MAJJIGA JASWANTH

LAB-8

PART-1

20BCD7171

Welcome to Jupyter!

In [1]:
```
import numpy as np
import pandas as pd
np_array = np.array([0.25, 0.5, 0.75, 1.0])
# Create a pandas series object
data = pd.Series(np_array)
data
```

Out[1]:
```
0    0.25
1    0.50
2    0.75
3    1.00
dtype: float64
```

In [2]:
```
data = pd.Series([0.25, 0.5, 0.75, 1.0])
print(data)
print("\n")
# Print the attributes of Series object data
print(data.index)
print("\n")
print(data.dtype)
```

```
0    0.25
1    0.50
2    0.75
3    1.00
dtype: float64


RangeIndex(start=0, stop=4, step=1)


float64
```

In [4]:
```
data = pd.Series([0.25, 0.5, 0.75, 1.0])
print("implicit indexing")
print(data)
print("\n")
data = pd.Series([0.25, 0.5, 0.75, 1.0],
 index=['a', 'b', 'c', 'd'])
print("explicit indexing")
data
```

```
implicit indexing
0    0.25
1    0.50
2    0.75
3    1.00
dtype: float64


explicit indexing
```

Out[4]:
```
a    0.25
b    0.50
c    0.75
```

```
d    1.00
dtype: float64
```

In [10]:

```
data = pd.Series([0.25, 0.5, 0.75, 1.0],
 index=[2, 5, 3, 7])
data
```

Out[10]:

```
2    0.25
5    0.50
3    0.75
7    1.00
dtype: float64
```

In [11]:

```
 population_dict = {'California': 38332521,
 'Texas': 26448193,
 'New York': 19651127,
'Florida': 19552860,
'Illinois': 12882135}
population = pd.Series(population_dict)
population
```

Out[11]:

```
California    38332521
Texas         26448193
New York      19651127
Florida       19552860
Illinois      12882135
dtype: int64
```

In [12]:

```
simple_df = pd.DataFrame(population)
simple_df
```

Out[12]:

	0
California	38332521
Texas	26448193
New York	19651127
Florida	19552860
Illinois	12882135

In [13]:

```
simple_df = pd.DataFrame(population,columns=['Population'])
simple_df
```

Out[13]:

	Population
California	38332521
Texas	26448193
New York	19651127
Florida	19552860
Illinois	12882135

In [14]:
```
population_dict = {'California': 38332521,
 'Texas': 26448193,
'New York': 19651127,
'Florida': 19552860,
'Illinois': 12882135}
population = pd.Series(population_dict)
population
```

Out[14]:
```
California    38332521
Texas         26448193
New York      19651127
Florida       19552860
Illinois      12882135
dtype: int64
```

In [21]:
```
simple_df=pd.DataFrame(population,columns=['Population'])
simple_df
```

Out[21]:

	Population
California	38332521
Texas	26448193
New York	19651127
Florida	19552860
Illinois	12882135

In [23]:
```
# create a pandas series object population_dist
population_dict={'California':38332521,'Texas':26448193,'New York':19651127,'I
population=pd.Series(population_dict)
# print the pandas series object population_dist
population
```

Out[23]:
```
California    38332521
Texas         26448193
New York      19651127
Florida       19552860
Illinois      12882135
dtype: int64
```

In [24]:
```
# create a pandas series object area_dist
area_dict={'California':423967,'Texas':695662,'New York':141297,'Florida':170
area=pd.Series(area_dict)
# create a pandas series object area_dist
area
```

Out[24]:
```
California    423967
Texas         695662
New York      141297
Florida       170312
Illinois      149995
dtype: int64
```

In [25]:
```
states=pd.DataFrame([population,area],columns=['population','area'])
states
```

Out[25]:

	population	area
0	NaN	NaN
1	NaN	NaN

In [26]:
```
states=pd.DataFrame({'population':population,'area':area})
states
```

Out[26]:

	population	area
California	38332521	423967
Texas	26448193	695662
New York	19651127	141297
Florida	19552860	170312
Illinois	12882135	149995

In [28]:
```
# From a two-dimensional NumPy array
# create a 2-d numpy array
data=np.random.rand(3,2)
# create a pandas data frame my_df from 2-d numpy array data
my_df=pd.DataFrame(data,columns=['foo','bar'],index=['a','b','c'])
# print the data frame my_df
my_df
```

Out[28]:

	foo	bar
a	0.462295	0.395934
b	0.525427	0.328054
c	0.812782	0.765844

In [33]:
```
states=pd.DataFrame({'population':population,'area':area})
states
print(states)
print("\n")
# Print the attributes of Series object data
print(states.index)
print("\n")
print(states.columns)
print("\n")
print(states.population.dtype)
print("\n")
print(states.area.dtype)
```

```
            population    area
```

```
California    38332521  423967
Texas         26448193  695662
New York      19651127  141297
Florida       19552860  170312
Illinois      12882135  149995


Index(['California', 'Texas', 'New York', 'Florida', 'Illinois'], dtype='objec
t')


Index(['population', 'area'], dtype='object')


int64


int64
```

In [34]:

```
ind=pd.Index([2,3,5,7,11])
ind
```

Out[34]:

```
Int64Index([2, 3, 5, 7, 11], dtype='int64')
```

In [35]:

```
# create a pandas index object
ind=pd.Index([2,3,5,7,11])
ind
# print the index object
print("size of given index is")
print(ind.size)
print("\n")
print("shape of given index is")
print(ind.shape)
print("\n")
print("No of dimensions of given index is")
print(ind.ndim)
print("\n")
print("datatype of given index is")
print(ind.dtype)
```

```
size of given index is
5


shape of given index is
(5,)


No of dimensions of given index is
1


datatype of given index is
int64
```

In [36]:

```
# index objects are immutable
ind[1]=0
```

```
---------------------------------------------------------------------------
```

```
TypeError                                 Traceback (most recent call last)
<ipython-input-36-d12e5ae11de8> in <module>
      1 # index objects are immutable
----> 2 ind[1]=0

/srv/conda/envs/notebook/lib/python3.6/site-packages/pandas/core/indexes/base.
py in __setitem__(self, key, value)
   4082
   4083     def __setitem__(self, key, value):
-> 4084         raise TypeError("Index does not support mutable operations")
   4085
   4086     def __getitem__(self, key):
```

In [37]:

```
indA=pd.Index([1,3,5,7,9])
indB=pd.Index([2,3,5,7,11])
```

In [39]:

```
indA&indB # intersection
```

Out[39]:

```
Int64Index([3, 5, 7], dtype='int64')
```

In [40]:

```
indA|indB# union
```

Out[40]:

```
Int64Index([1, 2, 3, 5, 7, 9, 11], dtype='int64')
```

In [42]:

```
import pandas as pd
data=pd.Series([0.25,0.5,0.75,1.0],index=['a','b','c','d'])
data
```

Out[42]:

```
a    0.25
b    0.50
c    0.75
d    1.00
dtype: float64
```

In [43]:

```
data['b']
```

Out[43]:

```
0.5
```

In [44]:

```
data[0]
```

Out[44]:

```
0.25
```

In [45]:

```
# adding new value to a Series object
data['e']=1.25
data
```

Out[45]:

```
a    0.25
b    0.50
c    0.75
d    1.00
e    1.25
```