1Create an array with zero and ones and print the output?

In [2]:
```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-pac
kages (1.20.1)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
import numpy as np
array_size = 10
zeros_and_ones_array = np.random.randint(2, size=array_size)
print(zeros_and_ones_array)
```

```
[0 0 1 0 1 0 1 1 0 1]
```

2 Create an array and print the output?

In [5]:
```

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

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

3 Create an array whose initial content is random and print the output?

In [7]:
```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-pac
kages (1.20.1)
Note: you may need to restart the kernel to use updated packages.
```

In [8]:
```
import numpy as np
array_size = 5
random_array = np.random.random(array_size)
print(random_array)
```

```
[0.63798602 0.16032906 0.92968192 0.13070191 0.11576051]
```

4 create an array with the range of values with even intervals?

In [10]:
```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-pac
kages (1.20.1)
Note: you may need to restart the kernel to use updated packages.
```

In [11]:
```
import numpy as np
start_value = 0
end_value = 10
interval = 2
even_interval_array = np.arange(start_value, end_value, interval)
print(even_interval_array)
```

```
[0 2 4 6 8]
```

5 Create an array with values that are spaced linearly in a specified interval?

In [13]:
```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-pac
kages (1.20.1)
Note: you may need to restart the kernel to use updated packages.
```

In [14]:
```
import numpy as np
start_value = 0
end_value = 10
num_values = 6
linearly_spaced_array = np.linspace(start_value, end_value, num_values)
print(linearly_spaced_array)
```

```
[ 0.  2.  4.  6.  8. 10.]
```

6 Access and manipulate elements in the array?

In [16]:
```
import numpy as np
my_array = np.array([1, 2, 3, 4, 5])
first_element = my_array[0]
print(first_element)
```

```
1
```

7 Create a two dimentional array and check the shape of the array?

In [18]:
```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-pac
kages (1.20.1)
Note: you may need to restart the kernel to use updated packages.
```

In [20]:
```
import numpy as np
two_dimensional_array = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
shape = two_dimensional_array.shape
print(two_dimensional_array)
print("Shape of the array:", shape)
```

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
Shape of the array: (3, 3)
```

8 Using the range() and linspace() function to evenly space values in a specified interval?

In [22]:
```

start_value = 0
end_value = 10
step = 2
evenly_spaced_integers = range(start_value, end_value, step)
evenly_spaced_integers_list = list(evenly_spaced_integers)
print(evenly_spaced_integers_list)
```

```
[0, 2, 4, 6, 8]
```

9 Create an array of random values between 0 and 1 in a given shape?

In [23]:
```

start_value = 0
end_value = 10
step = 2
evenly_spaced_integers = range(start_value, end_value, step)
evenly_spaced_integers_list = list(evenly_spaced_integers)

print(evenly_spaced_integers_list)
```

```
[0, 2, 4, 6, 8]
```

10 Repeat each element of an array by a

In [25]:
```
import numpy as np
original_array = np.array([1, 2, 3])
repeat_times = 3
repeated_array = np.repeat(original_array, repeat_times)
print(repeated_array)
```

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

11 How do you know the shape and size of an array?

In [26]:
```
import numpy as np
my_array = np.array([[1, 2, 3], [4, 5, 6]])
shape = my_array.shape
size = my_array.size

print(shape)
print(size)
```

```
(2, 3)
6
```

12 Create ann array that indicates the total number of elements in an array?

In [27]:
```
import numpy as np
original_array = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
total_elements_array = np.array([original_array.size])
print(total_elements_array)
```

```
[9]
```

13 To find the number of dimensions of the array?

In [28]:
```
import numpy as np
my_array = np.array([[1, 2, 3], [4, 5, 6]])
a = my_array.ndim
print(a)
```

```
2
```

14 Create an array and reshape into a new array?

In [29]:
```
import numpy as np
original_array = np.array([1, 2, 3, 4, 5, 6, 7, 8])
new_shape = (2, 4)
reshaped_array = original_array.reshape(new_shape)
print("Original Array:")
print(original_array)
print("Reshaped Array:")
print(reshaped_array)
```

```
Original Array:
[1 2 3 4 5 6 7 8]
Reshaped Array:
[[1 2 3 4]
 [5 6 7 8]]
```

15 Create a null array of size 10?

In [30]:
```

null_array = [0] * 10
print(null_array)
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

16 Array with numbers divisible by 7?

In [31]:
```
array_divisible_by_7 = np.arange(10, 50)[np.arange(10, 50) % 7 == 0]
print("\n15. Numbers divisible by 7:")
print(array_divisible_by_7)
```

```

15. Numbers divisible by 7:
[14 21 28 35 42 49]
```

16 Perform operations using arrays?

In [32]:
```
array_op1 = np.array([1, 2, 3])
array_op2 = np.array([3, 2, 1])
print("\n16. Arithmetic operations:")
print("Addition:", array_op1 + array_op2)
print("Subtraction:", array_op1 - array_op2)
print("Multiplication:", array_op1 * array_op2)
print("Division:", array_op1 / array_op2)
```

```

16. Arithmetic operations:
Addition: [4 4 4]
Subtraction: [-2  0  2]
Multiplication: [3 4 3]
Division: [0.33333333 1.         3.        ]
```

17: Relational operations using arrays?

In [33]:
```
array_rel1 = np.array([1, 2, 3])
array_rel2 = np.array([2, 2, 2])
print("\n17. Relational operations:")
print("Equal:", array_rel1 == array_rel2)
print("Not Equal:", array_rel1 != array_rel2)
print("Greater than:", array_rel1 > array_rel2)
print("Less than:", array_rel1 < array_rel2)
```

```

17. Relational operations:
Equal: [False  True False]
Not Equal: [ True False  True]
Greater than: [False False  True]
Less than: [ True False False]
```

18: Use Arithmetic operators and print the output using arrays?

In [34]:
```
array_arithmetic = np.array([1, 2, 3])
array_arithmetic += 2  # Adds 2 to each element of the array
print("\n18. Arithmetic operations on an array:")
print("After adding 2:", array_arithmetic)
```

```

18. Arithmetic operations on an array:
After adding 2: [3 4 5]
```

19: Use Relational operators and print the results using arrays (Repeats Problem Statement 17)

In [35]:
```
array_rel1 = np.array([1, 2, 3])
array_rel2 = np.array([2, 2, 2])
print("\n17. Relational operations:")
print("Equal:", array_rel1 == array_rel2)
print("Not Equal:", array_rel1 != array_rel2)
print("Greater than:", array_rel1 > array_rel2)
print("Less than:", array_rel1 < array_rel2)
```

```

17. Relational operations:
Equal: [False  True False]
Not Equal: [ True False  True]
Greater than: [False False  True]
Less than: [ True False False]
```