```
#write a program to display array indexing and fancy indexing
import numpy as np
# Create a NumPy array
arr = np.array([1, 2, 3, 4, 5])
# Array indexing
print("Array indexing:")
print(arr[0])
print(arr[2])
print(arr[-1])
# Fancy indexing
print("Fancy indexing:")
print(arr[[0, 2, 4]])
print(arr[arr > 2])
print(arr[np.arange(len(arr)) % 2 == 0])
```

```
Array indexing:
1
3
5
Fancy indexing:
[1 3 5]
[3 4 5]
[1 3 5]
```

```
#2 Execute a 2D array
import numpy as np
# Create a 2D array
arr = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
# 2D array slicing
print("2D array slicing:")
print(arr[0, 0])
print(arr[1, 2])
print(arr[2, :])
print(arr[:, 1])
print(arr[1:, :2])
```

```
2D array slicing:
1
6
[7 8 9]
[2 5 8]
[[4 5]
 [7 8]]
```

```
#3.5D array with ndim
import numpy as np

# create a 5D array with shape (1, 2, 3, 4, 5)
arr = np.array([[[[[1, 2, 3, 4, 5],
                   [6, 7, 8, 9, 10],
                   [11, 12, 13, 14, 15]],
                  [[16, 17, 18, 19, 20],
                   [21, 22, 23, 24, 25],
                   [26, 27, 28, 29, 30]],
                  [[31, 32, 33, 34, 35],
                   [36, 37, 38, 39, 40],
                   [41, 42, 43, 44, 45]],
                  [[46, 47, 48, 49, 50],
                   [51, 52, 53, 54, 55],
                   [56, 57, 58, 59, 60]]]]], ndmin=5)

print(arr)

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

   [[16 17 18 19 20]
    [21 22 23 24 25]
    [26 27 28 29 30]]

   [[31 32 33 34 35]
    [36 37 38 39 40]
    [41 42 43 44 45]]

   [[46 47 48 49 50]
    [51 52 53 54 55]
    [56 57 58 59 60]]]]]

#4 Reshape the array from 1-D to 2-D array.
import numpy as np
# Create a 1-D array
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
print(arr)
# Reshape the array into a 2-D array
new_arr = arr.reshape((5, 2))
# Print the new array
print(new_arr)

[ 1  2  3  4  5  6  7  8  9 10]
[[ 1  2]
 [ 3  4]
 [ 5  6]
 [ 7  8]
 [ 9 10]]

#5 Perform the Stack functions in Numpy arrays – Stack(), hstack(),
vstack(), and dstack().

import numpy as np
```

```
# Create some arrays to stack
arr1 = np.array([1, 2, 3])
arr2 = np.array([4, 5, 6])
arr3 = np.array([7, 8, 9])

# Stack the arrays using different functions
stacked = np.stack((arr1, arr2, arr3))
hstacked = np.hstack((arr1, arr2, arr3))
vstacked = np.vstack((arr1, arr2, arr3))
dstacked = np.dstack((arr1, arr2, arr3))

# Print the stacked arrays
print(stacked)
print(hstacked)
print(vstacked)
print(dstacked)
```

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

```
#6 Perform the searchsort method in Numpy array.

import numpy as np

# Create a sorted array
arr = np.array([1, 3, 5, 7, 9])

# Search for the index where 6 should be inserted
index = np.searchsorted(arr, 6)

# Insert 6 into the array at the correct index
arr = np.insert(arr, index, 6)

# Print the sorted array with 6 inserted
print(arr)
```

```
[1 3 5 6 7 9]
```

```
#7 Create Numpy Structured array using your domain features.

import numpy as np

# Define the structured array data type
```

```
music_dtype = np.dtype([
    ('artist_name', 'U50'),
    ('artist_id', int),
    ('age', int),
    ('followers', int)
])

music_array = np.empty(6, dtype=music_dtype)

# Add data to the structured array
music_array[0] = ('The Weeknd', 123, 33, 5000)
music_array[1] = ('21 Savage', 112, 28, 1000)
music_array[2] = ('Aubrey', 122, 34, 1500)
music_array[3] = ('Lil Uzi Vertz', 121, 32, 5000)
music_array[4] = ('Lil wayne', 124, 55, 9000)
music_array[5] = ('Eminem', 129, 45, 9000)

# Print the structured array
print(music_array)
```

```
[('The Weeknd', 123, 33, 5000) ('21 Savage', 112, 28, 1000)
 ('Aubrey', 122, 34, 1500) ('Lil Uzi Vertz', 121, 32, 5000)
 ('Lil wayne', 124, 55, 9000) ('Eminem', 129, 45, 9000)]
```

```
#Q8. Create Data frame using List and Dictionary.

import pandas as pd

# Create a list of dictionaries
artist = [
    {'name': 'The Weeknd', 'real_name': 'Abel', 'age': 33,
'followers': 500.0},
    {'name': '21 Savage', 'real_name': 'Shéyaa Bin Abraham-Joseph',
'age': 22, 'followers': 1000.0},
    {'name': 'lil wayne', 'real_name': 'Dwayne Michael Carter Jr.',
'age': 41, 'followers': 300.0}
]

# Create a DataFrame using pd.DataFrame()
df1 = pd.DataFrame(artist)

# Create a DataFrame using pd.DataFrame.from_dict()
df2 = pd.DataFrame.from_dict(artist)

# Print the resulting DataFrames
print(df1)
print(df2)
```

```
         name                  real_name  age  followers
0  The Weeknd                       Abel   33      500.0
1   21 Savage  Shéyaa Bin Abraham-Joseph   22     1000.0
```

```
2   lil wayne  Dwayne Michael Carter Jr.   41      300.0
         name                  real_name  age  followers
0  The Weeknd                       Abel   33      500.0
1   21 Savage  Shéyaa Bin Abraham-Joseph   22     1000.0
2   lil wayne  Dwayne Michael Carter Jr.   41      300.0
```

```
#Q9. Create Data frame on your Domain area and perform the following
operations to find and eliminate the  missing data from the dataset.

import pandas as pd
import numpy as np

# Create a dictionary of pets
music = {
    'name': ['The Weeknd', '21 Savage', 'Drake',  np.nan],
    'real_name': ['Abel', 'Shéyaa Bin Abraham-Joseph', 'Aubrey',
'Travis Scott'],
    'age': [33, 29, 41, np.nan],
    'folllowers': [5000.0, 1000.0, 1300.0, np.nan]
}

# Create a DataFrame from the dictionary
df = pd.DataFrame(music)

# Check for missing data
print(df.isnull())

# Check for non-missing data
print(df.notnull())

# Drop rows with missing data
df = df.dropna()

# Fill missing data with a value
df = df.fillna(0)

# Replace missing data with a value
df = df.replace(np.nan, 0)

# Interpolate missing data
df = df.interpolate()

# Print the resulting DataFrame
print(df)
```

```
    name  real_name    age  folllowers
0  False      False  False       False
1  False      False  False       False
2  False      False  False       False
3   True      False   True        True
    name  real_name    age  folllowers
```

```
0   True       True   True         True
1   True       True   True         True
2   True       True   True         True
3  False       True  False        False
         name                  real_name   age  folllowers
0  The Weeknd                       Abel  33.0      5000.0
1   21 Savage  Shéyaa Bin Abraham-Joseph  29.0      1000.0
2       Drake                     Aubrey  41.0      1300.0

C:\Users\alwin\AppData\Local\Temp\ipykernel_16564\292552399.py:33:
FutureWarning: DataFrame.interpolate with object dtype is deprecated
and will raise in a future version. Call obj.infer_objects(copy=False)
before interpolating instead.
  df = df.interpolate()
```

```
#10 Perform the Hierarchical Indexing in the above created dataset.

import pandas as pd

# Create a dictionary of pets
pets = {
    'name': ['The Weeknd', '21 Savage', 'Drake',  np.nan],
    'real_name': ['Abel', 'Shéyaa Bin', 'Aubrey', 'Travis Scott'],
    'age': [33, 29, 41, np.nan],
    'folllowers': [5000.0, 1000.0, 1300.0, np.nan]
}

# Create a DataFrame from the dictionary with hierarchical indexing
df = pd.DataFrame(pets, index=[['Abel', 'Shéyaa Bin', 'Aubrey',
'Travis Scott'], [1, 2, 3, 4]])

# Print the resulting DataFrame with hierarchical indexing
print(df)
```

```
                      name     real_name   age  folllowers
Abel         1  The Weeknd          Abel  33.0      5000.0
Shéyaa Bin   2   21 Savage    Shéyaa Bin  29.0      1000.0
Aubrey       3       Drake        Aubrey  41.0      1300.0
Travis Scott 4         NaN  Travis Scott   NaN         NaN
```