Python Programming

Lesson 7 – Introduction to Numpy and Pandas

Lesson 7 - Outline

- Review List, Tuple, Dictionaries, Set
- Introduction to Numpy
- Introduction to Pandas

Review on List, Tuple, Dictionaries, Set

Review on List, Tuple, Set, Dictionary

- **Usage**
  Store the collection of items

- **Syntax / Properties**
  Ordered / Unordered?
  Same / Different datatype?
  Separate the items with comma?
  Enclosed with brackets?

(i) List

(ii) Tuple

(iii) Set

(iv) Dictionary

List

Review on List, Tuple, Set, Dictionary

- **List**
  Store multiple items in a single variable
- **Properties**
  - Ordered (i.e. new item will be placed at the end)
  - Changeable (i.e. change, add, remove items)
  - Allow duplicate values
  - Allow different data types
  - Indexed (i.e. first item is [0])
  - Declaration with **square brackets**
- **Example**

```
sampleList = ["Peter", "Paul", "Mary"]
print(sampleList)
```

Review on List, Tuple, Set, Dictionary

- **<u>Find the number of items in List</u>**

```
sampleList = ["Peter", "Paul", "Mary"]
print(len(sampleList))
```

- **<u>Print the list item (By index)</u>**

  First item: `list[0]`
  Second item: `list[1]`
  Last item: `list[-1]`
  Second last item: `list[-2]`

```
sampleList = ["Peter", "Paul", "Mary"]
print(sampleList[-1])
```

Review on List, Tuple, Set, Dictionary

- **<u>Print the list items (By range of indexes)</u>**
  `list[x:y]`
  (x: start index (included), y: end index (excluded)
  `list[:]`
  (Empty start index means start from first item, empty end index means end at the last item)

```
sampleList = ["0Peter", "1Paul",
"2Mary","3June", "4Larry",
"5May","6Yan", "7Sean"]
print(sampleList[2:6])
```

Review on List, Tuple, Set, Dictionary

- **Use ‘in’ keyword**

```
sampleList = ["Peter", "Paul", "Mary"]
if "Mary" in sampleList:
    print("Mary is on list")
for x in sampleList:
    print(x)
```

- **Use insert, append, remove, pop**

```
sampleList = ["Peter", "Paul", "Mary"]
sampleList.insert(0,"Wendy")
sampleList.append("Fred")
sampleList.remove("Paul")
sampleList.pop(0)
print(sampleList)
```

Tuple

Review on List, Tuple, Set, Dictionary

- **Tuple**
  Store multiple items in a single variable
- **Properties**
  - Ordered (i.e. new item will be placed at the end)
  - Unchangeable (i.e. Cannot change, add, remove items)
  - Allow duplicate values
  - Allow different data types
  - Indexed (i.e. first item is [0])
  - Declaration with **round brackets**
- **Example**

```
sampleTuple = ("Peter", "Paul", "Mary")
print(sampleTuple)
```

Review on List, Tuple, Set, Dictionary

- **Similar to List's operations (Try it!)**

1. Find the number of items
2. Print the item (By index)
3. Print the list items (By range of indexes)
4. Use 'in' keyword

- **Example of casting to List**

```
sampleTuple = ("Peter", "Paul", "Mary")
sampleList = list(sampleTuple)
print(sampleList)
```

Set

Review on List, Tuple, Set, Dictionary

- **Set**
  Store multiple items in a single variable

- **Properties**
  - Unordered (i.e. unknown of stored order)
  - Unchangeable (i.e. Cannot change, remove items but can be added)
  - Not allow duplicate values
  - Allow different data types
  - Unindexed
  - Declaration with **curly brackets**

- **Example**

```
sampleSet = {"Peter", "Paul", "Mary"}
print(sampleSet)
```

Dictionary

Review on List, Tuple, Set, Dictionary

- **Dictionary**
  Store multiple items in a key:value pairs

- **Properties**
  - Ordered (i.e. new item will be placed at the end)
  - Changeable (i.e. change, add, remove items)
  - Allow duplicate values
  - Allow different data types
  - Find value with key name
  - Declaration with **curly brackets**

- **Example**

```
sampleDict = {"Peter":20, "Paul":25,
"Mary":30}
print(sampleDict)
```

Review on List, Tuple, Set, Dictionary

- **Example of getting value**

```
sampleDict = {"Peter":20, "Paul":25,
"Mary":30}
print(sampleDict.get("Peter"))
```

- **Example of adding/ changing/ removing value**

```
sampleDict = {"Peter":20, "Paul":25,
"Mary":30}
sampleDict["Joe"] = 35
sampleDict["Peter"] = 25
sampleDict.pop("Paul")
print(sampleDict)
```

Introduction to Numpy

Numpy in Python

- **Usage of Numpy module**
  - Faster and more compact than Python lists
  - Consumes less memory and is convenient to use
  - The array object in NumPy is called `ndarray` *(which provides many useful function for data science)*
- **Install of Numpy module**

```
pip install numpy
```

Numpy in Python

- **<u>How to install module to another path (e.g. without admin rights)</u>**
  1. Open the command prompt (cmd) at your windows
  2. Type “pip install --user <module name>” to install the module at user folder
     e.g. `pip install --user numpy`
  3. Type “pip show <module name>” to locate the exact location of module
     e.g. `pip show numpy`
  4. Open the Python IDLE
  5. Type the following command to append the location of module to the path
     ```
     import sys
     sys.path.append(r'<location of module>')
     ```

Creation of Numpy Array

Numpy in Python

- **Import Numpy module**

  ```
  import numpy as np
  ```

- **Example – create numpy array (i.e. ndarray)**

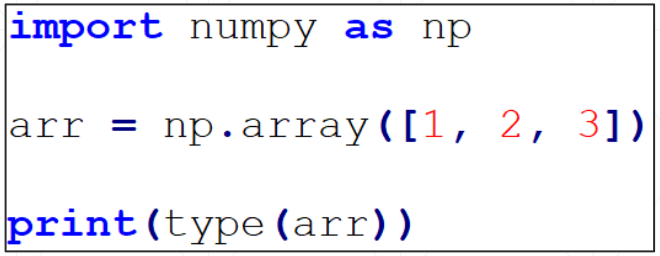
  ```
  import numpy as np

  arr = np.array(['ABC', 'DEF', 'GHI'])

  print(arr)
  ```

- **Explanation**

  Create the numpy array and print it out

Numpy in Python

- **Example – check the type of numpy array**



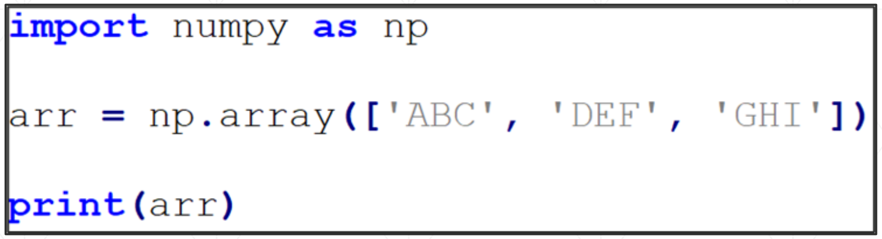
```
import numpy as np

arr = np.array([1, 2, 3])

print(type(arr))
```

- **Explanation**

Create the numpy array and print out its type

Numpy in Python

- **Example – 1D numpy array**



```
import numpy as np

arr = np.array(['ABC', 'DEF', 'GHI'])

print(arr)
```

- **Explanation**

  Create the numpy array (1D array) and print it out

Numpy in Python

- **Example – 2D numpy array**

```
import numpy as np

arr = np.array([['Col1', 'Col2', 'Col3'], [4, 5, 6]])

print(arr)
```

- **Explanation**

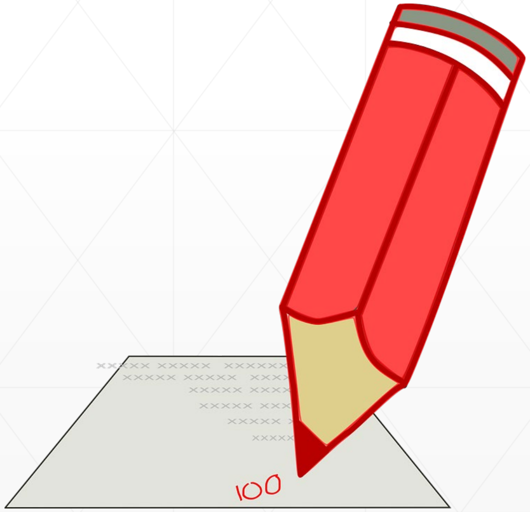
Create the numpy array (2D array) and print it out

Numpy in Python

- **Let's try what if the number of items do not match for each dimension**

- **For example**

```
arr = np.array([[1,2,3],[4,5]])
```



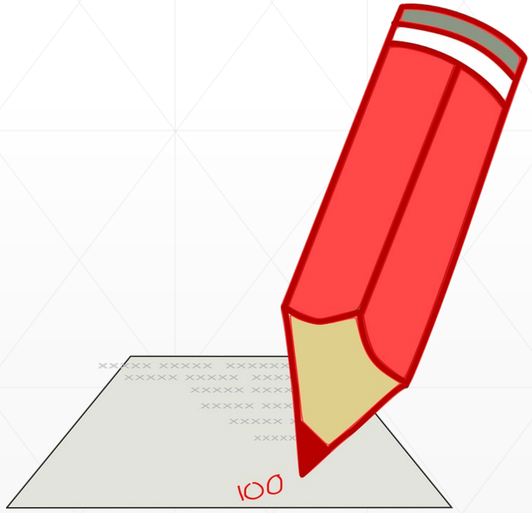
Numpy Array Indexing

Numpy in Python

- **Let's try how to print a particular item in a 1D numpy array**

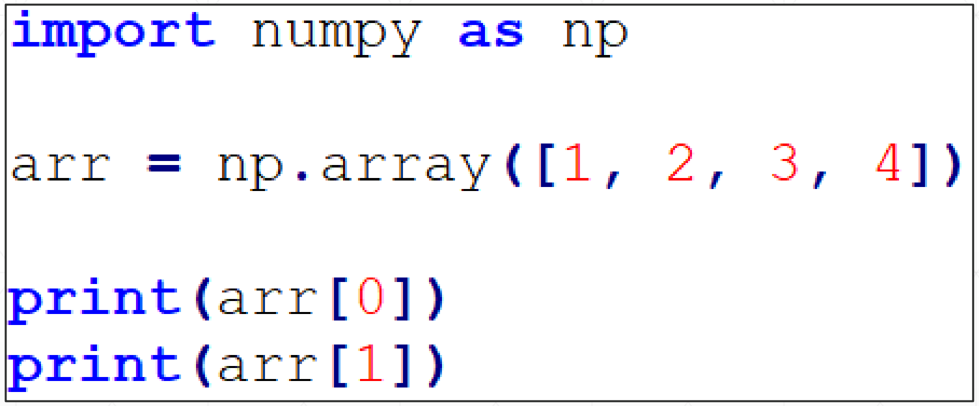
- **Hint: what is the index starting with?**

  0 or 1?



Numpy in Python

- **Example – 1D numpy array indexing**



```
import numpy as np

arr = np.array([1, 2, 3, 4])

print(arr[0])
print(arr[1])
```

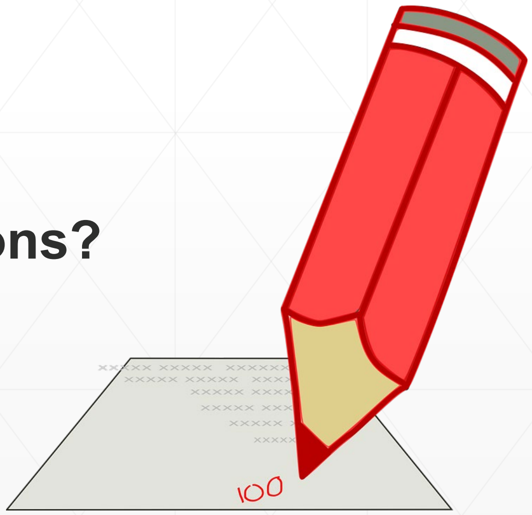
- **Explanation**

  Create the numpy array (1D array), print the first item and second item

Numpy in Python

- **Let's try how to print a particular item in a 2D numpy array**

- **Hint: how to indicate different dimensions?**



Numpy in Python

- **Example – 2D numpy array indexing**

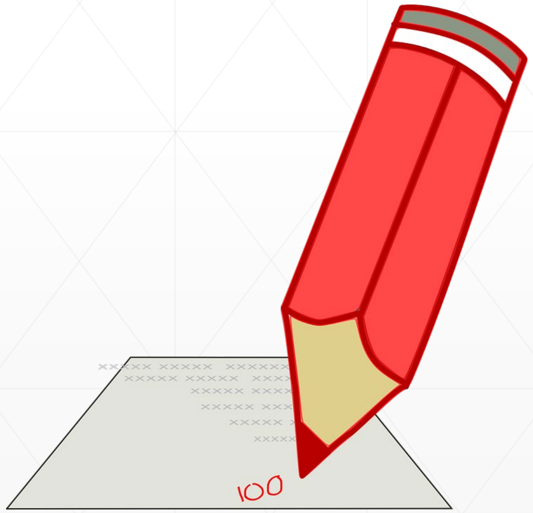
```
import numpy as np

arr = np.array([[1,2,3,4,5], [6,7,8,9,10]])

print('1st row and 1st column: ', arr[0, 0])
print('1st row and 3rd column: ', arr[0, 2])
print('2nd row and 1st column: ', arr[1, 0])
print('2nd row and 5th column: ', arr[1, 4])
```

- **Explanation**

Create the numpy array (2D array), print the items in both first row and second row.

Numpy in Python

- <u>Let's try the negative indexing</u>



Numpy in Python

- **Example – 2D numpy array with negative indexing**

```
import numpy as np

arr = np.array([[1,2,3,4,5],[6,7,8,9,10],[11,12,13,14,15]])

print('Last row and last column: ', arr[-1, -1])
print('Second last row and last column: ', arr[-2, -1])
```

- **Explanation**

  Create the numpy array (2D array), print the items with negative indexing.

Numpy Array Slicing

Numpy in Python

- **Understand Numpy Array Slicing**
  - Select a slice (subset) from the Numpy Array
- **Syntax**
  - `arr[start:end]`
- **Explanation**
  - Select all the items from start to end (exclude the end index)
- **Syntax**
  - `arr[start:end:step]`
- **Explanation**
  - Select all the items from start to end (exclude the end index) with steps.

Numpy in Python

- **Example – simple array slice**

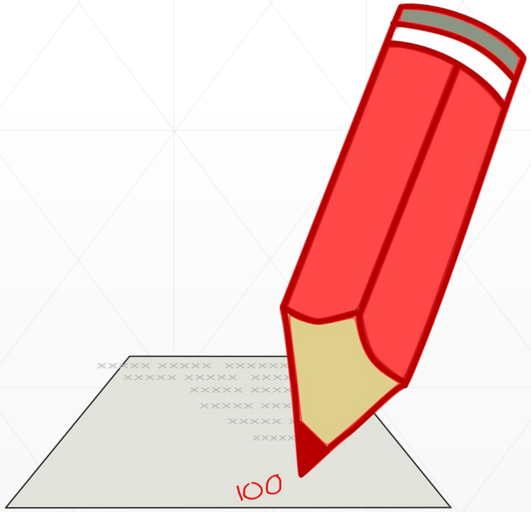
```
import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8])

print(arr[1:6])
```

- **Explanation**

Create the numpy array (1D array), print the slice from index 1 to index 6 (excluded)

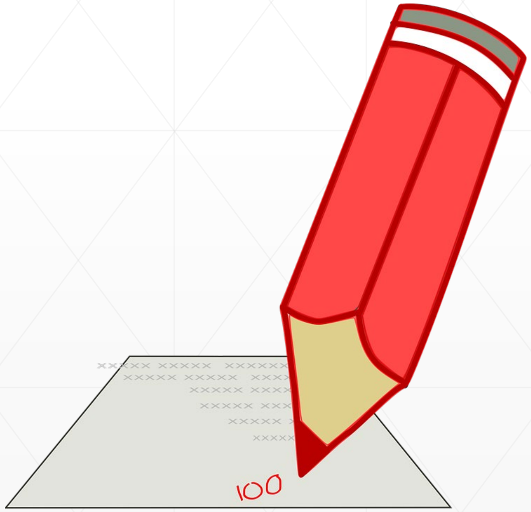
Numpy in Python

- **Let's try without passing the start index**
- e.g. `arr[:6]`



Numpy in Python

- **Let's try without passing the end index**
- e.g. `arr[1:]`



Numpy in Python

- **Example – simple array slice with steps**

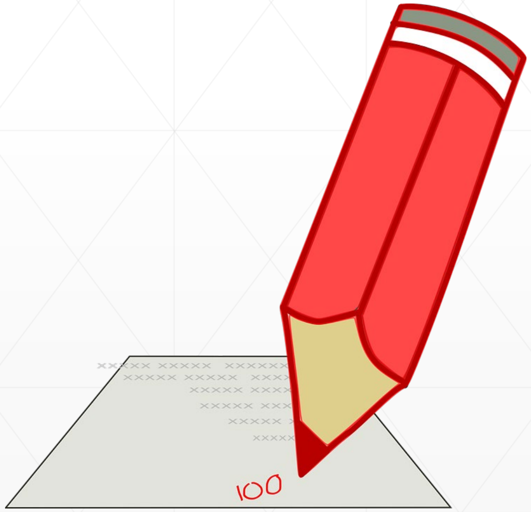
```
import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8])

print(arr[1:6:2])
```

- **Explanation**

Create the numpy array (1D array), print the slice from index 1 to index 6 *(excluded)* and steps with two items *(i.e. skip one item each time)*

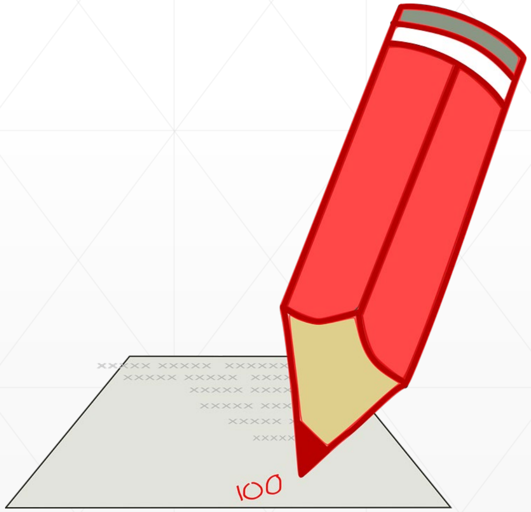
Numpy in Python

- **Let’s try without passing the step**
- e.g. `arr[1:6:]`



Numpy in Python

- **Let’s try the negative indexing**



Numpy in Python

- **Example – 1D numpy array slicing with negative indexing**

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8])

print(arr[-4:-2])
```

- **Explanation**

Create the numpy array (1D array), print the items with negative indexing.

Numpy in Python

- **Example – 2D numpy array slicing**

```
import numpy as np

arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])

print(arr[1, 1:4])
print(arr[0:2, 3])
print(arr[0:2, 1:4])
```

- **Explanation**

Create the numpy array (2D array), print the items with 2D slice.

Numpy Copy and View functions

Numpy in Python

- **Numpy Copy Function**
  - Make a copy of a numpy array.
  - The copy will not change when amending the original array.
- **Syntax**
  - `new_arr = arr.copy()`
- **Numpy View Function**
  - Make a view of a numpy array.
  - The view will change when amending the original array.
- **Syntax**
  - `new_arr = arr.view()`

Numpy in Python

- **Example – Numpy copy function**

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8])
new_arr = arr.copy()
arr[0] = 99

print(arr)
print(new_arr)
```

- **Explanation**

The copy will not change when amending the original array.

Numpy in Python

- **Example – Numpy view function**

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8])
new_arr = arr.view()
arr[0] = 99

print(arr)
print(new_arr)
```

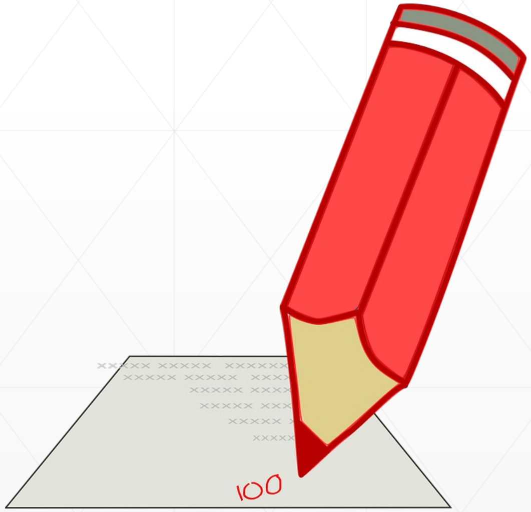
- **Explanation**

  The view will change when amending the original array.

Numpy in Python

- **Let's try if amending the value of new array for both copy and view functions**

- E.g. `new_arr[0] = 88`



Introduction to Pandas

Pandas in Python

- **Usage of Pandas module**
  - Wide spread of use in Data Science, Data Analysis
  - Can be used for data cleansing
  - Calculate the Max, Min, Average, etc.
- **Install of Pandas module**

```
pip install pandas
```

Create Pandas Series

Pandas in Python

- **Import Pandas module**

```
import pandas as pd
```

- **Example – create pandas series from a list**

```
import pandas as pd

list = [7, 8, 9]

pdseries = pd.Series(list)

print(pdseries)
```

- **Explanation**

Create the pandas series (~ a 1D column list)

Pandas in Python

- **Example – Index in pandas series**

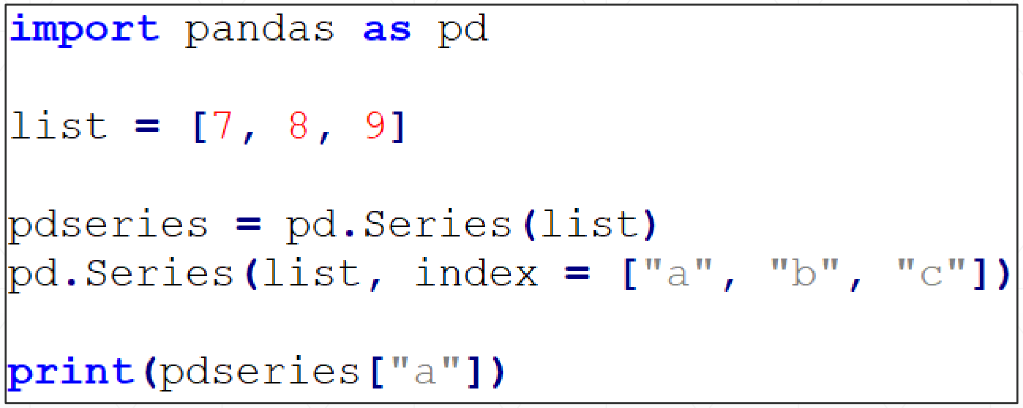
```
import pandas as pd

list = [7, 8, 9]

pdseries = pd.Series(list)

print(pdseries[0])      #return first item
#print(pdseries[3])     #invalid
#print(pdseries[-1])    #invalid
```

- **Explanation**

  Create the pandas series and print the first item

Pandas in Python

- **Example – labels in pandas series**



```
import pandas as pd

list = [7, 8, 9]

pdseries = pd.Series(list)
pd.Series(list, index = ["a", "b", "c"])

print(pdseries["a"])
```

- **Explanation**

Create the pandas series and print the with the labels

Create Pandas DataFrame

Pandas in Python

- **Example – create pandas DataFrame with key-pair**

```
import pandas as pd

data = {
  "Name": ['Peter', 'Paul', 'Mary'],
  "Marks": [80, 85, 90]
}

df = pd.DataFrame(data)

print(df)
```

- **Explanation**

Create the pandas DataFrame (~ a 2D column list with labels) and print the data

Pandas in Python

- **Print specific row(s) from pandas DataFrame**

  Use of `loc`

- **Example – print specific row(s) from pandas DataFrame**

```python
import pandas as pd

data = {
  "Name": ['Peter', 'Paul', 'Mary'],
  "Marks": [80, 85, 90]
}

df = pd.DataFrame(data)

print(df.loc[0])
```

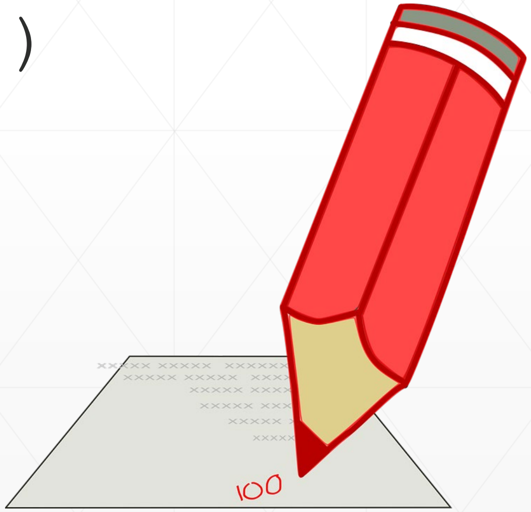
- **Explanation**

  Create the pandas DataFrame and print specific rows

Pandas in Python

- **<u>How to print multiple rows?</u>**

- E.g. `print(df.loc[[0,2]])`



Create Named Index for Pandas DataFrame

Pandas in Python

- **Use of Named Index for Pandas DataFrame**

```
df = pd.DataFrame(data, index = ["i1", "i2", "i3"])
```

- **Example – named Index for Pandas DataFrame**

```
import pandas as pd

data = {
  "Name": ['Peter', 'Paul', 'Mary'],
  "Marks": [80, 85, 90]
}

df = pd.DataFrame(data, index = ["stud1", "stud2", "stud3"])

print(df)
```

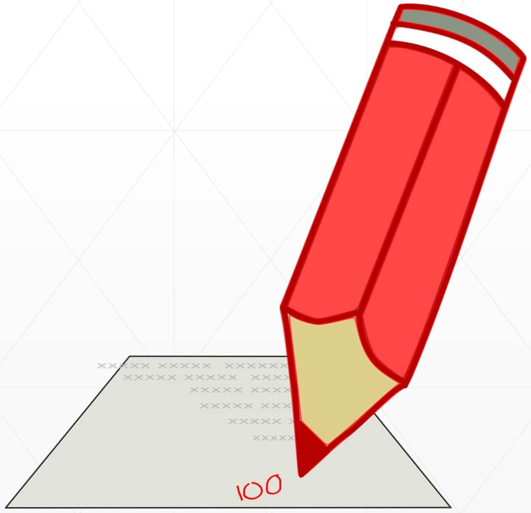
- **Explanation**

Create the pandas DataFrame and add the named index to the data

Pandas in Python

- **How to print DataFrames with named index?**

- E.g. `print(df.loc["i2"])`



Read CSV to DataFrame

Pandas in Python

- **Read CSV to Pandas DataFrame**

```
df = pd.read_csv('sample.csv')
```

- **Example – Read CSV to Pandas DataFrame**

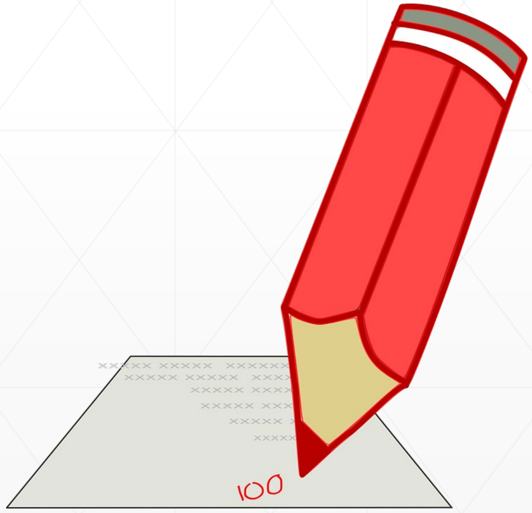
```
import pandas as pd

df = pd.read_csv('sample.csv')

print(df.to_string())
```

- **Explanation**

Read CSV to Pandas DataFrame and use print using to_string() to print all the content.

Pandas in Python

- **What if printing the dataframe without using to_string()?**
- **Let's try!**



Pandas in Python

- How to read Excel file?
- How to read JSON data?
- How make simple data analysis?

- Coming Soon!

Reference

Reference of Numpy

- https://numpy.org/

Reference of Pandas

- https://pandas.pydata.org/

Thank you