Numpy

August 23, 2022

[12]:
```
seq=(1,2,3,4,5,6,7,8,9)
```

[5]:
```
def quadrado(var):
    return var**3
```

[6]:
```
list(map(quadrado, seq))
```

[6]:
```
[1, 8, 27, 64, 125]
```

[8]:
```
list(map(quadrado, seq))
```

[8]:
```
[1, 8, 27, 64, 125]
```

[10]:
```
list(map(lambda x:x**4, seq))
```

[10]:
```
[1, 16, 81, 256, 625]
```

[13]:
```
list(filter(lambda x:x%2==0, seq))
```

[13]:
```
[2, 4, 6, 8]
```

[14]:
```
list(filter(lambda x:x%2!=0, seq))
```

[14]:
```
[1, 3, 5, 7, 9]
```

[15]:
```
str = 'Olá Mundo, tudo bem!!'
```

[16]:
```
str.lower()
```

[16]:
```
'olá mundo, tudo bem!!'
```

[17]:
```
str.upper()
```

[17]:
```
'OLÁ MUNDO, TUDO BEM!!'
```

[20]:
```
str.split('t')
```

[20]:
```
['Olá Mundo, ', 'udo bem!!']
```

[21]:
```
len(str.split())
```

[21]: 4

[22]:
```
lista = [1, 3, 5, 7, 9, 11]
```

[25]:
```
lista.append(15)
```

[26]:
```
lista
```

[26]: [1, 3, 5, 7, 9, 11, 15]

[27]:
```
lista.pop()
```

[27]: 15

[28]:
```
lista
```

[28]: [1, 3, 5, 7, 9, 11]

[29]:
```
last = lista.pop()
```

[30]:
```
last
```

[30]: 11

[31]:
```
primeiro=lista.pop(0)
```

[32]:
```
primeiro
```

[32]: 1

[33]:
```
lista
```

[33]: [3, 5, 7, 9]

[34]:
```

```

[35]:
```
import numpy as np
```

[36]:
```
my_array=np.arange(1000000)
```

[37]:
```
my_list = list(range(1000000))
```

[38]:
```
%time for _ in range(10):my_arr2=my_array*2
```

```
CPU times: user 33.6 ms, sys: 2.65 ms, total: 36.2 ms
Wall time: 33.6 ms
```

[39]:
```
%time for _ in range(10):my_list2=[x*2 for x in my_list]
```

```
CPU times: user 753 ms, sys: 128 ms, total: 882 ms
Wall time: 887 ms
```

[40]:
```
887/33.6
```

[40]:
```
26.398809523809522
```

[41]:
```
data = np.random.randn(2,3)
```

[42]:
```
data
```

[42]:
```
array([[-0.0416402 ,  1.63089983, -1.11338632],
       [-0.47514538,  1.28852064, -1.56626068]])
```

[43]:
```
data+data
```

[43]:
```
array([[-0.08328039,  3.26179967, -2.22677264],
       [-0.95029075,  2.57704127, -3.13252135]])
```

[44]:
```
data*20
```

[44]:
```
array([[ -0.83280394,  32.61799667, -22.26772636],
       [ -9.50290754,  25.77041272, -31.3252135 ]])
```

[45]:
```
1/data
```

[45]:
```
array([[-24.01525614,   0.61315844,  -0.89816085],
       [ -2.10461903,   0.77608381,  -0.63846333]])
```

[46]:
```
1/0
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
/tmp/ipykernel_82214/2354412189.py in <module>
----> 1 1/0

ZeroDivisionError: division by zero
```

[47]:
```
data+25
```

[47]:
```
array([[24.9583598 , 26.63089983, 23.88661368],
       [24.52485462, 26.28852064, 23.43373932]])
```

[48]:
```
data-data
```

[48]:
```
array([[0., 0., 0.],
       [0., 0., 0.]])
```

[49]:
```
data.shape
```

[49]:
```
(2, 3)
```

[51]:
```
data.dtype
```

[51]:
```
dtype('float64')
```

[52]:
```
# criando array de numpy
data1 = [1,3.5, 8, 6.2]
```

[53]:
```
array1= np.array(data1)
```

[54]:
```
array1.dtype
```

[54]:
```
dtype('float64')
```

[55]:
```
array1.shape
```

[55]:
```
(4,)
```

[57]:
```
array1.ndim
```

[57]:
```
1
```

[58]:
```
data1 = [[1,3.5, 8, 6.2],[1,3.5, 8, 6.2]]
```

[59]:
```
array2=np.array(data1)
```

[60]:
```
array2
```

[60]:
```
array([[1. , 3.5, 8. , 6.2],
       [1. , 3.5, 8. , 6.2]])
```

[61]:
```
array2.ndim
```

[61]:
```
2
```

[62]:
```
array2.shape
```

[62]:
```
(2, 4)
```

[63]:
```
np.zeros(10)
```

[63]:
```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

[64]:
```
np.zeros((3, 6))
```

[64]:
```
array([[0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0.]])
```

[65]:
```
np.zeros((3, 6, 3))
```

[65]:
```
array([[[0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.]],

       [[0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.]],

       [[0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.]]])
```

[66]:
```
np.empty((2,3,2))
```

[66]:
```
array([[[4.68777865e-310, 0.00000000e+000],
        [1.01855798e-312, 9.54898106e-313],
        [1.16709769e-312, 1.01855798e-312]],

       [[1.23075756e-312, 1.20953760e-312],
        [1.10343781e-312, 9.76118064e-313],
        [1.12465777e-312, 1.90979621e-312]]])
```

[67]:
```
np.ones((5,5))
```

[67]:
```
array([[1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.]])
```

[68]:
```
np.eye(4)
```

[68]:
```
array([[1., 0., 0., 0.],
       [0., 1., 0., 0.],
       [0., 0., 1., 0.],
       [0., 0., 0., 1.]])
```

[69]:
```
np.arange(20)
```

[69]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19])
```

[70]:
```
np.arange(0, 5)
```

[70]:
```
array([0, 1, 2, 3, 4])
```

[72]:
```
np.arange(0, 8, 3)
```

[72]:
```
array([0, 3, 6])
```

[73]:
```
array_new= np.array([1,3,6])
```

[74]:
```
array_new.dtype
```

[74]:
```
dtype('int64')
```

[75]:
```
arr2=np.array([1,3,6], dtype=np.float64)
```

[76]:
```
arr2.dtype
```

[76]:
```
dtype('float64')
```

[77]:
```
arr3=np.array([1,3,6], dtype=np.int32)
```

[78]:
```
arr3.dtype
```

[78]:
```
dtype('int32')
```

[79]:
```
array_new= np.array([1,3,6])
```

[80]:
```
array_new.dtype
```

[80]:
```
dtype('int64')
```

[81]:
```
float_new=array_new.astype(np.float32)
```

[82]:
```
float_new.dtype
```

[82]:
```
dtype('float32')
```

[83]:
```
arr4= np.array([3.7, -1.3, 3.5])
```

[84]:
```
arr4.dtype
```

[84]:
```
dtype('float64')
```

[85]:
```
arr4.astype(np.int32)
```

[85]:
```
array([ 3, -1,  3], dtype=int32)
```

[86]:
```
arr_string= np.array(['3.7', '-1.3', '3.5'], dtype=np.string_)
```

[87]:
```
arr_string.astype(float)
```

[87]:
```
array([ 3.7, -1.3,  3.5])
```

[88]:
```
np.linspace(0,100,5)
```

[88]:
```
array([  0.,  25.,  50.,  75., 100.])
```

[89]:
```
np.linspace(0,10,3)
```

[89]:
```
array([ 0.,  5., 10.])
```

[90]:
```
arr4==arr_string
```

```
/tmp/ipykernel_82214/3421481964.py:1: FutureWarning: elementwise comparison
failed; returning scalar instead, but in the future will perform elementwise
comparison
  arr4==arr_string
```

[90]:
```
False
```

[91]:
```
arr4 > 2
```

[91]:
```
array([ True, False,  True])
```

[92]:
```
bolean = arr4 > 2
```

[93]:
```
bolean
```

[93]:
```
array([ True, False,  True])
```

[94]:
```
arr5= np.arange(20)
```

[96]:
```
arr5[4:12]
```

[96]:
```
array([ 4,  5,  6,  7,  8,  9, 10, 11])
```

[97]:
```
array_slice=arr5[4:12]
```

[98]:
```
array_slice
```

[98]:
```
array([ 4,  5,  6,  7,  8,  9, 10, 11])
```

[99]:
```
arr5
```

[99]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19])
```

[100]:
```
array_slice[2]=36
```

[108]:
```
array_slice
```

[108]:
```
array([22, 22, 22, 22])
```

[109]:
```
arr5
```

[109]:
```
array([ 0,  1,  2,  3, 32, 32, 32, 32, 32, 32, 32, 32, 12, 13, 14, 15, 16,
       17, 18, 19])
```

[103]:
```
array_slice[:]=32
```

[106]:
```
array_slice=arr5[4:8].copy()
```

[107]:
```
array_slice[:]=22
```

[12]:
```
import numpy as np
arr2=np.array([[0.,4.,1.],[7., 2., 12.]])
```

[13]:
```
arr2
```

[13]:
```
array([[ 0.,  4.,  1.],
       [ 7.,  2., 12.]])
```

[14]:
```
bol=arr2>1.5
```

[15]:
```
bol
```

[15]:
```
array([[False,  True, False],
       [ True,  True,  True]])
```

[17]:
```
arr2[1,1]
```

[17]:
```
2.0
```

[20]:
```
arr2[1][2]
```

[20]:
```
12.0
```

[21]:
```
arr2
```

[21]:
```
array([[ 0.,  4.,  1.],
       [ 7.,  2., 12.]])
```

[24]:
```
arr2[:2,2:]
```

[24]:
```
array([[ 1.],
       [12.]])
```

[25]:
```
arr3=np.array([[0.,4.,1.],[7., 2., 12.], [5., 7., 16.]])
```

[ ]:
```
arr3
```

[29]:
```
arr3[:,1:]
```

[29]:
```
array([[ 4.,  1.],
       [ 2., 12.],
       [ 7., 16.]])
```

1 Funções Universais

[30]:
```
array=np.arange(6)
```

[31]:
```
array
```

[31]:
```
array([0, 1, 2, 3, 4, 5])
```

[32]:
```
np.sqrt(array)
```

[32]:
```
array([0.        , 1.        , 1.41421356, 1.73205081, 2.        ,
       2.23606798])
```

[33]:
```
np.exp(array)
```

[33]:
```
array([  1.        ,   2.71828183,   7.3890561 ,  20.08553692,
        54.59815003, 148.4131591 ])
```

[34]:
```
x=np.random.randn(8)
x
```

[34]:
```
array([ 0.07805864, -0.24254203, -1.06025933, -0.59064368, -0.10282507,
        0.66972538,  2.22974646,  0.25463292])
```

[35]:
```
y=np.random.randn(8)
y
```

[35]:
```
array([-1.09001616,  2.19900513, -0.78308249, -0.20690645,  0.79807914,
        0.74996121, -0.0398848 ,  0.69865195])
```

[36]:
```
np.maximum(x,y)
```

[36]:
```
array([ 0.07805864,  2.19900513, -0.78308249, -0.20690645,  0.79807914,
        0.74996121,  2.22974646,  0.69865195])
```

[38]:
```
arr=np.random.randn(5,4)
```

[39]:
```
arr
```

[39]:
```
array([[ 0.70701718,  0.46209266, -0.7146948 , -0.33421004],
       [-0.23185797,  0.65333815,  0.11748646,  1.62424065],
       [-1.00911326, -0.37383727,  0.41029615, -0.60534126],
       [ 0.09962542, -0.28757215, -0.05826101,  0.28315026],
       [ 1.20489126, -0.37661979,  1.46436934, -0.0300231 ]])
```

[40]:
```
arr.mean()
```

[40]:
```
0.15024884549701903
```

[41]:
```
np.mean(arr)
```

[41]:
```
0.15024884549701903
```

[42]:
```
arr.min()
```

[42]:
```
-1.0091132558679086
```

[43]:
```
arr.sum()
```

[43]:
```
3.0049769099403805
```

[44]:
```
arr.argmin()
```

[44]:
```
8
```

[45]:
```
arr.max()
```

[45]:
```
1.6242406487840042
```

[46]:
```
arr.argmax()
```

[46]:
```
7
```

[47]:
```
arr.var()
```

[47]:
```
0.481244191300792
```

[48]:
```
arr.std()
```

[48]:
```
0.6937176596431663
```

[49]:
```
np.sin(arr)
```

[49]:
```
array([[ 0.64956882,  0.44582227, -0.65538693, -0.32802302],
       [-0.22978618,  0.60784048,  0.11721637,  0.99857219],
       [-0.84635989, -0.36519035,  0.39888092, -0.56904273],
       [ 0.0994607 , -0.28362491, -0.05822805,  0.27938185],
       [ 0.93380032, -0.36777928,  0.99434199, -0.03001859]])
```

[50]:
```
np.cos(arr)
```

[50]:
```
array([[ 0.7603028 ,  0.8951215 ,  0.7552933 ,  0.94466973],
       [ 0.97324114,  0.79405916,  0.9931064 , -0.05341888],
       [ 0.53261144,  0.93093287,  0.91700273,  0.82230796],
       [ 0.99504149,  0.9589353 ,  0.99830331,  0.96018008],
       [ 0.35779459,  0.92991312,  0.10622619,  0.99954934]])
```

[51]:
```
arr
```

[51]:
```
array([[ 0.70701718,  0.46209266, -0.7146948 , -0.33421004],
       [-0.23185797,  0.65333815,  0.11748646,  1.62424065],
       [-1.00911326, -0.37383727,  0.41029615, -0.60534126],
       [ 0.09962542, -0.28757215, -0.05826101,  0.28315026],
       [ 1.20489126, -0.37661979,  1.46436934, -0.0300231 ]])
```

[52]:
```
arr.mean(axis=1)
```

[52]:
```
array([ 0.03005125,  0.54080182, -0.39449891,  0.00923563,  0.56565443])
```

[53]:
```
arr.mean(axis=0)
```

[53]:
```
array([0.15411253, 0.01548032, 0.24383923, 0.1875633 ])
```

[54]:
```
arr.sum(axis=1)
```

[54]:
```
array([ 0.120205  ,  2.16320729, -1.57799562,  0.03694253,  2.26261771])
```

[55]:
```
arr.sum(axis=0)
```

[55]:
```
array([0.77056263, 0.07740161, 1.21919616, 0.93781652])
```

[56]:
```
arr.cumsum()
```

[56]:
```
array([ 0.70701718,  1.16910984,  0.45441504,  0.120205  , -0.11165297,
        0.54168518,  0.65917164,  2.28341229,  1.27429904,  0.90046177,
```

```
        1.31075792,  0.70541667,  0.80504209,  0.51746994,  0.45920893,
        0.7423592 ,  1.94725046,  1.57063067,  3.03500001,  3.00497691])
```

[57]:
```
arr
```

[57]:
```
array([[ 0.70701718,  0.46209266, -0.7146948 , -0.33421004],
       [-0.23185797,  0.65333815,  0.11748646,  1.62424065],
       [-1.00911326, -0.37383727,  0.41029615, -0.60534126],
       [ 0.09962542, -0.28757215, -0.05826101,  0.28315026],
       [ 1.20489126, -0.37661979,  1.46436934, -0.0300231 ]])
```

[58]:
```
arr.cumprod()
```

[58]:
```
array([ 7.07017177e-01,  3.26707449e-01, -2.33496114e-01,  7.80367454e-02,
       -1.80934416e-02, -1.18211357e-02, -1.38882343e-03, -2.25578347e-03,
        2.27634101e-03, -8.50981097e-04, -3.49154271e-04,  2.11357485e-04,
        2.10565783e-05, -6.05528548e-06,  3.52787020e-07,  9.98917375e-08,
        1.20358682e-07, -4.53294618e-08, -6.63790742e-08,  1.99290558e-09])
```

[60]:
```
arr.cumsum(axis=1)
```

[60]:
```
array([[ 0.70701718,  1.16910984,  0.45441504,  0.120205  ],
       [-0.23185797,  0.42148018,  0.53896664,  2.16320729],
       [-1.00911326, -1.38295052, -0.97265437, -1.57799562],
       [ 0.09962542, -0.18794673, -0.24620774,  0.03694253],
       [ 1.20489126,  0.82827147,  2.29264081,  2.26261771]])
```

[61]:
```
arr.cumprod(axis=1)
```

[61]:
```
array([[ 7.07017177e-01,  3.26707449e-01, -2.33496114e-01,
         7.80367454e-02],
       [-2.31857973e-01, -1.51481660e-01, -1.77970445e-02,
        -2.89066832e-02],
       [-1.00911326e+00,  3.77244140e-01,  1.54781820e-01,
        -9.36958212e-02],
       [ 9.96254202e-02, -2.86494963e-02,  1.66914846e-03,
         4.72619824e-04],
       [ 1.20489126e+00, -4.53785898e-01, -6.64510157e-01,
         1.99506549e-02]])
```

[62]:
```
np.random.normal(size=(5,5))
```

[62]:
```
array([[-1.34720061, -2.41019578, -0.77685449,  0.33293232,  0.54912138],
       [-1.14812555, -0.91552847,  0.24153039, -0.31836711, -0.04140338],
       [-0.87987632,  1.37538998,  0.50950692,  2.07415703, -0.70660703],
       [-0.3239105 ,  0.53659958, -1.99675915,  1.03364244,  1.09561928],
       [ 0.23492983,  0.52168446,  0.87863333, -1.23846032, -0.28687538]])
```

[67]:
```
from random import normalvariate
n=1000000
```

1.0.1 Função random de python e criei 1 milhão de elementos

[70]:
```
%timeit samples=[normalvariate(0,1) for _ in range(n)]
```

```
1.09 s ± 70.3 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

1.0.2 Função random para uma distribuição normal e criei 1 milhão de elementos

[73]:
```
%timeit np.random.normal(size=n)
```

```
43.4 ms ± 853 µs per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

[74]:
```
np.random.seed(23082022)
```

[75]:
```
np.random.rand(10)
```

[75]:
```
array([0.9950068 , 0.03213636, 0.13331376, 0.92452319, 0.54759539,
       0.44709787, 0.33093033, 0.04784843, 0.85563495, 0.51744602])
```

[76]:
```
np.random.rand(10)
```

[76]:
```
array([0.46895624, 0.30376428, 0.13883955, 0.99894508, 0.35101724,
       0.11543095, 0.03803365, 0.1361207 , 0.61028589, 0.24120486])
```

[84]:
```
np.random.randint(0, 100, 15)
```

[84]:
```
array([94, 99, 42,  7, 53, 12, 50, 37,  7, 25, 66, 18, 65, 49, 40])
```

1.1 Reshape

[85]:
```
array10=np.random.rand(20)
```

[86]:
```
array10
```

[86]:
```
array([0.34906019, 0.17318441, 0.91449322, 0.1634006 , 0.45877218,
       0.83554532, 0.16701898, 0.86761334, 0.7891199 , 0.36272385,
       0.46075135, 0.88110305, 0.51847727, 0.82622975, 0.97164878,
       0.63018256, 0.64571532, 0.81730757, 0.38876882, 0.62808171])
```

[87]:
```
array10.reshape(5,4)
```

[87]:
```
array([[0.34906019, 0.17318441, 0.91449322, 0.1634006 ],
       [0.45877218, 0.83554532, 0.16701898, 0.86761334],
       [0.7891199 , 0.36272385, 0.46075135, 0.88110305],
       [0.51847727, 0.82622975, 0.97164878, 0.63018256],
       [0.64571532, 0.81730757, 0.38876882, 0.62808171]])
```

[88]:
```
!pip install pandas
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pandas in ./.local/lib/python3.10/site-packages
(1.4.3)
Requirement already satisfied: pytz>=2020.1 in /usr/lib/python3/dist-packages
(from pandas) (2022.1)
Requirement already satisfied: python-dateutil>=2.8.1 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: numpy>=1.21.0 in /usr/lib/python3/dist-packages
(from pandas) (1.21.5)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from
python-dateutil>=2.8.1->pandas) (1.16.0)
```

[89]:
```
import pandas as pd
```

[ ]:
```

```