

Experiment 11 - Creating and Manipulating 1D, 2D, and 3D Arrays in Python

Name: Mohammad Sayeed

Roll no : C56 Div: C Class: TY CSE

1. First 10 natural numbers and 5th element

Section: List

Problem Statement:

Create a Python list of the first 10 natural numbers. Print the 5th element using indexing.

Code:

```
lst = list(range(1,11))
print(lst[4])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
5
```

2. 10 random integers slice first 5

Section: List

Problem Statement:

Make a list of 10 random integers. Slice the list to print only the first 5 elements.

Code:

```
import random
lst = [random.randint(1,100) for _ in range(10)]
print(lst[:5])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[46, 64, 13, 79, 70]
```

3. Append 6 to list

Section: List

Problem Statement:

Create a list [1, 2, 3, 4, 5]. Append the number 6 and print the list.

Code:

```
lst = [1,2,3,4,5]
lst.append(6)
print(lst)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[1, 2, 3, 4, 5, 6]
```

4. Remove index 2

Section: List

Problem Statement:

Create a list [10, 20, 30, 40, 50]. Remove the element at index 2.

Code:

```
lst = [10,20,30,40,50]
lst.pop(2)
print(lst)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[10, 20, 40, 50]
```

5. Replace 3rd element with 99

Section: List

Problem Statement:

Create a list [1,2,3,4,5]. Replace the 3rd element with 99.

Code:

```
lst=[1,2,3,4,5]
lst[2]=99
print(lst)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[1, 2, 99, 4, 5]
```

6. Concatenate two lists

Section: List

Problem Statement:

Create two lists [1,2,3] and [4,5,6]. Concatenate them.

Code:

```
a=[1,2,3]; b=[4,5,6]
print(a+b)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[1, 2, 3, 4, 5, 6]
```

7. Nested list element 5

Section: List

Problem Statement:

Create a nested list [[1,2,3], [4,5,6], [7,8,9]]. Print the element 5.

Code:

```
nl = [[1,2,3], [4,5,6], [7,8,9]]
print(nl[1][1])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
5
```

8. Even numbers from 1-20

Section: List

Problem Statement:

Create a list of numbers from 1 to 20. Slice and print only the even numbers.

Code:

```
lst = list(range(1,21))
print(lst[1::2])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

9. Access 2nd and 4th strings

Section: List

Problem Statement:

Create a list of 5 strings. Access the second and fourth elements.

Code:

```
s=['a','b','c','d','e']
print(s[1], s[3])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
b d
```

10. Reverse list using slicing

Section: List

Problem Statement:

Create a list of 10 numbers. Reverse the list using slicing.

Code:

```
lst=list(range(1,11))
print(lst[::-1])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

11. Integer array and third element

Section: Array

Problem Statement:

Create an integer array [10,20,30,40,50]. Print the third element.

Code:

```
import array
arr = array.array('i', [10,20,30,40,50])
print(arr[2])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
30
```

12. array type i insert 1-5 slice 1-3

Section: Array

Problem Statement:

Create an array of type 'i' (integers). Insert numbers from 1 to 5. Slice and print elements at index 1–3.

Code:

```
import array
arr = array.array('i', [])
for i in range(1,6):
    arr.append(i)
print(arr[1:4])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
array('i', [2, 3, 4])
```

13. Append 12 to array

Section: Array

Problem Statement:

Create an integer array [2,4,6,8,10]. Append 12 to the array.

Code:

```
import array
arr = array.array('i',[2,4,6,8,10])
arr.append(12)
print(list(arr))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[2, 4, 6, 8, 10, 12]
```

14. Remove element 15

Section: Array

Problem Statement:

Create an array [5,10,15,20,25]. Remove the element 15.

Code:

```
import array
arr = array.array('i',[5,10,15,20,25])
arr.remove(15)
print(list(arr))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[5, 10, 20, 25]
```

15. Extend arrays

Section: Array

Problem Statement:

Create two arrays [1,2,3] and [4,5,6]. Extend the first array with the second

Code:

```
import array
a = array.array('i',[1,2,3])
b = array.array('i',[4,5,6])
a.extend(b)
print(list(a))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
[1, 2, 3, 4, 5, 6]
```

16. Update 2nd element to 99

Section: Array

Problem Statement:

Create an array [1,2,3,4,5]. Update the 2nd element to 99.

Code:

```
import array  
arr = array.array('i',[1,2,3,4,5])  
arr[1]=99  
print(list(arr))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
[1, 99, 3, 4, 5]
```

17. Slice first three elements

Section: Array

Problem Statement:

Create an array [10,20,30,40,50]. Use slicing to print the first three elements.

Code:

```
import array  
arr = array.array('i',[10,20,30,40,50])  
print(list(arr[:3]))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
[10, 20, 30]
```

18. Reverse array using slicing

Section: Array

Problem Statement:

Create an array [100,200,300,400]. Reverse the array using slicing.

Code:

```
import array
arr = array.array('i',[100,200,300,400])
print(list(arr[::-1]))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[400, 300, 200, 100]
```

19. Print every second element

Section: Array

Problem Statement:

Create an array [5,10,15,20,25,30]. Print every second element using slicing.

Code:

```
import array
arr = array.array('i',[5,10,15,20,25,30])
print(list(arr[::-2]))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[5, 15, 25]
```

20. Convert array to list

Section: Array

Problem Statement:

Create an array [2,4,6,8,10]. Convert it to a Python list and print the list.

Code:

```
import array
arr = array.array('i',[2,4,6,8,10])
print(list(arr))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
[2, 4, 6, 8, 10]
```

21. 1D array 0-20 5th element

Section: NumPy 1D

Problem Statement:

Create a 1D array of numbers from 0 to 20. Print the 5th element.

Code:

```
import numpy as np  
arr = np.arange(0,21)  
print(arr[4])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
4
```

22. First 15 odd numbers slice 3-8

Section: NumPy 1D

Problem Statement:

Create a 1D array of the first 15 odd numbers. Slice elements from index 3 to 8.

Code:

```
import numpy as np  
arr = np.arange(1,30,2) [:15]  
print(arr[3:9])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
[ 7  9 11 13 15 17]
```

23. 10 to 100 step10 last element negative index

Section: NumPy 1D

Problem Statement:

Create a 1D array of numbers from 10 to 100 with a step of 10. Print the last element using negative indexing.

Code:

```
import numpy as np
arr = np.arange(10,101,10)
print(arr[-1])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
100
```

24. 1 to 12 reshape 3x4

Section: NumPy 1D

Problem Statement:

Create a 1D array of numbers 1 to 12. Reshape it into a (3,4) array.

Code:

```
import numpy as np
arr = np.arange(1,13)
print(arr.reshape(3,4))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

25. Broadcast add 5

Section: NumPy 1D

Problem Statement:

Create an array [5,10,15,20,25]. Broadcast it by adding 5 to all elements.

Code:

```
import numpy as np
arr = np.array([5,10,15,20,25])
print(arr + 5)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
[10 15 20 25 30]
```

26. 1D array 12 reshape 2x6

Section: NumPy 1D

Problem Statement:

Create a 1D array of 12 elements and reshape it into (2,6).

Code:

```
import numpy as np  
arr = np.arange(12)  
print(arr.reshape(2,6))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
[[ 0  1  2  3  4  5]  
 [ 6  7  8  9 10 11]]
```

27. 50 to 60 slice first5

Section: NumPy 1D

Problem Statement:

Create a 1D array of numbers from 50 to 60. Slice the first 5 elements.

Code:

```
import numpy as np  
arr = np.arange(50,61)  
print(arr[:5])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
[50 51 52 53 54]
```

28. Broadcast multiply 3

Section: NumPy 1D

Problem Statement:

Create a 1D array [2,4,6,8,10]. Broadcast it by multiplying with 3.

Code:

```
import numpy as np
arr = np.array([2,4,6,8,10])
print(arr * 3)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[ 6 12 18 24 30]
```

29. 1 to 9 reshape 3x3

Section: NumPy 1D

Problem Statement:

Create a 1D array of numbers from 1 to 9. Reshape it into a (3,3) 2D array.

Code:

```
import numpy as np
arr = np.arange(1,10)
print(arr.reshape(3, 3))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

30. 1 to 20 slice 5 to end

Section: NumPy 1D

Problem Statement:

Create a 1D array of 20 numbers. Slice elements from index 5 till the end.

Code:

```
import numpy as np
arr = np.arange(1,21)
print(arr[5:])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[ 6  7  8  9 10 11 12 13 14 15 16 17 18 19 20]
```

31. 3x3 numbers 1-9 element row2 col3

Section: NumPy 2D

Problem Statement:

Create a 2D array of shape (3,3) with numbers 1–9. Print the element at row 2, col 3.

Code:

```
import numpy as np
arr = np.arange(1,10).reshape(3,3)
print(arr[1,2])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
6
```

32. 4x4 numbers 1-16 slice first two rows

Section: NumPy 2D

Problem Statement:

Create a 2D array of shape (4,4) with numbers 1–16. Slice the first two rows.

Code:

```
import numpy as np
arr = np.arange(1,17).reshape(4,4)
print(arr[:2])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[[1 2 3 4]
 [5 6 7 8]]
```

33. 3x5 numbers 10-24 last column

Section: NumPy 2D

Problem Statement:

Create a 2D array of shape (3,5) with numbers from 10 to 24. Slice the last column.

Code:

```
import numpy as np  
arr = np.arange(10, 25).reshape(3, 5)  
print(arr[:, -1])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
[14 19 24]
```

34. 2x6 reshape to 3x4

Section: NumPy 2D

Problem Statement:

Create a 2D array of shape (2,6). Reshape it into (3,4).

Code:

```
import numpy as np  
arr = np.arange(12).reshape(2, 6)  
print(arr.reshape(3, 4))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
[[ 0  1  2  3]  
 [ 4  5  6  7]  
 [ 8  9 10 11]]
```

35. 3x3 slice first row

Section: NumPy 2D

Problem Statement:

Create a 2D array (3×3). Slice the first row.

Code:

```
import numpy as np
arr = np.arange(1,10).reshape(3,3)
print(arr[0])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[1 2 3]
```

36. 4x4 bottom-right 2x2 block

Section: NumPy 2D

Problem Statement:

Create a 2D array (4×4). Slice the last two rows and last two columns (bottom-right block).

Code:

```
import numpy as np
arr = np.arange(1,17).reshape(4,4)
print(arr[-2:, -2:])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[[11 12]
 [15 16]]
```

37. 2x3 broadcast add [10,20,30]

Section: NumPy 2D

Problem Statement:

Create a 2D array of shape (2,3). Broadcast by adding [10,20,30] to each row.

Code:

```
import numpy as np
arr = np.array([[1,2,3],[4,5,6]])
print(arr + np.array([10,20,30]))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[[11 22 33]
 [14 25 36]]
```

38. 3x3 diagonal elements

Section: NumPy 2D

Problem Statement:

Create a 2D array (3×3). Print the diagonal elements using indexing.

Code:

```
import numpy as np
arr = np.arange(1,10).reshape(3,3)
print(np.diag(arr))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[1 5 9]
```

39. 5x2 reshape to 2x5

Section: NumPy 2D

Problem Statement:

Create a 2D array (5×2) with numbers 1–10. Reshape it into (2,5).

Code:

```
import numpy as np
arr = np.arange(1,11).reshape(5,2)
print(arr.reshape(2,5))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]]
```

40. 3x4 slice cols 1 and 2

Section: NumPy 2D

Problem Statement:

Create a 2D array (3×4). Slice all rows but only columns 1 and 2.

Code:

```
import numpy as np
arr = np.arange(1,13).reshape(3, 4)
print(arr[:, 1:3])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[[ 2  3]
 [ 6  7]
 [10 11]]
```

41. 3D 2x3x4 numbers 1-24 element [1,2,3]

Section: NumPy 3D

Problem Statement:

Create a 3D array of shape (2,3,4) with numbers 1–24. Print element at [1,2,3].

Code:

```
import numpy as np
arr = np.arange(1,25).reshape(2, 3, 4)
print(arr[1,2,3])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
24
```

42. 3D 2x2x3 slice first block

Section: NumPy 3D

Problem Statement:

Create a 3D array of shape (2,2,3). Slice the first 'block' (all rows/cols of index 0 along axis 0).

Code:

```
import numpy as np
arr = np.arange(1,13).reshape(2, 2, 3)
print(arr[0])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[[1 2 3]
 [4 5 6]]
```

43. 3D 3x3x3 reshape to 9x3

Section: NumPy 3D

Problem Statement:

Create a 3D array (3,3,3) with numbers 1–27. Reshape it into (9,3).

Code:

```
import numpy as np
arr = np.arange(1,28).reshape(3,3,3)
print(arr.reshape(9,3))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[[ 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]]
```

44. 3D 2x3x3 slice second row of first block

Section: NumPy 3D

Problem Statement:

Create a 3D array (2,3,3). Slice the second row of the first block.

Code:

```
import numpy as np
arr = np.arange(1,19).reshape(2,3,3)
print(arr[0,1])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
[4 5 6]
```

45. 3D 2x3x3 broadcast add [5,10,15]

Section: NumPy 3D

Problem Statement:

Create a 3D array (2,3,3). Broadcast by adding [5,10,15] to each row.

Code:

```
import numpy as np  
arr = np.arange(1,19).reshape(2,3,3)  
print(arr + np.array([5,10,15]))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py  
[[[ 6 12 18]  
 [ 9 15 21]  
 [12 18 24]]  
  
[[15 21 27]  
 [18 24 30]  
 [21 27 33]]]
```

46. 3D 2x2x4 slice last two columns of all blocks

Section: NumPy 3D

Problem Statement:

Create a 3D array (2,2,4) with numbers from 1 to 16. Slice the last two columns of all blocks.

Code:

```
import numpy as np  
arr = np.arange(1,17).reshape(2,2,4)  
print(arr[:, :, -2:])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[[[ 3  4]
 [ 7  8]]

 [[11 12]
 [15 16]]]
```

47. 3D 2x3x2 print element [0,1,1]

Section: NumPy 3D

Problem Statement:

Create a 3D array (2,3,2). Print the element at [0,1,1].

Code:

```
import numpy as np
arr = np.arange(1,13).reshape(2,3,2)
print(arr[0,1,1])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
4
```

48. 3D 3x2x2 reshape to 2x3x2

Section: NumPy 3D

Problem Statement:

Create a 3D array (3,2,2). Reshape it into (2,3,2).

Code:

```
import numpy as np
arr = np.arange(1,13).reshape(3,2,2)
print(arr.reshape(2,3,2))
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[[[ 1  2]
 [ 3  4]
 [ 5  6]]

 [[ 7  8]
 [ 9 10]
 [11 12]]]
```

49. 3D 2x3x3 slice last row of every block

Section: NumPy 3D

Problem Statement:

Create a 3D array (2,3,3). Slice the last row of every block.

Code:

```
import numpy as np
arr = np.arange(1,19).reshape(2,3,3)
print(arr[:, -1, :])
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[[ 7  8  9]
 [16 17 18]]
```

50. 3D 2x2x3 multiply by 10

Section: NumPy 3D

Problem Statement:

Create a 3D array (2,2,3). Broadcast by multiplying with 10.

Code:

```
import numpy as np
arr = np.arange(1,13).reshape(2,2,3)
print(arr * 10)
```

Output:

```
PS C:\Users\Mohammad Sayeed> python file.py
[[[ 10  20  30]
 [ 40  50  60]]

 [[ 70  80  90]
 [100 110 120]]]
```