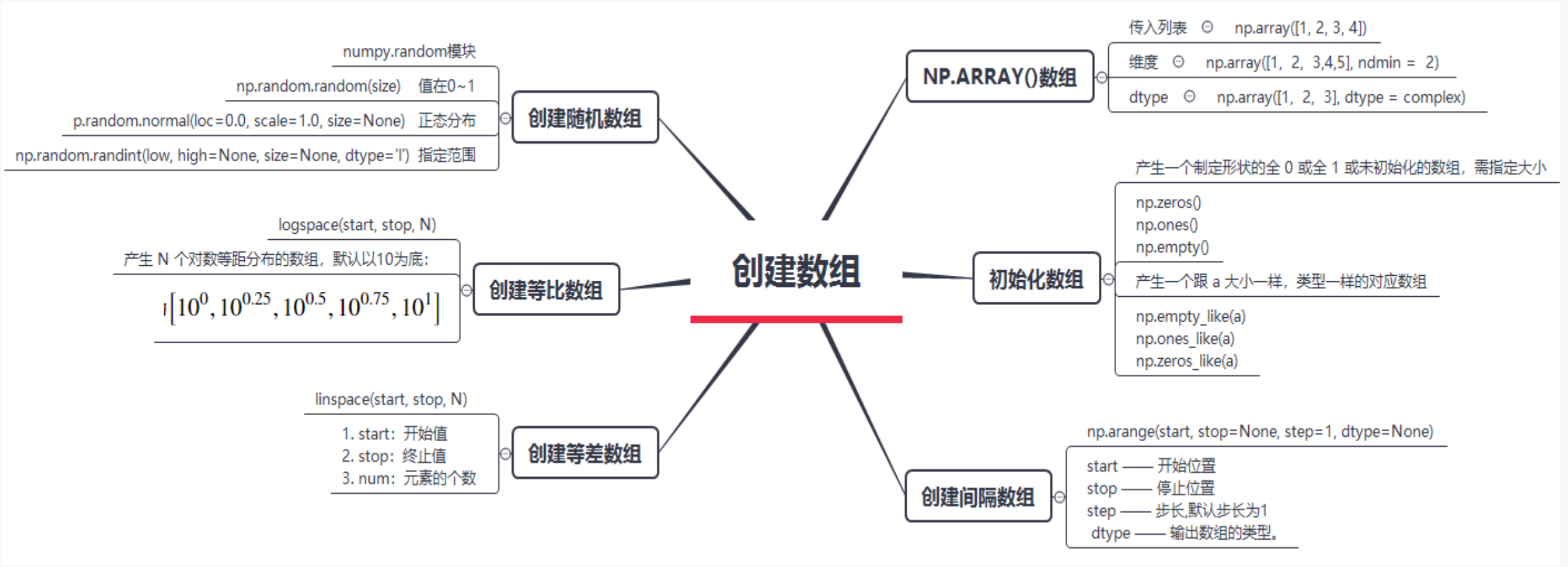
Python科学计算库Numpy之

02-创建Numpy数组

海豚数据科学实验室

知识结构图

海豚数据科学实验室



创建数组

使用np.array()创建一个NumPy数组，将python列表传递给NumPy对象

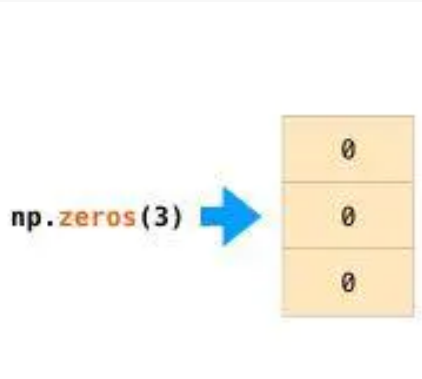
创造初始数组

一般情况，我们希望直接使用NumPy作为初始化的数组数据。

NumPy为这些情况提供了诸如ones()，zeros()和empty()类等方法。

```
np.zeros(shape, dtype=float, order='C')    # 全0
np.ones(shape, dtype=None, order='C')    # 全1
np.empty(shape, dtype=float, order='C')  # 未初始化
```

参数：
shape : 数组的形状
dtype : 数据类型。
order : {‘C’, ‘F’}, 可选规定返回数组元素在内存的存储顺序：C（C语言）

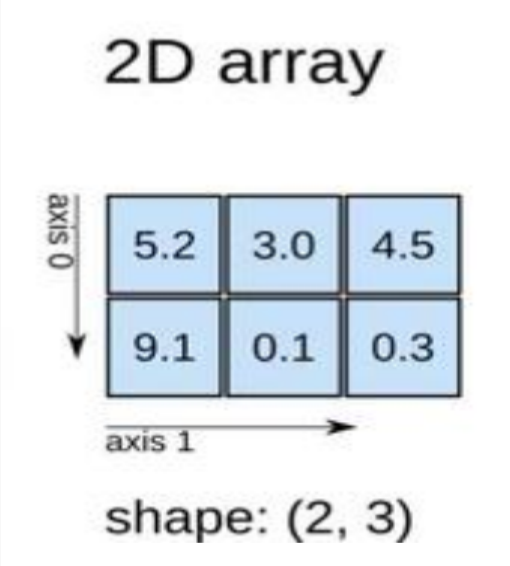


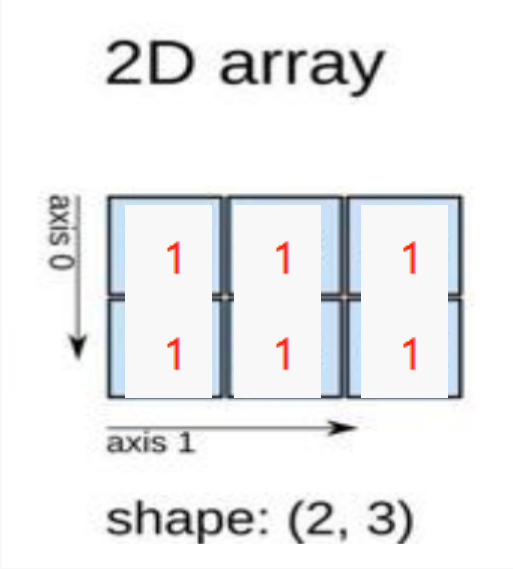

```
np.ones([2,3])      # 全是1
np.ones([2,3]) * 5  #全是 5
```

```
np.ones([2,3]) * 5
```

```
array([[5., 5., 5.],
       [5., 5., 5.]])
```

创造初始数组





np.ones_like(a, dtype=None, order='K)
np.zeros_like(a, dtype=None, order='K)
np.empty_like(a, dtype=None, order='K)

返回结果：

返回一个和a相同结构和数据类型的数组

参数：

a：想要复制结构的数组

dtype：元素类型（默认为a的元素类型）

order：改变数组的内存布局

默认为‘K'，意味着尽可能接近a的布局

创建间隔数组

语法:

numpy.arange(start, stop, step, dtype = None)

在给定间隔内返回均匀间隔的值，值在半开区间 [start，stop) 内生成

参数：

start —— 开始位置，数字，可选项，默认起始值为0

stop —— 停止位置，数字

step —— 步长，数字，可选项， 默认步长为1，如果指定了step，则还必须给出start。

dtype —— 输出数组的类型。 如果未给出dtype，则从其他输入参数推断数据类型。

```
A = np.arange(5)  # 只有结束项
print(A)  # 结果 [0 1 2 3 4] 结果不包含结束项
print(type(A))  # 结果 <class 'numpy.ndarray'>

A = np.arange(1, 5)  # 起点为1，步长默认为1
print(A)  # 结果 [1 2 3 4]

A = np.arange(1, 5, 2)  # 步长默认为2
print(A)  # 结果 [1 3]

A = np.arange(1, 5.2, 0.6)  # 浮点数参数，结果就不一定完全符合了
print(A)  # 结果 [1.  1.6 2.2 2.8 3.4 4.  4.6 5.2]
```

```
[0 1 2 3 4]
<class 'numpy.ndarray'>
[1 2 3 4]
[1 3]
[1.  1.6 2.2 2.8 3.4 4.  4.6 5.2]
```

创建等差数组

numpy.linspace(start, stop, num=50, endpoint=True, retstep=False, dtype=None)

创建一个以start为起点，stop为终点，中间有num项的等差数列：

1. start：开始值
2. stop：终止值
3. num：元素的个数（默认为50， 也就是生成50个元素，要是非负数）
4. endpoint: 如果为真，则终止值是最后一个元素，否则不包含终止值（默认为包含）
5. retstep: 如果为真，则以元组形式返回数组和步长，否则只返回数组（默认不返回步长）
6. dtype: 数据类型

```
>>> np.linspace(2.0, 3.0, num=5)
array([ 2.  , 2.25, 2.5 , 2.75, 3.  ])
>>> np.linspace(2.0, 3.0, num=5, endpoint=False)
array([ 2. , 2.2, 2.4, 2.6, 2.8])
>>> np.linspace(2.0, 3.0, num=5, retstep=True)
(array([ 2.  , 2.25, 2.5 , 2.75, 3.  ]), 0.25)
```

创建等比数组

np.logspace(start, stop, num=50, endpoint=True, base=10.0, dtype=None)

等比数列和等差数列的创建差不多，其中base 为对数 log 的底数。

```
In [106]: import numpy as np
          a = np.logspace(0,0,10)
          a
Out[106]: array([ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.])
```

以上例子的起始位和终止位都是0，元素个数是10，但是生成的等比数列全部是1，这是因为在logspace中，起始位和终止位代表的是10的幂（默认基数为10），0代表10的0次方

```
In [107]: a = np.logspace(0,9,10)
          a
Out[107]: array([  1.00000000e+00,   1.00000000e+01,   1.00000000e+02,
                   1.00000000e+03,   1.00000000e+04,   1.00000000e+05,
                   1.00000000e+06,   1.00000000e+07,   1.00000000e+08,
                   1.00000000e+09])
```

```
In [108]: a = np.logspace(0,9,10,base=2)
          a
Out[108]: array([   1.,    2.,    4.,    8.,   16.,   32.,   64.,  128.,  256.,  512.])
```

创建随机数组

使用numpy.random模块来生成随机数组

np.random.random(size)　创建一个指定size、值在0~1范围内的随机数组成的数组

np.random.normal(loc=0.0, scale=1.0, size=None) 创建一个指定size、均值为标准差为1的正态分布的随机数数组

np.random.randint(low, high=None, size=None, dtype='l') 创建一个指定size、值[low，high)区间的随机整型数组

创建一个3*3的在0~1范围内的随机数组成的数组

```
np.random.random((3,3))

array([[0.2131937 , 0.99476852, 0.12470957],
       [0.57641785, 0.00268608, 0.13124108],
       [0.0626813 , 0.73013671, 0.72995278]])
```

创建一个3*3均值为0，标准差为1的正态分布的随机数数组

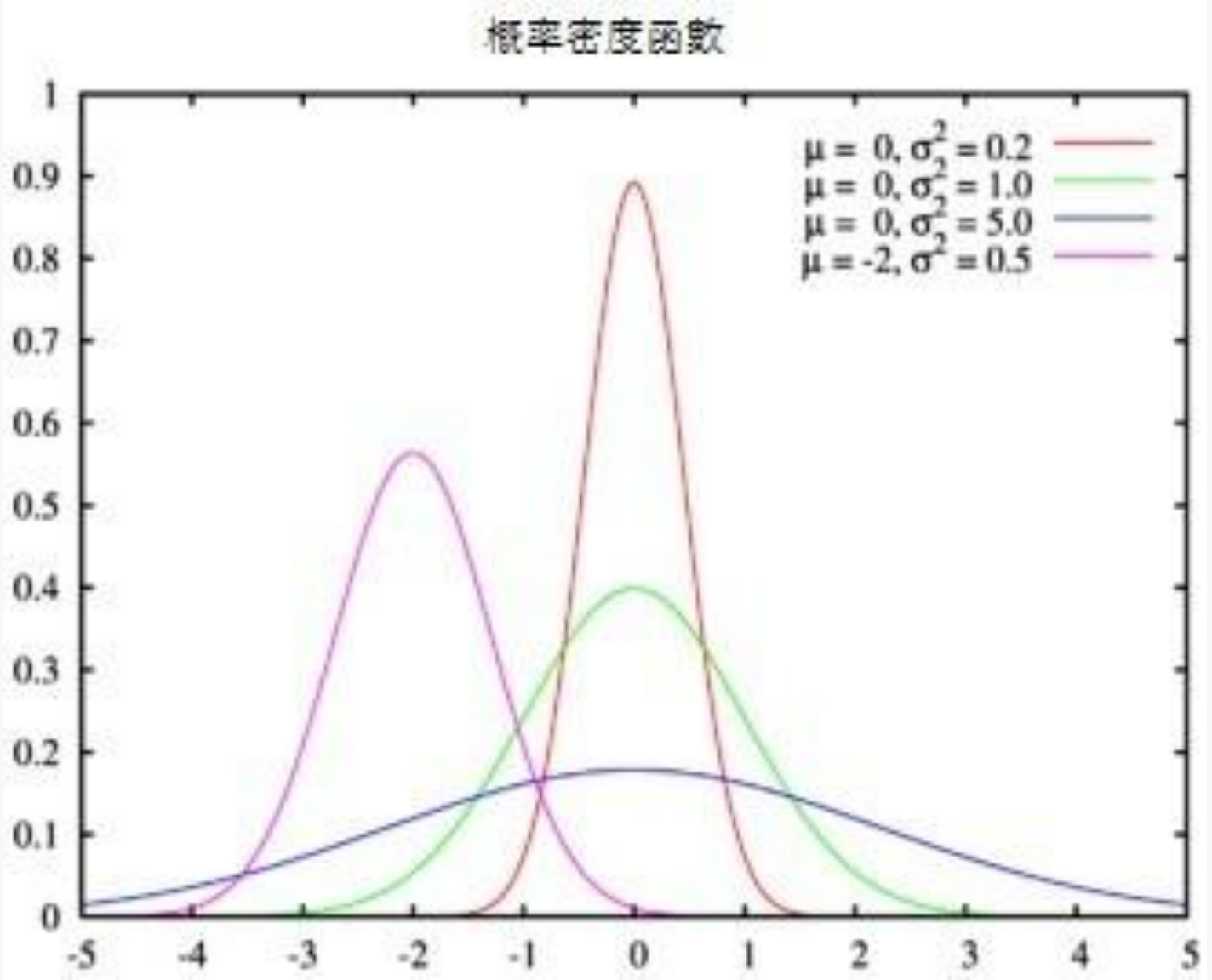
```
Int[7]: np.random.normal(0,1, (3,3))

Out[7]: array([[ 1.15213505,  3.70161801,  0.9126224 ],
               [-0.19543063,  1.54035371,  0.32575042],
               [ 0.16181688, -2.81311219, -0.86688313]])
```

创建一个3*3的，[0，10)区间的随机整型数

```
np.random.randint(0,10,(3,3))
array([[2, 6, 3],
       [4, 3, 8],
       [4, 3, 8]])
```

正态分布


概率密度函数
μ = 0, σ² = 0.2
μ = 0, σ² = 1.0
μ = 0, σ² = 5.0
μ = -2, σ² = 0.5
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-5
-4
-3
-2
-1
0
1
2
3
4
5

谢 谢!

海豚数据科学实验室