Write a program to distinguish between Array Indexing and Fancy Indexing.

```
import numpy as np

# Create a NumPy array with some random values
arr = np.array([1, 2, 3, 4, 5])

# Access a single element of the array using simple indexing
simple_indexing = arr[2]
print("Simple indexing:", simple_indexing)

# Access multiple elements of the array using fancy indexing
fancy_indexing = arr[[1, 3]]
print("Fancy indexing:", fancy_indexing)

# Print the type of the result of each indexing method
print("Type of simple indexing:", type(simple_indexing))
print("Type of fancy indexing:", type(fancy_indexing))
```

```
Simple indexing: 3
Fancy indexing: [2 4]
Type of simple indexing: <class 'numpy.int64'>
Type of fancy indexing: <class 'numpy.ndarray'>
```

Execute the 2D array Slicing.

```
import numpy as np

# Create a 2D NumPy array with some random values
arr = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

# Use array slicing to extract a subset of the array
sliced_arr = arr[1:, :2]

# Print the original array and the sliced array
print("Original array:")
print(arr)
print("Sliced array:")
print(sliced_arr)
```

```
Original array:
[[1 2 3]
 [4 5 6]
 [7 8 9]]
Sliced array:
[[4 5]
 [7 8]]
```

Create the 5-Dimensional arrays using ‘ndmin’.

```
import numpy as np

# Create a 1D NumPy array with some random values
arr = np.array([1, 2, 3, 4, 5])

# Use ndmin to create a 5-dimensional array
five_dim_arr = np.array(arr, ndmin=5)

# Print the original array and the 5-dimensional array
print("Original array:")
print(arr)
print("5-dimensional array:")
print(five_dim_arr)
```

```
Original array:
[1 2 3 4 5]
5-dimensional array:
[[[[[1 2 3 4 5]]]]]
```

Reshape the array from 1-D to 2-D array.

```
import numpy as np

# Create a 1D NumPy array with some random values
arr = np.array([1, 2, 3, 4, 5, 6])

# Use the reshape() method to convert the 1D array to a 2D array
reshaped_arr = arr.reshape(2, 3)

# Print the original array and the reshaped array
print("Original array:")
print(arr)
print("Reshaped array:")
print(reshaped_arr)
```

```
Original array:
[1 2 3 4 5 6]
Reshaped array:
[[1 2 3]
 [4 5 6]]
```

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

```
import numpy as np

# Create two 1D arrays
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])

# Stack the arrays vertically
vstacked = np.vstack((a, b))
print("Vertical stacking:")
print(vstacked)

# Stack the arrays horizontally
hstacked = np.hstack((a, b))
print("Horizontal stacking:")
print(hstacked)

# Create two 2D arrays
c = np.array([[1], [2], [3]])
d = np.array([[4], [5], [6]])

# Stack the arrays depth-wise
dstacked = np.dstack((c, d))
print("Depth-wise stacking:")
print(dstacked)
```

```
Vertical stacking:
[[1 2 3]
 [4 5 6]]
Horizontal stacking:
[1 2 3 4 5 6]
Depth-wise stacking:
[[[1 4]]

 [[2 5]]

 [[3 6]]]
```

Perform the searchsort method in Numpy array.

```
import numpy as np

# Create a sorted 1D NumPy array
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

# Use searchsorted() to find the index of a value in the array
index = np.searchsorted(arr, 5)

# Print the index
```

```
# Print the index
print("Index of 5 in the array:", index)
```

```
Index of 5 in the array: 4
```

Create Numpy Structured array using your domain features.

```
import numpy as np

# Create a structured array
data = np.array([
    ('WhatsApp', 'Communication', 4.4, 'Free'),
    ('Instagram', 'Social', 4.5, 'Free'),
    ('Spotify', 'Music', 4.6, 'Free'),
    ('Candy Crush Saga', 'Games', 4.3, 'Free'),
    ('Zoom', 'Business', 4.2, 'Free')],
    dtype=[('app_name', 'U20'), ('category', 'U20'), ('rating', 'f4'), ('price', 'U10')])

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

```
[('WhatsApp', 'Communication', 4.4, 'Free')
 ('Instagram', 'Social', 4.5, 'Free') ('Spotify', 'Music', 4.6, 'Free')
 ('Candy Crush Saga', 'Games', 4.3, 'Free')
 ('Zoom', 'Business', 4.2, 'Free')]
```

Create Data frame using List and Dictionary.

```
import pandas as pd

# Create a list of dictionaries
data = [
    {'app_name': 'WhatsApp', 'category': 'Communication', 'rating': 4.4, 'price': 'Free'},
    {'app_name': 'Instagram', 'category': 'Social', 'rating': 4.5, 'price': 'Free'},
    {'app_name': 'Spotify', 'category': 'Music', 'rating': 4.6, 'price': 'Free'},
    {'app_name': 'Candy Crush Saga', 'category': 'Games', 'rating': 4.3, 'price': 'Free'},
    {'app_name': 'Zoom', 'category': 'Business', 'rating': 4.2, 'price': 'Free'}
]

# Create a DataFrame from the list of dictionaries
df = pd.DataFrame(data)

# Print the DataFrame
print(df)
```

```
            app_name       category  rating price
0          WhatsApp  Communication     4.4  Free
1         Instagram         Social     4.5  Free
2           Spotify          Music     4.6  Free
3  Candy Crush Saga          Games     4.3  Free
4              Zoom       Business     4.2  Free
```

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
import numpy as np

# Create a list of dictionaries
data = [
    {'app_name': 'WhatsApp', 'category': 'Communication', 'rating': 4.4, 'price': 'Free'},
    {'app_name': 'Instagram', 'category': 'Social', 'rating': np.nan, 'price': 'Free'},
    {'app_name': 'Spotify', 'category': 'Music', 'rating': 4.6, 'price': 'Free'},
    {'app_name': 'Candy Crush Saga', 'category': 'Games', 'rating': 4.3, 'price': np.nan},
    {'app_name': 'Zoom', 'category': 'Business', 'rating': 4.2, 'price': 'Free'}
]

# Create a DataFrame from the list of dictionaries
df = pd.DataFrame(data)
```

```
# Print the DataFrame
print(df)

# 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 specified value
df = df.fillna({'rating': 0, 'price': 'Unknown'})

# Replace specific values in the DataFrame
df = df.replace({'category': {'Music': 'Audio'}})

# Fill missing data with interpolated values
df = df.interpolate()

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

```
             app_name       category  rating price
    0          WhatsApp  Communication     4.4  Free
    1         Instagram         Social     NaN  Free
    2           Spotify          Music     4.6  Free
    3  Candy Crush Saga          Games     4.3   NaN
    4              Zoom       Business     4.2  Free
       app_name  category  rating  price
    0     False     False   False  False
    1     False     False    True  False
    2     False     False   False  False
    3     False     False   False   True
    4     False     False   False  False
       app_name  category  rating  price
    0      True      True    True   True
    1      True      True   False   True
    2      True      True    True   True
    3      True      True    True  False
    4      True      True    True   True
       app_name       category  rating price
    0  WhatsApp  Communication     4.4  Free
    2   Spotify          Audio     4.6  Free
    4      Zoom       Business     4.2  Free
```

Perform the Hierarchical Indexing in the above created dataset.

```
import pandas as pd

# Create a DataFrame
data = {
    'app_name': ['WhatsApp', 'Instagram', 'Spotify', 'Candy Crush Saga', 'Zoom'],
    'category': ['Communication', 'Social', 'Music', 'Games', 'Business'],
    'rating': [4.4, 4.5, 4.6, 4.3, 4.2],
    'price': ['Free', 'Free', 'Free', 'Free', 'Free']
}

df = pd.DataFrame(data)

# Set hierarchical index
df = df.set_index(['app_name', 'category'])

# Print the DataFrame
print(df)
```

```
                                  rating price
    app_name         category
    WhatsApp         Communication     4.4  Free
    Instagram        Social            4.5  Free
    Spotify          Music             4.6  Free
    Candy Crush Saga Games             4.3  Free
    Zoom             Business          4.2  Free
```

Double-click (or enter) to edit