```
tensor.ndim
```

```
3
```

```
tensor.shape
```

```
torch.Size([1, 3, 3])
```

Random Tensor

```
random_tensor = torch.rand(3,4)
```

```
random_tensor
```

```
tensor([[0.8751, 0.4789, 0.6935, 0.0265],
        [0.5623, 0.9521, 0.4083, 0.0548],
        [0.5759, 0.3718, 0.2234, 0.6170]])
```

```
torch.rand(3,3)
```

```
tensor([[0.6737, 0.7784, 0.7657],
        [0.7395, 0.5793, 0.0958],
        [0.5229, 0.4830, 0.7211]])
```

```
zeros = torch.zeros(size=(3,4))
```

```
zeros
```

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

```
ones= torch.ones(size=(3,4))
```

```
ones
```

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

```
ones.dtype
```

```
torch.float32
```

```
torch.range(0,10)
```

```
<ipython-input-3-dfdec9b83f7d>:1: UserWarning: torch.range is deprecated and will be removed in a future release because its behavio
  torch.range(0,10)
tensor([ 0.,  1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

```
one_to_ten=torch.arange(start=1,end=11,step=1)
one_to_ten
```

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

```
ten_zeros=torch.zeros_like(input=one_to_ten)
ten_zeros
```

```
tensor([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

```
#manipulating Tensor
# Addition
# Subtraction
# Multiplication(element wise)
# DIvidation
# Matrix Multiplication
```

```
tensor=torch.tensor([1,2,3])
tensor+10
```

```
#find the sum
torch.sum(x)
```

```
tensor(450)
```

```
x.sum()
```

```
tensor(450)
```

```
# finding the positional min & max
x.argmin()
```

```
tensor(0)
```

```
x.argmax()
```

```
tensor(9)
```

Reshaping

```
# Lets Create a tensor
import torch
x=torch.arange(1.,10.)
x,x.shape
```

```
(tensor([1., 2., 3., 4., 5., 6., 7., 8., 9.]), torch.Size([9]))
```

```
# Add an extra Dimention
x_reshaped=x.reshape(1,9)
x_reshaped,x_reshaped.shape
```

```
(tensor([[1., 2., 3., 4., 5., 6., 7., 8., 9.]]), torch.Size([1, 9]))
```

```
# view
z=x.view(1,9)
z,z.shape
```

```
(tensor([[1., 2., 3., 4., 5., 6., 7., 8., 9.]]), torch.Size([1, 9]))
```

```
# changing Z changes x
z[:,0]=5
z,x
```

```
(tensor([[5., 2., 3., 4., 5., 6., 7., 8., 9.]]),
 tensor([5., 2., 3., 4., 5., 6., 7., 8., 9.]))
```

```
x_stacked=torch.stack([x,x,x,x],dim=0)
x_stacked
```

```
tensor([[5., 2., 3., 4., 5., 6., 7., 8., 9.],
        [5., 2., 3., 4., 5., 6., 7., 8., 9.],
        [5., 2., 3., 4., 5., 6., 7., 8., 9.],
        [5., 2., 3., 4., 5., 6., 7., 8., 9.]])
```

```
# torch.squeeze(). remove all single dimensions from a target tensor
x_reshaped
```

```
tensor([[5., 2., 3., 4., 5., 6., 7., 8., 9.]])
```

```
x_reshaped.shape
```

```
torch.Size([1, 9])
```

```
x_reshaped.squeeze()
```

```
tensor([5., 2., 3., 4., 5., 6., 7., 8., 9.])
```

```
x_reshaped.squeeze().shape
```

```
torch.Size([9])
```