

##1. How to extract all odd numbers from arr?

```
import numpy as np  
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [12]: import numpy as np  
arr=np.array([0,1,2,3,4,5,6,7,8,9])  
print(arr[arr%2!=0])  
  
[1 3 5 7 9]
```

##2. Replace all odd numbers in arr with -1 without changing

```
import numpy as np  
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8,9])
```

```
In [16]: import numpy as np  
arr=np.array([0,1,2,3,4,5,6,7,8,9])  
arr[arr%2!=0]=-1  
print(arr)  
  
[ 0 -1  2 -1  4 -1  6 -1  8 -1]
```

##3. Convert a 1D array to a 2D array with 2 rows

```
import numpy as np  
arr=np.arange(10)
```

#Desired Output:

```
#> array([[0, 1, 2, 3, 4],  
#>         [5, 6, 7, 8, 9]])
```

```
In [18]: import numpy as np  
arr=np.arange(10)  
arr.reshape(2,5)  
  
Out[18]: array([[0, 1, 2, 3, 4],  
                 [5, 6, 7, 8, 9]])
```

##4. Stack arrays a and b vertically

```
import numpy as np
a = np.arange(10).reshape(2,-1)
b = np.repeat(1, 10).reshape(2,-1)
```

#Desired Output:

```
#> array([[0, 1, 2, 3, 4],
#>        [5, 6, 7, 8, 9],
#>        [1, 1, 1, 1, 1],
#>        [1, 1, 1, 1, 1]])
```

```
In [32]: import numpy as np
         a=np.arange(10).reshape(2,-1)
         b=np.repeat(1,10).reshape(2,-1)
         arr=np.vstack((a,b))
         print(arr)

[[0 1 2 3 4]
 [5 6 7 8 9]
 [1 1 1 1 1]
 [1 1 1 1 1]]
```

##5. Stack the arrays a and b horizontally.

```
import numpy as np
a = np.arange(10).reshape(2,-1)
b = np.repeat(1, 10).reshape(2,-1)
```

#Desired Output:

```
#> array([[0, 1, 2, 3, 4, 1, 1, 1, 1, 1],
#>        [5, 6, 7, 8, 9, 1, 1, 1, 1, 1]])
```

```
In [33]: import numpy as np  
a=np.arange(10).reshape(2,-1)  
b=np.repeat(1,10).reshape(2,-1)  
arr=np.hstack((a,b))  
print(arr)
```

```
[[0 1 2 3 4 1 1 1 1 1]  
 [5 6 7 8 9 1 1 1 1 1]]
```

##6.How to get the common items between two python numpy arrays?

Get the common items between a and b

```
import numpy as np  
a = np.array([1,2,3,2,3,4,3,4,5,6])  
b = np.array([7,2,10,2,7,4,9,4,9,8])
```

```
In [54]: import numpy as np  
a=np.array([1,2,3,2,3,4,3,4,5,6])  
b=np.array([7,2,10,2,7,4,9,4,9,8])  
x=np.where(a==b)  
print(x)
```

```
(array([1, 3, 5, 7], dtype=int64),)
```

##7.How to remove from one array those items that exist in another?

Q. From array a remove all items present in array b

```
import numpy as np
```

```
a = np.array([1,2,3,4,5])  
b = np.array([5,6,7,8,9])
```

```
In [55]: import numpy as np  
a=np.array([1,2,3,4,5])  
b=np.array([5,6,7,8,9])  
x=np.setdiff1d(a,b)  
print(x)
```

```
[1 2 3 4]
```

##8. How to get the positions where elements of two arrays match?

Q. Get the positions where elements of a and b match

```
import numpy as np  
a = np.array([1,2,3,2,3,4,3,4,5,6])  
b = np.array([7,2,10,2,7,4,9,4,9,8])
```

#Desired Output:

```
#> (array([1, 3, 5, 7]),)
```

```
In [54]: import numpy as np  
a=np.array([1,2,3,2,3,4,3,4,5,6])  
b=np.array([7,2,10,2,7,4,9,4,9,8])  
x=np.where(a==b)  
print(x)  
  
(array([1, 3, 5, 7], dtype=int64),)
```

##9. How to extract all numbers between a given range from a numpy array?

Get all items between 5 and 10 from a

```
import numpy as np
```

```
a = np.array([2, 6, 1, 9, 10, 3, 27])
```

#Desired Output:

```
#[array([6, 9, 10]),]
```

```
In [57]: import numpy as np  
a=np.array([2,6,1,9,10,3,27])  
x=(a>=5)&(a<=10)  
a[x]  
  
Out[57]: array([ 6,  9, 10])
```

##Pandas Data Series

##1. Write a Pandas program to convert a dictionary to a Pandas series.

...

Sample Series:

Original dictionary:

```
{'a': 100, 'b': 200, 'c': 300, 'd': 400, 'e': 800}
```

Converted series:

```
a    100  
b    200  
c    300  
d    400  
e    800  
dtype: int64
```

...

```
In [59]: import pandas as pd  
dict={'a':100,'b':200,'c':300,'d':400,'e':800}  
x=pd.Series(dict)  
print(x)
```

  

```
a    100  
b    200  
c    300  
d    400  
e    800  
dtype: int64
```

##2. Write a Pandas program to convert a NumPy array to a Pandas series.

Sample Series:

NumPy array:

```
[10 20 30 40 50]
```

Converted Pandas series:

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

```
In [60]: import pandas as pd
numpy_array=[10,20,30,40,50]
x=pd.Series(numpy_array)
print(x)

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

##3. Write a Pandas program to change the data type of given a column or a Series.

Sample Series:

Original Data Series:

```
0    100
1    200
2    python
3    300.12
4    400
dtype: object
```

Change the said data type to numeric:

```
0    100.00
1    200.00
2    NaN
3    300.12
4    400.00
dtype: float64
```

```
In [1]: import pandas as pd
s=pd.Series([100,200,'python',300.12,400])
print(s)
s=pd.to_numeric(s,errors='coerce')
print(s)

0    100
1    200
2    python
3    300.12
4    400
dtype: object
0    100.00
1    200.00
2    NaN
3    300.12
4    400.00
dtype: float64
```

##4. Write a Pandas program to convert the first column of a DataFrame as a Series.

Sample Output:

Original DataFrame

```
col1 col2 col3
0    1    4    7
1    2    5    5
2    3    6    8
3    4    9   12
4    7    5    1
5   11    0   11
```

1st column as a Series:

```
0    1
1    2
2    3
3    4
4    7
5   11
```

Name: col1, dtype: int64

<class 'pandas.core.series.Series'>

```
In [5]: import pandas as pd
x=pd.DataFrame({'col1':[1,2,3,4,7,11],'col2':[4,5,6,9,5,0],'col3':[7,5,8,12,1,11]})
s=x.iloc[:,0]
s

Out[5]: 0    1
1    2
2    3
3    4
4    7
5   11
Name: col1, dtype: int64
```

##5. Write a Pandas program to convert a given Series to an array.

Sample Output:

Original Data Series:

```
0    100
1    200
2  python
3  300.12
4    400
dtype: object
```

Series to an array

```
['100' '200' 'python' '300.12' '400']
```

<class 'numpy.ndarray'>

```
In [6]: import pandas as pd
s=pd.Series([100,200,'python',300.12,400])
a=s.to_numpy()
print(a)
[100 200 'python' 300.12 400]
```

##6 Write a Pandas program to convert Series of lists to one Series.

Sample Output:

Original Series of list

0 [Red, Green, White]

1 [Red, Black]

2 [Yellow]

dtype: object

One Series

0 Red

1 Green

2 White

3 Red

4 Black

5 Yellow

dtype: object

```
In [16]: import pandas as pd
s=pd.Series([['Red','Green','White'],['Red','Black'],['Yellow']])
print(s)
s=s.explode()
print(s)

0   [Red, Green, White]
1           [Red, Black]
2           [Yellow]
dtype: object
0     Red
0     Green
0     White
1     Red
1     Black
2     Yellow
dtype: object
```

##7. Write a Pandas program to sort a given Series.

Sample Output:

Original Data Series:

0 100

```
1    200
2  python
3  300.12
4    400
dtype: object
0    100
1    200
3  300.12
4    400
2  python
dtype: object
```

In [1]:

```
import pandas as pd
s=pd.Series([100,200,'python',300.12,400])
print(s)
x= s.sort_values()
print(x)
```

0 100  
1 200  
2 python  
3 300.12  
4 400  
dtype: object

##Pandas DataFrame

##1. Write a Pandas program to create a dataframe from a dictionary and display it.

...

Sample data: {'X':[78,85,96,80,86], 'Y':[84,94,89,83,86],'Z':[86,97,96,72,83]}

Expected Output:

	X	Y	Z
0	78	84	86
1	85	94	97
2	96	89	96
3	80	83	72
4	86	86	83

```
In [100]: import pandas as pd  
d={'X':[78,85,96,80,86],'Y':[84,94,89,83,86],'Z':[86,97,96,72,83]}  
x=pd.DataFrame(d)  
print(x)
```

	X	Y	Z
0	78	84	86
1	85	94	97
2	96	89	96
3	80	83	72
4	86	86	83

##2. Write a Pandas program to create and display a DataFrame from a specified dictionary data which has the index labels.

Sample Python dictionary data and list labels:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',  
'Laura', 'Kevin', 'Jonas'],  
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],  
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],  
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

Expected Output:

	attempts	name	qualify	score
a	1	Anastasia	yes	12.5
b	3	Dima	no	9.0
....				
i	2	Kevin	no	8.0
j	1	Jonas	yes	19.0

```
In [10]:
import pandas as pd
import numpy as np

data={'name':['Anastasia','Dima','Katherine','James','Emily','Michael','Matthew','Laura','Kevin','Jonas'],
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],
      'attempts':[1,3,2,3,2,3,1,1,2,1],
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}
labels=['a','b','c','d','e','f','g','h','i','j']
x=pd.DataFrame(data,index = ['a','b','c','d','e','f','g','h','i','j'])
print(x)

      name  score  attempts  qualify
a Anastasia    12.5        1     yes
b Dima         9.0        3      no
c Katherine    16.5        2     yes
d James          NaN        3      no
e Emily          9.0        2      no
f Michael       20.0        3     yes
g Matthew       14.5        1     yes
h Laura          NaN        1      no
i Kevin           8.0        2      no
j Jonas          19.0        1     yes
```

##3. Write a Pandas program to get the first 3 rows of a given DataFrame.

...

Sample Python dictionary data and list labels:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',
'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

Expected Output:

First three rows of the data frame:

	attempts	name	qualify	score
a	1	Anastasia	yes	12.5
b	3	Dima	no	9.0
c	2	Katherine	yes	16.5

```
In [15]: import pandas as pd
import numpy as np

data={'name':['Anastasia','Dima','Katherine','James','Emily','Michael','Matthew','Laura','Kevin','Jonas'],
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],
      'attempts':[1,3,2,3,2,3,1,1,2,1],
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}
labels=['a','b','c','d','e','f','g','h','i','j']
x=pd.DataFrame(data,index = ['a','b','c','d','e','f','g','h','i','j'])
x.head(3)
```

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Katherine	16.5	2	yes

##4. Write a Pandas program to select the 'name' and 'score' columns from the following DataFrame.

Sample Python dictionary data and list labels:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',
'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

Expected Output:

Select specific columns:

	name	score
a	Anastasia	12.5
b	Dima	9.0
c	Katherine	16.5
...		
h	Laura	NaN
i	Kevin	8.0
j	Jonas	19.0

```
In [16]: import pandas as pd
import numpy as np

data={'name':['Anastasia','Dima','Katherine','James','Emily','Michel','Matthew','Laura','Kevin','Jonas'],
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],
      'attempts':[1,3,2,3,2,3,1,1,2,1],
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}
labels=['a','b','c','d','e','f','g','h','i','j']
x=pd.DataFrame(data,columns=['name','score'],index = ['a','b','c','d','e','f','g','h','i','j'])
print(x)

   name  score
a Anastasia    12.5
b Dima         9.0
c Katherine    16.5
d James        NaN
e Emily         9.0
f Michel        20.0
g Matthew       14.5
h Laura        NaN
i Kevin          8.0
j Jonas         19.0
```

##5. Write a Pandas program to select the specified columns and rows from a given data frame.

...

Sample Python dictionary data and list labels:

Select 'name' and 'score' columns in rows 1, 3, 5, 6 from the following data frame.

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',
'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

Expected Output:

Select specific columns and rows:

```
score qualify
b  9.0  no
d  NaN  no
f  20.0 yes
g  14.5 yes
```

```
In [7]: import pandas as pd
import numpy as np

data={'name':['Anastasia','Dima','Katherine','James','Emily','Michel','Matthew','Laura','Kevin','Jonas'],
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],
      'attempts':[1,3,2,3,2,3,1,1,2,1],
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}
labels=['a','b','c','d','e','f','g','h','i','j']
x=pd.DataFrame(data,columns=['score','qualify'],index =labels)
s=x.loc[['b','d','f','g'],['name','score']]
print(s)
```

##6. Write a Pandas program to select the rows where the number of attempts in the examination is greater than 2.

Sample Python dictionary data and list labels:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',  
'Laura', 'Kevin', 'Jonas'],  
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],  
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],  
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']  
...  
...  
...
```

Expected Output:

Number of attempts in the examination is greater than 2:

	name	score	attempts	qualify
b	Dima	9.0	3	no
d	James	NaN	3	no
f	Michael	20.0	3	yes

```
n [34]: import pandas as pd  
import numpy as np  
  
data={'name':['Anastasia','Dima','Katherine','James','Emily','Michael','Matthew','Laura','Kevin','Jonas'],  
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],  
      'attempts':[1,3,2,3,2,3,1,1,2,1],  
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}  
labels=['a','b','c','d','e','f','g','h','i','j']  
x=pd.DataFrame(data,index =labels)  
x=x[x['attempts']>2]  
print(x)  
  
      name  score  attempts  qualify  
b    Dima    9.0        3     no  
d   James    NaN        3     no  
f  Michael   20.0        3    yes
```

##7. Write a Pandas program to select the rows where the score is missing, i.e. is NaN.

Sample Python dictionary data and list labels:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',  
'Laura', 'Kevin', 'Jonas'],  
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],  
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],  
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']  
...  
...  
...
```

Expected Output:

Rows where score is missing:

```
  attempts  name qualify  score
d      3  James    no   NaN
h      1  Laura    no   NaN
```

...

```
In [35]: import pandas as pd
import numpy as np

data={'name':['Anastasia','Dima','Katherine','James','Emily','Michel','Matthew','Laura','Kevin','Jonas'],
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],
      'attempts':[1,3,2,3,2,3,1,1,2,1],
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}
labels=['a','b','c','d','e','f','g','h','i','j']
x=pd.DataFrame(data,index =labels)
x=x[x['score'].isnull()]
print(x)

      name  score  attempts  qualify
d  James    NaN        3     no
h  Laura    NaN        1     no
```

##8. Write a Pandas program to select the rows the score is between 15 and 20 (inclusive).

...

Sample Python dictionary data and list labels:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',
'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
...
```

```

Expected Output:

Rows where score between 15 and 20 (inclusive):

```
  attempts  name qualify  score
c      2  Katherine  yes  16.5
f      3  Michael   yes  20.0
j      1  Jonas     yes  19.0
```

```

...

```
In [36]: import pandas as pd
import numpy as np

data={'name':['Anastasia','Dima','Katherine','James','Emily','Michel','Matthew','Laura','Kevin','Jonas'],
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],
      'attempts':[1,3,2,3,2,3,1,1,2,1],
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}
labels=['a','b','c','d','e','f','g','h','i','j']
x=pd.DataFrame(data,index =labels)
x=x[x['score'].between(15,20)]
print(x)

      name  score  attempts  qualify
c  Katherine  16.5        2     yes
f    Michel   20.0        3     yes
j    Jonas    19.0        1     yes
```

##9. Write a Pandas program to select the rows where number of attempts in the examination is less than 2 and score greater than 15.

...

Sample Python dictionary data and list labels:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',  
'Laura', 'Kevin', 'Jonas'],  
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],  
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],  
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']  
...  
...
```

Expected Output:

Number of attempts in the examination is less than 2 and score greater than 15 :

name	score	attempts	qualify
j Jonas	19.0	1	yes

```
In [2]: import pandas as pd  
import numpy as np  
  
data={'name':['Anastasia','Dima','Katherine','James','Emily','Michael','Matthew','Laura','Kevin','Jonas'],  
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],  
      'attempts':[1,3,2,3,2,3,1,1,2,1],  
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}  
labels=['a','b','c','d','e','f','g','h','i','j']  
x=pd.DataFrame(data,index =labels)  
x=x[(x['attempts']<2)&(x['score']>15)]  
print(x)  
  
          name  score  attempts qualify  
j  Jonas    19.0        1     yes
```

##10. Write a Pandas program to change the score in row 'd' to 11.5.

...

Sample Python dictionary data and list labels:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',  
'Laura', 'Kevin', 'Jonas'],  
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],  
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],  
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

Expected Output:

Change the score in row 'd' to 11.5:

attempts	name	qualify	score
----------	------	---------	-------

a	1	Anastasia	yes	12.5
b	3	Dima	no	9.0
c	2	Katherine	yes	16.5
...				
i	2	Kevin	no	8.0
j	1	Jonas	yes	19.0

```
In [38]: import pandas as pd
import numpy as np

data={'name':['Anastasia','Dima','Katherine','James','Emily','Michel','Matthew','Laura','Kevin','Jonas'],
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],
      'attempts':[1,3,2,3,2,3,1,1,2,1],
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}
labels=['a','b','c','d','e','f','g','h','i','j']
x=pd.DataFrame(data,index =labels)
x.loc['d','score']=11.5
print(x)

   name  score  attempts  qualify
a Anastasia    12.5        1     yes
b Dima         9.0        3      no
c Katherine    16.5        2     yes
d James        11.5        3      no
e Emily         9.0        2      no
f Michel        20.0        3     yes
g Matthew       14.5        1     yes
h Laura        NaN        1      no
i Kevin         8.0        2      no
j Jonas         19.0        1     yes
```

##11. Write a Pandas program to calculate the sum of the examination attempts by the students.

...

Sample Python dictionary data and list labels:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',
'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

Expected Output:

Sum of the examination attempts by the students:

```
In [41]: import pandas as pd
import numpy as np

data={'name':['Anastasia','Dima','Katherine','James','Emily','Michel','Matthew','Laura','Kevin','Jonas'],
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],
      'attempts':[1,3,2,3,2,3,1,1,2,1],
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}
labels=['a','b','c','d','e','f','g','h','i','j']
x=pd.DataFrame(data,index =labels)
x['attempts'].sum()

Out[41]: 19
```

##12. Write a Pandas program to calculate the mean of all students' scores. Data is stored in a dataframe.

...

Sample Python dictionary data and list labels:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',
'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

Expected Output:

Mean score for each different student in data frame:

13.5625

```
In [42]: import pandas as pd
import numpy as np

data={'name':['Anastasia','Dima','Katherine','James','Emily','Michel','Matthew','Laura','Kevin','Jonas'],
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],
      'attempts':[1,3,2,3,2,3,1,1,2,1],
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}
labels=['a','b','c','d','e','f','g','h','i','j']
x=pd.DataFrame(data,index =labels)
x['score'].mean()

Out[42]: 13.5625
```

##13. Write a Pandas program to replace the 'qualify' column contains the values 'yes' and 'no' with True and False.

...

Sample Python dictionary data and list labels:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',
'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

Expected Output:

Replace the 'qualify' column contains the values 'yes' and 'no' with True and False:

	attempts	name	qualify	score
a	1	Anastasia	True	12.5
b	3	Dima	False	9.0
.....				
i	2	Kevin	False	8.0
j	1	Jonas	True	19.0

```
In [44]:  
import pandas as pd  
import numpy as np  
  
data={'name':['Anastasia','Dima','Katherine','James','Emily','Michel','Matthew','Laura','Kevin','Jonas'],  
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],  
      'attempts':[1,3,2,3,2,3,1,1,2,1],  
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}  
labels=['a','b','c','d','e','f','g','h','i','j']  
x=pd.DataFrame(data,index =labels)  
x[['qualify']]>=x[['qualify']].map({'yes':True,'no':False})  
print(x)  
  
          name  score  attempts  qualify  
a  Anastasia    12.5        1     True  
b      Dima      9.0        3    False  
c  Katherine    16.5        2     True  
d      James      NaN        3    False  
e      Emily      9.0        2    False  
f      Michel     20.0        3     True  
g      Matthew    14.5        1     True  
h      Laura      NaN        1    False  
i      Kevin      8.0        2    False  
j      Jonas     19.0        1     True
```

##14. Write a Pandas program to change the name 'James' to 'Suresh' in name column of the DataFrame.

Sample Python dictionary data and list labels:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',  
'Laura', 'Kevin', 'Jonas'],  
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],  
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],  
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}  
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']  
...  
...
```

Expected Output:

Change the name 'James' to \?Suresh\?:

	attempts	name	qualify	score
a	1	Anastasia	yes	12.5

b 3 Dima no 9.0

.....

i 2 Kevin no 8.0

j 1 Jonas yes 19.0

...

```
In [47]: import pandas as pd
import numpy as np

data={'name':['Anastasia','Dima','Katherine','James','Emily','Michel','Matthew','Laura','Kevin','Jonas'],
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],
      'attempts':[1,3,2,3,2,3,1,1,2,1],
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}
labels=['a','b','c','d','e','f','g','h','i','j']
x=pd.DataFrame(data,index =labels)
x['name']=x['name'].replace('James','Suresh')
print(x)

      name  score  attempts  qualify
a  Anastasia    12.5        1     yes
b       Dima     9.0        3      no
c  Katherine    16.5        2     yes
d      Suresh     NaN        3      no
e      Emily     9.0        2      no
f      Michel    20.0        3     yes
g      Matthew   14.5        1     yes
h      Laura     NaN        1      no
i      Kevin     8.0        2      no
j      Jonas    19.0        1     yes
```

##15. Write a Pandas program to delete the 'attempts' column from the DataFrame.

...

Sample Python dictionary data and list labels:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',
'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
...
```

Expected Output:

Delete the 'attempts' column from the data frame:

```
      name  qualify  score
a  Anastasia    yes   12.5
b      Dima     no    9.0
.....
i      Kevin     no    8.0
j      Jonas    yes   19.0
...
```

```
In [48]: import pandas as pd
import numpy as np

data={'name':['Anastasia','Dima','Katherine','James','Emily','Michel','Matthew','Laura','Kevin','Jonas'],
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],
      'attempts':[1,3,2,3,2,3,1,1,2,1],
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}
labels=['a','b','c','d','e','f','g','h','i','j']
x=pd.DataFrame(data,index =labels)
x.pop('attempts')
print(x)

   name  score qualify
a Anastasia    12.5     yes
b Dima         9.0      no
c Katherine    16.5     yes
d James        NaN      no
e Emily        9.0      no
f Michel       20.0     yes
g Matthew      14.5     yes
h Laura        NaN      no
i Kevin         8.0      no
j Jonas        19.0     yes
```

##16. Write a Pandas program to insert a new column in existing DataFrame.

Sample Python dictionary data and list labels:

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew',
'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

Expected Output:

New DataFrame after inserting the 'color' column

	attempts	name	qualify	score	color
a	1	Anastasia	yes	12.5	Red
b	3	Dima	no	9.0	Blue
.....					
i	2	Kevin	no	8.0	Green
j	1	Jonas	yes	19.0	Red

```
In [51]: import pandas as pd
import numpy as np

data={'name':['Anastasia','Dima','Katherine','James','Emily','Michel','Matthew','Laura','Kevin','Jonas'],
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],
      'attempts':[1,3,2,3,2,3,1,1,2,1],
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}
labels=['a','b','c','d','e','f','g','h','i','j']
color=['red','black','blue','red','green','blue','yellow','red','red','blue']
x=pd.DataFrame(data,index =labels)
x['color']= color
print(x)

   name  score  attempts qualify  color
a Anastasia    12.5        1     yes    red
b Dima         9.0         3      no  black
c Katherine    16.5        2     yes    blue
d James          NaN        3      no    red
e Emily         9.0         2      no  green
f Michel        20.0        3     yes    blue
g Matthew       14.5        1     yes  yellow
h Laura          NaN        1      no    red
i Kevin          8.0         2      no    red
j Jonas         19.0        1     yes    blue
```

##17. Write a Pandas program to rename columns of a given DataFrame

...

Sample data:

Original DataFrame

	col1	col2	col3
0	1	4	7
1	2	5	8
2	3	6	9

New DataFrame after renaming columns:

	Column1	Column2	Column3
0	1	4	7
1	2	5	8
2	3	6	9

...

```
In [61]: x=pd.DataFrame({'col1':[1,2,3], 'col2':[4,5,6], 'col3':[7,8,9]})
print(x)
x.rename(columns={'col1': 'column1','col2':'column2','col3':'column3'},inplace=True)
print(x)

   col1  col2  col3
0     1     4     7
1     2     5     8
2     3     6     9
   column1  column2  column3
0       1       4       7
1       2       5       8
2       3       6       9
```

##18. Write a Pandas program to select rows from a given DataFrame based on values in some columns.

...

Sample data:

Original DataFrame

	col1	col2	col3
0	1	4	7
1	4	5	8
2	3	6	9
3	4	7	0
4	5	8	1

Rows for column1 value == 4

	col1	col2	col3
1	4	5	8
3	4	7	0

..

```
In [63]: x=pd.DataFrame({'col1':[1,4,3,4,5], 'col2':[4,5,6,7,8], 'col3':[7,8,9,0,1]})  
print(x)  
rows=x[x['col1']==4]  
print('rows for col1 value==4')  
print(rows)  
  
          col1  col2  col3  
0        1      4      7  
1        4      5      8  
2        3      6      9  
3        4      7      0  
4        5      8      1  
rows for col1 value==4  
          col1  col2  col3  
1        4      5      8  
3        4      7      0
```

##19. Write a Pandas program to add one row in an existing DataFrame.

...

Sample data:

Original DataFrame

	col1	col2	col3
0	1	4	7
1	4	5	8
2	3	6	9
3	4	7	0
4	5	8	1

After add one row:

	col1	col2	col3
0	1	4	7
1	4	5	8
2	3	6	9
3	4	7	0

```
4   5   8   1  
5   10  11  12  
...
```

```
In [14]: x=pd.DataFrame({'col1':[1,4,3,4,5],'col2':[4,5,6,7,8],'col3':[7,8,9,0,1]})  
print(x)  
  
new_row={'col1': 10,'col2':11,'col3':12}  
s=x.append(new_row,ignore_index=True)  
  
print(x)
```

	col1	col2	col3
0	1	4	7
1	4	5	8
2	3	6	9
3	4	7	0
4	5	8	1
5	10	11	12

##20. Write a Pandas program to replace all the NaN values with Zero's in a column of a dataframe.

...

Sample data:

Original DataFrame

```
attempts      name qualify  score  
0      1  Anastasia    yes  12.5  
1      3      Dima     no   9.0  
2      2 Katherine    yes  16.5  
.....  
8      2      Kevin     no   8.0  
9      1      Jonas    yes  19.0
```

New DataFrame replacing all NaN with 0:

```
attempts      name qualify  score  
0      1  Anastasia    yes  12.5  
1      3      Dima     no   9.0  
2      2 Katherine    yes  16.5  
.....  
8      2      Kevin     no   8.0  
9      1      Jonas    yes  19.0  
...
```

```
In [74]: import pandas as pd
import numpy as np

data={'name':['Anastasia','Dima','Katherine','James','Emily','Michel','Matthew','Laura','Kevin','Jonas'],
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],
      'attempts':[1,3,2,3,2,3,1,1,2,1],
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}
labels=['a','b','c','d','e','f','g','h','i','j']
x=pd.DataFrame(data)
print(x)
x=x.fillna(0)
print(x)

   name  score  attempts qualify
0 Anastasia    12.5        1     yes
1 Dima         9.0        3      no
2 Katherine    16.5        2     yes
3 James          NaN        3      no
4 Emily         9.0        2      no
5 Michel        20.0        3     yes
6 Matthew       14.5        1     yes
7 Laura          NaN        1      no
8 Kevin          8.0        2      no
9 Jonas         19.0        1     yes
   name  score  attempts qualify
0 Anastasia    12.5        1     yes
1 Dima         9.0        3      no
2 Katherine    16.5        2     yes
3 James          0.0        3      no
```

##21. Write a Pandas program to count the NaN values in one or more columns in DataFrame.

...

Sample data:

Original DataFrame

	attempts	name	qualify	score
0	1	Anastasia	yes	12.5
1	3	Dima	no	9.0
2	2	Katherine	yes	16.5
3	3	James	no	NaN
4	2	Emily	no	9.0
5	3	Michael	yes	20.0
6	1	Matthew	yes	14.5
7	1	Laura	no	NaN
8	2	Kevin	no	8.0
9	1	Jonas	yes	19.0

Number of NaN values in one or more columns:

2

```
In [79]: import pandas as pd
import numpy as np

data={'name':['Anastasia','Dima','Katherine','James','Emily','Michel','Matthew','Laura','Kevin','Jonas'],
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],
      'attempts':[1,3,2,3,2,3,1,1,2,1],
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}
labels=['a','b','c','d','e','f','g','h','i','j']
x=pd.DataFrame(data)
print(x)
x.isnull().values.sum()

          name  score  attempts  qualify
0  Anastasia    12.5        1      yes
1       Dima     9.0        3       no
2  Katherine    16.5        2      yes
3       James     NaN        3       no
4       Emily     9.0        2       no
5       Michel    20.0        3      yes
6     Matthew    14.5        1      yes
7       Laura     NaN        1       no
8       Kevin     8.0        2       no
9       Jonas    19.0        1      yes

Out[79]: 2
```

##22. Write a Pandas program to drop a list of rows from a specified DataFrame.

Sample data:

Original DataFrame

	col1	col2	col3
0	1	4	7
1	4	5	8
2	3	6	9
3	4	7	0
4	5	8	1

New DataFrame after removing 2nd & 4th rows:

	col1	col2	col3
0	1	4	7
1	4	5	8
3	4	7	0

```
In [80]: x=pd.DataFrame({'col1':[1,4,3,4,5],'col2':[4,5,6,7,8],'col3':[7,8,9,0,1]})
drope=[2,4]
x=x.drop(drope)
print(x)

          col1  col2  col3
0           1     4     7
1           4     5     8
3           4     7     0
```

##23. . Write a Pandas program to convert DataFrame column type from string to datetime.

Sample data:

String Date:

```
0 3/11/2000  
1 3/12/2000  
2 3/13/2000  
dtype: object
```

Original DataFrame (string to datetime):

```
0  
0 2000-03-11  
1 2000-03-12  
2 2000-03-13  
...
```

```
In [84]: import pandas as pd  
x=pd.DataFrame({'date':['3/11/2000','3/12/2000','3/13/2000']})  
print(x)  
x['date']=pd.to_datetime(x['date'])  
print(x)
```

	date
0	3/11/2000
1	3/12/2000
2	3/13/2000

	date
0	2000-03-11
1	2000-03-12
2	2000-03-13

##24. Write a Pandas program to find the row for where the value of a given column is maximum.

Sample Output:

Original DataFrame

	col1	col2	col3
0	1	4	7
1	2	5	8
2	3	6	12
3	4	9	1
4	7	5	11

Row where col1 has maximum value:

4

Row where col2 has maximum value:

3

Row where col3 has maximum value:

2

...

```
In [85]: import pandas as pd
x=pd.DataFrame({'col1':[1,2,3,4,7], 'col2':[4,5,6,9,5], 'col3':[7,8,12,1,11]})
print(x)
print("max value for col1",x['col1'].argmax())
print("max value for col2",x['col2'].argmax())
print("max value for col3",x['col3'].argmax())

      col1  col2  col3
0       1      4      7
1       2      5      8
2       3      6     12
3       4      9      1
4       7      5     11
max value for col1 4
max value for col2 3
max value for col3 2
```

##25. Write a Pandas program to get the datatypes of columns of a DataFrame.

## Sample data:

### Original DataFrame:

	attempts	name	qualify	score
0	1	Anastasia	yes	12.5
1	3	Dima	no	9.0
.....				
8	2	Kevin	no	8.0
9	1	Jonas	yes	19.0

Data types of the columns of the said DataFrame:

```
attempts    int64
name        object
qualify     object
score      float64
dtype: object
..
```

```
In [86]: import pandas as pd
import numpy as np

data={'name':['Anastasia','Dima','Katherine','James','Emily','Michel','Matthew','Laura','Kevin','Jonas'],
      'score':[12.5,9,16.5,np.nan,9,20,14.5,np.nan,8,19],
      'attempts':[1,3,2,3,2,3,1,1,2,1],
      'qualify':['yes','no','yes','no','no','yes','yes','no','no','yes']}
labels=['a','b','c','d','e','f','g','h','i','j']

x=pd.DataFrame(data)
print(x.dtypes)

name      object
score     float64
attempts   int64
qualify    object
dtype: object
```

##26. Write a Pandas program to group by the first column and get second column as lists in rows.

...

Sample data:

Original DataFrame

col1 col2

0 C1 1

1 C1 2

2 C2 3

3 C2 3

4 C2 4

5 C3 6

6 C2 5

Group on the col1:

col1

C1 [1, 2]

C2 [3, 3, 4, 5]

C3 [6]

Name: col2, dtype: object

```
In [90]: import pandas as pd
x=pd.DataFrame({'col2':['c1','c1','c2','c2','c3','c1'],'col2':[1,2,3,3,4,5]})
print(x)
x=x.groupby('col1')['col2'].apply(list)
print(x)
```

col1	col2
0	[1]
1	[2]
2	[3]
3	[3]
4	[4]
5	[6]
6	[5]

##27 Write a Pandas program to count number of columns of a DataFrame.

...

Sample Output:

Original DataFrame

col1 col2 col3

0 1 4 7

1 2 5 8

2 3 6 12

3 4 9 1

4 7 5 11

Number of columns:

3

```
In [92]: import pandas as pd
x=pd.DataFrame({'col1':[1,2,3,4,7], 'col2':[4,5,6,9,5], 'col3':[7,8,12,1,11]})
print(x)
number_of_columns=len(x.columns)
print(number_of_columns)

    col1  col2  col3
0      1      4      7
1      2      5      8
2      3      6     12
3      4      9      1
4      7      5     11
5
```

##28. Write a Pandas program to get first n records of a DataFrame.

...

Sample Output:

Original DataFrame

col1 col2 col3

0 1 4 7

1 2 5 5

2 3 6 8

3 4 9 12

4 7 5 1

5 11 0 11

First 3 rows of the said DataFrame':

col1 col2 col3

0 1 4 7

1 2 5 5

2 3 6 8

...

```
In [95]: import pandas as pd
x=pd.DataFrame({'col1':[1,2,3,4,7,11], 'col2':[4,5,6,9,5,0], 'col3':[7,5,8,12,1,11]})
print(x)
y=x.iloc[0:3]
print(y)

    col1  col2  col3
0      1      4      7
1      2      5      5
2      3      6      8
3      4      9     12
4      7      5      1
5     11      0     11
    col1  col2  col3
0      1      4      7
1      2      5      5
2      3      6      8
```

##29. Write a Pandas program to get last n records of a DataFrame.

Sample Output:

Original DataFrame

```
col1 col2 col3
0 1 4 7
1 2 5 5
2 3 6 8
3 4 9 12
4 7 5 1
5 11 0 11
```

Last 3 rows of the said DataFrame':

```
col1 col2 col3
3 4 9 12
4 7 5 1
5 11 0 11
'''
```

```
In [96]: import pandas as pd
x=pd.DataFrame({'col1':[1,2,3,4,7,11], 'col2':[4,5,6,9,5,0], 'col3':[7,5,8,12,1,11]})
print(x)
y=x.iloc[3:]
print(y)

      col1  col2  col3
0      1     4     7
1      2     5     5
2      3     6     8
3      4     9    12
4      7     5     1
5     11     0    11
      col1  col2  col3
3      4     9    12
4      7     5     1
5     11     0    11
```

##30. Write a Pandas program to get topmost n records within each group of a DataFrame.

Sample Output:

Original DataFrame

```
col1 col2 col3
0 1 4 7
1 2 5 5
2 3 6 8
3 4 9 12
4 7 5 1
5 11 0 11
```

topmost n records within each group of a DataFrame:

col1 col2 col3

5 11 0 11

4 7 5 1

3 4 9 12

col1 col2 col3

3 4 9 12

2 3 6 8

1 2 5 5

4 7 5 1

col1 col2 col3

3 4 9 12

5 11 0 11

2 3 6 8

...

```
In [99]: import pandas as pd  
x=pd.DataFrame({'col1':[1,2,3,4,7,11],'col2':[4,5,6,9,5,0],'col3':[7,5,8,12,1,11]})
```

```
print(x)
```

```
x1=x.nlargest(3,'col1')
```

```
print(x1)
```

```
x2=x.nlargest(3,'col2')
```

```
print(x2)
```

```
x3=x.nlargest(3,'col3')
```

```
print(x3)
```

	col1	col2	col3
0	1	4	7
1	2	5	5
2	3	6	8
3	4	9	12
4	7	5	1
5	11	0	11

```
2 86 96 89 96  
3 80 80 83 72  
4 66 86 86 83
```

Add prefix:

```
A_W A_X A_Y A_Z  
0 68 78 84 86  
1 75 85 94 97  
2 86 96 89 96  
3 80 80 83 72  
4 66 86 86 83
```

Add suffix:

```
W_1 X_1 Y_1 Z_1  
0 68 78 84 86  
1 75 85 94 97  
2 86 96 89 96  
3 80 80 83 72  
4 66 86 86 83
```

...

```
In [105]: import pandas as pd  
d={'W':[78,85,96,80,66], 'X':[78,85,96,80,86], 'Y':[84,94,89,83,86], 'Z':[86,97,96,72,83]}  
x=pd.DataFrame(d)  
print(x)  
print(x.add_prefix("A_"))  
  
print(x.add_suffix("_1"))
```

	W	X	Y	Z
0	78	78	84	86
1	85	85	94	97
2	96	96	89	96
3	80	80	83	72
4	66	86	86	83

	A_W	A_X	A_Y	A_Z
0	78	78	84	86
1	85	85	94	97
2	96	96	89	96
3	80	80	83	72
4	66	86	86	83

	W_1	X_1	Y_1	Z_1
0	78	78	84	86
1	85	85	94	97
2	96	96	89	96
3	80	80	83	72
4	66	86	86	83

##32. Write a Pandas program to convert continuous values of a column in a given DataFrame to categorical.

...

Input:

```
{ 'Name': ['Alberto Franco','Gino Mcneill','Ryan Parkes', 'Eesha Hinton', 'Syed Wharton'],
```

```
'Age': [18, 22, 40, 50, 80, 5] }
```

Output:

Age group:

0 kids

1 adult

2 elderly

3 adult

4 elderly

5 kids

Name: age\_groups, dtype: category

Categories (3, object): [kids < adult < elderly]

...

```
In [9]: import pandas as pd
x=pd.DataFrame({'Name':['Alberto Franco','Gino McNeill','Ryan Parkes','Eesha Hinton','Syed Wharton','Kierra Gentry'],
'Age':[18, 22, 40, 50, 80,5]})

age_groups=[0,18,60,100]
age_labels=['kids', 'adult', 'elderly']
x['age_groups']=pd.cut(x['Age'],bins=age_groups,labels=age_labels)
print(x['age_groups'])

0      kids
1    adult
2    adult
3    adult
4  elderly
5      kids
Name: age_groups, dtype: category
Categories (3, object): ['kids' < 'adult' < 'elderly']
```

##33. Write a Pandas program to append rows to an existing DataFrame and display the combined data.

...

Test Data:

student\_data1

	student_id	name	marks
0	S1	Danniella Fenton	200
1	S2	Ryder Storey	210
2	S3	Bryce Jensen	190
3	S4	Ed Bernal	222
4	S5	Kwame Morin	199

New Row(s)

student_id	name	marks
S6	Scarlette Fisher	205

dtype: object

```
In [5]: import pandas as pd
stdata1=pd.DataFrame({'student_id':['S1','S2','S3','S4','S5'],
                      'name':['Danniella Fenton','Ryder Storey','Bryce Fisher','Ed Bernal','Kwame Morine'],
                      'marks':[200,210,198,222,199]})
new_row=pd.Series({'student_id':'S6','name':'Scarlette Fisher','marks':205})
stdata2=stdata1.append(new_row,ignore_index=True)
print(stdata2)
```

##34 Write a Pandas program to join the two given dataframes along rows and merge with another dataframe along the common column id.

Test Data:

student\_data1:

	student_id	name	marks
0	S1	Danniella Fenton	200
1	S2	Ryder Storey	210
2	S3	Bryce Jensen	190
3	S4	Ed Bernal	222
4	S5	Kwame Morin	199

student\_data2:

	student_id	name	marks
0	S4	Scarlette Fisher	201
1	S5	Carla Williamson	200
2	S6	Dante Morse	198
3	S7	Kaiser William	219
4	S8	Madeeha Preston	201

...

exam\_data:

student\_id exam\_id

0	S1	23
1	S2	45
2	S3	12
3	S4	67
4	S5	21
5	S7	55
6	S8	33
7	S9	14
8	S10	56
9	S11	83
10	S12	88
11	S13	12

```
In [12]: import pandas as pd
stdata1=pd.DataFrame({'student_id':['S1','S2','S3','S4','S5'],
                      'name':['Danniella Fenton','Ryder Storey','Bryce Fisher','Ed Bernal','Kwame Morine'],
                      'marks':[200,210,190,222,199]})
stdata2=pd.DataFrame({'student_id':['S4','S5','S6','S7','S8'],
                      'name':['Scarlette Fisher','Carla Williamson','Dante Morse','Kaiser William','Madeeha Preston'],
                      'marks':[201,200,198,219,201]})
exame_data=pd.DataFrame({'student_id':['S1','S2','S3','S4','S5','S7','S8','S9','S10','S11','S12','S13'],
                         'exame_id':[23,45,12,67,21,55,33,14,56,83,88,12]})

print(stdata1)
print(stdata2)
new=pd.concat([stdata1,stdata2])
print(new)
result=pd.merge(new,exame_data,on='student_id')
print(result)
```

	student_id	name	marks
0	S1	Danniella Fenton	200
1	S2	Ryder Storey	210
2	S3	Bryce Fisher	190
3	S4	Ed Bernal	222
4	S5	Kwame Morine	199

  

	student_id	name	marks
0	S4	Scarlette Fisher	201
1	S5	Carla Williamson	200
2	S6	Dante Morse	198
3	S7	Kaiser William	219
4	S8	Madeeha Preston	201

	student_id	name	marks
0	S1	Danniella Fenton	200
1	S2	Ryder Storey	210
2	S3	Bryce Fisher	190
3	S4	Ed Bernal	222
4	S5	Kwame Morine	199

  

	student_id	name	marks
0	S4	Scarlette Fisher	201
1	S5	Carla Williamson	200
2	S6	Dante Morse	198
3	S7	Kaiser William	219
4	S8	Madeeha Preston	201

  

	student_id	name	marks
0	S1	Danniella Fenton	200
1	S2	Ryder Storey	210
2	S3	Bryce Fisher	190
3	S4	Ed Bernal	222
4	S5	Kwame Morine	199

  

	student_id	name	marks	exame_id
0	S1	Danniella Fenton	200	23
1	S2	Ryder Storey	210	45
2	S3	Bryce Fisher	190	12
3	S4	Ed Bernal	222	67
4	S4	Scarlette Fisher	201	67
5	S5	Kwame Morine	199	21

The image shows a Jupyter Notebook cell containing two data frames. The first data frame has four columns: student\_id, name, marks, and exame\_id. The second data frame has three columns: student\_id, name, and marks.

	student_id	name	marks	exame_id
0	S1	Danniella Fenton	200	23
1	S2	Ryder Storey	210	45
2	S3	Bryce Fisher	190	12
3	S4	Ed Bernal	222	67
4	S5	Kwame Morine	199	21
0	S4	Scarlette Fisher	201	21
1	S5	Carla Williamson	200	55
2	S6	Dante Morse	198	33
3	S7	Kaiser William	219	
4	S8	Madeeha Preston	201	

	student_id	name	marks
0	S1	Danniella Fenton	200
1	S2	Ryder Storey	210
2	S3	Bryce Fisher	190
3	S4	Ed Bernal	222
4	S5	Scarlette Fisher	201
5	S5	Kwame Morine	199
6	S5	Carla Williamson	200
7	S7	Kaiser William	219
8	S8	Madeeha Preston	201