In [1]:
```
import numpy as np
```

In [2]:
```
np.__version__
```

Out[2]:
```
'2.2.6'
```

In [3]:
```
np.show_config()
```

```
Build Dependencies:
  blas:
    detection method: pkgconfig
    found: true
    include directory: C:/Users/runneradmin/AppData/Local/Temp/cibw-run-gnu2g9dg/
cp313-win_amd64/build/venv/Lib/site-packages/scipy_openblas64/include
    lib directory: C:/Users/runneradmin/AppData/Local/Temp/cibw-run-gnu2g9dg/cp31
3-win_amd64/build/venv/Lib/site-packages/scipy_openblas64/lib
    name: scipy-openblas
    openblas configuration: OpenBLAS 0.3.29  USE64BITINT DYNAMIC_ARCH NO_AFFINITY
      Haswell MAX_THREADS=24
    pc file directory: C:/a/numpy/numpy/.openblas
    version: 0.3.29
  lapack:
    detection method: pkgconfig
    found: true
    include directory: C:/Users/runneradmin/AppData/Local/Temp/cibw-run-gnu2g9dg/
cp313-win_amd64/build/venv/Lib/site-packages/scipy_openblas64/include
    lib directory: C:/Users/runneradmin/AppData/Local/Temp/cibw-run-gnu2g9dg/cp31
3-win_amd64/build/venv/Lib/site-packages/scipy_openblas64/lib
    name: scipy-openblas
    openblas configuration: OpenBLAS 0.3.29  USE64BITINT DYNAMIC_ARCH NO_AFFINITY
      Haswell MAX_THREADS=24
    pc file directory: C:/a/numpy/numpy/.openblas
    version: 0.3.29
Compilers:
  c:
    commands: cl
    linker: link
    name: msvc
    version: 19.29.30159
  c++:
    commands: cl
    linker: link
    name: msvc
    version: 19.29.30159
  cython:
    commands: cython
    linker: cython
    name: cython
    version: 3.1.0
Machine Information:
  build:
    cpu: x86_64
    endian: little
    family: x86_64
    system: windows
  host:
    cpu: x86_64
    endian: little
    family: x86_64
    system: windows
Python Information:
  path: C:\Users\runneradmin\AppData\Local\Temp\build-env-iv6bfc05\Scripts\pytho
n.exe
  version: '3.13'
SIMD Extensions:
  baseline:
  - SSE
  - SSE2
```

```
- SSE3
found:
- SSSE3
- SSE41
- POPCNT
- SSE42
- AVX
- F16C
- FMA3
- AVX2
not found:
- AVX512F
- AVX512CD
- AVX512_SKX
- AVX512_CLX
- AVX512_CNL
- AVX512_ICL
```

Reshape

In [4]:
```
a = np.array([1,2,3,4,5,6,7,8,9,10,11,12])
a
```

Out[4]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

In [5]:
```
a.shape
```

Out[5]:
```
(12,)
```

In [6]:
```
a.ndim
```

Out[6]:
```
1
```

In [ ]:
```
12 = 1 x 12
     2 x 6
     3 x 4
     4 x 3
     6 x 2
     12 x 1
```

In [12]:
```
a.reshape(4,3)
```

Out[12]:
```
array([[ 1,  2,  3],
       [ 4,  5,  6],
       [ 7,  8,  9],
       [10, 11, 12]])
```

In [13]:
```
a.reshape(3,4)
```

Out[13]:
```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12]])
```

In [14]:
```
a.reshape(2,6)
```

Out[14]:
```
array([[ 1,  2,  3,  4,  5,  6],
       [ 7,  8,  9, 10, 11, 12]])
```

In [15]:
```
b = np.array([1,2,3,4]) # 1d
b
```

Out[15]:
```
array([1, 2, 3, 4])
```

In [ ]:
```
4 =  1 x 4
     4 x 1
     2 x 2
```

In [16]:
```
b.reshape(1,4)
```

Out[16]:
```
array([[1, 2, 3, 4]])
```

In [17]:
```
b.reshape(2,2)
```

Out[17]:
```
array([[1, 2],
       [3, 4]])
```

In [19]:
```
b.reshape(4,1)
```

Out[19]:
```
array([[1],
       [2],
       [3],
       [4]])
```

In [21]:
```
c = np.array([[1,2,3,4]]) # 2d
c
```

Out[21]:
```
array([[1, 2, 3, 4]])
```

In [22]:
```
c.reshape(1,4)
```

Out[22]:
```
array([[1, 2, 3, 4]])
```

In [23]:
```
c.reshape(2, 2)
```

Out[23]:
```
array([[1, 2],
       [3, 4]])
```

In [24]:
```
c.reshape(4, 1)
```

Out[24]:
```
array([[1],
       [2],
       [3],
       [4]])
```

In [26]:
```
d = np.array([1,2,3])
d
```

Out[26]:
```
array([1, 2, 3])
```

In [30]:
```
d.reshape(3,1)
```

Out[30]:
```
array([[1],
       [2],
       [3]])
```

how to convert n- dimension array to 1 dimension

In [31]:
```
x = np.array([[[[2,3,4,5]]]])
x
```

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

In [33]:
```
x.ndim
```

Out[33]:
```
4
```

In [35]:
```
# convert n dim to 1 dim
y = x.flatten()
y
```

Out[35]:
```
array([2, 3, 4, 5])
```

In [36]:
```
y.ndim
```

Out[36]:
```
1
```

In [38]:
```
z = x.ravel()
z
```

Out[38]:
```
array([2, 3, 4, 5])
```

In [39]:
```
z.ndim
```

Out[39]:
```
1
```

flatten

In [50]:
```
arr = np.array([[1,2],[3,4]])
flat_arr = arr.flatten()
flat_arr[0] = 99
flat_arr
```

Out[50]:
```
array([99,  2,  3,  4])
```

In [51]:
```
arr
```

Out[51]:
```
array([[1, 2],
       [3, 4]])
```

Ravel

In [52]:
```
arr1 = np.array([[1,2],[3,4]])
ravel_arr = arr1.ravel()
```

```
ravel_arr[0] = 99
ravel_arr
```

Out[52]:
```
array([99,  2,  3,  4])
```

In [53]:
```
arr1
```

Out[53]:
```
array([[99,  2],
       [ 3,  4]])
```

asarray

In [54]:
```
lst = [1,2,3]

s = np.asarray(lst)
s
```

Out[54]:
```
array([1, 2, 3])
```

Hstack and Vstack

In [55]:
```
a = np.array([1,2,3,4])
b = np.array([5,6,7,8])
c = np.hstack((a,b))
c
```

Out[55]:
```
array([1, 2, 3, 4, 5, 6, 7, 8])
```

In [56]:
```
c = np.vstack((a,b))
c
```

Out[56]:
```
array([[1, 2, 3, 4],
       [5, 6, 7, 8]])
```

In [62]:
```
# load the file

data = np.genfromtxt("demo.txt",delimiter=",",dtype = "int32")
data
```

Out[62]:
```
array([  1, 100,  24], dtype=int32)
```

In [63]:
```
data[data > 20]
```

Out[63]:
```
array([100,  24], dtype=int32)
```

In [64]:
```
data[data > 50]
```

Out[64]:
```
array([100], dtype=int32)
```

In [67]:
```
data1 = np.genfromtxt("test.csv",delimiter=",",dtype = "int32")
data1
```

Out[67]:
```
array([[ 1, 10, 20, 30],
       [ 1, 10, 20, 30],
       [ 1, 10, 20, 30],
       [ 1, 10, 20, 30],
       [ 1, 10, 20, 30],
       [ 1, 10, 20, 30],
       [ 1, 10, 20, 30]], dtype=int32)
```

In [ ]:
```

```