add() –> adds an item
- .remove() –> removes an item
- set_1 & set_2 –> And operation. Intersection of two sets
- set_1.union(set_2) –> Or operation. Union of two sets.
- set_2.issubset(set_1) –> Boolean value showing if set_2 is a subset of set_1.

Week 3

For loops:

- enumerate() –> gets the index each time as well as the item in a tuple, list, etc.

Functions:

- sorted() –> creates a new sorted list. sort() –> modifies the list.

Classes:

- dir(name of the object) –> lists all the attributes.

Week 4

Opening file:

- open(".txt", "w") –> "w" can be "r" for read, "a" for append, or "w" for write.
- after we assign a txt file to a variable, **variable.name** –> Get the name attribute.
- Should always close the file using .close() method.

Using "with" statement –> better practice, as it automatically closes the file.

- Look at the example. with open("", "r") as Name:
- .readlines() –> everytime this is called, reads the next line in the file. Can put a number in it to specify how many characters to print out.

Writing files with open:

- .write("") to write things inside the opened file
- with open("file name", "w") as Variable_name –> to open the file, and then write things inside.

Pandas

- read.excel("…xlsx") --> reads excel and make it into a DataFrame.
- From dictionary to DataFrame –> pd.DataFrame(dictionary).
  - Keys correspond to headers, and values correspond to rows.
- .ix[row, column] –> Accesses the unique value in a DataFrame. The row and column label can be an index or name (str type). Also can slice the dataframe by using range.
- .unique() --> to get all the unique values in a column or columns

- Can use inequality operator to get specific columns according to the range (ex: results after the year 1980)

Numpy - One dimensional

- np.array([….])
  - .size –> size attribute
  - .ndim –> dimension of the array
  - .shape –> (rows, columns) # axis 0: vertical axis 1: horizontal
- Vector addition, vector substraction, vector multiplication by a scalar.
- Multiplication of two array –> be careful of the shape
- Dot product –> shows how similar two vectors are.
- Broadcasting
- np.linspace(starting number, ending number, number of evenly spaced numbers to generate)

Numpy - Two dimensional

- A[n][m] –> row, then column.
- For matrix multiplication –> np.dot(matrix, matrix)