In [ ]:
```python
#Write a program to distinguish between Array Indexing and Fancy Indexing.
import numpy as np

# Array Indexing
arr = np.array([1, 2, 3, 4, 5])
print("Array Indexing:")
print(arr[0])
print(arr[1:3])

# Fancy Indexing
print("\nFancy Indexing:")
arr = np.array([1, 2, 3, 4, 5])
indices = np.array([0, 2, 3])
print(arr[indices])
```

```
Array Indexing:
1
[2 3]

Fancy Indexing:
[1 3 4]
```

In [ ]:
```python
#Execute the 2D array Slicing.
import numpy as np

# Create a 2D array
arr = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])

# Slice the array to get the first two rows and columns
subarray1 = arr[:2, :2]

# Slice the array to get the last two rows and columns
subarray2 = arr[1:, 2:]

# Print the subarrays
print("First Two Rows and Columns:\n", subarray1)
print("Last Two Rows and Columns:\n", subarray2)
```

```
First Two Rows and Columns:
 [[1 2]
 [5 6]]
Last Two Rows and Columns:
 [[ 7  8]
 [11 12]]
```

In [ ]:
```python
#5d array
import numpy as np

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

# Print the array and its shape
print("Array:\n", arr)
print("Shape of Array:", arr.shape)
```

```
Array:
 [[[[[1 2 3 4]]]]]
Shape of Array: (1, 1, 1, 1, 4)
```

In [ ]:
```python
#Reshape the array from 1-D to 2-D array.
import numpy as np

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

# Reshape the array to a 2D array with 2 rows and 3 columns
arr_2d = np.reshape(arr, (2, 3))

# Print the original and reshaped arrays
print("Original Array:\n", arr)
print("Reshaped Array:\n", arr_2d)
```

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

In [ ]:
```python
#Perform the Stack functions in Numpy arrays - Stack(), hstack(), vstack(), and dst
import numpy as np

# Create two 2D arrays
arr1 = np.array([[1, 2], [3, 4]])
arr2 = np.array([[5, 6], [7, 8]])

# Stack them using stack()
print("Stack():")
print(np.stack((arr1, arr2), axis=0))

# Stack them using hstack()
print("\nhstack():")
print(np.hstack((arr1, arr2)))

# Stack them using vstack()
print("\nvstack():")
print(np.vstack((arr1, arr2)))

# Stack them using dstack()
print("\ndstack():")
print(np.dstack((arr1, arr2)))
```

```
Stack():
[[[1 2]
  [3 4]]

 [[5 6]
  [7 8]]]

hstack():
[[1 2 5 6]
 [3 4 7 8]]

vstack():
[[1 2]
 [3 4]
 [5 6]
 [7 8]]

dstack():
[[[1 5]
  [2 6]]

 [[3 7]
  [4 8]]]
```

In [ ]:
```python
#Perform the searchsort method in Numpy array.
import numpy as np

# Create a sorted array
arr = np.array([1, 2, 3, 4, 5])

# Find the index where 3 should be inserted
index = np.searchsorted(arr, 3)

# Print the index
print("Index of 3:", index)
```

```
Index of 3: 2
```

In [ ]:
```python
#Create Numpy Structured array using your domain features.
#Photography Prop Store
import numpy as np

# Define the data types for the fields
dt = np.dtype([('Customer_name', 'U12'), ('Customer_id', 'i4'), ('Products', 'U10')

# Create a structured array with 2 records
arr = np.array([('Pavi', 25, 'Basket'), ('Boo', 30, 'Wraps')], dtype=dt)

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

```
[('Pavi', 25, 'Basket') ('Boo', 30, 'Wraps')]
```

In [ ]:
```python
#Create Data frame using List and Dictionary.
import pandas as pd

# Define the data as a list of dictionaries
```

```
data = [{'name': 'John', 'age': 25, 'gender': 'M'},
        {'name': 'Jane', 'age': 30, 'gender': 'F'},
        {'name': 'Bob', 'age': 35, 'gender': 'M'}]

# Create the DataFrame
df = pd.DataFrame(data)

# Print the DataFrame
print(df)
```

```
   name  age gender
0  John   25      M
1  Jane   30      F
2   Bob   35      M
```

In [ ]:

```
#Create Data frame on your Domain
import pandas as pd
import numpy as np

# Create a dictionary with missing values
data = {'Customer_id': [1, 2, np.nan, 4],
        'Customer_name': ['Pavi', 'Boo', np.nan, 'Priya'],
        'Product': ['Bamboo Basket', 'Cheese Wrap', np.nan, 'Maternity Gowns'],
        'Category': ['Basket', 'Wraps', np.nan, 'Gowns'],
        'Price': [np.nan, '30$', '120$', '400$'],
        }

# Create the DataFrame
df = pd.DataFrame(data)

# Check for missing values using isnull()
print("Missing Values:\n", df.isnull())

# Check for non-missing values using notnull()
print("\nNon-Missing Values:\n", df.notnull())

# Drop rows with missing values using dropna()
df_dropped = df.dropna()
print("\nDataFrame after dropping rows with missing values:\n", df_dropped)

# Fill missing values using fillna()
df_filled = df.fillna(0)
print("\nDataFrame after filling missing values with 0:\n", df_filled)

# Replace missing values using replace()
df_replaced = df.replace(np.nan, -1)
print("\nDataFrame after replacing missing values with -1:\n", df_replaced)

# Interpolate missing values using interpolate()
df_interpolated = df.interpolate()
print("\nDataFrame after interpolating missing values:\n", df_interpolated)
```

```
Missing Values:
    Customer_id  Customer_name  Product  Category  Price
0         False          False    False     False   True
1         False          False    False     False  False
2          True           True     True      True  False
3         False          False    False     False  False

Non-Missing Values:
    Customer_id  Customer_name  Product  Category  Price
0          True           True     True      True  False
1          True           True     True      True   True
2         False          False    False     False   True
3          True           True     True      True   True

DataFrame after dropping rows with missing values:
    Customer_id Customer_name          Product Category Price
1           2.0           Boo      Cheese Wrap    Wraps   30$
3           4.0         Priya  Maternity Gowns    Gowns  400$

DataFrame after filling missing values with 0:
    Customer_id Customer_name          Product Category Price
0           1.0          Pavi    Bamboo Basket   Basket     0
1           2.0           Boo      Cheese Wrap    Wraps   30$
2           0.0             0                0        0  120$
3           4.0         Priya  Maternity Gowns    Gowns  400$

DataFrame after replacing missing values with -1:
    Customer_id Customer_name          Product Category Price
0           1.0          Pavi    Bamboo Basket   Basket    -1
1           2.0           Boo      Cheese Wrap    Wraps   30$
2          -1.0            -1               -1       -1  120$
3           4.0         Priya  Maternity Gowns    Gowns  400$

DataFrame after interpolating missing values:
    Customer_id Customer_name          Product Category Price
0           1.0          Pavi    Bamboo Basket   Basket   NaN
1           2.0           Boo      Cheese Wrap    Wraps   30$
2           3.0           NaN              NaN      NaN  120$
3           4.0         Priya  Maternity Gowns    Gowns  400$
C:\Users\rpdpr\AppData\Local\Temp\ipykernel_18904\2998112668.py:35: FutureWarning: D
ataFrame.interpolate with object dtype is deprecated and will raise in a future vers
ion. Call obj.infer_objects(copy=False) before interpolating instead.
  df_interpolated = df.interpolate()
```

In [ ]:

```
#Perform the Hierarchical Indexing in the above created dataset.
import pandas as pd

# Create a dictionary with data
data = {'Customer_id': [1, 2, 3, 4],
        'Customer_name': ['Pavi', 'Boo', 'Priya', 'Malz'],
        'Product': [' Bamboo Basket', 'Cheese Wrap', 'Maternity Gown', 'Flower Tiar
        'Category': ['Basket', 'Wraps', 'Gowns', 'Tiaras'],
        'Price': ['30$', '120$', '400$', '150$'],
        }

# Create the DataFrame
```

```
df = pd.DataFrame(data)

# Set hierarchical index
df.set_index(['Customer_name', 'Category'], inplace=True)

# Print the DataFrame
print(df)
```

```
                        Customer_id         Product Price
Customer_name Category
Pavi          Basket              1   Bamboo Basket   30$
Boo           Wraps               2     Cheese Wrap  120$
Priya         Gowns               3  Maternity Gown  400$
Malz          Tiaras              4    Flower Tiara  150$
```