From Python to Numpy
0% COMPLETED
Search Course
Introduction
What is NumPy?
Creation in NumPy
Reshaping in NumPy
Indexing in NumPy
Broadcasting in NumPy
NumPy Vectorization
Readability vs. Speed
Anatomy of an Array
Introduction
Memory layout

Creation in NumPy

This lesson helps you learn how to create a NumPy array in different ways.

We'll cover the following
- Create an Array of Zeros
- Create an Array of Ones
- Create an Array of 0's and 1's
- Create an Array of 2's
- Create a NumPy Array of any Length
- Reshape a NumPy Array into a Column Vector
- Generate Array of Random Numbers and in Grid Format
- Create a Linspace
- Create a Mesh Grid

For using numpy, import the numpy library.

```
import numpy
```

Create an Array of Zeros

To create a numpy array containing zeros, write: `np.zeros(size)`

To create an array of size 9 write: `np.zeros(9)`

Here is how this array is stored in memory:

```
import numpy as np
Z=np.zeros(9)
print(Z)
```

RUN SAVE RESET

Output 0.9101s

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

Create an Array of Ones

To create a numpy array containing ones, write: `np.ones(size)`.

To create an array of size 9 write: `np.ones(9)`

Here is how this array is stored in memory:

```
import numpy as np
Z = np.ones(9)
print(Z)
```

RUN SAVE RESET

Output 0.3661s

```
[1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

Create an Array of 0's and 1's

To create an array of zeros and ones, use `np.array([1,0,0,0,0,0,1,0])`:

Here is how the array is stored in memory:

```
import numpy as np
Z = np.array([1,0,0,0,0,0,1,0])
print(Z)
```

RUN SAVE RESET

Output 1.341s

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

Create an Array of 2's

To create an array of 2's write: `2*np.ones(size)`.

To create an array of 2's of size 9 write: `2*np.ones(9)`.

Here is how the array is stored in memory:

```
import numpy as np
Z = 2*np.ones(9)
print(Z)
```

RUN SAVE RESET

Output 0.3971s

```
[2. 2. 2. 2. 2. 2. 2. 2. 2.]
```

Create a NumPy Array of any Length

To create an array of any length write: `np.arange(size)`.

To create an array of size 9 write : `np.arange(9)`.

Here is how the array is stored in memory:

```
import numpy as np
Z = np.arange(9)
print(Z)
```

RUN SAVE RESET

Output 0.4991s

```
[0 1 2 3 4 5 6 7 8]
```

Reshape a NumPy Array into a Column Vector

To reshape a numpy array,write: `np.arange(size).reshape(size,1)`.

To reshape a numpy array into 9 rows and 1 column ,write: `np.arange(9).reshape(9,1)`.

Here is how the array is stored is stored in memory:

```
import numpy as np
Z = np.arange(9).reshape(9,1)
print(Z)
```

RUN SAVE RESET

Output 0.8861s

```
[[0]
 [1]
 [2]
 [3]
 [4]
 [5]
 [6]
 [7]
 [8]]
```

Generate Array of Random Numbers and in Grid Format

To generate an array of random size, write: `np.random.randint(0,size,(x_dimension,y_dimension))`.

To generate an array of random numbers from 0 to 9 and x dimension 3 and y dimension 3, write: `np.random.randint(0,9,(3,3))`.

Here is how the array is stored in memory:

```
import numpy as np
Z=np.random.randint(0,9,(3,3))
print(Z)
```

RUN SAVE RESET

Output 0.6771s

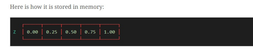
```
[[4 5 7]
 [3 0 2]
 [2 8 0]]
```

Create a Linspace

To create evenly spaced numbers over a specified interval write : `np.linspace(start, stop, size)`

To create a linspace of range 0-1 and size 5 , write : `Z = np.linspace(0, 1, 5)`.

Here is how it is stored in memory:




```
import numpy as np
Z = np.linspace(0, 1, 5)
print(Z)
```

RUN SAVE RESET

Output 0.8661s

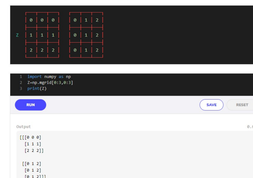
```
[0.  0.25 0.5  0.75 1.  ]
```

Create a Mesh Grid

To create a dense multi-dimensional "meshgrid" , write: `np.mgrid[0:x_dimension,0:y_dimension]`

To create a grid in numpy of size(3*3),write: `np.mgrid[0:3,0:3]`

Here is how a mesh grid is stored in memory:




```
import numpy as np
Z=np.mgrid[0:3,0:3]
print(Z)
```

RUN SAVE RESET

Output 0.6161s

```
[[[0 0 0]
  [1 1 1]
  [2 2 2]]

 [[0 1 2]
  [0 1 2]
  [0 1 2]]]
```

Solve this Quiz!

1 How would you create a null vector of size 10?

A)
```
import numpy as np
np.zeros(10)
```

B)
```
import numpy as np
np.ones(10)
```

COMPLETED 0%

1 of 3

Now that we have learned to create a NumPy, let's move on to the next lesson "Reshaping in NumPy".

Mark as Completed

Back
What is NumPy?

Next
Reshaping in NumPy