In [ ]:
```
"""Write a program to distinguish between Array Indexing and Fancy Indexing."""
import numpy as np
arr = np.array((20,40,29,48,71,63))
print("Array Indexing")
print("3rd Element is ",arr[2])
print("6rd Element is ",arr[5])
print("\nFancy Indexing")
condition=arr>20
filterElements=arr[condition]
print("Elements greater than 20 ",filterElements)
customIndices=[1,5]
filterElements=arr[customIndices]
print("Elements with Custom Indices ",filterElements)
```

```
Array Indexing
3rd Element is  29
6rd Element is  63

Fancy Indexing
Elements greater than 20  [40 29 48 71 63]
Elements with Custom Indices  [40 63]
```

In [ ]:
```
"""Execute the 2D array Slicing."""
import numpy as np
arr= np.array([[3,5,6,3,9,5,7,1],[34,54,12,39,92,5,45,78],[4,5,6,3,6,3,3,2]])
sr = arr[0:2]
sc = arr[:,2:4]
print("Elements of First 2 Rows\n",sr)
print("\nElements of Third and Fourth Colomn of All Rows\n",sc)
```

```
Elements of First 2 Rows
 [[ 3  5  6  3  9  5  7  1]
 [34 54 12 39 92  5 45 78]]

Elements of Third and Fourth Colomn of All Rows
 [[ 6  3]
 [12 39]
 [ 6  3]]
```

In [ ]:
```
"""Create the 5-Dimensional arrays using ‘ndmin’."""
import numpy as np
arr = np.array([[2,3,4,5],[4,3,9]])


print(arr)
```

In [ ]:
```
"""Reshape the array from 1-D to 2-D array."""
import numpy as np
arr = np.array([43, 23, 64, 45, 21, 78, 90, 98, 23, 41])
newarr = arr.reshape(5, 2)
print(newarr)
```

```
[[43 23]
 [64 45]
 [21 78]
 [90 98]
 [23 41]]
```

In [ ]:
```
"""Perform the Stack functions in Numpy arrays – Stack(), hstack(), vstack(), and dstack()."""
import numpy as np
arr1 = np.array([[2,3,4,5],[4,3,9,5]])
arr2 = np.array([[13,12,56,78],[89,23,90,87]])

stack= np.stack((arr1,arr2))
hstack=np.hstack((arr1,arr2))
vstack=np.vstack=((arr1,arr2))
dstack=np.dstack((arr1,arr2))

print("Stacking:\n",stack)
print("\nVertical  Stacking:\n",vstack)
print("\nHorizontal  Stacking:\n",hstack)
print("\nDepth  Stacking:\n",dstack)
```

```
Stacking:
 [[[ 2  3  4  5]
  [ 4  3  9  5]]

 [[13 12 56 78]
  [89 23 90 87]]]

Vertical  Stacking:
 (array([[2, 3, 4, 5],
       [4, 3, 9, 5]]), array([[13, 12, 56, 78],
       [89, 23, 90, 87]]))

Horizontal  Stacking:
 [[ 2  3  4  5 13 12 56 78]
 [ 4  3  9  5 89 23 90 87]]

Depth  Stacking:
 [[[ 2 13]
  [ 3 12]
  [ 4 56]
  [ 5 78]]

 [[ 4 89]
  [ 3 23]
  [ 9 90]
  [ 5 87]]]
```

In [ ]:
```
"""Perform the searchsort method in Numpy array."""
import numpy as np
arr = np.array([43, 23, 64, 45, 21, 78, 90, 98, 23, 41])
arr = np.sort(arr)
print("Given Array ",arr)
newnum = input("Enter Number to insert")
sortedarr = np.searchsorted(arr,newnum)
print(newnum," should be in position ",sortedarr)
```

```
Given Array  [21 23 23 41 43 45 64 78 90 98]
45  should be in position  5
```

In [ ]:
```
"""Create Numpy Structured array using your domain features."""
import numpy as np
dtype=np.dtype([('workerId','U10'),('workerName','U20'),('workerAge','i4'),('hourlyRate','i4'),('rating','i4'),('averageDuration','i4'),('location','U10')])
data=np.array([
    ('W1','Vivek Gupta','28',220,4.2,7,'Patna'),
    ('W10','Olivia Martin','24',200,4.3,10,'New York'),
    ('W21','William','20',160,4.0,8,'California'),
    ('W5','Rebecca Turner','22',210,4.9,6,'New Jearsy'),
    ('W1','Madhav Menon','25',205,4.4,6,'Brooklyn'),
     ],dtype=dtype)
print(data)
```

```
[('W1', 'Vivek Gupta', 28, 220, 4,  7, 'Patna')
 ('W10', 'Olivia Martin', 24, 200, 4, 10, 'New York')
 ('W21', 'William', 20, 160, 4,  8, 'California')
 ('W5', 'Rebecca Turner', 22, 210, 4,  6, 'New Jearsy')
 ('W1', 'Madhav Menon', 25, 205, 4,  6, 'Brooklyn')]
```

In [ ]:
```
"""Create Data frame using List and Dictionary."""
import pandas as pd
dict1={
    'name':["Samantha","Tom","John","Serah"],
    'age':[23,24,21,25],
    'gender':["Female","Male","Male","Female"]
}
df = pd.DataFrame(dict1)
#df = pd.DataFrame.from_dict(dict1)
print(df)
```

```
      name  age  gender
0  Samantha   23  Female
1       Tom   24    Male
2      John   21    Male
3     Serah   25  Female
```

In [ ]:
```
"""Create Data frame on your Domain area and perform the following operations to find and eliminate the
missing data from the dataset.
• isnull()
• notnull()
• dropna()
• fillna()
• replace()
• interpolate()
"""
import pandas as pd
data=pd.read_csv("LaborData.csv")
df= pd.DataFrame(data)
print("Original Data Frame")
print(df)

print("To Check Null Data")
print(df.isnull())

print("To Check NotNull Data")
print(df.notnull())

print("Dropping Rows with Null datas")
dropped_df=df.dropna()
print(dropped_df)

print("Filling Rows with Null datas")
filled_df=df.fillna({'workerId':'W0','workerName':'Dummy Worker','workerAge':25,'hourlyRate':200,'rating':4,'averageDuration':10,'location':'India'})
print(filled_df)

print("Replace To specific Data John Doe to Johnny")
replace_df=df.replace('John Doe','Johnny')
print(replace_df)

print("Interpolate Data")
interpolated_df=df.interpolate()
print(interpolated_df)
```

```
Original Data Frame
   workerId       workerName  workerAge  hourlyRate  rating  averageDuration
0        W1         John Doe        NaN        20.0     4.5              6.0  \
1        W2       Jane Smith       30.0        18.0     4.7              5.0
2        W3  Michael Johnson       27.0        22.0     4.2              7.0
3        W4      Alice Brown       29.0        25.0     NaN              6.0
4        W5     David Wilson        NaN        20.0     4.6              6.0
..      ...              ...        ...         ...     ...              ...
95      W96       Neha Yadav       32.0       220.0     4.5              NaN
96      W97      Rahul Verma       28.0       250.0     4.7              5.0
97      W98    Sneha Sharma        30.0         NaN     4.2              7.0
98      W99  Abhishek Tiwari       27.0       200.0     4.9              6.0
99     W100      Kirti Singh       29.0       210.0     NaN              6.0

         location
0        New York
1   San Francisco
2     Los Angeles
3         Chicago
4         Houston
..            ...
95         Rajkot
96       Jabalpur
97        Gwalior
98       Amritsar
99       Guwahati

[100 rows x 7 columns]
To Check Null Data
    workerId  workerName  workerAge  hourlyRate  rating  averageDuration
0      False       False       True       False   False            False  \
1      False       False      False       False   False            False
2      False       False      False       False   False            False
3      False       False      False       False    True            False
4      False       False       True       False   False            False
..       ...         ...        ...         ...     ...              ...
95     False       False      False       False   False             True
96     False       False      False       False   False            False
97     False       False      False        True   False            False
98     False       False      False       False   False            False
99     False       False      False       False    True            False

    location
0      False
1      False
2      False
3      False
4      False
..       ...
95     False
96     False
97     False
98     False
99     False

[100 rows x 7 columns]
To Check NotNull Data
    workerId  workerName  workerAge  hourlyRate  rating  averageDuration
0       True        True      False        True    True             True  \
1       True        True       True        True    True             True
2       True        True       True        True    True             True
3       True        True       True        True   False             True
4       True        True      False        True    True             True
..       ...         ...        ...         ...     ...              ...
95      True        True       True        True    True            False
96      True        True       True        True    True             True
97      True        True       True       False    True             True
98      True        True       True        True    True             True
99      True        True       True        True   False             True

    location
0       True
1       True
2       True
3       True
4       True
..       ...
95      True
96      True
97      True
98      True
99      True

[100 rows x 7 columns]
Dropping Rows with Null datas
   workerId       workerName  workerAge  hourlyRate  rating  averageDuration
1        W2       Jane Smith       30.0        18.0     4.7              5.0  \
2        W3  Michael Johnson       27.0        22.0     4.2              7.0
6        W7       Kevin Lee        28.0        19.0     4.7              7.0
7        W8      Emily Clark       31.0        24.0     4.8              6.0
9       W10    Olivia Martin       27.0        18.0     4.9              7.0
..      ...              ...        ...         ...     ...              ...
91      W92     Rajesh Kumar       27.0       190.0     4.7              7.0
92      W93    Ananya Tiwari       29.0       210.0     4.8              6.0
94      W95     Shivam Gupta       30.0       200.0     4.9              7.0
96      W97      Rahul Verma       28.0       250.0     4.7              5.0
98      W99  Abhishek Tiwari       27.0       200.0     4.9              6.0

         location
1   San Francisco
2     Los Angeles
6          Boston
7          Dallas
9         Atlanta
..            ...
91     Coimbatore
92       Ludhiana
94       Vadodara
96       Jabalpur
98       Amritsar

[77 rows x 7 columns]
Filling Rows with Null datas
   workerId       workerName  workerAge  hourlyRate  rating  averageDuration
0        W1         John Doe       25.0        20.0     4.5              6.0  \
1        W2       Jane Smith       30.0        18.0     4.7              5.0
2        W3  Michael Johnson       27.0        22.0     4.2              7.0
3        W4      Alice Brown       29.0        25.0     4.0              6.0
4        W5     David Wilson       25.0        20.0     4.6              6.0
..      ...              ...        ...         ...     ...              ...
95      W96       Neha Yadav       32.0       220.0     4.5             10.0
96      W97      Rahul Verma       28.0       250.0     4.7              5.0
97      W98    Sneha Sharma        30.0       200.0     4.2              7.0
98      W99  Abhishek Tiwari       27.0       200.0     4.9              6.0
99     W100      Kirti Singh       29.0       210.0     4.0              6.0

         location
0        New York
```

```
1   San Francisco
2     Los Angeles
3         Chicago
4         Houston
..            ...
95         Rajkot
96       Jabalpur
97        Gwalior
98       Amritsar
99       Guwahati

[100 rows x 7 columns]
Replace To specific Data John Doe to Johnny
   workerId       workerName  workerAge  hourlyRate  rating  averageDuration  \
0        W1           Johnny        NaN        20.0     4.5              6.0
1        W2       Jane Smith       30.0        18.0     4.7              5.0
2        W3  Michael Johnson       27.0        22.0     4.2              7.0
3        W4      Alice Brown       29.0        25.0     NaN              6.0
4        W5     David Wilson        NaN        20.0     4.6              6.0
..      ...              ...        ...         ...     ...              ...
95       W96       Neha Yadav       32.0       220.0     4.5              NaN
96       W97      Rahul Verma       28.0       250.0     4.7              5.0
97       W98     Sneha Sharma       30.0         NaN     4.2              7.0
98       W99  Abhishek Tiwari       27.0       200.0     4.9              6.0
99      W100      Kirti Singh       29.0       210.0     NaN              6.0

         location
0        New York
1   San Francisco
2     Los Angeles
3         Chicago
4         Houston
..            ...
95         Rajkot
96       Jabalpur
97        Gwalior
98       Amritsar
99       Guwahati

[100 rows x 7 columns]
Interpolate Data
   workerId       workerName  workerAge  hourlyRate  rating  averageDuration  \
0        W1         John Doe        NaN        20.0     4.5              6.0
1        W2       Jane Smith       30.0        18.0     4.7              5.0
2        W3  Michael Johnson       27.0        22.0     4.2              7.0
3        W4      Alice Brown       29.0        25.0     4.4              6.0
4        W5     David Wilson       27.5        20.0     4.6              6.0
..      ...              ...        ...         ...     ...              ...
95       W96       Neha Yadav       32.0       220.0     4.5              6.0
96       W97      Rahul Verma       28.0       250.0     4.7              5.0
97       W98     Sneha Sharma       30.0       225.0     4.2              7.0
98       W99  Abhishek Tiwari       27.0       200.0     4.9              6.0
99      W100      Kirti Singh       29.0       210.0     4.9              6.0

         location
0        New York
1   San Francisco
2     Los Angeles
3         Chicago
4         Houston
..            ...
95         Rajkot
96       Jabalpur
97        Gwalior
98       Amritsar
99       Guwahati

[100 rows x 7 columns]
```

In [ ]:
```
"""Perform the Hierarchical Indexing in the above created dataset."""
import pandas as pd
data=pd.read_csv("LaborData.csv")
df= pd.DataFrame(data)
print("Original Data Frame")
print(df)
df.set_index(['workerId','location'])
print("Dataframe with Hierarchical Indexing")
print(df)
```

```
Original Data Frame
   workerId       workerName  workerAge  hourlyRate  rating  averageDuration
0        W1         John Doe        NaN        20.0     4.5              6.0  \
1        W2       Jane Smith       30.0        18.0     4.7              5.0
2        W3  Michael Johnson       27.0        22.0     4.2              7.0
3        W4      Alice Brown       29.0        25.0     NaN              6.0
4        W5     David Wilson        NaN        20.0     4.6              6.0
..      ...              ...        ...         ...     ...              ...
95      W96       Neha Yadav       32.0       220.0     4.5              NaN
96      W97      Rahul Verma       28.0       250.0     4.7              5.0
97      W98     Sneha Sharma       30.0         NaN     4.2              7.0
98      W99  Abhishek Tiwari       27.0       200.0     4.9              6.0
99     W100      Kirti Singh       29.0       210.0     NaN              6.0

         location
0        New York
1   San Francisco
2     Los Angeles
3         Chicago
4         Houston
..            ...
95         Rajkot
96       Jabalpur
97        Gwalior
98       Amritsar
99       Guwahati

[100 rows x 7 columns]
Dataframe with Hierarchical Indexing
   workerId       workerName  workerAge  hourlyRate  rating  averageDuration
0        W1         John Doe        NaN        20.0     4.5              6.0  \
1        W2       Jane Smith       30.0        18.0     4.7              5.0
2        W3  Michael Johnson       27.0        22.0     4.2              7.0
3        W4      Alice Brown       29.0        25.0     NaN              6.0
4        W5     David Wilson        NaN        20.0     4.6              6.0
..      ...              ...        ...         ...     ...              ...
95      W96       Neha Yadav       32.0       220.0     4.5              NaN
96      W97      Rahul Verma       28.0       250.0     4.7              5.0
97      W98     Sneha Sharma       30.0         NaN     4.2              7.0
98      W99  Abhishek Tiwari       27.0       200.0     4.9              6.0
99     W100      Kirti Singh       29.0       210.0     NaN              6.0

         location
0        New York
1   San Francisco
2     Los Angeles
3         Chicago
4         Houston
..            ...
95         Rajkot
96       Jabalpur
97        Gwalior
98       Amritsar
99       Guwahati

[100 rows x 7 columns]
```