numpy

August 10, 2025

[2]:
```
import numpy as np
```

[2]:
```
a = np.array([[1, 2, 3],
              [4, 5, 6]])
print(a, a[1], a[1, 1], sep = ";\n")
```

```
[[1 2 3]
 [4 5 6]];[4 5 6];5
```

typing

[ ]:
```
from numpy.typing import NDArray

def func(a : NDArray) -> NDArray : return a
```

This section covers `np.zeros()`, `np.ones()`, `np.empty()`, `np.arange()`, `np.linspace()`

[7]:
```
a = np.zeros(2)
b = np.ones((2, 2))
c = np.eye(3)
d = np.arange(4)
e = np.arange(8, 2, -2)
print(a, b, c, d, e, sep = ";\n")
```

```
[0. 0.];
[[1. 1.]
 [1. 1.]];
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]];
[0 1 2 3];
[8 6 4]
```

[ ]:
```
a = np.linspace(0, 1, 4)
b = np.linspace(0, 1, num = 4, endpoint=False)
c, step = np.linspace(0, 1, num = 4, retstep=True)
print(a, b, sep = ";")
print(c, step)
```

This section covers the `ndim`, `shape`, `size`, and `dtype` attributes of an array.

[ ]:
```
print(a.ndim, a.shape, a.size, a.dtype, sep = ";")
```

[5]:
```
a = np.reshape(np.arange(24), (2, 3, 4))
print(a)
```

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

 [[12 13 14 15]
  [16 17 18 19]
  [20 21 22 23]]]
```

Boolean Indexing

[9]:
```
num = np.reshape(np.arange(12), (3, 4))
mask = (num>=5) | (num%2==0)
print(mask)
res = num[mask]
print(res)
```

```
[[ True False  True False]
 [ True  True  True  True]
 [ True  True  True  True]]
[ 0  2  4  5  6  7  8  9 10 11]
```

[12]:
```
num = np.reshape(np.arange(12), (3, 4))
print(num[(num>=5) | (num%2==0)])
```

```
[ 0  2  4  5  6  7  8  9 10 11]
```

[8]:
```
num = np.random.randn(5)              #
print(num)
num = np.random.uniform(2, 3, 5)      #  5 (2,3)
print(num)
```

```
[ 0.65690067 -0.37845734  0.01391865 -0.98910409 -0.91937114]
[2.95215298 2.79381034 2.63837973 2.74441253 2.19779344]
```

[18]:
```
num = np.random.randint(0, 10)         #    [0,10)
print(num)
```

```
2
```

[19]:
```
num = np.random.randint(0, 100, 5)     #  5
print(num)
```

```
[42 84 74 35 35]
```

[4]:
```
a = np.random.rand(2, 3)                #  2*3
print(a)
```

```
[[0.84459842 0.44269828 0.24859259]
 [0.32864026 0.96146871 0.82595084]]
```