

```

import torch
import numpy as np

data = [
    [1,2],
    [3,4]
]
x_data = torch.tensor(data)
print(x_data)
print(type(x_data))

    tensor([[1, 2],
           [3, 4]])
    <class 'torch.Tensor'>

np_array = np.array(data)
x_np = torch.from_numpy(np_array)
print(x_np)
print(type(x_np))

    tensor([[1, 2],
           [3, 4]])
    <class 'torch.Tensor'>

x_ones = torch.ones_like(x_data)
print("One Tensor: \n",x_ones)

    One Tensor:
    tensor([[1, 1],
           [1, 1]])

x_rand = torch.rand_like(x_data, dtype=torch.float)
print(x_rand)

    tensor([[0.6020, 0.9099],
           [0.4514, 0.8904]])

shape = (2,3)
random_tensor = torch.rand(shape)
print(random_tensor)
print(type(random_tensor))

    tensor([[0.1073, 0.6264, 0.7049],
           [0.2508, 0.5489, 0.8232]])
    <class 'torch.Tensor'>

ones_tensor = torch.ones(shape)
print(ones_tensor)
print(type(ones_tensor))

    tensor([[1., 1., 1.],
           [1., 1., 1.]])
    <class 'torch.Tensor'>

zeros_tensor = torch.zeros(shape)
print(zeros_tensor)
print(type(zeros_tensor))

    tensor([[0., 0., 0.],
           [0., 0., 0.]])
    <class 'torch.Tensor'>

tensor = torch.rand(3,4)
print(tensor)
tensor.shape
tensor.dtype
tensor.device

    tensor([[0.2296, 0.9300, 0.1502, 0.1718],
           [0.6704, 0.5065, 0.6198, 0.1189],
           [0.3555, 0.8748, 0.7588, 0.5139]])
    device(type='cpu')

if torch.cuda.is_available():
    tensor = tensor.to('cuda')
    print("Device tensor is stored in ", tensor.device)

```

```
# Indexing, Slicing
tensor = torch.ones(3,3)

print(tensor)

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

tensor1 = torch.zeros(3,3)
print(tensor1)

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

tensor2 = torch.cat([tensor,tensor1])
print(tensor2)

      tensor([[1., 1., 1.],
              [1., 1., 1.],
              [1., 1., 1.],
              [0., 0., 0.],
              [0., 0., 0.],
              [0., 0., 0.]])

# Multiply Operation
tensor.mul(tensor1)
tensor * tensor1
tensor.T

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

# inplace - change the original tensor
tensor.add_(5)
print(tensor)

      tensor([[6., 6., 6.],
              [6., 6., 6.],
              [6., 6., 6.]])

t = torch.ones(5)
print(t)

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

n = t.numpy()
print(n)
print(type(n))

      [1. 1. 1. 1. 1.]
      <class 'numpy.ndarray'>
```