numpy中的数据类型

numpy.arange

numpy.zeros

numpy.ones

numpy.full

numpy.linspace

numpy.random

```
In [2]:  import numpy as np
```

```
In [3]:  np.array([1, 2, 3])
Out[3]:  array([1, 2, 3])
```

```
In [4]:  range(10)
Out[4]:  range(0, 10)
```

```
In [5]:  list(range(10))
Out[5]:  [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
In [7]:  np.array(list(range(10)))
Out[7]:  array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [8]:  np.array(range(10))
Out[8]:  array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [9]:  np.array(range(0, 10, 2))
Out[9]:  array([0, 2, 4, 6, 8])
```

1. numpy.arange

```
In [10]: np.arange(10)

Out[10]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [11]: np.arange(0, 10, 2)

Out[11]: array([0, 2, 4, 6, 8])
```

numpy.arange 与 python中的range的不同在于：numpy.arange的步长可以是小数！

```
In [12]: np.arange(0, 10, 0.4)

Out[12]: array([0. , 0.4, 0.8, 1.2, 1.6, 2. , 2.4, 2.8, 3.2, 3.6, 4. , 4.4,
         4.8,
                5.2, 5.6, 6. , 6.4, 6.8, 7.2, 7.6, 8. , 8.4, 8.8, 9.2, 9.6]
         )
```

```
In [13]: range(0, 10, 0.4)

         ------------------------------------------------------------------
         ---------
         TypeError                                 Traceback (most recent c
         all last)
         <ipython-input-13-e138d611d5e1> in <module>()
         ----> 1 range(0, 10, 0.4)

         TypeError: 'float' object cannot be interpreted as an integer
```

2. numpy.zeros

```
In [14]: np.zeros(10)

Out[14]: array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

```
In [15]: np.zeros(10, dtype=int)

Out[15]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

```
In [16]: np.zeros((3, 5))

Out[16]: array([[0., 0., 0., 0., 0.],
                [0., 0., 0., 0., 0.],
                [0., 0., 0., 0., 0.]])
```

```
In [17]: np.zeros(shape=(3, 5), dtype=int)
```

```
Out[17]: array([[0, 0, 0, 0, 0],
                [0, 0, 0, 0, 0],
                [0, 0, 0, 0, 0]])
```

3. numpy.ones

```
In [18]: np.ones(10)
```

```
Out[18]: array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

```
In [19]: np.ones((2, 3))
```

```
Out[19]: array([[1., 1., 1.],
                [1., 1., 1.]])
```

numpy.ones 与 numpy.zeros用法相同

4. numpy.full

```
In [21]: np.full(10, 66)
```

```
Out[21]: array([66, 66, 66, 66, 66, 66, 66, 66, 66, 66])
```

```
In [22]: np.full((2, 3), 66)
```

```
Out[22]: array([[66, 66, 66],
                [66, 66, 66]])
```

```
In [23]: np.full((2, 3), 66.0)
```

```
Out[23]: array([[66., 66., 66.],
                [66., 66., 66.]])
```

```
In [24]: np.full((2, 3), fill_value=66.0)
```

```
Out[24]: array([[66., 66., 66.],
                [66., 66., 66.]])
```

5. numpy.linspace

numpy.linspace 用来生成等差数列

```
In [28]: np.linspace(0, 20, 10)    # 与arange等不同，linspace默认包含endpoint-20

Out[28]: array([ 0.        ,  2.22222222,  4.44444444,  6.66666667,  8.8888
         8889,
                11.11111111, 13.33333333, 15.55555556, 17.77777778, 20.
         ])
```

```
In [29]: np.linspace(0, 20, 11)

Out[29]: array([ 0.,  2.,  4.,  6.,  8., 10., 12., 14., 16., 18., 20.])
```

```
In [30]: np.linspace(0, 20, 10, endpoint=False)

Out[30]: array([ 0.,  2.,  4.,  6.,  8., 10., 12., 14., 16., 18.])
```

6. numpy.random

```
In [34]: np.random.randint(0, 10)           # 生成0到10（不包含10）之间的随机整数

Out[34]: 9
```

```
In [35]: np.random.randint(0, 10, 5)

Out[35]: array([1, 1, 8, 7, 6])
```

```
In [36]: np.random.randint(0, 10, size=5)

Out[36]: array([0, 1, 1, 6, 6])
```

```
In [37]: np.random.randint(0, 10, size=(2, 5))

Out[37]: array([[9, 0, 2, 7, 2],
                [2, 5, 2, 2, 3]])
```

```
In [39]: np.random.seed(20)
         np.random.randint(0, 10, 5)

Out[39]: array([3, 9, 4, 6, 7])
```

```
In [40]: np.random.seed(20)
         np.random.randint(0, 10, 5)

Out[40]: array([3, 9, 4, 6, 7])
```

上面两个代码段说明，计算机是伪随机，只要指定了种子，每次随机出来的结果是相同的

```
In [41]: np.random.random()       # 随机生成一个0-1之间的随机数

Out[41]: 0.95045165254683
```

```
In [42]: np.random.random(5)
```

```
Out[42]: array([0.67286566, 0.04117333, 0.84157413, 0.8165543 , 0.78251141]
         )
```

```
In [44]: np.random.random(size = 5)
```

```
Out[44]: array([0.98711439, 0.92971622, 0.18492277, 0.453444  , 0.44734283]
         )
```

```
In [43]: np.random.random((2, 5))
```

```
Out[43]: array([[0.73526734, 0.4773877 , 0.86620232, 0.31552421, 0.90653395
         ],
                [0.00137097, 0.57548998, 0.47452438, 0.27205127, 0.26839872
         ]])
```