

▼ Instructions

Make a duplicate copy of google colab by clicking **File/ Save a copy in Drive option**. A duplicate copy will get created. It will open up in the new tab on your web browser .After that change **share settings**

▼ CO-1 Review of Python Programming

This colab is designed for you to practice and solve the activities that are based on the following concepts:

1. Python Lists
 2. NumPy Arrays
-

Double-click (or enter) to edit

▼ Activities

▼ Activity 1: Create a 3 X 3 Matrix

To create a 3x3 matrix with values ranging from 2 to 10.

For Example:

```
[[ 2  3  4]
 [ 5  6  7]
 [ 8  9 10]]
```

Follow the steps given below to achieve the desired result:

- **Step 1:** Import `numpy` module.
- **Step 2:** Use `arange()` function to create array of numbers from 2 to 10 and `reshape()` function to reshape your array into another array having 3 rows and 3 columns. Store this reshaped array in a variable `x`.
- **Step 3:** Print variable `x` to get the output.

```
1 import numpy as np
2 x = np.arange(2,11).reshape(3,3)
3 print(x)
4
```

5

```
[[ 2  3  4]
 [ 5  6  7]
 [ 8  9 10]]
```

▼ Activity 2: Change Dimension of an Array and Convert the NumPy Array into a List

Write a program to change the dimension of an array (say `my_arr = [1, 2, 3, 4, 5, 6, 7, 8, 9]`) into a 3 X 3 (3 rows and 3 columns) array and convert this NumPy array into a list.

For Example:

```
Original array is [1 2 3 4 5 6 7 8 9]
Dimension is (9,)
Change array shape to (3, 3) -> 3 rows and 3 columns
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

The data type of the converted variable is
list

```
1 import numpy as np
2 x = np.array([1,2,3,4,5,6,7,8,9])
3 print("CHANGE DIMENSION TO 3 x 3")
4 x.shape = (3,3)
5 print(x)
6 print(type(x))
7 l1 = x.tolist()
8 print(l1)
9 print(type(l1))
10
```

```
CHANGE DIMENSION TO 3 x 3
[[1 2 3]
 [4 5 6]
 [7 8 9]]
<class 'numpy.ndarray'>
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
<class 'list'>
```

▼ Activity 3: Find Square Root

Write a program to perform following task:

- Print the square root of numbers in the list.

For Example:

```
list1 = [4, 16, 9, 1, 25]
[2  4  3  1  5 ]
```

Hint: Use `np.sqrt()` function.

```
1 import numpy as np
2 l1 = [4,16,9,1,25]
3 a = np.sqrt(l1)
4 print(a)
5
```

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

▼ Activity 4: Create and Update a Null NumPy Array

Create a null NumPy array of size 10 and update the sixth value to 11.

A null array is basically an array with all elements as 0.

Follow the steps given below to achieve the desired result:

- **Step 1:** Import the Numpy module as `np`.
- **Step 2:** Create a null array by passing the size i.e. 10 inside the `np.zeros()` function and store it in a variable `null_arr`.
- **Step 3:** Print the null array.
- **Step 4:** Now update the sixth element of the array by using **list indexing** method. As you need to update the sixth element, the index must be 5.
- **Step 5:** Print the updated array in the output.

```
1 import numpy as np
2 nullarray = np.zeros(10)
3 print(nullarray)
4 print("Update 6th Element")
5 nullarray[6] = 11
6 print(nullarray)
7
8
9
```

```
[0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
```

```
Update 6th Element
[ 0.  0.  0.  0.  0.  0. 11.  0.  0.  0.]
```

In the above program we have created a null array by using the `np.zeros()` function of the `numpy` module.

We have updated the 6th element to 11 by using list indexing method.

▼ Activity 5: Populate a Number List

Write a program that populates a list by numbers that lies in the range of 0 - 49 and also divisible by 5. Use List Comprehension method.

Output: [0, 5, 10, 15, 20, 25, 30, 35, 40, 45]

```
1 # Write a program to populate a number list divisible by 5 in a range 0 - 49
2 a = [x for x in range(0,49) if x % 5 ==0]
3 print(a)
4
```

```
[0, 5, 10, 15, 20, 25, 30, 35, 40, 45]
```

Here, using list comprehension method we are running a `for` loop in range 0 - 49 and checking whether the number is divisible by 5 or not, using an `if` condition.

If number satisfies the condition, then that number is appended to `number_list`.

▼ Activity 6: Convert List into Array

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

For Example:

Input: `mylist = [1.23, 23.32, 300, 16.37]`

Data type of `mylist` = `list`

Output: `numpy_array = [1.23, 23.32, 300, 16.37]`

Data type of `numpy_array` = `numpy.ndarray`

```
1 # Program to convert a list into one dimensional NumPy array
2 import numpy as np
3 mylist = [1.23,23.32,300,16.37]
4 print(mylist)
```

```
5 print(type(mylist))
6 numpy_array = np.array(mylist)
7 print(numpy_array)
8 print(type(numpy_array))
9
10

[1.23, 23.32, 300, 16.37]
<class 'list'>
[ 1.23 23.32 300. 16.37]
<class 'numpy.ndarray'>
```

