jupyter Numpy tutorial Last Checkpoint: 7 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

Array creation : conversion from other pyhton structure

```
In [1]: 
        import numpy

In [2]: import numpy as np

In [3]: cvalues = [20.1, 20.8, 21.9, 22.5, 22.7, 22.3, 21.8, 21.2, 20.9, 20.1]

In [4]: C = np.array(cvalues)
        print(C)

        [20.1 20.8 21.9 22.5 22.7 22.3 21.8 21.2 20.9 20.1]

In [5]: print(C * 9 / 5 + 32)
        [68.18 69.44 71.42 72.5  72.86 72.14 71.24 70.16 69.62 68.18]

In [6]: print(C)
        [20.1 20.8 21.9 22.5 22.7 22.3 21.8 21.2 20.9 20.1]

In [7]: fvalues = [ x*9/5 + 32 for x in cvalues]
        print(fvalues)

        [68.18, 69.44, 71.42, 72.5, 72.86, 72.14, 71.24000000000001, 70.16, 69.62, 68.18]

In [8]: type(C)
Out[8]: numpy.ndarray

In [9]: myarr=np.array([[3,6,37,7]],np.int64)

In [10]: myarr
Out[10]: array([[ 3,  6, 37,  7]], dtype=int64)

In [11]: myarr[0, 1]
Out[11]: 6

In [12]: myarr.shape
Out[12]: (1, 4)

In [13]: myarr.dtype
Out[13]: dtype('int64')

In [14]: myarr[0, 1]=45

In [15]: myarr
Out[15]: array([[ 3, 45, 37,  7]], dtype=int64)

In [16]: myarr.size
Out[16]: 4

In [17]: zeros=np.zeros((2,5))

In [18]: zeros
Out[18]: array([[0., 0., 0., 0., 0.],
                [0., 0., 0., 0., 0.]])

In [19]: zeros.size
Out[19]: 10

In [21]: zeros.dtype
Out[21]: dtype('float64')

         zeros.shape

In [22]: zeros.shape
Out[22]: (2, 5)

In [24]: rng=np.arange(15)

In [25]: rng
Out[25]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14])

In [26]: lspace=np.linspace(1,5,12)

In [27]: lspace
Out[27]: array([1.        , 1.36363636, 1.72727273, 2.09090909, 2.45454545,
                2.81818182, 3.18181818, 3.54545455, 3.90909091, 4.27272727,
                4.63636364, 5.        ])

In [28]: emp=np.empty((4,6))

In [29]: emp
Out[29]: array([[1.42417221e-306, 7.56595733e-307, 8.90071135e-308,
                 8.01007880e-307, 1.78020169e-306, 7.56601165e-307]
        8.01097889e-307, 1.78020109e-306, 7.56001105e-307],
       [1.02359984e-306, 8.90092016e-307, 1.02361342e-306,
        6.89804133e-307, 8.90104239e-307, 8.90098127e-307],
       [8.90111708e-307, 6.23054633e-307, 9.34598926e-307,
        1.42417629e-306, 1.11260687e-306, 9.34593493e-307],
       [1.33511290e-306, 1.33511969e-306, 2.22523140e-306,
        6.23059726e-307, 1.33511562e-306, 6.89805151e-307]])
```

```
In [30]: emp_like=np.empty_like(lspace)

In [31]: emp_like

Out[31]: array([1.        , 1.36363636, 1.72727273, 2.09090909, 2.45454545,
                2.81818182, 3.18181818, 3.54545455, 3.90909091, 4.27272727,
                4.63636364, 5.        ])

In [32]: id=np.identity(4)

In [33]: id

Out[33]: array([[1., 0., 0., 0.],
                [0., 1., 0., 0.],
                [0., 0., 1., 0.],
                [0., 0., 0., 1.]])

In [35]: arr=np.arange(99)

In [36]: arr

Out[36]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
                34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
                51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
                68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84,
                85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98])

In [37]: arr.shape

Out[37]: (99,)

In [38]: arr=arr.reshape(3,33)

In [39]: arr

Out[39]: array([[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15,
                 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
                 32],
                [33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48,
                 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64,
                 65],
                [66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81,
                 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97,
                 98]])

In [40]: arr.shape

Out[40]: (3, 33)

In [46]: arr=arr.ravel()

In [47]: arr.shape

Out[47]: (99,)

In [48]: a = np.arange(1, 10)

In [49]: a

Out[49]: array([1, 2, 3, 4, 5, 6, 7, 8, 9])

In [50]: x = range(1, 10)

In [51]: x

Out[51]: range(1, 10)

In [52]: list(x)

Out[52]: [1, 2, 3, 4, 5, 6, 7, 8, 9]

In [53]: x = np.arange(10.4)

In [54]: x

Out[54]: array([ 0.,  1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])

In [55]: x = np.arange(0.5, 10.4, 0.8)

In [56]: x

Out[56]: array([ 0.5,  1.3,  2.1,  2.9,  3.7,  4.5,  5.3,  6.1,  6.9,  7.7,  8.5,
                 9.3, 10.1])

In [57]: y=np.arange(12.04, 12.84, 0.08)

In [58]: y

Out[58]: array([12.04, 12.12, 12.2 , 12.28, 12.36, 12.44, 12.52, 12.6 , 12.68,
               12.76, 12.84])

In [59]: x = np.arange(0.5, 10.4, 0.8, int)
         print(x)

         [ 0  1  2  3  4  5  6  7  8  9 10 11 12]

In [61]: ak=np.linspace(1, 10)

In [62]: ak

Out[62]: array([ 1.        ,  1.18367347,  1.36734694,  1.55102041,  1.73469388,
                 1.91836735,  2.10204082,  2.28571429,  2.46938776,  2.65306122,
        2.83673469,  3.02040816,  3.20408163,  3.3877551 ,  3.57142857,
        3.75510204,  3.93877551,  4.12244898,  4.30612245,  4.48979592,
        4.67346939,  4.85714286,  5.04081633,  5.2244898 ,  5.40816327,
        5.59183673,  5.7755102 ,  5.95918367,  6.14285714,  6.32653061,
        6.51020408,  6.69387755,  6.87755102,  7.06122449,  7.24489796,
        7.42857143,  7.6122449 ,  7.79591837,  7.97959184,  8.16326531,
        8.34693878,  8.53061224,  8.71428571,  8.89795918,  9.08163265,
        9.26530612,  9.44897959,  9.63265306,  9.81632653, 10.        ])
```

```
In [63]: ak.shape
```

```
Out[63]: (50,)
```

```
In [65]: print(np.linspace(1, 10, 7))
```

```
[ 1.   2.5  4.   5.5  7.   8.5 10. ]
```

```
In [66]: print(np.linspace(1, 10, 7, endpoint=False))
```

```
[1.         2.28571429 3.57142857 4.85714286 6.14285714 7.42857143
 8.71428571]
```

```
In [67]: samples, spacing = np.linspace(1, 10, retstep=True)
         print(spacing)
```

```
0.1836734693877551
```

```
In [68]: samples, spacing = np.linspace(1, 10, 20, endpoint=True, retstep=True)
         print(spacing)
```

```
0.47368421052631576
```

```
In [69]: samples, spacing = np.linspace(1, 10, 20, endpoint=False, retstep=True)
         print(spacing)
```

```
0.45
```

```
In [ ]:
```