

▼ TENSORS

- A torch.Tensor is a multi-dimensional matrix containing elements of a single data type.
- Similar To Numpy Arrays, but full of fun things that make them work better on GPU's (vs regular CPU's).
- default data type of float32.
- More suitable for deep learning than a numpy array.

```
1 import torch
2 import numpy as np
```

▼ LISTS

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

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

```
1 my_list = [[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]]
2 my_list
3
```

```
[[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]]
```

▼ NUMPY ARRAYS

```
1 np1 = np.random.rand(3, 4)
2 np1
```

```
array([[0.58965169, 0.14007069, 0.46423785, 0.76590038],
       [0.58988162, 0.96104904, 0.10133875, 0.49474744],
       [0.47704003, 0.97830051, 0.69495892, 0.6912239 ]])
```

```
1 np1.dtype
```

```
dtype('float64')
```

▼ TENSORS

```
1 tensor_2d1 = torch.randn(3, 4)
2 tensor_2d1
```

```
tensor([[ -0.0508,  0.8274, -0.6107, -0.4827],
        [-1.1635,  0.4372, -1.0339,  0.0574],
        [-0.8250, -0.2743, -1.0665, -0.3182]])
```

```
1 tensor_3d = torch.zeros(2, 3, 4)
2 tensor_3d
```

```
tensor([[[0., 0., 0., 0.],
```

```
[0., 0., 0., 0.],  
[0., 0., 0., 0.]],  
  
[[0., 0., 0., 0.],  
 [0., 0., 0., 0.],  
 [0., 0., 0., 0.]])
```

1