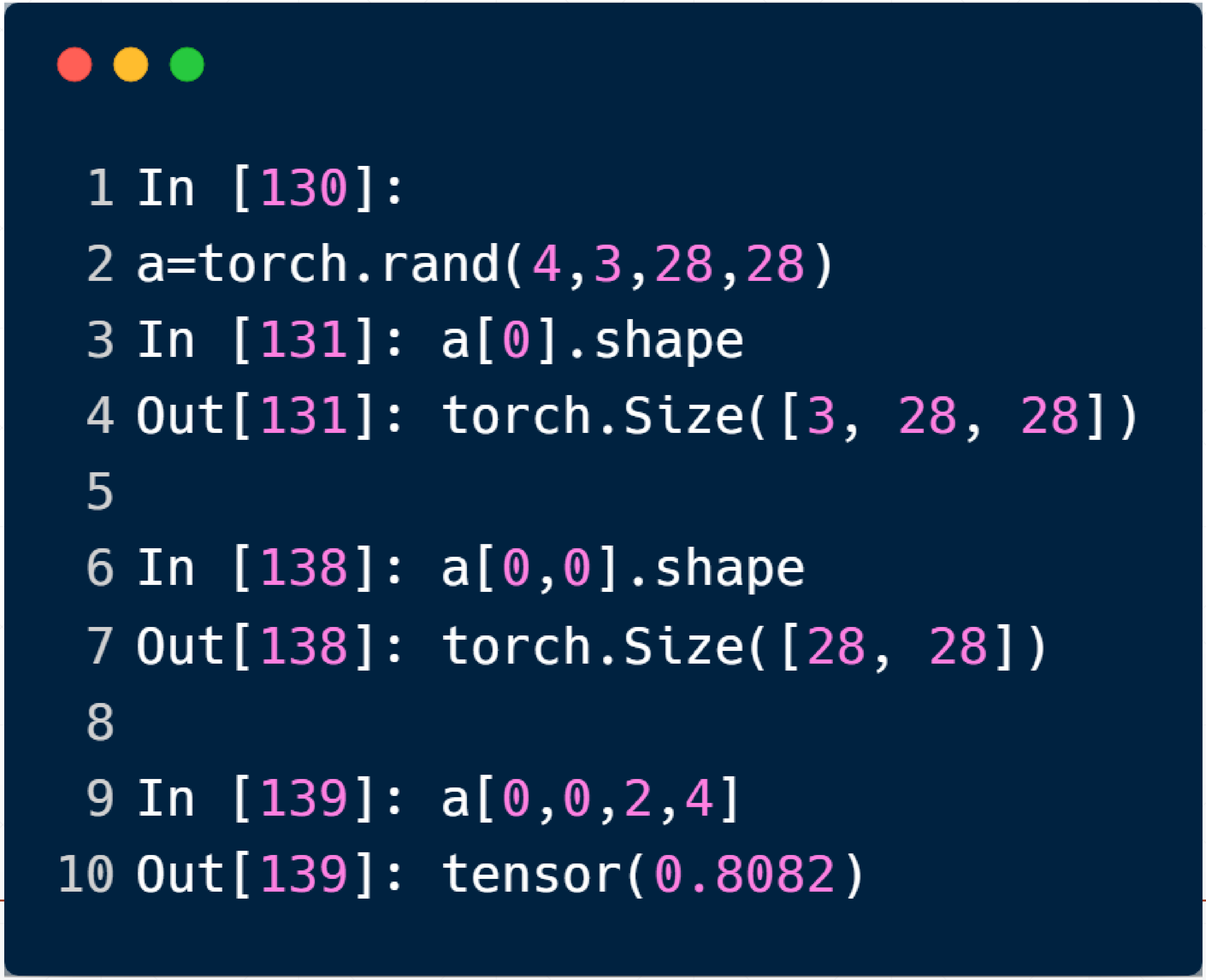
PyTorch

索引与切片

主讲人：龙良曲

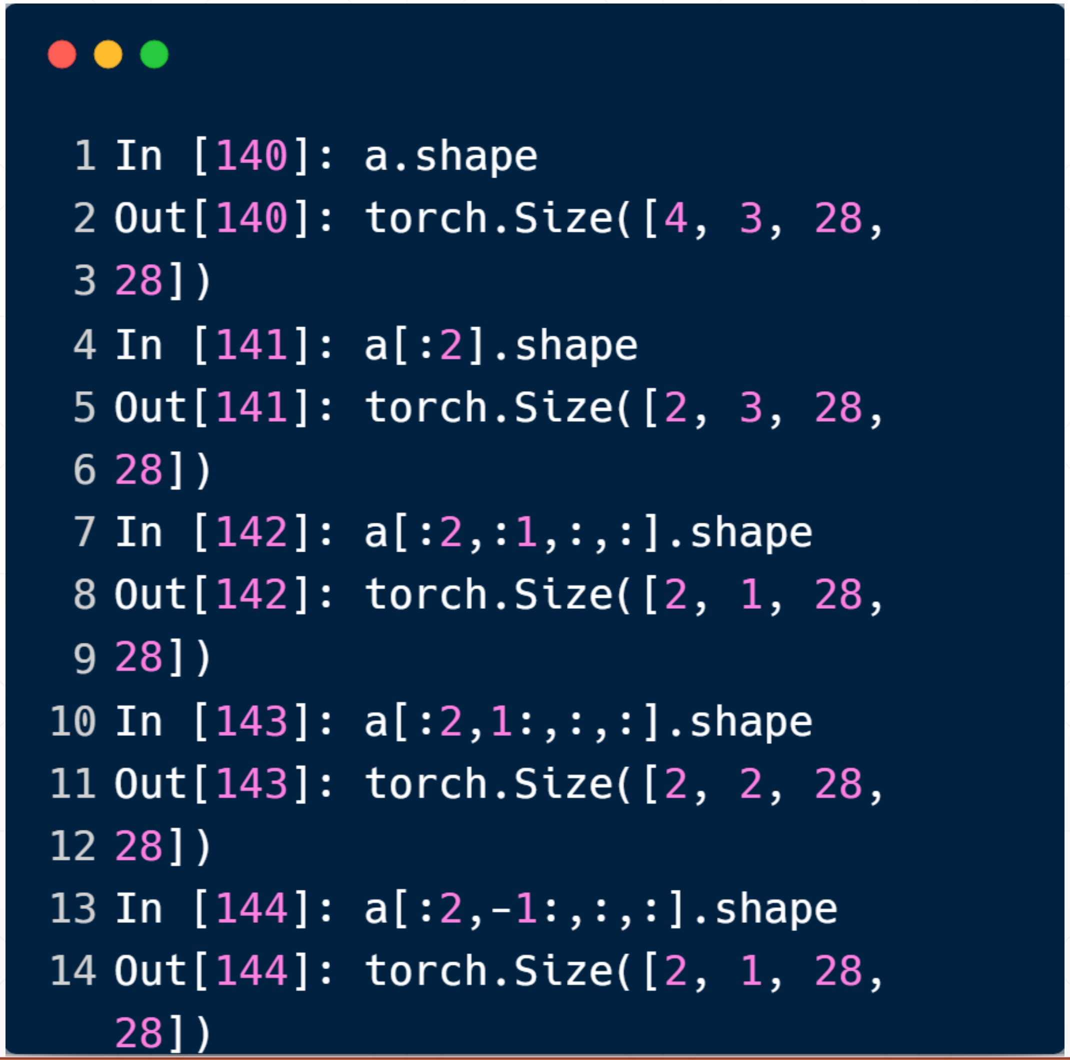
Indexing

- dim 0 first



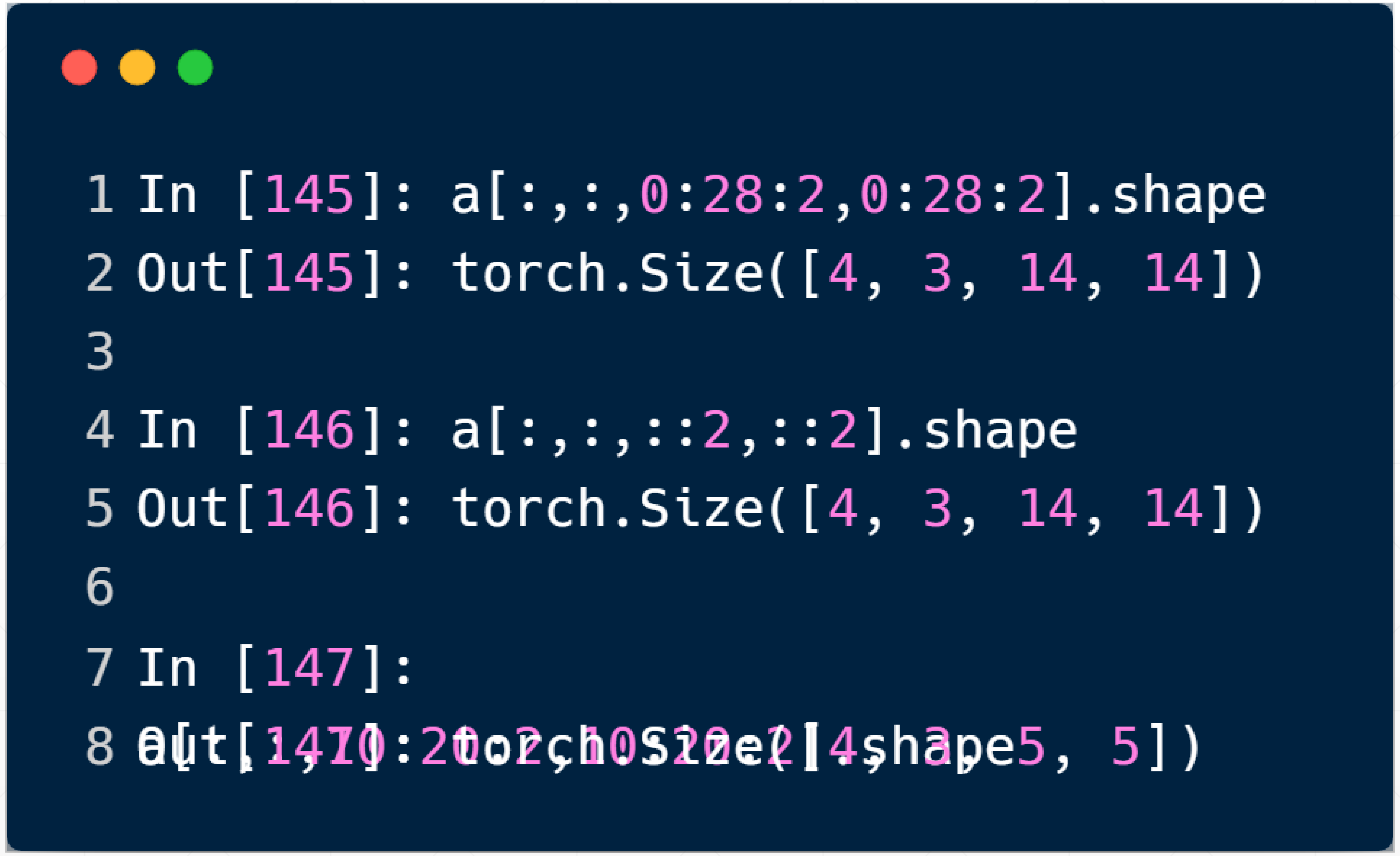
```
In [130]:
a=torch.rand(4,3,28,28)
In [131]: a[0].shape
Out[131]: torch.Size([3, 28, 28])

In [138]: a[0,0].shape
Out[138]: torch.Size([28, 28])

In [139]: a[0,0,2,4]
Out[139]: tensor(0.8082)
```

select first/last N



```
In [140]: a.shape
Out[140]: torch.Size([4, 3, 28,
28])
In [141]: a[:2].shape
Out[141]: torch.Size([2, 3, 28,
28])
In [142]: a[:2,:1,:,:].shape
Out[142]: torch.Size([2, 1, 28,
28])
In [143]: a[:2,1:,:,:].shape
Out[143]: torch.Size([2, 2, 28,
28])
In [144]: a[:2,-1:,:,:].shape
Out[144]: torch.Size([2, 1, 28,
28])
```

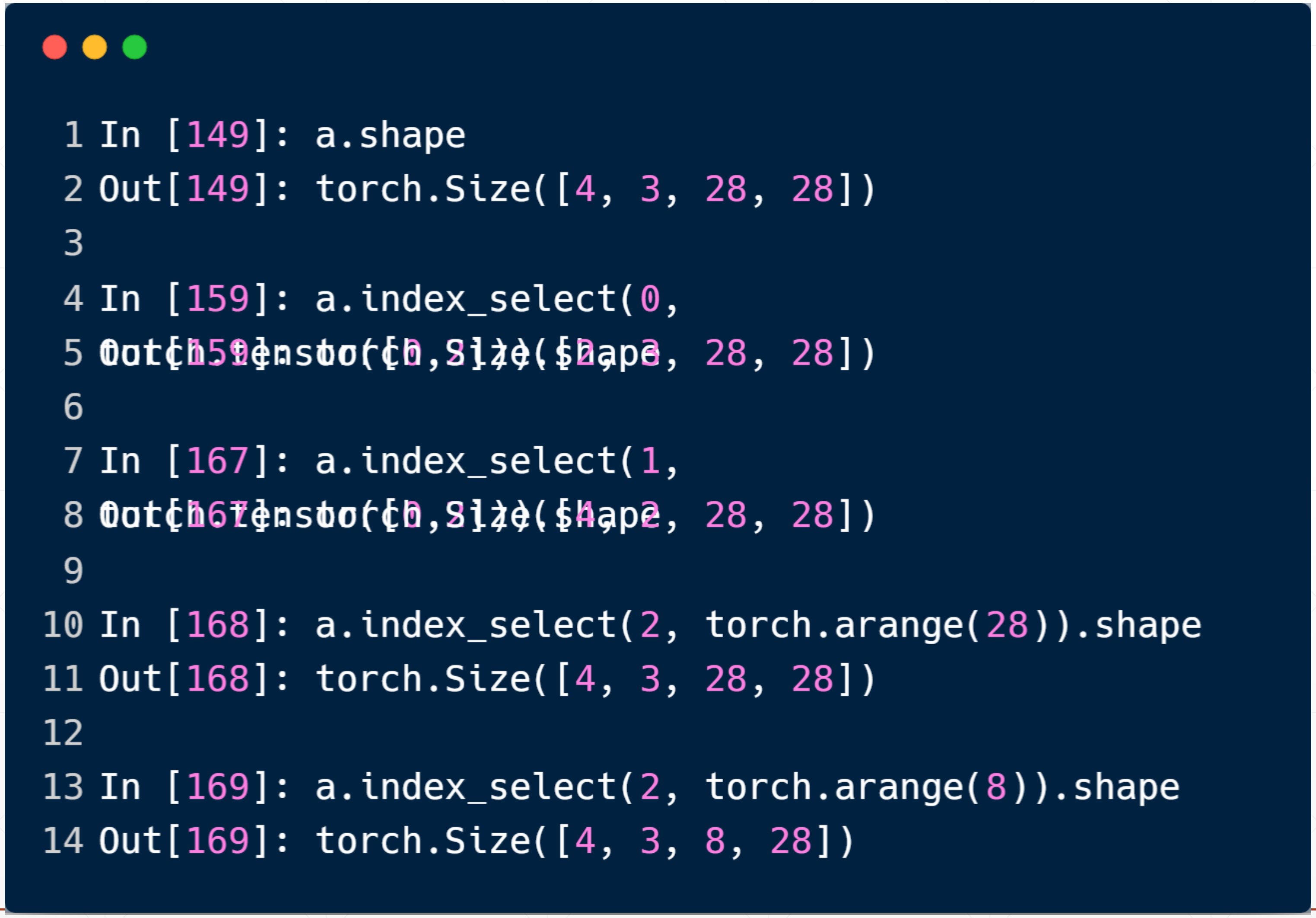
select by steps



```
In [145]: a[:,:,0:28:2,0:28:2].shape
Out[145]: torch.Size([4, 3, 14, 14])

In [146]: a[:,:,::2,::2].shape
Out[146]: torch.Size([4, 3, 14, 14])

In [147]:
a[:,:,10:20:2,10:20:2].shape
Out[147]: torch.Size([4, 3, 5, 5])
```

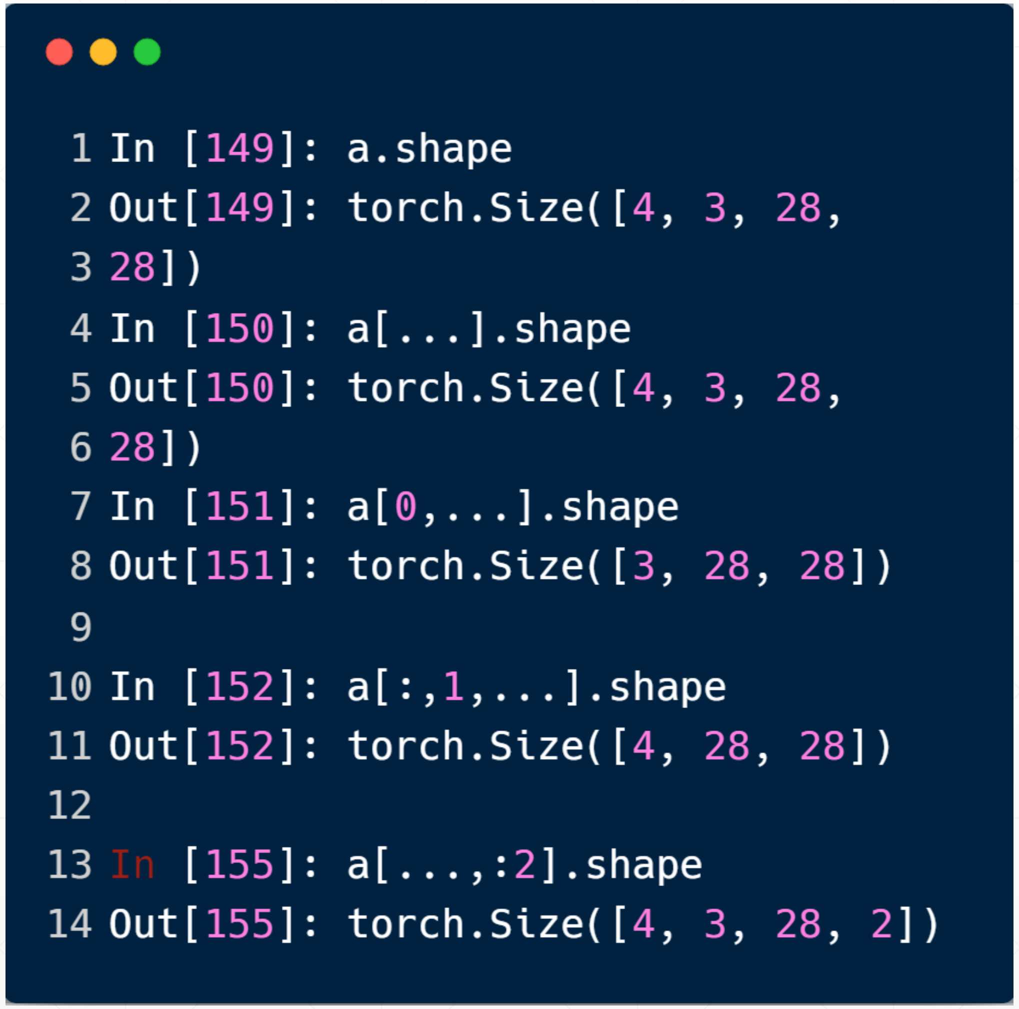
select by specific index

index_select(dim, index) → Tensor
index (LongTensor) – the 1-D tensor containing the indices to index



```
In [149]: a.shape
Out[149]: torch.Size([4, 3, 28, 28])

In [159]: a.index_select(0,
torch.tensor([0,2])).shape
Out[159]: torch.Size([2, 3, 28, 28])

In [167]: a.index_select(1,
torch.tensor([0,2])).shape
Out[167]: torch.Size([4, 2, 28, 28])

In [168]: a.index_select(2, torch.arange(28)).shape
Out[168]: torch.Size([4, 3, 28, 28])

In [169]: a.index_select(2, torch.arange(8)).shape
Out[169]: torch.Size([4, 3, 8, 28])
```

indices 需要是 tensor，取的就是准确的序列号



```
In [149]: a.shape
Out[149]: torch.Size([4, 3, 28,
28])
In [150]: a[...].shape
Out[150]: torch.Size([4, 3, 28,
28])
In [151]: a[0,...].shape
Out[151]: torch.Size([3, 28, 28])

In [152]: a[:,1,...].shape
Out[152]: torch.Size([4, 28, 28])

In [155]: a[...,:2].shape
Out[155]: torch.Size([4, 3, 28, 2])
```

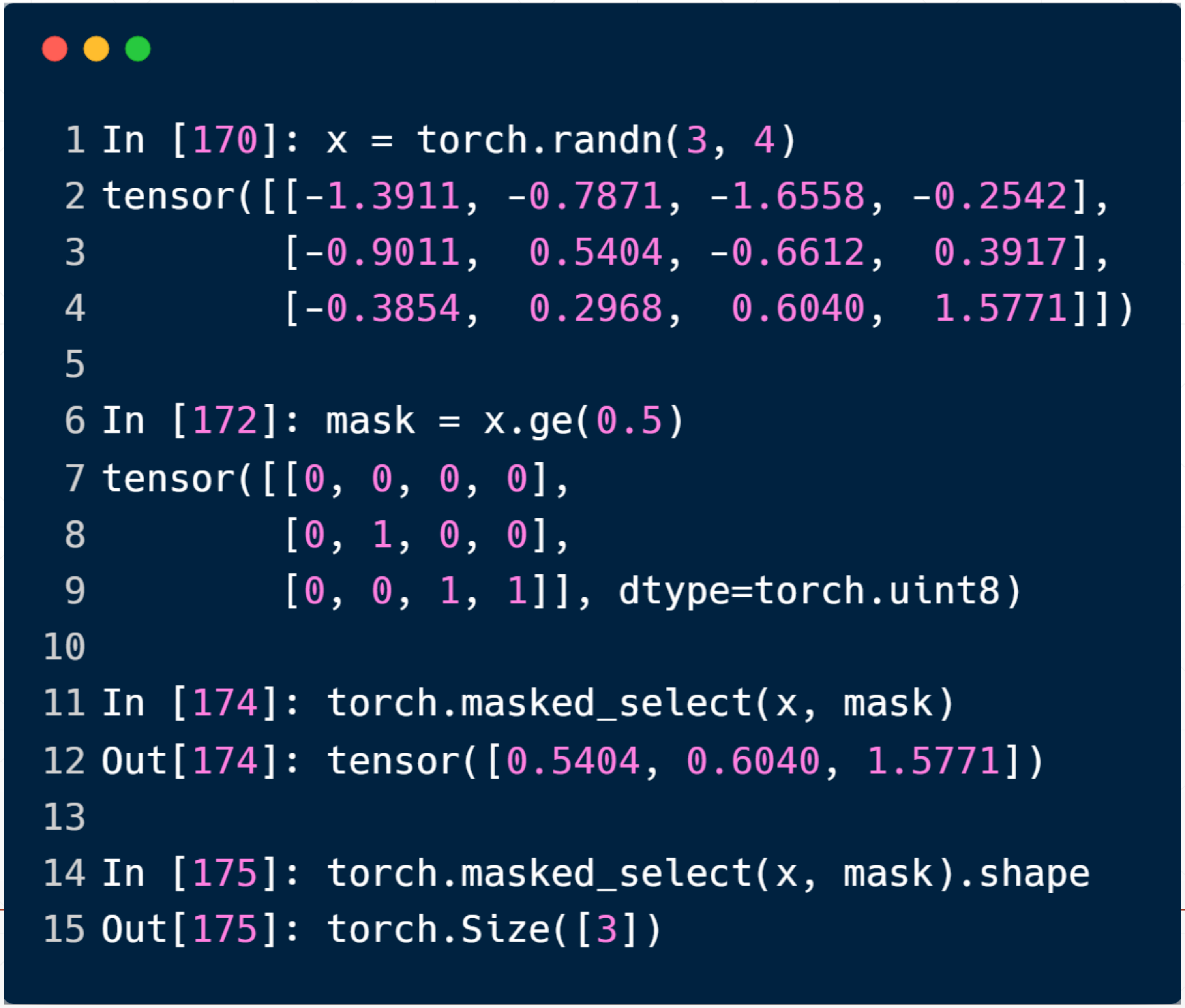
select by mask

- .masked_select()

torch.ge(input, other, out=None) → Tensor

Computes input≥other element-wise.

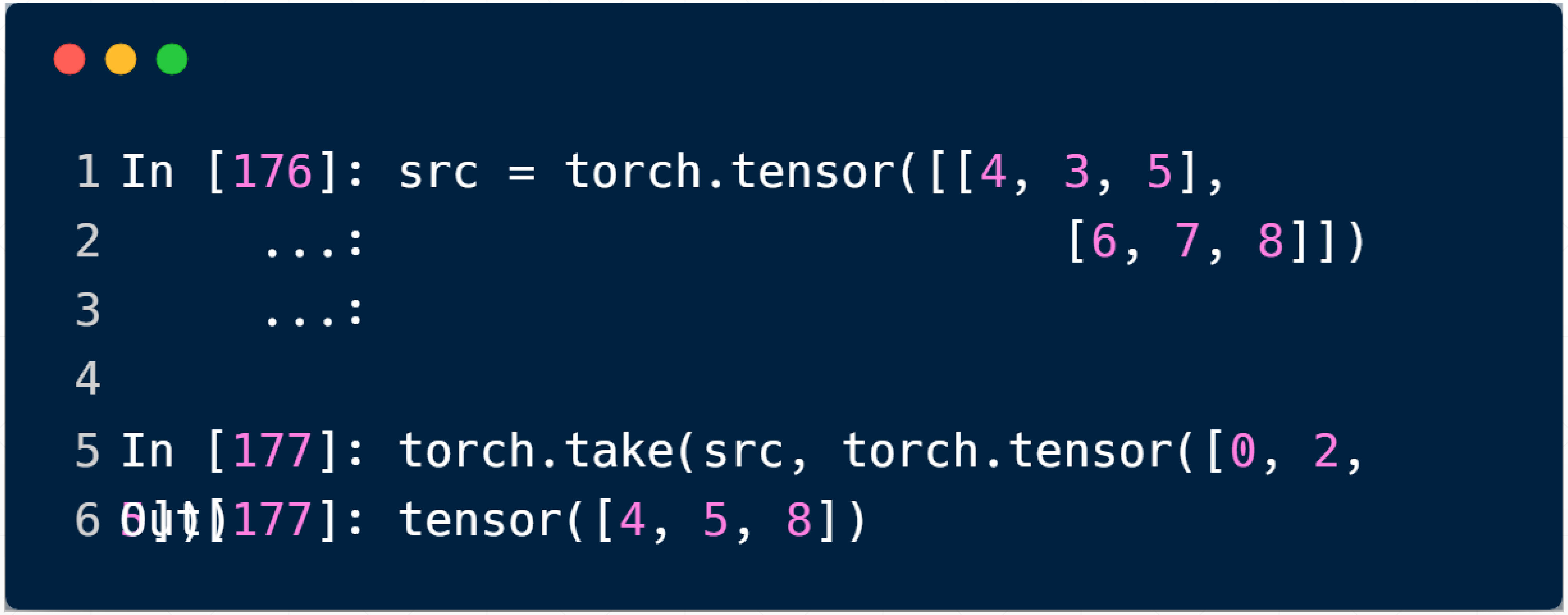
The second argument can be a number or a tensor whose shape is broadcastable with the first argument.



```
In [170]: x = torch.randn(3, 4)
tensor([[-1.3911, -0.7871, -1.6558, -0.2542],
        [-0.9011,  0.5404, -0.6612,  0.3917],
        [-0.3854,  0.2968,  0.6040,  1.5771]])

In [172]: mask = x.ge(0.5)
tensor([[0, 0, 0, 0],
        [0, 1, 0, 0],
        [0, 0, 1, 1]], dtype=torch.uint8)

In [174]: torch.masked_select(x, mask)
Out[174]: tensor([0.5404, 0.6040, 1.5771])

In [175]: torch.masked_select(x, mask).shape
Out[175]: torch.Size([3])
```

select by flatten index

The input tensor is treated as if it were viewed as a 1–D tensor. The result takes the same shape as the indices.
先变成1D按序号全部取出来，再按照take里面的index的形状reshape



```
In [176]: src = torch.tensor([[4, 3, 5],
     ...:                     [6, 7, 8]])
     ...:

In [177]: torch.take(src, torch.tensor([0, 2,
Out[177]: tensor([4, 5, 8])
```

下一课时

Tensor变换

Thank You.