

1. <https://teachablemachine.withgoogle.com/models/QGjRGStf/>

2. Numpy kutubxonasi orqali massiv yarating va massiv turlari haqida ma'lumot yozing . Array, arrange, ndim, shape, size, zeros, ones, dtype, astype, random, metodlari haqida ma'lumot yozing va har bir metodni ishlatish bo'yicha misollar ishlab natija oling.

Array: NumPy-da bir xil turdagi ma'lumotlar elementlarini saqlaydigan asosiy ma'lumotlar tuzilmasi. `arange(start, stop, step)`: teng intervalli qiymatlarni o'z ichiga olgan massiv yaratadi. `ndim`: massivdagi o'lchamlar (o'qlar) sonini qaytaradi. `shape`: Massiv o'lchamlarini ifodalovchi kortejni qaytaradi (masalan, 3x4 matritsa uchun (3, 4)). `size`: massivdagi elementlarning umumiy sonini qaytaradi. `zeros(shape, dtype=float)`: Nollar bilan to'ldirilgan massiv hosil qiladi. `ones(shape, dtype=float)`: Birlar bilan to'ldirilgan massivni yaratadi. `dtype`: massivdagi elementlarning ma'lumotlar turini qaytaradi. `astype(dtype)`: Massivdagi elementlarning ma'lumotlar turini o'zgartiradi. `random.randint(past, baland, o'lcham)`: low (shu jumladan) dan high (ekslyuziv) gacha bo'lgan tasodifiy butun sonlar massivini yaratadi .

```
import numpy as np
```

3. Numpy kutubxonasi orqali 2 o'lchovli va 3 o'lchovli massiv yarating, size, zeros, ones, dtype, astype metodlari bo'yicha misollar tayyorlang va ishlatib ko'rsating

```
ruyxat=[[8,9,6,7],[41,85,63,87]]
massiv_2=np.array(ruixat)
massiv_2 # 2 o'lchovli massiv yaratish

array([[ 8,  9,  6,  7],
       [41, 85, 63, 87]])

massiv_2.size #size metodi massiv elementlar sonini chiqarib beradi

8

ruyxat2=[[7,8,5,9],[10,12,14,16]], [[741,852,963,102],[321,654,987,102]]
massiv_3=np.array(ruixat2)
massiv_3 # 3 o'lchamli massiv yaratish

array([[[ 7,  8,  5,  9],
        [10, 12, 14, 16]],
       [[741, 852, 963, 102],
        [321, 654, 987, 102]]])

massiv_3.size #size metodi massiv elementlar sonini chiqarib beradi

16

massiv=np.zeros(shape=(4,6))
massiv # barcha elementlari 0 dan iborat bo'lgan massiv yaratib beradi

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.]])

massiv=np.ones(shape=(3,5))
massiv # barcha elementlari 1 dan iborat bo'lgan massiv yaratib beradi

array([[1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.]])

massiv.dtype # massiv elementlari turini aniqlovchi metod
dtype('float64')
massiv.astype #Massivdagi elementlarning ma'lumotlar turini o'zgartirad
```

https://colab.research.google.com/drive/1U_RTt09hO7ypM77MPCIFkPdvG_06QsOE#scrollTo=yGsdGYvUKW5D&line=1&uniqifier=1

4. Numpy kutubxonasi `arange` va `random.randint` metodi orqali elementlari 1 dan 200 gacha bo'lgan massiv yarating. Massiv elementlari orasidan 35 dan 105 gacha bo'lgan elementlarni 9 raqami bilan almashtiring va massivni 25 dan 50 gacha bo'lgan elementlari yig'indisi toping.

```
mm=np.random.randint(1, 200, size=(200))
mm # random rand int yordamida elementlari 35 105 oraliq'idagi sonlardan iborat massiv yaratib beraadi
```

```
array([ 12, 152,  94,  71, 149,  48, 150,  61,  16, 179,   3,  14, 168,
        78, 112,  22, 151,   5,  62, 199, 147,  74, 161,  34, 136, 139,
       133,  64, 133, 102,  84, 185,  17,  85, 181,   6,  19,  33, 143,
        19,  31,  95,  43, 179, 133, 173, 171, 162, 126,  90,  17,  92,
        89, 175,  57, 128, 107,  50,  41,  26,  26, 193, 121,  43,  30,
       157,   6, 128,  36, 142, 149,  95,  55,  85,  95,  83, 120, 124,
       192, 105, 156,  74, 169, 160, 190,  91, 153,  40,  13,  27, 157,
        25, 143,  62, 122, 181, 158,  41, 130, 188,  61, 155,  93, 174,
        34, 158, 194,  49,  88, 131, 146,  19,  33, 160, 187,  25,  33,
       155, 118,  94,  21, 102, 182, 103, 178, 135,  88,  93,  57,  71,
        55, 119,  26, 199,  44, 115, 128,  95,  48, 185,   6,  17,  42,
        91, 152, 101,  59, 166, 161,  71,  63,  38,  48,  65,  18, 104,
        61,  49,  49, 103, 167,  38,   2, 112, 120, 103, 184,  69,  78,
        41,  31,  43,  50, 146,  90,  11, 142, 107, 177,  50, 107,   4,
        32,   6,  83, 104,   2, 163, 103, 133, 178, 164,  59, 167,  33,
       151,  35, 134, 191,  72])
```

```
mm[35:105]=9
mm # 35 105 oraliqdagi indekslardagi elementlarni 9 bilan almashtirish
```

```
array([ 12, 152,  94,  71, 149,  48, 150,  61,  16, 179,   3,  14, 168,
        78, 112,  22, 151,   5,  62, 199, 147,  74, 161,  34, 136, 139,
       133,  64, 133, 102,  84, 185,  17,  85, 181,   9,   9,   9,   9,
         9,   9,   9,   9,   9,   9,   9,   9,   9,   9,   9,   9,
         9,   9,   9,   9,   9,   9,   9,   9,   9,   9,   9,   9,
         9,   9,   9,   9,   9,   9,   9,   9,   9,   9,   9,   9,
         9,   9,   9,   9,   9,   9,   9,   9,   9,   9,   9,   9,
         9, 158, 194,  49,  88, 131, 146,  19,  33, 160, 187,  25,  33,
       155, 118,  94,  21, 102, 182, 103, 178, 135,  88,  93,  57,  71,
        55, 119,  26, 199,  44, 115, 128,  95,  48, 185,   6,  17,  42,
        91, 152, 101,  59, 166, 161,  71,  63,  38,  48,  65,  18, 104,
        61,  49,  49, 103, 167,  38,   2, 112, 120, 103, 184,  69,  78,
        41,  31,  43,  50, 146,  90,  11, 142, 107, 177,  50, 107,   4,
        32,   6,  83, 104,   2, 163, 103, 133, 178, 164,  59, 167,  33,
       151,  35, 134, 191,  72])
```

```
mm[25:50].sum() # massivning 25 50 oraliqdagi indekslardagi elementlarni yig'indisi
```

```
1258
```

```
nn=np.arange(1,200,1)
nn # arange yordamida 1 dan 200 gacha bo'lgan elementlardan iborat bo'lgan massiv yaratdishi
```

```
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, 101, 102, 103, 104,
       105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117,
       118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130,
       131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143,
       144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156,
       157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169,
       170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182,
       183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195,
       196, 197, 198, 199])
```

```
nn[35:105]=9
nn # 35 105 oraliqdagi indekslardagi elementlarni 9 bilan almashtirishi
```

```
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,   9,   9,   9,   9,
         9,   9,   9,   9,   9,   9,   9,   9,   9,   9,   9,   9,
         9,   9,   9,   9,   9,   9,   9,   9,   9,   9,   9,   9,
         9,   9,   9,   9,   9,   9,   9,   9,   9,   9,   9,   9,
         9,   9,   9,   9,   9,   9,   9,   9,   9,   9,   9,   9,
         9, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117,
       118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130,
       131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143,
       144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156,
       157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169,
       170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182,
       183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195,
       196, 197, 198, 199])
```

https://colab.research.google.com/drive/1U_RTt09hO7ypM77MPCIFkPdvG_06QsOE#scrollTo=MBBCG1rPL2rS&line=1&uniqifier=1

5. Numpy kutubxonasi orqali massiv yarating, yaratgan massivingizni transporterlang, yaratgan massivingizda quyidagi funksiyalarni ishlatib: sqrt,square,exp,log, modf,sign, isnan

```
ss=np.random.randint(10, size=(3,4))
ss
array([[9, 2, 9, 2],
       [6, 7, 1, 8],
       [8, 9, 6, 4]])

ss.T # Massivlarni transporterlash
array([[9, 6, 8],
       [2, 7, 9],
       [9, 1, 6],
       [2, 8, 4]])

np.sqrt(ss) # massiv elementlarini kvadrat ildizlarini qaytaradi
array([[3.         , 1.41421356, 3.         , 1.41421356],
       [2.44948974, 2.64575131, 1.         , 2.82842712],
       [2.82842712, 3.         , 2.44948974, 2.         ]])

np.square(ss) # massiv elementlarini kvadratga ko'taradi
array([[81, 4, 81, 4],
       [36, 49, 1, 64],
       [64, 81, 36, 16]])

np.exp(ss) # massiv ning barcha elementlarini eksponentasini aniqlash
array([[8.10308393e+03, 7.38905610e+00, 8.10308393e+03, 7.38905610e+00],
       [4.03428793e+02, 1.09663316e+03, 2.71828183e+00, 2.98095799e+03],
       [2.98095799e+03, 8.10308393e+03, 4.03428793e+02, 5.45981500e+01]])

np.log(ss) # massiv ning barcha elementlarini logarifmini aniqlash (loge)
<ipython-input-21-55e82fe60c28>:1: RuntimeWarning: divide by zero encountered in log
np.log(ss) # massiv ning barcha elementlarini logarifmini aniqlash (loge)
array([[ -inf, -inf, -inf, -inf, -inf, -inf],
       [-inf, -inf, -inf, -inf, -inf, -inf],
       [-inf, -inf, -inf, -inf, -inf, -inf],
       [-inf, -inf, -inf, -inf, -inf, -inf]])

xx=np.random.randn(6)
xx # elementlari 6 ta ixtiyoriy haqiqiy sonlardan iborat bo'lgan massiv yaratadi
array([-0.76789131, 0.58434053, -0.92152009, 0.97360341, -0.94983684,
       -0.01552716])

m1, m2=np.modf(xx) # Modf massiv elementlarinig butun va qoldiq qismini chiqarib beradi
print(m2)

[-0.  0. -0.  0. -0. -0.]

m1, m2=np.modf(xx) # Modf massiv elementlarinig butun va qoldiq qismini chiqarib beradi
print(m2)

[-0.  0. -0.  0. -0. -0.]

np.isnan(xx) # massiv elementlari orasida bo'sh joy borligini aniqlaydi, bo'sh joy bo'lsa true bo'lmasa false qaytaradi
array([False, False, False, False, False, False])
```

https://colab.research.google.com/drive/1U_RTt09hO7ypM77MPCIFkPdvG_06QsOE#scrollTo=f5Kug3e7M2mm&line=1&uniqifier=1

