How can we make tensor.

```
import tensorflow as tf
print(tf.version)
```

```
<module 'tensorflow._api.v2.version' from '/usr/local/lib/python3.10/dist-packages/tensorflow/_api/v2/version/__init__.py'>
```

```
tf.__version__
```

```
'2.17.0'
```

```
tf.test.is_gpu_available()
```

```
WARNING:tensorflow:From <ipython-input-4-17bb7203622b>:1: is_gpu_available (from tensorflow.python.framework.test_util) is deprecate
Instructions for updating:
Use `tf.config.list_physical_devices('GPU')` instead.
False
```

```
x=tf.constant(10)
x
```

```
<tf.Tensor: shape=(), dtype=int32, numpy=10>
```

```
y=tf.constant(10.5)
y
```

```
<tf.Tensor: shape=(), dtype=float32, numpy=10.5>
```

```
z=tf.constant("India")
z
```

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

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

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

```
import numpy as np
x=np.array([[1,2,3],[4,5,6]])
tf.constant(x)
```

```
<tf.Tensor: shape=(2, 3), dtype=int64, numpy=
array([[1, 2, 3],
       [4, 5, 6]])>
```

```
t_2d=tf.constant([[1,2],[3,4]])
t_2d
```

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

```
t_1d=tf.constant([1,2,3,4])
t_1d
```

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

```
t_2d_1=tf.constant([1,2,3,4],shape=(2,2),dtype="int32")
t_2d_1
```

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

```
t_3d_1=tf.constant([[[1,2],[3,4],[5,6]]],dtype="float")
t_3d_1
```

```
<tf.Tensor: shape=(1, 3, 2), dtype=float32, numpy=
array([[[1., 2.],
        [3., 4.],
        [5., 6.]]], dtype=float32)>
```

```
type(t_3d_1)
```

```
tensorflow.python.framework.ops.EagerTensor
```

```
t_3d_1.shape
```

```
TensorShape([1, 3, 2])
```

Tensorflow Variable

```
# Variable constructor requires an initial value for the variables which can be a tensor of any type & shape.
# This initial value defines the type & shape of the variable
# After construction the type & shape of the variables are fixed.
```

```
import tensorflow as tf
```

```
tf.Variable(1)
```

```
<tf.Variable 'Variable:0' shape=() dtype=int32, numpy=1>
```

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

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

```
x.name
```

```
'Variable:0'
```

```
x.shape
```

```
TensorShape([4])
```

```
x.dtype
```

```
tf.int32
```

```
y=tf.Variable([1.0,2,3.3,4])
y
```

```
<tf.Variable 'Variable:0' shape=(4,) dtype=float32, numpy=array([1. , 2. , 3.3, 4. ], dtype=float32)>
```

```
z=tf.Variable(True,False)
z
```

```
<tf.Variable 'Variable:0' shape=() dtype=bool, numpy=True>
```

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

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

```
tf.Variable(t_con)
```

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

Create TensorFlow variable with Different Shape

```
t_2d=tf.Variable([[2,3],[4,5]])
t_2d
```

```
<tf.Variable 'Variable:0' shape=(2, 2) dtype=int32, numpy=
array([[2, 3],
       [4, 5]], dtype=int32)>
```

```
x
```

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

```
tf.reshape(x,[2,2])
```

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

```
# Get index of highest value
```

```
t_2d
```

```
<tf.Variable 'Variable:0' shape=(2, 2) dtype=int32, numpy=
array([[2, 3],
       [4, 5]], dtype=int32)>
```

```
tf.argmax(t_2d)
```

```
<tf.Tensor: shape=(2,), dtype=int64, numpy=array([1, 1])>
```

```
# Viewed / convert as a tensor
```

```
x
```

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

```
tf.convert_to_tensor(x)
```

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

```
tf.convert_to_tensor(t_2d)
```

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

```
# Change/Assign New value to Tensor
```

```
x
```

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

```
x.assign([4,6,2,8])
```

```
<tf.Variable 'UnreadVariable' shape=(4,) dtype=int32, numpy=array([4, 6, 2, 8], dtype=int32)>
```

```
x
```

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

```
x.assign([4,6,2,9])
```

```
<tf.Variable 'UnreadVariable' shape=(4,) dtype=int32, numpy=array([4, 6, 2, 9], dtype=int32)>
```

```
x
```

```
<tf.Variable 'Variable:0' shape=(4,) dtype=int32, numpy=array([4, 6, 2, 9], dtype=int32)>
```

```
# Assign Variable with another Memory
```

```
x.assign_add([4,6,2,8])
```

```
<tf.Variable 'UnreadVariable' shape=(4,) dtype=int32, numpy=array([ 8, 12,  4, 17], dtype=int32)>
```

```
x
```

```
<tf.Variable 'Variable:0' shape=(4,) dtype=int32, numpy=array([ 8, 12,  4, 17], dtype=int32)>
```

TensorFlow Math Modules

```
import tensorflow as tf
import numpy as np
```

```
a=10
b=20
a+b
```

```
30
```

```
tf.add(a,b)
```

```
<tf.Tensor: shape=(), dtype=int32, numpy=30>
```

```
# Addition of two List
l1=[1,2,3,4]
l2=[5,6,7,8]
tf.add(l1,l2)
```

```
<tf.Tensor: shape=(4,), dtype=int32, numpy=array([ 6,  8, 10, 12], dtype=int32)>
```

```
# Addition of two tuple
t1=(2,3,4,5)
t2=(6,7,8,9)
tf.add(t1,t2)
```

```
<tf.Tensor: shape=(4,), dtype=int32, numpy=array([ 8, 10, 12, 14], dtype=int32)>
```

```
l1=[1,2,3,4]
a=10
tf.add(l1,a)
```

```
<tf.Tensor: shape=(4,), dtype=int32, numpy=array([11, 12, 13, 14], dtype=int32)>
```

```
# Given a tensor x and y,this operation x^y computers for corresponding elements in x and y.
x=tf.constant([[2,2],[3,3]])
y=tf.constant([[8,16],[2,3]])
tf.pow(x,y)
```

```
<tf.Tensor: shape=(2, 2), dtype=int32, numpy=
array([[  256, 65536],
       [    9,    27]], dtype=int32)>
```

```
# tf.math.abs
# real number
x=tf.constant([-2.25,3.25])
tf.abs(x)
```

```
<tf.Tensor: shape=(2,), dtype=float32, numpy=array([2.25, 3.25], dtype=float32)>
```

```
# complex number
x=tf.constant([[-2.25+4.75j],[-3.25+5.75j]])
tf.abs(x)
```

```
<tf.Tensor: shape=(2, 1), dtype=float64, numpy=
array([[5.25594901],
       [6.60492241]])>
```

```
# tf.math.subtract
x=[1,2,3,4,5]
y=1
tf.subtract(x,y)
```

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

```
x=[1,2,3,4]
y=1
tf.subtract(y,x)
```

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

```
# tf.math.multiply
x=[1,2,3,4]
y=2
tf.multiply(x,y)
```

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

```
# tf.math.multiply
x=tf.constant(([1,2,3,4]))
tf.math.multiply(x,2)
```

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

```
tf.multiply(x,x)
```

```
<tf.Tensor: shape=(4,), dtype=int32, numpy=array([ 1,  4,  9, 16], dtype=int32)>
```

```
tf.math.multiply(7,6)
```

```
<tf.Tensor: shape=(), dtype=int32, numpy=42>
```

```
x=tf.ones([1,2])
y=tf.ones([2,1])
x*y
```

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

```
# tf.math.maximum
x=tf.constant([0.,0.,0.,0.])
y=tf.constant([-2.,0.,2.,5.])
tf.math.maximum(x,y)
```

```
<tf.Tensor: shape=(4,), dtype=float32, numpy=array([0., 0., 2., 5.], dtype=float32)>
```

```
x=tf.constant([-5,0,0,0])
y=tf.constant([-3])
tf.math.maximum(x,y)
```

```
<tf.Tensor: shape=(4,), dtype=int32, numpy=array([-3,  0,  0,  0], dtype=int32)>
```

```
# tf.math.minimum
x=tf.constant([0,0,0,0])
y=tf.constant([-5,-2,0,3])
tf.math.minimum(x,y)
```

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

```
x=tf.constant([-5,0,0,0])
y=tf.constant([-3])
tf.math.maximum(x,y)
```

```
<tf.Tensor: shape=(4,), dtype=int32, numpy=array([-3,  0,  0,  0], dtype=int32)>
```

```
# tf.math.log
x=tf.constant([0,0.5,1,5])
tf.math.log(x)
```

```
<tf.Tensor: shape=(4,), dtype=float32, numpy=array([      -inf, -0.6931472,  0.       ,  1.609438 ], dtype=float32)>
```

```
# tf.math.divide
x=tf.constant([16,12,11])
y=tf.constant([4,6,2])
tf.divide(x,y)
```

```
<tf.Tensor: shape=(3,), dtype=float64, numpy=array([4. , 2. , 5.5])>
```

```
# tf.math.pow
x=tf.constant([[2,2],[3,3]])
y=tf.constant([[8,16],[2,3]])
tf.pow(x,y)
```

```
<tf.Tensor: shape=(2, 2), dtype=int32, numpy=
array([[  256, 65536],
       [    9,    27]], dtype=int32)>
```

```
# tf.math.sqrt
x=tf.constant([[4.0],[16.0]])
tf.sqrt(x)
```

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

```
y=tf.constant([[-4.0],[16.0]])
tf.sqrt(y)
```

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

```
tf.ones(1)
```

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

```
tf.ones((1)),tf.float32
```

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

```
ones_2d=tf.ones((2,2)),tf.float32
ones_2d
```

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

```
ones_2d=tf.ones((2,2,4),tf.float32)
ones_2d
```

```
<tf.Tensor: shape=(2, 2, 4), dtype=float32, numpy=
array([[[1., 1., 1., 1.],
        [1., 1., 1., 1.]],

       [[1., 1., 1., 1.],
        [1., 1., 1., 1.]]], dtype=float32)>
```

```
tf.zeros(1)
```

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

```
zeros_2d=tf.zeros((2,2),dtype=tf.int32)
zeros_2d
```

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