

```
In [1]: import torch
```

## Basics of Tensor

```
In [2]: torch.empty(size=(2,3,4))
```

```
Out[2]: tensor([[[ -1.5369e+29,  3.0882e-41,  7.0065e-44,  7.0065e-44],
                  [ 6.3058e-44,  6.7262e-44,  6.8664e-44,  6.3058e-44],
                  [ 6.8664e-44,  6.8664e-44,  1.1771e-43,  6.7262e-44]],
                [[ 7.8473e-44,  8.1275e-44,  7.1466e-44,  7.1466e-44],
                  [ 8.1275e-44,  6.7262e-44,  7.4269e-44,  6.4460e-44],
                  [ 7.0065e-44,  7.9874e-44,  7.5670e-44,  7.0065e-44]])])
```

```
In [3]: torch.ones(size=(3,4))*2
```

```
Out[3]: tensor([[2., 2., 2., 2.],
                [2., 2., 2., 2.],
                [2., 2., 2., 2.]])
```

```
In [4]: t=torch.arange(6)
        t.dtype
```

```
Out[4]: torch.int64
```

```
In [5]: t.double().dtype
```

```
Out[5]: torch.float64
```

```
In [6]: torch.linspace(start = 1 , end=20, steps=30).shape
```

```
Out[6]: torch.Size([30])
```

```
In [7]: import numpy as np
```

```
In [8]: arr=np.arange(16)
        arr
```

```
Out[8]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15])
```

```
In [9]: t=torch.tensor(arr)
        t
```

```
Out[9]: tensor([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15])
```

```
In [10]: t=torch.from_numpy(arr)
```

```
In [11]: t.numpy()
```

```
Out[11]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15])
```

```
In [12]: device = "cuda" if torch.cuda.is_available() else "cpu"  
device
```

```
Out[12]: 'cuda'
```

```
In [13]: torch.tensor([1,2,3,4],dtype=torch.float32,device=device,requires_grad=True)
```

```
Out[13]: tensor([1., 2., 3., 4.], device='cuda:0', requires_grad=True)
```

```
In [14]: torch.eye(5)
```

```
Out[14]: tensor([[1., 0., 0., 0., 0.],  
                [0., 1., 0., 0., 0.],  
                [0., 0., 1., 0., 0.],  
                [0., 0., 0., 1., 0.],  
                [0., 0., 0., 0., 1.]])
```

```
In [15]: torch.diag(torch.tensor([1,2,3,5]))
```

```
Out[15]: tensor([[1, 0, 0, 0],  
                [0, 2, 0, 0],  
                [0, 0, 3, 0],  
                [0, 0, 0, 5]])
```

```
In [17]: t= torch.rand(size=(4,4))  
t
```

```
Out[17]: tensor([[0.9226, 0.8020, 0.0982, 0.4110],  
                [0.3395, 0.5142, 0.0340, 0.7705],  
                [0.9272, 0.9321, 0.5185, 0.3325],  
                [0.1675, 0.0060, 0.5868, 0.8137]])
```

```
In [18]: torch.diag(t)
```

```
Out[18]: tensor([0.9226, 0.5142, 0.5185, 0.8137])
```

```
In [ ]:
```

## Tensor Operation

```
In [20]: x = torch.ones(size=(3,4))  
y = torch.ones(size=(3,4))*4
```

```
In [21]: print(x)  
print(y)
```

```
tensor([[1., 1., 1., 1.],  
        [1., 1., 1., 1.],  
        [1., 1., 1., 1.]])  
tensor([[4., 4., 4., 4.],  
        [4., 4., 4., 4.],  
        [4., 4., 4., 4.]])
```

```
In [22]: z=torch.empty(size=(3,4))
```

```
In [23]: torch.add(x,y,out=z)
z
```

```
Out[23]: tensor([[5., 5., 5., 5.],
                 [5., 5., 5., 5.],
                 [5., 5., 5., 5.]])
```

```
In [25]: z=torch.add(x,y)
z
```

```
Out[25]: tensor([[5., 5., 5., 5.],
                 [5., 5., 5., 5.],
                 [5., 5., 5., 5.]])
```

```
In [26]: z=x+y
```

```
In [27]: #x+=y
x.add_(y)
x
```

```
Out[27]: tensor([[5., 5., 5., 5.],
                 [5., 5., 5., 5.],
                 [5., 5., 5., 5.]])
```

```
In [32]: torch.arange(4)*3 <5
```

```
Out[32]: tensor([ True,  True, False, False])
```

```
In [31]: x.pow(2)
```

```
Out[31]: tensor([[25., 25., 25., 25.],
                 [25., 25., 25., 25.],
                 [25., 25., 25., 25.]])
```

```
In [35]: x = torch.ones(size=(3,4))
y = torch.ones(size=(4,5))*3
```

```
In [36]: torch.matmul(x,y)
```

```
Out[36]: tensor([[12., 12., 12., 12., 12.],
                 [12., 12., 12., 12., 12.],
                 [12., 12., 12., 12., 12.]])
```

```
In [38]: batch=32
m=3
n=4
p=5
```

```
In [40]: x = torch.ones(size=(batch,m,n))
y = torch.ones(size=(batch,n,p))
torch.bmm(x,y).shape
```

```
Out[40]: torch.Size([32, 3, 5])
```

## Reshaping a tensor

```
In [41]: t=torch.tensor([1,2,3,4,5,6])  
t.shape
```

```
Out[41]: torch.Size([6])
```

```
In [47]: t.unsqueeze(axis=1).squeeze()
```

```
Out[47]: tensor([1, 2, 3, 4, 5, 6])
```

```
In [51]: t=torch.arange(15)  
t
```

```
Out[51]: tensor([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14])
```

```
In [52]: t.view(1,3,5)
```

```
Out[52]: tensor([[ 0,  1,  2,  3,  4],  
                [ 5,  6,  7,  8,  9],  
                [10, 11, 12, 13, 14]])
```

```
In [61]: tensor=torch.linspace(0,10,24).reshape(2,3,4)
```

```
In [62]: tensor
```

```
Out[62]: tensor([[[ 0.0000,  0.4348,  0.8696,  1.3043],  
                 [ 1.7391,  2.1739,  2.6087,  3.0435],  
                 [ 3.4783,  3.9130,  4.3478,  4.7826]],  
                [[ 5.2174,  5.6522,  6.0870,  6.5217],  
                 [ 6.9565,  7.3913,  7.8261,  8.2609],  
                 [ 8.6957,  9.1304,  9.5652, 10.0000]]])
```

```
In [64]: tensor[0][:,:2]
```

```
Out[64]: tensor([[0.0000, 0.4348],  
                [1.7391, 2.1739],  
                [3.4783, 3.9130]])
```

```
In [66]: tensor.shape
```

```
Out[66]: torch.Size([2, 3, 4])
```

```
In [69]: tensor.sum(axis=2)
```

```
Out[69]: tensor([[ 2.6087,  9.5652, 16.5217],  
                [23.4783, 30.4348, 37.3913]])
```