AKTU B.Tech Second Year

PYTHON PROGRAMMING

Unit 5

Python packages

LEC-1

Gateway classes

Today's Target

- **NUMPY**
- **AKTU PYQs**

By
Pragya Rajvanshi

Numpy

- NumPy is an open-source Python library that provides support for large, multi-dimensional arrays and matrices.
- It also have a collection of high-level mathematical functions to operate on arrays.

```
Array is of type:  <class 'numpy.ndarray'>
No. of dimensions:  2
Shape of array:  (2, 3)
Size of array:  6
Array stores elements of type:  int64
```

main.py

```python
import numpy as np
# Creating array object
arr = np.array( [[ 1, 2, 3],
                 [ 4, 2, 5]] )
# Printing type of arr object
print("Array is of type: ", type(arr))

# Printing array dimensions (axes)
print("No. of dimensions: ", arr.ndim)

# Printing shape of array
print("Shape of array: ", arr.shape)

# Printing size (total number of elements) of array
print("Size of array: ", arr.size)

# Printing type of elements in array
print("Array stores elements of type: ", arr.dtype)
```

Gateway classes

NumPy Basic Operations

1. Operations on a single NumPy array

main.py

```
# Python program to demonstrate
# basic operations on single array
import numpy as np
a = np.array([1, 2, 5, 3])
# add 1 to every element
print ("Adding 1 to every element:", a+1)
# subtract 3 from each element
print ("Subtracting 3 from each element:", a-3)
# multiply each element by 10
print ("Multiplying each element by 10:", a*10)
# square each element
print ("Squaring each element:", a**2)
# modify existing array
a *= 2
print ("Doubled each element of original array:", a)
# transpose of array
a = np.array([[1, 2, 3], [3, 4, 5], [9, 6, 0]])

print ("\nOriginal array:\n", a)
print ("Transpose of array:\n", a.T)

```

Gateway classes

```
Adding 1 to every element: [2 3 6 4]
Subtracting 3 from each element: [-2 -1  2  0]
Multiplying each element by 10: [10 20 50 30]
Squaring each element: [ 1  4 25  9]
Doubled each element of original array: [ 2  4 10  6]

Original array:
 [[1 2 3]
 [3 4 5]
 [9 6 0]]
Transpose of array:
 [[1 3 9]
 [2 4 6]
 [3 5 0]]
```

NumPy – Unary Operators

Many unary operations are provided as a method of ndarray class. This includes sum, min, max, etc. These functions can also be applied row-wise or column-wise by setting an axis parameter

main.py

```python
# Python program to demonstrate
# unary operators in numpy
import numpy as np

arr = np.array([[1, 5, 6],
                [4, 7, 2],
                [3, 1, 9]])
# maximum element of array
print ("Largest element is:", arr.max())
print ("Row-wise maximum elements:",
                    arr.max(axis = 1))
# minimum element of array
print ("Column-wise minimum elements:",
                        arr.min(axis = 0))
# sum of array elements
print ("Sum of all array elements:",
                            arr.sum())
                            # cumulative sum along each row
print ("Cumulative sum along each row:\n",
                        arr.cumsum(axis = 1))
```

```
Largest element is: 9
Row-wise maximum elements: [6 7 9]
Column-wise minimum elements: [1 1 2]
Sum of all array elements: 38
Cumulative sum along each row:
 [[ 1  6 12]
 [ 4 11 13]
 [ 3  4 13]]
```

Gateway classes

NumPy – Binary Operators

These operations apply to the array elementwise and a new array is created. You can use all basic arithmetic operators like +, -, /, etc. In the case of *+=, -=,* = operators, the existing array is modified

main.py

```
1  # Python program to demonstrate
2  # binary operators in Numpy
3  import numpy as np
4
5  a = np.array([[1,2],
6                [2,4]])
7  b = np.array([[4, 3],
8                [2, 1]])
9
10 # add arrays
11 print ("Array sum:\n", a + b)
12
13 # multiply arrays (elementwise multiplication)
14 print ("Array multiplication:\n", a*b)
15
16 # matrix multiplication
17 print ("Matrix multiplication:\n", a.dot(b))
18
```

```
Array sum:
 [[5 5]
 [4 5]]
Array multiplication:
 [[4 6]
 [4 4]]
Matrix multiplication:
 [[ 8  5]
 [16 10]]
```

Gateway classes

Numpy sorting array

main.py

```
1  import numpy as np
2  a = np.array([[1, 4, 2],
3                [3, 4, 6],
4                [0, -1, 5]])
5  # sorted array
6  print ("Array elements in sorted order:\n",
7                  np.sort(a, axis = None))
8  # sort array row-wise
9  print ("Row-wise sorted array:\n",
10                 np.sort(a, axis = 1))
11 # specify sort algorithm
12 print ("Column wise sort by applying merge-sort:\n",
13                 np.sort(a, axis = 0, kind = 'mergesort'))
```

```
Array elements in sorted order:
 [-1  0  1  2  3  4  4  5  6]
Row-wise sorted array:
 [[ 1  2  4]
 [ 3  4  6]
 [-1  0  5]]
Column wise sort by applying merge-sort:
 [[ 0 -1  2]
 [ 1  4  5]
 [ 3  4  6]]
```

Gateway Classes

Write a NumPy program to convert a list of numeric values into a one-dimensional NumPy array.

main.py

```
1  import numpy as np
2  # Original list
3  original_list = [12.23, 13.32, 100, 36.32]
4
5  # Convert to one-dimensional NumPy array
6  numpy_array = np.array(original_list)
7
8  # Display results
9  print("Original List:", original_list)
10 print("One-dimensional NumPy array:", numpy_array)
```

```
Original List: [12.23, 13.32, 100, 36.32]
One-dimensional NumPy array: [ 12.23  13.32 100.    36.32]
```

gateway classes

Write a NumPy program to create a 3x3 matrix with values ranging from 2 to 10.

main.py

```
1 import numpy as np
2 # Create a 3x3 matrix with values from 2 to 10
3 matrix = np.arange(2, 11).reshape(3, 3)
4 # Display the matrix
5 print("3x3 Matrix with values ranging from 2 to 10:")
6 print(matrix)
7
```

```
3x3 Matrix with values ranging from 2 to 10:
[[ 2  3  4]
 [ 5  6  7]
 [ 8  9 10]]
```

Gateway Cl

Write a NumPy program to reverse an array (the first element becomes the last)

main.py

```python
import numpy as np

# Create an example array
original_array = np.array([1, 2, 3, 4, 5])

# Reverse the array
reversed_array = original_array[::-1]

# Display the result
print("Original array:", original_array)
print("Reversed array:", reversed_array)

```

```
Original array: [1 2 3 4 5]
Reversed array: [5 4 3 2 1]
```

Gateway classes

main.py

```python
# Write a NumPy program to append values to the end of an array.
import numpy as np

# Original array
original_array = np.array([1, 2, 3])

# Values to append
values_to_append = [4, 5, 6]

# Append values to the array
result_array = np.append(original_array, values_to_append)

# Display the result
print("Original array:", original_array)
print("Appended array:", result_array)

```

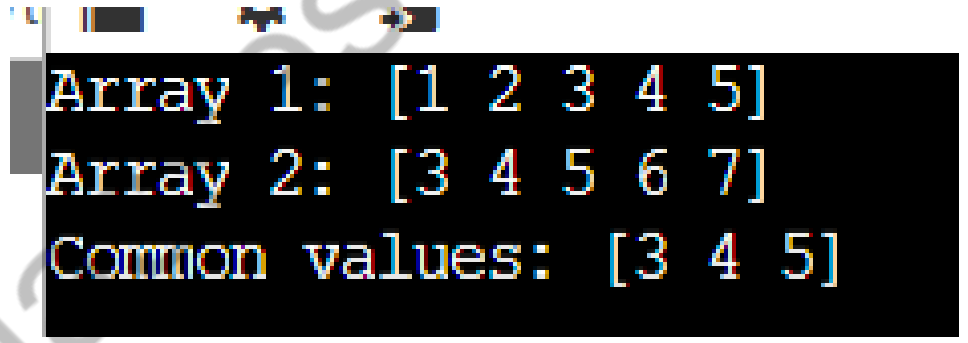
```
Original array: [1 2 3]
Appended array: [1 2 3 4 5 6]
```

Gateway classes

Write a NumPy program to find common values between two arrays.

main.py

```
1  import numpy as np
2  # Define two arrays
3  array1 = np.array([1, 2, 3, 4, 5])
4  array2 = np.array([3, 4, 5, 6, 7])
5  # Find common values
6  common_values = np.intersect1d(array1, array2)
7  # Display the result
8  print("Array 1:", array1)
9  print("Array 2:", array2)
10 print("Common values:", common_values)
```



Gateway classes

Write a NumPy program to get the unique elements of an array.

main.py

```
1  import numpy as np
2  # Define an array
3  array = np.array([1, 2, 2, 3, 4, 4, 5, 5, 5])
4  # Get unique elements
5  unique_elements = np.unique(array)
6  # Display the result
7  print("Original array:", array)
8  print("Unique elements:", unique_elements)
9
```

```
Original array: [1 2 2 3 4 4 5 5 5]
Unique elements: [1 2 3 4 5]
```

Gateway classes

Write a NumPy program to find the set difference between two arrays. The set difference will return sorted, distinct values in array1 that are not in array2

main.py

```
1  import numpy as np
2
3  # Define two arrays
4  array1 = np.array([1, 2, 3, 4, 5])
5  array2 = np.array([3, 4, 5, 6, 7])
6
7  # Find the set difference
8  set_difference = np.setdiff1d(array1, array2)
9
10 # Display the result
11 print("Array 1:", array1)
12 print("Array 2:", array2)
13 print("Set difference", set_difference)
```

```
Array 1: [1 2 3 4 5]
Array 2: [3 4 5 6 7]
Set difference (elements in Array 1 not in Array 2): [1 2]
```

Gateway Classes

Union of two array

main.py

```
1  import numpy as np
2
3  # Define two arrays
4  array1 = np.array([0, 10, 20, 40, 60, 80])
5  array2 = np.array([10, 30, 40, 50, 70])
6
7  # Find the union of the two arrays
8  union_array = np.union1d(array1, array2)
9
10 # Display the result
11 print("Array 1:", array1)
12 print("Array 2:", array2)
13 print("Union of Array 1 and Array 2:", union_array)
14
```

```
Array 1: [ 0 10 20 40 60 80]
Array 2: [10 30 40 50 70]
Union of Array 1 and Array 2: [ 0 10 20 30 40 50 60 70 80]
```

Gateway Classes


Thank
you