ch50sbvi4

January 23, 2025

[ ]:
```
# accessing values in a array in numpy
import numpy as np
array=np.array([[1,2,3],[4,5,6],[7,8,9]])
array[2,2]
```

[ ]:
```
9
```

[ ]:
```
#array slicing
array=np.array([1,2,3,4,5,6,7])
array[0:6:2] #index 0 to 6 for every second digit
```

[ ]:
```
array([1, 3, 5])
```

[ ]:
```
array=np.array([[1,2,3,4],[5,6,7,8],[9,10,11,12],[13,14,15,16]])
print(array)
array[3,1] #for multi dimentional array we use multi level indexing
array[0:1]
array[1,:] #prints entire 1th index row
array[:,2] #prints every value in 2nd index column
```

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]]
```

[ ]:
```
array([ 3,  7, 11, 15])
```

[ ]:
```
array=np.array([[[1,2,3,4],[5,6,7,8]],[[9,10,11,12],[13,14,15,16]]]) #3
 ↪dimentional array
array[1,1,1] #for multi dimentional array we use multi level indexing
array
```

[ ]:
```
array([[[ 1,  2,  3,  4],
        [ 5,  6,  7,  8]],

       [[ 9, 10, 11, 12],
        [13, 14, 15, 16]]])
```

[ ]:
```
array=np.array([1,2,3])
array^2
```

[ ]:
```
array([3, 0, 1])
```

[ ]:
```
start=time.time()
array=np.array([[1,2,3],[4,5,6],[7,8,9]])
result=array**2 #vectorization -accessing individual items
end=time.time()
print(end-start)
result
```

```
0.0002779960632324219
```

[ ]:
```
array([[ 1,  4,  9],
       [16, 25, 36],
       [49, 64, 81]])
```

[ ]:
```
import time
start=time.time()
array=np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14])
for i in range(0,len(array)):
  array[i]=array[i]**2
end=time.time()
print(end-start)
array
```

```
0.00031256675720214844
```

[ ]:
```
array([  1,   4,   9,  16,  25,  36,  49,  64,  81, 100, 121, 144, 169,
       196])
```

Pandas

[ ]:
```
import pandas as pd
#series , dataframe in pandas --collection of series is dataframe
data=[10,20,30,40,50]
s=pd.Series(data)
s
```

[ ]:
```
0    10
1    20
2    30
3    40
4    50
dtype: int64
```

[ ]:
```
data=[10,20,30,40]
s=pd.Series(data, index=['a','b','c','d'])
s
s[3]
```

```
<ipython-input-33-5e3cdb3b4cf7>:4: FutureWarning: Series.__getitem__ treating
keys as positions is deprecated. In a future version, integer keys will always
be treated as labels (consistent with DataFrame behavior). To access a value by
position, use `ser.iloc[pos]`
  s[3]
```

[ ]:
```
40
```

[ ]:
```
grades={"Sem1":3.4,"Sem2":4,"Sem3":2.5}
type(grades) #dict type
s=pd.Series(grades)
print(s)
```

```
Sem1    3.4
Sem2    4.0
Sem3    2.5
dtype: float64
```

[ ]:
```
lst=[]
mathmark=float(input("Enter marks in math: "))
lst.append(mathmark)
sciencemark=float(input("Enter marks in science"))
lst.append(sciencemark)
cmark=float(input("Enter mark in c: "))
lst.append(cmark)
cpp=float(input("Enter mark in cpp"))
lst.append(cpp)

s=pd.Series(lst, index=['Math','Science','C','Cpp'])

print(s)
```

```
Enter marks in math: 99
ENter marks in science90
Enter mark in c: 98
Enter mark in cpp97
Math       99.0
Science    90.0
C          98.0
Cpp        97.0
dtype: float64
9.805348873138428
```

#DataFrame

```
[ ]: data={
         "Name":['Raman','Sajin','Santosh'],
         "Address":['LT','BKT','BT'],
         "Age":[1,2,3]
     }
     df=pd.DataFrame(data)
     df.head()
```

```
[ ]:       Name Address  Age
     0    Raman      LT    1
     1    Sajin     BKT    2
     2  Santosh      BT    3
```

```
[ ]: data=[["Raman","sajin","suraj"],["1","T","P"]]
     df=pd.DataFrame(data, columns=["Name","Code","Age"]).T #transposes a value
     df.head()
```

```
[ ]:          0  1
     Name  Raman  1
     Code  sajin  T
     Age   suraj  P
```

```
[ ]: df=pd.read_csv("data.csv")
     print(df)
     gpa=[3.2,4.0,3.5,2.0,3.2]
     df['gpa']=gpa
     df
```

```
        Name    Address Faculty Semester
     0  Raman   Lalitpur   Steam      III
     1   Ram   bhaktapur  are-23       II
     2   Hari       kath    dect       IV
     3  Gopal      nupur    rict        V
     4  Suraj       dnag    beat      VII
```

```
[ ]:     Name    Address Faculty Semester  gpa
     0  Raman   Lalitpur   Steam      III  3.2
     1   Ram   bhaktapur  are-23       II  4.0
     2   Hari       kath    dect       IV  3.5
     3  Gopal      nupur    rict        V  2.0
     4  Suraj       dnag    beat      VII  3.2
```

```
[ ]: df.iloc[2]  #prints specific column
```

```
[ ]: Name        Hari
     Address     kath
```

```
Faculty     dect
Semester      IV
gpa          3.5
Name: 2, dtype: object
```

[ ]:
```
df.head(2) #
```

[ ]:
```
    Name    Address Faculty Semester  gpa
0  Raman   Lalitpur   Steam      III  3.2
1   Ram   bhaktapur  are-23       II  4.0
```

[ ]:
```
df.tail(2)
```

[ ]:
```
    Name Address Faculty Semester  gpa
3  Gopal   nupur    rict        V  2.0
4  Suraj    dnag    beat      VII  3.2
```

[ ]:
```
dv1=pd.read_csv("sample_data/california_housing_train.csv")
len(dv1) #lenth of the csv file
dv1.head()
```

[ ]:
```
   longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
0    -114.31     34.19                15.0       5612.0          1283.0
1    -114.47     34.40                19.0       7650.0          1901.0
2    -114.56     33.69                17.0        720.0           174.0
3    -114.57     33.64                14.0       1501.0           337.0
4    -114.57     33.57                20.0       1454.0           326.0

   population  households  median_income  median_house_value
0      1015.0       472.0         1.4936             66900.0
1      1129.0       463.0         1.8200             80100.0
2       333.0       117.0         1.6509             85700.0
3       515.0       226.0         3.1917             73400.0
4       624.0       262.0         1.9250             65500.0
```

[ ]:
```
df=pd.read_excel("test.xlsx")
df.head()
```

[ ]:
```
      Name     Address    Faculty  GPA Semester
0    Raman    Lalitpur   B Tech E  3.8      III
1    Sajin    Lalitpur   B Tech D  3.6       II
2  Santosh  Kanchanpur  B Tech ED  3.7       IV
3    Suraj      Siraha     B Tech  3.9        V
```

[ ]:
```
df.loc[2,'GPA']=2.0
df.to_excel("test.xlsx",index=False)
df
```

[ ]:

	Name	Address	Faculty	GPA	Semester
0	Raman	Lalitpur	B Tech E	3.8	III
1	Sajin	Lalitpur	B Tech D	3.6	II
2	Santosh	Kanchanpur	B Tech ED	2.0	IV
3	Suraj	Siraha	B Tech	3.9	V