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

```

creating array in the given range

- range
- arange
- linspace

In [3]:
```
for i in range(1,10):
    print(i, end = " ")
```

```
1 2 3 4 5 6 7 8 9
```

In [5]:
```
np.array(range(1,10,2))
```

Out[5]:
```
array([1, 3, 5, 7, 9])
```

In [6]:
```
np.array(range(11,0,-2))
```

Out[6]:
```
array([11,  9,  7,  5,  3,  1])
```

In [8]:
```
# arange
np.arange(1,10,3)
```

Out[8]:
```
array([1, 4, 7])
```

In [9]:
```
# linspace

np.linspace(1,10,5)


# (ending value - starting value)/(size-1)
# (10-1)/4
```

Out[9]:
```
array([ 1.  ,  3.25,  5.5 ,  7.75, 10.  ])
```

In [10]:
```
np.linspace(1,30,5)
```

Out[10]:
```
array([ 1.  ,  8.25, 15.5 , 22.75, 30.  ])
```

Reshape

In [13]:
```
s = np.array(range(1,101))
```

In [19]:
```
1 s.shape
```

Out[19]:
```
(100,)
```

In [20]:
```
1 s.shape[0]
```

Out[20]:
```
100
```

In [22]:
```
1 s1 = np.array([[[1,2,3]],[4,5,6],[6,7,8]]])
2 s1.shape
```

Out[22]:
```
(1, 3)
```

In [23]:
```
1 s1
```

Out[23]:
```
array([[list([1, 2, 3]]), list([4, 5, 6]), list([6, 7, 8])]],
      dtype=object)
```

In [24]:
```
1 s.shape
```

Out[24]:
```
(100,)
```

In [25]:
```
1 s.size
```

Out[25]:
```
100
```

In [29]:
```
1 #print(s)
2 s.reshape(25,4)
3 #s.reshape(30,2)
```

...

In [33]:
```
s.reshape(2,2,5,5)
```

Out[33]:
```
array([[[[  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,  99, 100]]]])
```

Ramdom to create arrays

- random.randint
- random.randn
- random.rand
- random.random

In [49]:
```
# randint
np.random.randint(10)

```

Out[49]:
```
7
```

In [50]:
```
np.random.randint(10,100)
```

Out[50]:
```
65
```

In [44]:
```
np.random.randint(10,100,3)
```

Out[44]:
```
array([79, 69, 11])
```

In [57]:
```
np.random.randint(10,100,(10,4))
```

...

In [62]:
```
# randn

print(np.random.randn(10))
print()
np.random.randn(3,4)
```

```
[ 1.13537083 -1.84260707 -1.66554766  0.69522936 -1.96011022  2.7979229
 -1.81128875  0.56287244  1.52315046 -0.34478046]
```

Out[62]:
```
array([[ 1.38998332,  0.11493115, -0.65428065,  0.64679895],
       [ 1.44611102,  0.96187684, -0.09966883, -3.1356072 ],
       [ 1.85811942, -0.24073039,  1.19984556,  0.64018528]])
```

In [67]:
```
print(np.random.rand(3,5))
print("===========")
print(np.random.random((4,3,5)))
```

```
[[0.80457955 0.10681298 0.2649445  0.01094595 0.02000766]
 [0.42295306 0.13003126 0.278211   0.16960972 0.16439762]
 [0.53435898 0.02180105 0.50599773 0.56935337 0.42048227]]
===========
[[[0.81003791 0.79420003 0.26205463 0.3514356  0.06017977]
  [0.79615713 0.47875321 0.0093424  0.23350414 0.89916815]
  [0.64998803 0.48327112 0.78581189 0.1621409  0.25424691]]

 [[0.62737495 0.28381799 0.3517482  0.67361635 0.37299061]
  [0.57881205 0.51342993 0.7086484  0.3275092  0.86615042]
  [0.18133068 0.75152101 0.62669351 0.69496283 0.03199719]]

 [[0.27097496 0.65997293 0.48015462 0.45783783 0.64991772]
  [0.08902582 0.65677527 0.53405602 0.41950992 0.18000075]
  [0.76170932 0.46426152 0.53206331 0.69425062 0.71943471]]

 [[0.21437111 0.40065391 0.02682994 0.28596119 0.44163055]
  [0.91819103 0.97716797 0.86486285 0.110914   0.94313214]
  [0.10787365 0.96433408 0.52563291 0.65593471 0.12454711]]]
```

In [69]:
```
s =np.random.randint(10,100,(6,6))
s[1:4,2:5]
```

Out[69]:
```
array([[49, 70, 78, 41, 14, 19],
       [57, 63, 45, 18, 99, 86],
       [45, 89, 44, 38, 42, 31],
       [78, 29, 13, 39, 75, 48],
       [69, 43, 22, 72, 94, 59],
       [38, 23, 85, 74, 82, 91]])
```

Accessing and Slincing

In [70]:
```
s[2]
```

Out[70]:
```
array([45, 89, 44, 38, 42, 31])
```

In [71]:
```
s[2:4]  # 2,3
```

Out[71]:
```
array([[45, 89, 44, 38, 42, 31],
       [78, 29, 13, 39, 75, 48]])
```

In [72]:
```
s[1:5]
```

Out[72]:
```
array([[57, 63, 45, 18, 99, 86],
       [45, 89, 44, 38, 42, 31],
       [78, 29, 13, 39, 75, 48],
       [69, 43, 22, 72, 94, 59]])
```

In [73]:
```
s[1:5,2:6]

# 1,2,3,4 rows
# 2,3,4,5
```

Out[73]:
```
array([[45, 18, 99, 86],
       [44, 38, 42, 31],
       [13, 39, 75, 48],
       [22, 72, 94, 59]])
```

In [74]:
```
s[0:1,0:1]
```

Out[74]:
```
array([[49]])
```

In [75]:
```
s[2][3]
```

Out[75]:
```
38
```

In [78]:
```
print(s)
s[::2,::2]
#0,2,4,6,8
```

```
[[49 70 78 41 14 19]
 [57 63 45 18 99 86]
 [45 89 44 38 42 31]
 [78 29 13 39 75 48]
 [69 43 22 72 94 59]
 [38 23 85 74 82 91]]
```

Out[78]:
```
array([[49, 78, 14],
       [45, 44, 42],
       [69, 22, 94]])
```

```
### Stacking

]* hstack
* vstack
```

In [79]:
```
a = np.array([[1,2,3],[1,2,3]])
b = np.array([[1,2],[3,4]])
print(a)
print(b)
```

```
[[1 2 3]
 [1 2 3]]
[[1 2]
 [3 4]]
```

In [81]:
```
np.hstack((a,b))
```

Out[81]:
```
array([[1, 2, 3, 1, 2],
       [1, 2, 3, 3, 4]])
```

In [82]:
```
np.vstack((a,b))
```

...

In [92]:
```
import math
math.log2(10)
math.log2(20)
```

Out[92]:
```
4.321928094887363
```

In [84]:
```
np.log2([10,20,30])
```

Out[84]:
```
array([3.32192809, 4.32192809, 4.9068906 ])
```

In [86]:
```
np.tan(45)
```

Out[86]:
```
1.6197751905438615
```

In [91]:
```
np.sin([30,45,60])
```

Out[91]:
```
array([-0.98803162,  0.85090352, -0.30481062])
```

In [ ]:
```

```