**Numpy**

**1.Sort a given array of shape 2 along the first axis, last axis and on flattened array**

**Original array:**

**[[10 40]**

**[30 20]]**

**Expected Output:**

**Sort the array along the first axis:**

**[[10 20]**

**[30 40]]**

**Sort the array along the last axis:**

**[[10 40]**

**[20 30]]**

**Sort the flattened array:**

**[10 20 30 40]**

**Python Code:**

import numpy as np

a = np.array([[10,40],[30,20]])

print("Original array:")

print(a)

print("Sort the array along the first axis:")

print(np.sort(a, axis=0))

print("Sort the array along the last axis:")

print(np.sort(a))

print("Sort the flattened array:")

print(np.sort(a, axis=None))

2.**Write a NumPy program to repeat all the elements three times of a given array of string**

**Original Array:**

**['Python' 'PHP' 'Java' 'C++']**

**Expected Output:**

**New array:**

**['PythonPythonPython' 'PHPPHPPHP' 'JavaJavaJava' 'C++C++C++']**

import numpy as np

x1 = np.array(['Python', 'PHP', 'Java', 'C++'], dtype=np.str)

print("Original Array:")

print(x1)

new\_array = np.char.multiply(x1, 3)

print("New array:")

print(new\_array)

**3.Write a NumPy program to count the number of "P" in a given array, element-wise.**

Original Array:  
['Python' 'PHP' 'JS' 'examples' 'html']

Sample Output:

Number of 'P':  
[1 2 0 0 0]

**Sample Solution**:-

**Python Code:**

import numpy as np

x1 = np.array(['Python', 'PHP', 'JS', 'examples', 'html'], dtype=np.str)

print("\nOriginal Array:")

print(x1)

print("Number of ‘P’:")

r = np.char.count(x1, "P")

print(r)

4.Write a NumPy program to count the number of days of specific month.   
Expected Output:  
Number of days, February, 2016:  
29 days  
Number of days, February, 2017:  
28 days  
Number of days, February, 2018:  
28 days

**Sample Solution**:-

**Python Code:**

import numpy as np

print("Number of days, February, 2016: ")

print(np.datetime64('2016-03-01') - np.datetime64('2016-02-01'))

print("Number of days, February, 2017: ")

print(np.datetime64('2017-03-01') - np.datetime64('2017-02-01'))

print("Number of days, February, 2018: ")

print(np.datetime64('2018-03-01') - np.datetime64('2018-02-01'))

**5.** Write a NumPy program to concatenate element-wise two arrays of string.   
Expected Output:  
Array1:  
['Python' 'PHP']  
Array2:  
[' Java' ' C++']  
new array:  
['Python Java' 'PHP C++']

**Sample Solution**:-

**Python Code:**

import numpy as np

x1 = np.array(['Python', 'PHP'], dtype=np.str)

x2 = np.array([' Java', ' C++'], dtype=np.str)

print("Array1:")

print(x1)

print("Array2:")

print(x2)

new\_array = np.char.add(x1, x2)

print("new array:")

print(new\_array)

6.Write a NumPy program to create a structured array from given student name, height, class and their data types. Now sort by class, then height if class are equal.   
Expected Output:  
Original array:  
[(b'James', 5, 48.5 ) (b'Nail', 6, 52.5 ) (b'Paul', 5, 42.1 ) (b'Pit', 5, 40.11)]  
Sort by age, then height if class are equal:  
[(b'Pit', 5, 40.11) (b'Paul', 5, 42.1 ) (b'James', 5, 48.5 ) (b'Nail', 6, 52.5 )]

**Sample Solution**:

**Python Code:**

import numpy as np

data\_type = [('name', 'S15'), ('class', int), ('height', float)]

students\_details = [('James', 5, 48.5), ('Nail', 6, 52.5),('Paul', 5, 42.10), ('Pit', 5, 40.11)]

# create a structured array

students = np.array(students\_details, dtype=data\_type)

print("Original array:")

print(students)

print("Sort by class, then height if class are equal:")

print(np.sort(students, order=['class', 'height']))

7.Write a NumPy program to get the dates of yesterday, today and tomorrow.   
Sample Output:  
Yestraday: 2017-03-24  
Today: 2017-03-25  
Tomorrow: 2017-03-26

**Sample Solution**:-

**Python Code:**

import numpy as np

yesterday = np.datetime64('today', 'D') - np.timedelta64(1, 'D')

print("Yestraday: ",yesterday)

today = np.datetime64('today', 'D')

print("Today: ",today)

tomorrow = np.datetime64('today', 'D') + np.timedelta64(1, 'D')

print("Tomorrow: ",tomorrow)

8.Write a NumPy program to make all the elements of a given string a numeric string of 5 digits with zeros on its left

**Sample Solution**:-

**Python Code:**

import numpy as np

x = np.array(['2', '11', '234', '1234', '12345'], dtype=np.str)

print("\nOriginal Array:")

print(x)

r = np.char.zfill(x, 5)

print("\nNumeric string of 5 digits with zeros:")

print(r)

Sample Input:

['2', '11', '234', '1234', '12345']

Sample Output:

Numeric string of 5 digits with zeros:

['00002' '00011' '00234' '01234' '12345']

9.Write a NumPy program to convert numpy datetime64 to Timestamp.   
Sample output:  
Current date:  
2017-04-01 08:01:12.722055  
Timestamp:  
1491033672.72  
UTC from Timestamp:  
2017-04-01 08:01:12.722055

**Python Code:**

import numpy as np

from datetime import datetime

dt = datetime.utcnow()

print("Current date:")

print(dt)

dt64 = np.datetime64(dt)

ts = (dt64 - np.datetime64('1970-01-01T00:00:00Z')) / np.timedelta64(1, 's')

print("Timestamp:")

print(ts)

print("UTC from Timestamp:")

print(datetime.utcfromtimestamp(ts))

**Panda**

Write a Python 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

2. Write a Python 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

3. Write a Python program to add, subtract, multiple and divide two Pandas Series.   
Sample Series: [2, 4, 6, 8, 10], [1, 3, 5, 7, 9]

4. 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

**01Dec**

1.Write a Pandas program to display a summary of the basic information about a specified DataFrame and its data. [Go to the editor](https://www.w3resource.com/python-exercises/pandas/index-dataframe.php#EDITOR)  
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:  
Summary of the basic information about this DataFrame and its data:  
<class 'pandas.core.frame.DataFrame'>  
Index: 10 entries, a to j  
Data columns (total 4 columns):  
.... dtypes: float64(1), int64(1), object(2)  
memory usage: 400.0+ bytes  
None

2.Write a Pandas program to select the 'name' and 'score' columns from the following DataFrame. [Go to the editor](https://www.w3resource.com/python-exercises/pandas/index-dataframe.php#EDITOR)  
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

3.Write a Pandas program to select the rows where the score is missing, i.e. is NaN.  
*Sample DataFrame*:  
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']

4.Write a Pandas program to calculate the mean score for each different student in data frame.  
*Sample DataFrame*:  
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']

5.Write a Pandas program to convert the datatype of a given column(floats to ints).

6.Write a Pandas program to get column index from column name of a given DataFrame.

7.Write a Pandas program to replace the ‘qualify' column contains the values 'yes' and 'no' with True and False.  
*Sample DataFrame*:  
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']  
Values for each column will be:  
name : ‘Suresh’, score: 15.5, attempts: 1, qualify: ‘yes’, label: ‘k’

8.Write a Pandas program to write a DataFrame to CSV file using tab separator.  
*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  
Data from new\_file.csv file:  
col1\tcol2\tcol3  
0 1\t4\t7  
1 4\t5\t8  
2 3\t6\t9  
3 4\t7\t0  
4 5\t8\t1

9.Write a Pandas program to convert index in a column of the given 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  
After converting index in a column:  
index attempts name qualify score  
0 0 1 Anastasia yes 12.5  
1 1 3 Dima no 9.0  
2 2 2 Katherine yes 16.5  
3 3 3 James no NaN  
4 4 2 Emily no 9.0  
5 5 3 Michael yes 20.0  
6 6 1 Matthew yes 14.5  
7 7 1 Laura no NaN  
8 8 2 Kevin no 8.0  
9 9 1 Jonas yes 19.0  
Hiding index:  
index 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

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