```
import tensorflow as tf

# Scalar (Rank 0 Tensor)
scalar = tf.constant(7)
print("Scalar:", scalar, "Shape:", scalar.shape)

# Vector (Rank 1 Tensor)
vector = tf.constant([1, 2, 3])
print("Vector:", vector, "Shape:", vector.shape)

# Matrix (Rank 2 Tensor)
matrix = tf.constant([[1, 2], [3, 4]])
print("Matrix:", matrix, "Shape:", matrix.shape)

# Higher-Dimensional Tensor (Rank 3)
tensor_3d = tf.constant([[[1, 2], [3, 4]], [[5, 6], [7, 8]]])
print("3D Tensor:", tensor_3d, "Shape:", tensor_3d.shape)
```

```
Scalar: tf.Tensor(7, shape=(), dtype=int32) Shape: ()
Vector: tf.Tensor([1 2 3], shape=(3,), dtype=int32) Shape: (3,)
Matrix: tf.Tensor(
[[1 2]
 [3 4]], shape=(2, 2), dtype=int32) Shape: (2, 2)
3D Tensor: tf.Tensor(
[[[1 2]
  [3 4]]

 [[5 6]
  [7 8]]], shape=(2, 2, 2), dtype=int32) Shape: (2, 2, 2)
```

```
a=1.2
aa=tf.constant(1.2)
type(a), type(aa), tf.is_tensor(aa)
```

```
(float, tensorflow.python.framework.ops.EagerTensor, True)
```

```
tensor= tf.constant(3.14)
print(type(tensor))
```

```
<class 'tensorflow.python.framework.ops.EagerTensor'>
```

```
x=tf.constant([1,2.,3.3])
x
```

```
<tf.Tensor: shape=(3,), dtype=float32, numpy=array([1. , 2. , 3.3], dtype=float32)>
```

```
x.numpy()
```

```
array([1. , 2. , 3.3], dtype=float32)
```

```
a=tf.constant([1.2])
a,a.shape
```

```
(<tf.Tensor: shape=(1,), dtype=float32, numpy=array([1.2], dtype=float32)>,
 TensorShape([1]))
```

```
a=tf.constant([1,2,3.])
a,a.shape
```

```
(<tf.Tensor: shape=(3,), dtype=float32, numpy=array([1., 2., 3.], dtype=float32)>,
 TensorShape([3]))
```

```
a=tf.constant([[1,2],[3,4]])
a,a.shape
```

```
(<tf.Tensor: shape=(2, 2), dtype=int32, numpy=
 array([[1, 2],
        [3, 4]], dtype=int32)>,
 TensorShape([2, 2]))
```

```
a=tf.constant([[1,2,3],[4,5,6],[7,8,9]])
a,a.shape
```

```
(<tf.Tensor: shape=(3, 3), dtype=int32, numpy=
 array([[1, 2, 3],
        [4, 5, 6],
        [7, 8, 9]], dtype=int32)>,
 TensorShape([3, 3]))
```

```
a= tf.constant([[[1,2],[3,4]],[[5,6],[7,8]]])
a
```

```
<tf.Tensor: shape=(2, 2, 2), dtype=int32, numpy=
array([[[1, 2],
        [3, 4]],

       [[5, 6],
        [7, 8]]], dtype=int32)>
```

```
a= tf.constant([[[1,2],[3,4],[10,11]],[[5,6],[7,8],[11,12]]])
a
```

```
<tf.Tensor: shape=(2, 3, 2), dtype=int32, numpy=
array([[[ 1,  2],
        [ 3,  4],
        [10, 11]],

       [[ 5,  6],
        [ 7,  8],
        [11, 12]]], dtype=int32)>
```

```
a=tf.constant('Hello, Deep Learning')
a
```

```
<tf.Tensor: shape=(), dtype=string, numpy=b'Hello, Deep Learning'>
```

```
tf.strings.lower(a)
```

```
<tf.Tensor: shape=(), dtype=string, numpy=b'hello, deep learning'>
```

```
a=tf.constant(["Hello","Tensorflow","World"])
tf.strings.join(a)
```

```
<tf.Tensor: shape=(), dtype=string, numpy=b'HelloTensorflowWorld'>
```

```
a=tf.constant('Hello Deep Learning')
tf.strings.length(a)
```

```
<tf.Tensor: shape=(), dtype=int32, numpy=19>
```

```
a=tf.constant('Hello Deep Learning')
tf.strings.split(a)
```

```
<tf.Tensor: shape=(3,), dtype=string, numpy=array([b'Hello', b'Deep', b'Learning'], dtype=object)>
```

```
a=tf.constant(False)
a
```

```
<tf.Tensor: shape=(), dtype=bool, numpy=False>
```

```
a=tf.constant([True, False])
a
```

```
<tf.Tensor: shape=(2,), dtype=bool, numpy=array([ True, False])>
```

```
a=tf.constant(True)
a is True
```

```
False
```

```
a==True
a
```

```
<tf.Tensor: shape=(), dtype=bool, numpy=True>
```

```
tf.constant(123456789, dtype=tf.int16)
```

```
<tf.Tensor: shape=(), dtype=int16, numpy=-13035>
```

```
tf.constant(123456789, dtype=tf.int32)
```

```
<tf.Tensor: shape=(), dtype=int32, numpy=123456789>
```

```
import numpy as np
tf.constant(np.pi, dtype=tf.float32)
```

```
<tf.Tensor: shape=(), dtype=float32, numpy=3.1415927410125732>
```

```
tf.constant(np.pi,dtype=tf.float64)
```

```
<tf.Tensor: shape=(), dtype=float64, numpy=3.141592653589793>
```

```
a=tf.constant(3.14,dtype=tf.float16)
print('before ', a.dtype)
if a.dtype!=tf.float32:
  a=tf.cast(a,tf.float32)
  print('after ', a.dtype)
```

```
before  <dtype: 'float16'>
after  <dtype: 'float32'>
```

```
a=tf.constant(123456789, dtype=tf.int32)
tf.cast(a,tf.int16)
```

```
<tf.Tensor: shape=(), dtype=int16, numpy=-13035>
```

```
a=tf.constant([True,False])
tf.cast(a,tf.int32)
```

```
<tf.Tensor: shape=(2,), dtype=int32, numpy=array([1, 0], dtype=int32)>
```

```
a=tf.constant([-1,0,1,2])
aa=tf.Variable(a)
aa.name, aa.trainable
```

```
('Variable:0', True)
```

```
a=tf.Variable([[1,2],[3,4]])
a
```

```
<tf.Variable 'Variable:0' shape=(2, 2) dtype=int32, numpy=
array([[1, 2],
       [3, 4]], dtype=int32)>
```

```
tf.convert_to_tensor([1,2.])
```

```
<tf.Tensor: shape=(2,), dtype=float32, numpy=array([1., 2.], dtype=float32)>
```

```
tf.convert_to_tensor(np.array([[1,2.],[3,4]]))
```

```
<tf.Tensor: shape=(2, 2), dtype=float64, numpy=
array([[1., 2.],
       [3., 4.]])>
```

```
tf.zeros([]), tf.ones([])
```

```
(<tf.Tensor: shape=(), dtype=float32, numpy=0.0>,
 <tf.Tensor: shape=(), dtype=float32, numpy=1.0>)
```

```
tf.zeros([1]), tf.ones([1])
```

```
(<tf.Tensor: shape=(1,), dtype=float32, numpy=array([0.], dtype=float32)>,
 <tf.Tensor: shape=(1,), dtype=float32, numpy=array([1.], dtype=float32)>)
```

```
tf.zeros([2,2])
```

```
<tf.Tensor: shape=(2, 2), dtype=float32, numpy=
array([[0., 0.],
       [0., 0.]], dtype=float32)>
```

```
tf.ones([3,2])
```

```
<tf.Tensor: shape=(3, 2), dtype=float32, numpy=
array([[1., 1.],
       [1., 1.],
       [1., 1.]], dtype=float32)>
```

```
a=tf.ones([2,3])
tf.zeros_like(a)
```

```
<tf.Tensor: shape=(2, 3), dtype=float32, numpy=
array([[0., 0., 0.],
       [0., 0., 0.]], dtype=float32)>
```

```
a=tf.zeros([3,2])
tf.ones_like(a)
```

```
<tf.Tensor: shape=(3, 2), dtype=float32, numpy=
array([[1., 1.],
       [1., 1.],
       [1., 1.]], dtype=float32)>
```

```
tf.fill([],-1)
```

```
<tf.Tensor: shape=(), dtype=int32, numpy=-1>
```

```
tf.fill([1],-1)
```

```
<tf.Tensor: shape=(1,), dtype=int32, numpy=array([-1], dtype=int32)>
```

```
tf.fill([2,2],99)
```

```
<tf.Tensor: shape=(2, 2), dtype=int32, numpy=
array([[99, 99],
       [99, 99]], dtype=int32)>
```

```
tf.random.normal([2,2])
```

```
<tf.Tensor: shape=(2, 2), dtype=float32, numpy=
array([[ 0.14233631, -1.1423683 ],
       [ 0.61008525,  0.3722208 ]], dtype=float32)>
```

```
tf.random.normal([2,2],mean=1,stddev=2)
```

```
<tf.Tensor: shape=(2, 2), dtype=float32, numpy=
array([[-0.10383248,  2.1312323 ],
       [-1.3579569 ,  1.4205459 ]], dtype=float32)>
```

```
tf.random.uniform([2,2])
```

```
<tf.Tensor: shape=(2, 2), dtype=float32, numpy=
array([[0.63056767, 0.33726573],
       [0.01888847, 0.37670958]], dtype=float32)>
```

```
tf.random.uniform([2,2],maxval=10)
```

```
<tf.Tensor: shape=(2, 2), dtype=float32, numpy=
array([[4.5839014, 1.056267 ],
       [7.737305 , 0.6368363]], dtype=float32)>
```

```
tf.random.uniform([2,2],maxval=100, dtype=tf.int32)
```

```
<tf.Tensor: shape=(2, 2), dtype=int32, numpy=
array([[72, 51],
       [44, 11]], dtype=int32)>
```

```
tf.range(10)
```

```
<tf.Tensor: shape=(10,), dtype=int32, numpy=array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], dtype=int32)>
```

```
tf.range(10,delta=2)
```

```
<tf.Tensor: shape=(5,), dtype=int32, numpy=array([0, 2, 4, 6, 8], dtype=int32)>
```

```
tf.range(1,10,delta=2)
```

```
<tf.Tensor: shape=(5,), dtype=int32, numpy=array([1, 3, 5, 7, 9], dtype=int32)>
```

```
out=tf.random.uniform([4,10])
y=tf.constant([2,3,2,0])
y=tf.one_hot(y,depth=10)
loss=tf.keras.losses.mse(y,out)
loss=tf.reduce_mean(loss)
print(loss)
```

```
tf.Tensor(0.38115382, shape=(), dtype=float32)
```

```
o=tf.random.normal([4,2])
b=tf.zeros([2])
o=o+b
o
```

```
<tf.Tensor: shape=(4, 2), dtype=float32, numpy=
array([[ 0.32005203, -0.63538855],
       [-0.14683501, -0.966589  ],
       [ 1.0750834 , -2.032762  ],
       [-0.43547243, -1.9074225 ]], dtype=float32)>
```

```
from tensorflow.keras import layers
fc= layers.Dense(3)
fc.build(input_shape=(2,4))
fc.bias
```

```
<Variable path=dense/bias, shape=(3,), dtype=float32, value=[0. 0. 0.]>
```

```
x=tf.random.normal([2,4])
x
```

```
<tf.Tensor: shape=(2, 4), dtype=float32, numpy=
array([[-0.02900418,  0.6099714 , -1.5118811 , -0.08705422],
       [ 0.6290019 ,  0.7165651 , -0.48481372, -0.22970662]],
      dtype=float32)>
```

```
w=tf.ones([4,3])
b=tf.zeros([3])
o=x@w+b
o
```

```
<tf.Tensor: shape=(2, 3), dtype=float32, numpy=
array([[-1.0179682 , -1.0179682 , -1.0179682 ],
       [ 0.63104665,  0.63104665,  0.63104665]], dtype=float32)>
```

```
x=tf.random.normal([4,32,32,3])
x[0]
```

```
<tf.Tensor: shape=(32, 32, 3), dtype=float32, numpy=
array([[[ 0.8990563 , -0.8247917 ,  0.03674725],
        [-0.31099567,  0.50334716, -1.0773557 ],
        [-0.10894333,  0.5021071 , -0.12461568],
        ...,
        [-2.2282634 , -0.6551524 ,  0.960816  ],
        [ 1.6623789 ,  0.6023733 , -0.14287531],
        [ 0.51655793,  0.406914  ,  0.9314251 ]],

       [[ 1.6559993 ,  0.01356253,  0.4688979 ],
        [-0.56629044, -2.8021178 ,  0.71825   ],
```

```
 [-0.28684804, -0.15508671,  0.7951721 ],
 ...,
 [-1.3489555 , -0.13011225, -0.94987977],
 [ 0.8226971 ,  0.8902097 , -0.6111554 ],
 [-0.8623008 , -0.13536602,  1.5664788 ]],

[[ 0.01775992,  0.5500005 , -1.0406972 ],
 [-1.4126251 ,  1.3615392 ,  0.4377927 ],
 [ 0.45624495, -0.5785752 , -0.64903104],
 ...,
 [ 0.05876148,  0.07207233,  1.5670661 ],
 [ 0.24493064, -0.43719876,  1.6845024 ],
 [-0.0689787 ,  0.8823174 ,  1.0952557 ]],

...,

[[ 1.6885024 , -0.58800095,  0.05116628],
 [-2.3214154 ,  0.10137639,  0.01648552],
 [-0.58316034,  0.22185223,  1.6835934 ],
 ...,
 [-0.12686643,  0.04454844,  0.757897  ],
 [-0.92277926,  0.345654  , -0.1985716 ],
 [ 2.4187155 , -1.4997629 , -0.22331586]],

[[-0.34000304,  1.2188934 , -1.4248092 ],
 [-2.9490306 ,  0.6794512 , -1.3332825 ],
 [ 0.066569  ,  0.146378  ,  0.66645586],
 ...,
 [-0.58605057, -0.5725485 , -1.6801622 ],
 [-0.58956367,  0.41098946,  1.9651045 ],
 [-0.34518084,  0.32613873, -1.6988238 ]],

[[-0.94879645,  1.8744773 ,  0.65635103],
 [-1.7179883 ,  0.16811264,  0.9944725 ],
 [ 1.7038059 ,  1.6702785 , -2.2512145 ],
 ...,
 [-1.2343988 , -0.36488134, -0.927459  ],
 [-0.30551013, -1.0326229 , -0.35943624],
 [-0.6274894 , -0.79294294, -1.6068864 ]]], dtype=float32)>
```

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

```
<tf.Tensor: shape=(32, 3), dtype=float32, numpy=
array([[ 1.65599930e+00,  1.35625303e-02,  4.68897909e-01],
       [-5.66290438e-01, -2.80211782e+00,  7.18249977e-01],
       [-2.86848038e-01, -1.55086711e-01,  7.95172095e-01],
       [-7.10083544e-01,  1.54462361e+00, -1.39539087e+00],
       [-4.43654060e-02,  1.34817541e+00,  1.40054882e+00],
       [ 4.65628982e-01,  7.01586366e-01,  3.25020045e-01],
       [-1.58954844e-01,  5.22216439e-01, -2.85790831e-01],
       [-1.23926893e-01,  2.07273811e-01,  5.94534516e-01],
       [-6.21617794e-01,  6.28185689e-01,  4.08773363e-01],
       [-1.14778966e-01, -1.42328039e-01,  2.53339350e-01],
       [-5.13043404e-01,  9.65908706e-01, -2.15824223e+00],
       [ 8.54894698e-01,  8.63893211e-01,  8.36776257e-01],
       [-1.72498858e+00, -1.03178096e+00, -2.14974260e+00],
       [ 4.09178346e-01,  8.73549402e-01, -6.21881425e-01],
       [-7.96108723e-01, -6.40694499e-01, -5.69701552e-01],
       [ 1.52367532e+00, -2.24548262e-02,  4.06905949e-01],
       [-4.29811507e-01,  9.77528691e-01,  6.57115936e-01],
       [-1.27561819e-02, -1.50837719e-01,  8.30350935e-01],
       [-7.13693440e-01,  9.60969090e-01, -4.22435641e-01],
       [-8.55616033e-02, -4.56031114e-01,  4.23917502e-01],
       [-1.19294751e+00, -1.58869356e-01,  7.91377544e-01],
       [-4.16934073e-01, -1.07398653e+00, -5.51711321e-01],
       [ 3.28073353e-01,  9.64485109e-01,  4.21063691e-01],
       [-2.90485710e-01,  6.92187622e-02, -7.24098310e-02],
       [-2.23003328e-03,  8.19521248e-01,  8.17143857e-01],
       [-3.21971327e-02, -2.05962211e-01, -2.46438801e-01],
```

```
[ 1.05879045e+00,  1.38781989e+00,  1.97745115e-01],
[ 1.07349062e+00, -6.79703534e-01, -2.53756851e-01],
[-4.27120715e-01, -1.84326482e+00, -2.12904468e-01],
[-1.34895551e+00, -1.30112246e-01, -9.49879766e-01],
[ 8.22697103e-01,  8.90209675e-01, -6.11155391e-01],
[-8.62300813e-01, -1.35366023e-01,  1.56647885e+00]], dtype=float32)>
```

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

```
<tf.Tensor: shape=(3,), dtype=float32, numpy=array([-0.28684804, -0.15508671,  0.7951721 ],
dtype=float32)>
```

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

```
<tf.Tensor: shape=(), dtype=float32, numpy=-0.5583078265190125>
```

```
x[1,9,2]
```

```
<tf.Tensor: shape=(3,), dtype=float32, numpy=array([-1.4485155, -1.0754294, -0.5795421],
dtype=float32)>
```

```
x[1:3]
```

```
  [-6.90675437e-01, -2.17964816e+00,  3.36026520e-01],
  [ 9.07947779e-01, -6.28533244e-01, -1.19047868e+00],
  ...,
  [ 8.84019375e-01, -1.06710184e+00,  7.49200046e-01],
  [ 1.02993488e+00, -5.74553251e-01, -2.34003499e-01],
  [ 1.73054922e+00, -1.30930805e+00,  8.99983644e-01]],

 [[ 1.51931775e+00,  5.71227551e-01, -8.11442554e-01],
  [ 1.22813754e-01,  1.13894629e+00, -1.20405006e+00],
  [ 1.47964358e+00,  2.21263456e+00,  1.48384440e+00],
  ...,
  [ 4.60944057e-01, -5.59206665e-01, -3.75813544e-01],
  [ 1.23541772e+00, -1.03303857e-01,  1.11681890e+00],
  [-9.23112988e-01,  2.11454257e-01,  4.97951567e-01]]]],
dtype=float32)>
```

```
x[0,::]
```

```
<tf.Tensor: shape=(32, 32, 3), dtype=float32, numpy=
array([[[ 0.8990563 , -0.8247917 ,  0.03674725],
        [-0.31099567,  0.50334716, -1.0773557 ],
        [-0.10894333,  0.5021071 , -0.12461568],
        ...,
        [-2.2282634 , -0.6551524 ,  0.960816  ],
        [ 1.6623789 ,  0.6023733 , -0.14287531],
        [ 0.51655793,  0.406914  ,  0.9314251 ]],

       [[ 1.6559993 ,  0.01356253,  0.4688979 ],
        [-0.56629044, -2.8021178 ,  0.71825   ],
        [-0.28684804, -0.15508671,  0.7951721 ],
        ...,
        [-1.3489555 , -0.13011225, -0.94987977],
        [ 0.8226971 ,  0.8902097 , -0.6111554 ],
        [-0.8623008 , -0.13536602,  1.5664788 ]],

       [[ 0.01775992,  0.5500005 , -1.0406972 ],
        [-1.4126251 ,  1.3615392 ,  0.4377927 ],
        [ 0.45624495, -0.5785752 , -0.64903104],
        ...,
        [ 0.05876148,  0.07207233,  1.5670661 ],
        [ 0.24493064, -0.43719876,  1.6845024 ],
        [-0.0689787 ,  0.8823174 ,  1.0952557 ]],

       ...,

       [[ 1.6885024 , -0.58800095,  0.05116628],
        [-2.3214154 ,  0.10137639,  0.01648552],
        [-0.58316034,  0.22185223,  1.6835934 ],
        ...,
        [-0.12686643,  0.04454844,  0.757897  ],
        [-0.92277926,  0.345654  , -0.1985716 ],
        [ 2.4187155 , -1.4997629 , -0.22331586]],

       [[-0.34000304,  1.2188934 , -1.4248092 ],
        [-2.9490306 ,  0.6794512 , -1.3332825 ],
        [ 0.066569  ,  0.146378  ,  0.66645586],
        ...,
        [-0.58605057, -0.5725485 , -1.6801622 ],
        [-0.58956367,  0.41098946,  1.9651045 ],
        [-0.34518084,  0.32613873, -1.6988238 ]],

       [[-0.94879645,  1.8744773 ,  0.65635103],
        [-1.7179883 ,  0.16811264,  0.9944725 ],
        [ 1.7038059 ,  1.6702785 , -2.2512145 ],
        ...,
        [-1.2343988 , -0.36488134, -0.927459  ],
        [-0.30551013, -1.0326229 , -0.35943624],
        [-0.6274894 , -0.79294294, -1.6068864 ]]], dtype=float32)>
```

```
x[:,0:28:2,0:28:2,:]
```

```
          [[-0.5441349 , -2.0803385 , -1.1904532 ],
           [-0.8275847 , -0.5686187 , -1.0989127 ],
           [ 0.32576784, -1.4470346 , -1.8477999 ],
           ...,
           [-0.89810526, -0.9151176 , -0.50407   ],
           [-0.7376212 ,  0.7493374 ,  1.1837064 ],
           [-0.27368304, -0.33658606, -0.31725723]]],


         [[[ 1.428389  , -1.0106564 ,  0.4321332 ],
           [-1.3694679 ,  0.11740626, -1.786343  ],
           [ 0.96806294,  1.6510272 ,  0.9772494 ],
           ...,
           [ 0.71096736,  0.4505352 ,  1.2668408 ],
           [ 1.0101787 ,  1.8411117 , -1.3023686 ],
           [-0.03188907,  2.0103223 ,  1.3840399 ]],

          [[-0.4190765 , -0.6157906 , -0.18749954],
           [ 0.42605427,  1.3328013 , -0.44784054],
           [-0.56957716, -0.5996533 ,  0.18873025],
           ...,
           [-0.09750228, -1.0407015 , -2.117747  ],
           [ 1.5277071 ,  0.47632205,  1.4883982 ],
           [-1.5725746 , -0.6130953 , -1.5901762 ]],

          [[-0.29703975,  0.9620423 , -0.57836807],
           [-0.7086887 , -1.5652897 ,  0.09523854],
           [-0.37639946, -0.8685678 , -1.1856197 ],
           ...,
           [ 0.3409369 ,  0.93588924, -1.1446068 ],
           [ 1.3130832 ,  0.77980083,  0.02138465],
           [ 0.6843247 , -0.49246395, -0.46489057]],

          ...,

          [[ 1.4107744 , -0.5198269 , -0.29459816],
           [-0.5386635 ,  1.8076097 ,  0.12410137],
           [-2.416304  , -0.43407   ,  0.11742891],
           ...,
           [ 0.5676454 , -0.11426787,  0.88615364],
           [ 0.26025197,  0.06547154,  0.68838406],
           [ 0.34458172, -1.4319303 , -2.3481722 ]],

          [[ 1.8621323 ,  0.8288549 , -0.8258257 ],
           [ 0.67562175, -0.70260143,  0.60823596],
           [ 0.05760941, -1.2709367 , -0.73971283],
           ...,
           [ 0.7344534 ,  0.12776388,  0.9173569 ],
           [ 0.5780885 ,  0.56489104, -0.6426949 ],
           [-0.8241067 ,  1.0518272 , -0.5502524 ]],

          [[ 0.2318302 ,  1.3650792 ,  0.2998244 ],
           [ 0.3005176 ,  0.7207237 , -2.0310626 ],
           [ 0.39600435, -1.7756342 ,  0.92595273],
           ...,
           [ 0.19814928,  0.27587172,  0.87592936],
           [ 0.81773204, -0.0880958 , -0.2218313 ],
           [-0.72682846, -1.3551997 ,  0.07219852]]]], dtype=float32)>
```

```
x=tf.range(9)
x
```

```
<tf.Tensor: shape=(9,), dtype=int32, numpy=array([0, 1, 2, 3, 4, 5, 6, 7, 8], dtype=int32)>
```

```
x[8:0:-1]
```

```
<tf.Tensor: shape=(8,), dtype=int32, numpy=array([8, 7, 6, 5, 4, 3, 2, 1], dtype=int32)>
```

```
x[::-1]
```

```
<tf.Tensor: shape=(9,), dtype=int32, numpy=array([8, 7, 6, 5, 4, 3, 2, 1, 0], dtype=int32)>
```

```
x[::-2]
```

```
<tf.Tensor: shape=(5,), dtype=int32, numpy=array([8, 6, 4, 2, 0], dtype=int32)>
```

Start coding or generate with AI.