```
import torch
import numpy as np
```

```
torch.__version__
```

```
'2.5.1+cu124'
```

```
arr = np.array([1,2,3,4,5])
print(arr)
print(arr.dtype)
print(type(arr))
```

```
[1 2 3 4 5]
int64
<class 'numpy.ndarray'>
```

```
x = torch.from_numpy(arr)
# Equivalent to x = torch.as_tensor(arr)

print(x)
```

```
tensor([1, 2, 3, 4, 5])
```

```
print(x.dtype)
```

```
torch.int64
```

```
print(type(x))
print(x.type())
```

```
<class 'torch.Tensor'>
torch.LongTensor
```

```
arr2 = np.arange(0.,12.).reshape(4,3)
print(arr2)
```

```
[[ 0.  1.  2.]
 [ 3.  4.  5.]
 [ 6.  7.  8.]
 [ 9. 10. 11.]]
```

```
x2 = torch.from_numpy(arr2)
print(x2)
print(x2.type())
```

```
tensor([[ 0.,  1.,  2.],
        [ 3.,  4.,  5.],
        [ 6.,  7.,  8.],
        [ 9., 10., 11.]], dtype=torch.float64)
torch.DoubleTensor
```

```
# Using torch.from_numpy()
arr = np.arange(0,5)
t = torch.from_numpy(arr)
print(t)
```

```
tensor([0, 1, 2, 3, 4])
```

```
arr[2]=77
print(t)
```

```
tensor([ 0,  1, 77,  3,  4])
```

```
data = np.array([1,2,3])
```

```
a = torch.Tensor(data)  # Equivalent to cc = torch.FloatTensor(data)
print(a, a.type())
```

```
tensor([1., 2., 3.]) torch.FloatTensor
```

```
b = torch.tensor(data)
print(b, b.type())
```

```
tensor([1, 2, 3]) torch.LongTensor
```

```
c = torch.tensor(data, dtype=torch.long)
print(c, c.type())
```

```
tensor([1, 2, 3]) torch.LongTensor
```

```
x = torch.empty(4, 3)
print(x)
```

```
tensor([[4.3800e-17, 4.4266e-41, 1.5382e-34],
        [0.0000e+00, 0.0000e+00, 0.0000e+00],
        [0.0000e+00, 0.0000e+00, 0.0000e+00],
        [0.0000e+00, 1.8788e+31, 1.7220e+22]])
```

```
x = torch.zeros(4, 3, dtype=torch.int64)
print(x)
```

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

```
x = torch.arange(0,18,2).reshape(3,3)
print(x)
```

```
tensor([[ 0,  2,  4],
        [ 6,  8, 10],
        [12, 14, 16]])
```

```
x = torch.linspace(0,18,12).reshape(3,4)
print(x)
```

```
tensor([[ 0.0000,  1.6364,  3.2727,  4.9091],
        [ 6.5455,  8.1818,  9.8182, 11.4545],
        [13.0909, 14.7273, 16.3636, 18.0000]])
```

```
x = torch.tensor([1, 2, 3, 4])
print(x)
print(x.dtype)
print(x.type())
```

```
tensor([1, 2, 3, 4])
torch.int64
torch.LongTensor
```

```
x = torch.tensor([5,6,7])
print(x.dtype)
print(x.type())
```

```
torch.int64
torch.LongTensor
```

```
x = torch.tensor([8,9,-3],dtype=torch.int)
print(x)
print(x.dtype)
print(x.type())
```

```
tensor([ 8,  9, -3], dtype=torch.int32)
torch.int32
torch.IntTensor
```

```
print('Old:', x.type())

x = x.type(torch.int64)

print('New:', x.type())
```

```
Old: torch.IntTensor
New: torch.LongTensor
```

```
x = torch.rand(4, 3)
print(x)
```

```
tensor([[0.8938, 0.6093, 0.4876],
        [0.6277, 0.5419, 0.8038],
        [0.2548, 0.8183, 0.4880],
        [0.2684, 0.0975, 0.0769]])
```

```
x = torch.randn(4, 3)
print(x)
```

```
tensor([[ 0.5333,  1.8884, -0.7853],
        [-0.3200, -1.0474, -0.3794],
        [ 2.3827, -0.2504,  0.3897],
        [ 0.3213,  0.1602, -1.0768]])
```

```
x = torch.randint(0, 5, (4, 3))
print(x)
```

```
tensor([[4, 2, 3],
        [0, 4, 4],
        [4, 4, 3],
        [2, 1, 3]])
```

```
x = torch.zeros(2,5)
print(x)
```

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

```
x2 = torch.randn_like(x)
print(x2)
```

```
tensor([[ 0.1490, -0.2397, -1.3528, -1.3118,  1.4034],
        [ 0.5629, -1.5053,  0.2032, -0.7624,  1.4253]])
```

```
x3 = torch.ones_like(x2)
print(x3)
```

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

```
torch.manual_seed(42)
x = torch.rand(2, 3)
print(x)
```

```
tensor([[0.8823, 0.9150, 0.3829],
        [0.9593, 0.3904, 0.6009]])
```

```
x.shape
```

```
torch.Size([2, 3])
```

```
x.size()
```

```
torch.Size([2, 3])
```

```
x.device
```

```
device(type='cpu')
```

```
x.layout
```

```
torch.strided
```

Start coding or generate with AI.