In [1]:

```
import torch

x = [2, 9, 16, 23, 35]
```

In [2]:

```
# 1 检查x是否为tensor对象
torch.is_tensor(x)
```

Out[2]:

```
False
```

In [3]:

```
torch.is_storage(x)
```

Out[3]:

```
False
```

In [10]:

```
y = torch.rand(1, 2)
```

In [11]:

```
y
```

Out[11]:

```
tensor([[0.2609, 0.0116]])
```

In [8]:

```
torch.is_tensor(y)
```

Out[8]:

```
True
```

In [9]:

```
torch.is_storage(y)
```

Out[9]:

```
False
```

In [12]:

```
torch.numel(y) # 统计有多少个元素
```

Out[12]:

```
2
```

In [14]:

```
# 2 创建全0的tensor
z = torch.zeros(3, 3)
z
```

Out[14]:

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

In [15]:

```
torch.numel(z)
```

Out[15]:

```
9
```

In [16]:

```
# 3 创建对角线为1的tensor
torch.eye(3, 3)
```

Out[16]:

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

In [18]:

```
# 4 numpy 转 tensor
import numpy as np

x = np.array([3,4,5,6,7])
x
```

Out[18]:

```
array([3, 4, 5, 6, 7])
```

In [20]:

```
torch.from_numpy(x)
```

Out[20]:

```
tensor([3, 4, 5, 6, 7])
```

In [21]:

```
# 5 切分
torch.linspace(2, 10, steps = 5)
```

Out[21]:

```
tensor([ 2.,  4.,  6.,  8., 10.])
```

In [22]:

```
# 6 均匀分布：值在0和1之间
torch.rand(10)
```

Out[22]:

```
tensor([0.4174, 0.8224, 0.1174, 0.8415, 0.1034, 0.2841, 0.6364, 0.546
6, 0.0994,
        0.7993])
```

In [23]:

```
# 7 正态分布：均值为0，方差为1
torch.randn(10)
```

Out[23]:

```
tensor([ 0.6295,  0.7187, -0.1227,  1.1522, -1.4574, -0.7598,  0.1972,
0.9423,
         0.8848,  0.5814])
```

In [24]:

```
# 8 选择随机数
torch.randperm(10)
```

Out[24]:

```
tensor([3, 5, 2, 4, 8, 7, 0, 9, 6, 1])
```

In [26]:

```
# 9 生成一个区间的数
torch.arange(10, 30, 5)
```

Out[26]:

```
tensor([10, 15, 20, 25])
```

In [29]:

```
# 10 获取行或列的最小值和最大值的索引
x = torch.rand(3, 3)
x
```

Out[29]:

```
tensor([[0.9178, 0.4254, 0.9267],
        [0.1193, 0.7301, 0.3858],
        [0.1571, 0.9794, 0.3650]])
```

In [30]:

```
torch.argmin(x, dim=0)
```

Out[30]:

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

In [31]:

```
torch.argmax(x, dim=0)
```

Out[31]:

```
tensor([0, 2, 0])
```

In [33]:

```
# 11 连接
x = torch.randint(1, 10, (2, 3))
x
```

Out[33]:

```
tensor([[5, 1, 8],
        [5, 8, 6]])
```

In [34]:

```
torch.cat((x, x))
```

Out[34]:

```
tensor([[5, 1, 8],
        [5, 8, 6],
        [5, 1, 8],
        [5, 8, 6]])
```

In [38]:

```
torch.cat((x, x), 1) # 横轴连接
```

Out[38]:

```
tensor([[5, 1, 8, 5, 1, 8],
        [5, 8, 6, 5, 8, 6]])
```

In [48]:

```
# 12 chunk 切块
a = torch.randint(10, 100, (3, 3))
a
```

Out[48]:

```
tensor([[41, 49, 20],
        [65, 65, 42],
        [37, 79, 29]])
```

In [49]:

```
torch.chunk(a, 2, 0) # 横轴切分为两块
```

Out[49]:

```
(tensor([[41, 49, 20],
         [65, 65, 42]]), tensor([[37, 79, 29]]))
```

In [50]:

```
torch.chunk(a, 2, 1) # 纵轴切分为两块
```

Out[50]:

```
(tensor([[41, 49],
         [65, 65],
         [37, 79]]), tensor([[20],
         [42],
         [29]]))
```

In [51]:

```
# 13 index_select 根据索引选择
x = torch.randn(4, 4)
x
```

Out[51]:

```
tensor([[ 1.1549,  0.4992,  0.8953, -0.0380],
        [-0.8262, -0.9964, -1.2510,  1.0942],
        [ 0.7331, -0.5104,  0.6739, -1.1947],
        [-0.7761, -2.4222, -0.5131, -0.6125]])
```

In [52]:

```
indices = torch.tensor([0, 2])
```

In [53]:

```
torch.index_select(x, 0, indices) # 0 是横轴
```

Out[53]:

```
tensor([[ 1.1549,  0.4992,  0.8953, -0.0380],
        [ 0.7331, -0.5104,  0.6739, -1.1947]])
```

In [54]:

```
torch.index_select(x, 1, indices) # 1 是纵轴
```

Out[54]:

```
tensor([[ 1.1549,  0.8953],
        [-0.8262, -1.2510],
        [ 0.7331,  0.6739],
        [-0.7761, -0.5131]])
```

In [55]:

```
# 14 split 分割
x = torch.tensor([1,2,3,4,5,6,7,8])
torch.split(x, 2)
```

Out[55]:

```
(tensor([1, 2]), tensor([3, 4]), tensor([5, 6]), tensor([7, 8]))
```

In [56]:

```
torch.split(x, 3)
```

Out[56]:

```
(tensor([1, 2, 3]), tensor([4, 5, 6]), tensor([7, 8]))
```

In [59]:

```
# 15 .t 和 .transpose 转置
x = torch.tensor([[1,2],[3,4]])
x
```

Out[59]:

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

In [60]:

```
x.t()
```

Out[60]:

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

In [62]:

```
x.transpose(1, 0)
```

Out[62]:

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

In [64]:

```
# 16 tensor运算
x = torch.tensor([[1,2],[3,4]])
x
```

Out[64]:

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

In [65]:

```
torch.add(x, 1) # 加法
```

Out[65]:

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

In [66]:

```
torch.mul(x, 2) # 乘法
```

Out[66]:

```
tensor([[2, 4],
        [6, 8]])
```

In [ ]:

```

```

In [ ]:

```

```