

# NumPy exercises

## Array creation

1. Create a numpy array of size 10, filled with zeros
2. Create a numpy array with values ranging from 10 to 49 and set its datatype to float

*(Hint: Remember that you can set a specific dtype while defining an array for e.g.*

*Arr1 = np.array([1,2,3,4,5]) #defines an int array by default*

*Arr2 = np.array([1,2,3,4,5], dtype = float) #specifying the dtype as an argument will create a float array).*

3. Create a numpy matrix of 2\*2 integers, filled with ones.
4. Create a numpy matrix of 3\*2 float numbers, filled with ones.
5. Given the X numpy array, create a new numpy array with the same shape and type as X, filled with ones. (*#hint: search the function np.ones\_like()*)
6. Given the X numpy matrix, create a new numpy matrix with the same shape and type as X, filled with zeros. (*hint: similar function as above exists for zeros as well*)

7. Create a numpy matrix of 4\*4 integers, filled with fives.
8. Given the X numpy matrix, create a new numpy matrix with the same shape and type as X, filled with sevens.
9. Create a 3\*3 identity numpy matrix with ones on the diagonal and zeros elsewhere.
10. Create a numpy array, filled with 3 random integer values between 1 and 10.
11. Create a 3\*3\*3 numpy matrix, filled with random float values.
12. Create a numpy array with numbers from 1 to 10
13. Create a numpy array with the odd numbers between 1 to 10
14. Create a numpy array with numbers from 1 to 10, in descending order.
15. Create a 3\*3 numpy matrix, filled with values ranging from 0 to 8

## **Array indexing:**

1. Given the X numpy array, show its first element

```
X = np.array(['A', 'B', 'C', 'D', 'E'])
```

2. Given the X numpy array, show its last element

```
X = np.array(['A', 'B', 'C', 'D', 'E'])
```

3. Given the X numpy array, show its first three elements

```
X = np.array(['A', 'B', 'C', 'D', 'E'])
```

4. Given the X numpy array, show all middle elements

```
X = np.array(['A', 'B', 'C', 'D', 'E'])
```

5. Given the X numpy array, show the elements in reverse position.

```
X = np.array(['A', 'B', 'C', 'D', 'E'])
```

6. Given the X numpy array, show the elements in an odd position.

```
X = np.array(['A', 'B', 'C', 'D', 'E'])
```

7. Given the X numpy matrix, show the first row's elements.

```
X = np.array([
    [1, 2, 3, 4],
    [5, 6, 7, 8],
    [9, 10, 11, 12],
])
```

```
    [13, 14, 15, 16]  
])
```

8. Given the X numpy matrix, show the last row elements.

```
X = np.array([  
    [1, 2, 3, 4],  
    [5, 6, 7, 8],  
    [9, 10, 11, 12],  
    [13, 14, 15, 16]  
])
```

9. Given the X numpy matrix, show the first element on first row

```
X = np.array([  
    [1, 2, 3, 4],  
    [5, 6, 7, 8],  
    [9, 10, 11, 12],  
    [13, 14, 15, 16]  
])
```

10. Given the X numpy matrix, show the last element on last row.

```
X = np.array([  
    [1, 2, 3, 4],  
    [5, 6, 7, 8],  
    [9, 10, 11, 12],  
    [13, 14, 15, 16]  
])
```

11. Given the X numpy matrix, show the first two elements on the first two rows.

```
X = np.array([  
    [1, 2, 3, 4],  
    [5, 6, 7, 8],  
    [9, 10, 11, 12],  
])
```

```
        [13, 14, 15, 16]
    ])
```

12. Given the X numpy matrix, show the last two elements on the last two rows

```
X = np.array([
    [1, 2, 3, 4],
    [5, 6, 7, 8],
    [9, 10, 11, 12],
    [13, 14, 15, 16]
])
```

## **Array manipulation**

1. Convert the given integer numpy array to float (*you need to google the function that will help you do this*)

```
X = [-5, -3, 0, 10, 40]
```

2. Reverse the given numpy array (first element becomes last) through indexing

```
X = [-5, -3, 0, 10, 40]
```

3. Given the X numpy array, set the fifth element equal to 1

```
X = np.zeros(10)
```

4. Given the X numpy matrix, change the last row with all 1

```
X = np.array([
    [1, 2, 3, 4],
    [5, 6, 7, 8],
    [9, 10, 11, 12],
])
```

```
    [13, 14, 15, 16]  
])
```

5. Given the X numpy matrix, add 5 to every element

```
X = np.array([  
    [1, 2, 3, 4],  
    [5, 6, 7, 8],  
    [9, 10, 11, 12],  
    [13, 14, 15, 16]  
])
```