labexcercise9-2347223

September 16, 2023

[10]:
```
#Q1. Write a program to distinguish between Array Indexing and Fancy Indexing.
import numpy as np
a = np.array([[1, 2, 3, 4, 5], [2, 2, 3, 4, 6]])
print(a[0, 0])
print(a[-1, -1])
print(a[[0, -1], [0, -1]])
print(a)
row_indices = np.array([0, 1])
column_indices = np.array([0])
print(a[row_indices, column_indices])
```

```
1
6
[1 6]
[[1 2 3 4 5]
 [2 2 3 4 6]]
[1 2]
```

[11]:
```
#Q2. Execute the 2D array Slicing.
import numpy as np


a = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])

print(a[0])

print(a[:, 1])


print(a[1:3, 1:4])
```

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

[2]:
```
#Q3. Create the 5-Dimensional arrays using 'ndmin'.
import numpy as np
```

```
a = np.array([1, 2, 3])

b = np.array(a, ndmin=5)

print(b)
```

```
[[[[[1 2 3]]]]]
```

[3]:
```
#Q4. Reshape the array from 1-D to 2-D array.
import numpy as np

array1 = np.array([1, 3, 5, 7, 2, 4, 6, 8])

result = np.reshape(array1, (2, 4))
print(result)
```

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

[5]:
```
#Q5. 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])


c = np.vstack((a, b))
d = np.hstack((a, b))
e = np.dstack((a, b))
f = np.stack((a,b))

print("\n",c)
print("\n",d)
print("\n",e)
print("\n",f)
```

```
 [[1 2 3]
 [4 5 6]]

 [1 2 3 4 5 6]

 [[[1 4]
  [2 5]
  [3 6]]]
```

```
[[1 2 3]
 [4 5 6]]
```

[8]:
```
#Q6. Perform the searchsort method in Numpy array.
import numpy as np
a = np.array([1, 2, 3, 4, 5])
indices = np.searchsorted(a, [2, 4])
print(indices)
```

```
[1 3]
```

[9]:
```
#Q7. Create Numpy Structured array using your domain features.
import numpy as np

dtypes = [('name', 'U50'), ('age', int), ('membership_type', 'U20'),␣
 ↪('membership_duration', int)]

gym_members = np.zeros(100, dtype=dtypes)

gym_members[0] = ('John Doe', 30, 'Gold', 6)
gym_members[1] = ('Jane Smith', 25, 'Silver', 3)
gym_members[2] = ('Bob Johnson', 35, 'Bronze', 1)

print(f"Member 1: Name={gym_members[0]['name']}, Age={gym_members[0]['age']},␣
 ↪Type={gym_members[0]['membership_type']},␣
 ↪Duration={gym_members[0]['membership_duration']}")

print(f"Member 2: Name={gym_members[1]['name']}, Age={gym_members[1]['age']},␣
 ↪Type={gym_members[1]['membership_type']},␣
 ↪Duration={gym_members[1]['membership_duration']}")
```

```
Member 1: Name=John Doe, Age=30, Type=Gold, Duration=6
Member 2: Name=Jane Smith, Age=25, Type=Silver, Duration=3
```

[10]:
```
#Q8. Create Data frame using List and Dictionary.
import pandas as pd


names = ['John Doe', 'Jane Smith', 'Bob Johnson']
ages = [30, 25, 35]
membership_types = ['Gold', 'Silver', 'Bronze']
membership_durations = [6, 3, 1]

data = {
    'Name': names,
    'Age': ages,
    'Membership Type': membership_types,
```

```
    'Membership Duration (months)': membership_durations
}

df = pd.DataFrame(data)


print(df)
```

```
          Name  Age Membership Type  Membership Duration (months)
0     John Doe   30            Gold                             6
1   Jane Smith   25          Silver                             3
2  Bob Johnson   35          Bronze                             1
```

[18]:
```
#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

data = {
    'Name': ['John Doe', 'Jane Smith', 'Bob Johnson', None],
    'Age': [30, 25, None, 35],
    'Membership Type': ['Gold', 'Silver', 'Bronze', 'Platinum'],
    'Membership Duration (months)': [6, None, 1, 12]
}

df = pd.DataFrame(data)

print("isnull():")
print(df.isnull())
print()


print("notnull():")
print(df.notnull())
print()


print("dropna():")
print(df.dropna())
print()


print("fillna():")
print(df.fillna(0))
print()

print("interpolate():")
```

```
print(df.interpolate(method='linear'))
```

```
isnull():
    Name    Age  Membership Type  Membership Duration (months)
0  False  False            False                         False
1  False  False            False                          True
2  False   True            False                         False
3   True  False            False                         False

notnull():
    Name    Age  Membership Type  Membership Duration (months)
0   True   True             True                          True
1   True   True             True                         False
2   True  False             True                          True
3  False   True             True                          True

dropna():
       Name   Age Membership Type  Membership Duration (months)
0  John Doe  30.0            Gold                           6.0

fillna():
          Name   Age Membership Type  Membership Duration (months)
0     John Doe  30.0            Gold                           6.0
1   Jane Smith  25.0          Silver                           0.0
2  Bob Johnson   0.0          Bronze                           1.0
3            0  35.0        Platinum                          12.0

interpolate():
          Name   Age Membership Type  Membership Duration (months)
0     John Doe  30.0            Gold                           6.0
1   Jane Smith  25.0          Silver                           3.5
2  Bob Johnson  30.0          Bronze                           1.0
3         None  35.0        Platinum                          12.0

C:\Users\91918\AppData\Local\Temp\ipykernel_15644\2751162846.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.
  print(df.interpolate(method='linear'))
```

[24]:
```
#Q10. Perform the Hierarchical Indexing in the above created dataset.
import pandas as pd
data = {
    'Name': ['John Doe', 'Jane Smith', 'Bob Johnson', None],
    'Age': [30, 25, None, 35],
    'Membership Type': ['Gold', 'Silver', 'Bronze', 'Platinum'],
    'Membership Duration (months)': [6, None, 1, 12]
}
```

```
df = pd.DataFrame(data)


df.set_index(['Membership Type', 'Name'], inplace=True)


print(df)

print(df.loc['Silver', 'Jane Smith'])
```

```
                              Age  Membership Duration (months)
Membership Type Name
Gold            John Doe     30.0                           6.0
Silver          Jane Smith   25.0                           NaN
Bronze          Bob Johnson   NaN                           1.0
Platinum        NaN          35.0                          12.0
Age                             25.0
Membership Duration (months)     NaN
Name: (Silver, Jane Smith), dtype: float64
```