Python 과학 프로그래밍 기초
7. NumPy (4)

한림대학교 소프트웨어융합대학
박섭형

2021년 1학기

NumPy 다차원 배열의 속성 변경

[1]:
```
import numpy as np
n = np.arange(5)
print(n.dtype)
```

```
int32
```

[2]:
```
n[0] = 5
print(n, n.dtype)
```

```
[5 1 2 3 4] int32
```

[3]:
```
n[2] = 10.8
print(n, n.dtype)
```

```
[ 5  1 10  3  4] int32
```

[4]:
```
m = n.astype(np.int64)
print(m, m.dtype)
print(n, n.dtype)
```

```
[ 5  1 10  3  4] int64
[ 5  1 10  3  4] int32
```

[5]:
```python
f = n.astype(np.float64)
print(f, f.dtype)
```

```
[ 5.  1. 10.  3.  4.] float64
```

[6]:
```python
c = n.astype(np.complex128)
print(c, c.dtype)
```

```
[ 5.+0.j  1.+0.j 10.+0.j  3.+0.j  4.+0.j] complex128
```

[7]:
```python
print(n.dtype)
```

```
int32
```

[8]:
```python
n.flags.writeable = False
n[0] = 10
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-8-e12f5b469656> in <module>
      1 n.flags.writeable = False
----> 2 n[0] = 10

ValueError: assignment destination is read-only
```

NumPy 다차원 배열 변수와 할당 연산자

[9]:
```python
import numpy as np
x = np.arange(5)
print(x)
print(id(x))
x += 1
print(x)
print(id(x))
```

```
[0 1 2 3 4]
2352194966912
[1 2 3 4 5]
2352194966912
```

[10]:
```python
x = np.arange(5)
print(x)
print(id(x))
x = x + 1
print(x)
print(id(x))
```

```
[0 1 2 3 4]
2352194911328
[1 2 3 4 5]
2352194911568
```

[11]:
```python
import numpy as np
def add1(x):
    x += 1

a = np.arange(5)
print(a)
add1(a)
print(a)
```

```
[0 1 2 3 4]
[1 2 3 4 5]
```

[12]:
```python
import numpy as np
def add2(x):
    x = x + 1

a = np.arange(5)
print(a)
add2(a)
```

```python
print(a)
```

```
[0 1 2 3 4]
[0 1 2 3 4]
```

[13]:
```python
import numpy as np
def mult1(x):
    x *= 2

a = np.arange(5)
print(a)
mult1(a)
print(a)
```

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

[14]:
```python
import numpy as np
def mult2(x):
    x = x * 2

a = np.arange(5)
print(a)
mult2(a)
print(a)
```

```
[0 1 2 3 4]
[0 1 2 3 4]
```

Broadcasting

- Shape이 다른 두 개의 ndarray들을 이용해서 산술 연산을 할 때 numpy가 ndarrays를 처리하는 방법

Shape이 같은 두 ndarray의 산술 연산

- 원소끼리 연산이 이루어짐

```
[15]: a = np.array([1, 2, 3])
      b = np.array([10, 20, 30])
      a + b
```

```
[15]: array([11, 22, 33])
```

```
[16]: a * b
```

```
[16]: array([10, 40, 90])
```

```
[17]: a = np.arange(12).reshape(3,4)
      a
```

```
[17]: array([[ 0,  1,  2,  3],
             [ 4,  5,  6,  7],
             [ 8,  9, 10, 11]])
```

```
[18]: b = np.arange(10, 22).reshape(3,4)
      b
```

```
[18]: array([[10, 11, 12, 13],
             [14, 15, 16, 17],
             [18, 19, 20, 21]])
```

```
[19]: a + b
```

```
[19]: array([[10, 12, 14, 16],
             [18, 20, 22, 24],
             [26, 28, 30, 32]])
```

```
[20]: a = np.array([1, 2, 3])
      b = np.array([1, 2])
      a + b
```

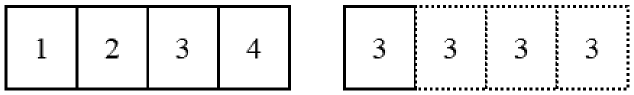
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-20-f5a67fd89fb2> in <module>
      1 a = np.array([1, 2, 3])
      2 b = np.array([1, 2])
----> 3 a + b
```

```
ValueError: operands could not be broadcast together with shapes (3,) (2,)
```

ndarray와 상수의 산술 연산

[21]:
```python
x = np.arange(1,5)
print(x)
print(x + 3)
```

```
[1 2 3 4]
[4 5 6 7]
```



[22]:
```python
print(x * 3)
```

```
[ 3  6  9 12]
```

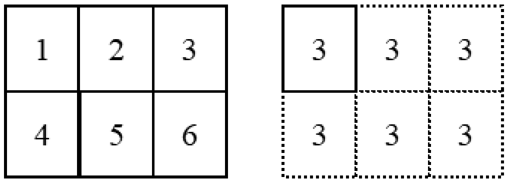
[23]:
```python
print(x / 3)
```

```
[0.33333333 0.66666667 1.         1.33333333]
```

[24]:
```python
x = np.arange(1,7).reshape(2,3)
print(x + 3)
```

```
[[4 5 6]
 [7 8 9]]
```

[25]:
```python
print(3 * x)
```



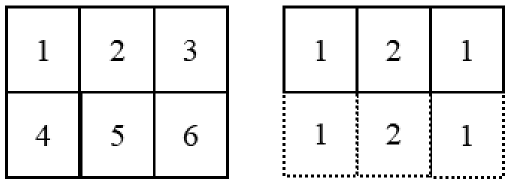
```
[[ 3  6  9]
 [12 15 18]]
```

Shape이 서로 다른 두 ndarray 사이의 산술 연산

axis의 원소의 갯수가 같은 두 ndarray 사이의 산술 연산

[26]:
```python
a = np.arange(1,7).reshape(2,3)
b = np.array([[1, 2, 1]])
print(a + b)
```

```
[[2 4 4]
 [5 7 7]]
```



[27]:
```python
a = np.arange(1,7).reshape(2,3)
b = np.array([1, 2, 1])
print(a + b)
```

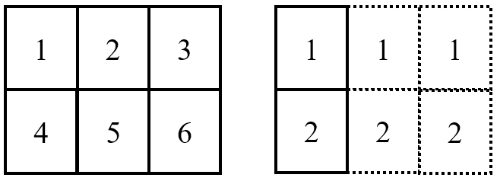
```
[[2 4 4]
 [5 7 7]]
```

[28]:
```python
a = np.arange(1,7).reshape(2,3)
b = np.array([1, 2])
print(a + b)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-28-5a11c48ce658> in <module>
      1 a = np.arange(1,7).reshape(2,3)
      2 b = np.array([1, 2])
----> 3 print(a + b)

ValueError: operands could not be broadcast together with shapes (2,3) (2,)
```

[29]:
```python
a = np.arange(1,7).reshape(2,3)
b = np.array([[1], [2]])
print(a + b)
```

```
[[2 3 4]
 [6 7 8]]
```



[30]:
```python
a = np.arange(1,7).reshape(2,3)
b = np.array([[1, 1], [2, 2]])
print(a + b)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-30-a8b431da9202> in <module>
      1 a = np.arange(1,7).reshape(2,3)
      2 b = np.array([[1, 1], [2, 2]])
----> 3 print(a + b)

ValueError: operands could not be broadcast together with shapes (2,3) (2,2)
```

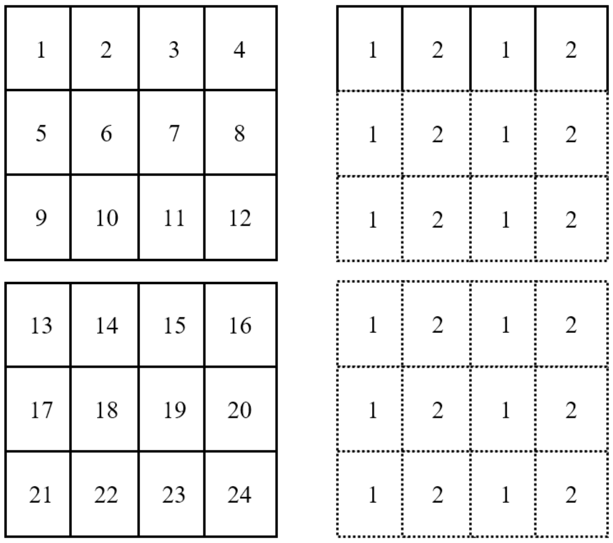
[31]:
```python
a = np.arange(1,9).reshape(2,4)
b = np.array([[1, 1], [2, 2]])
print(a + b)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-31-7f68e1658272> in <module>
      1 a = np.arange(1,9).reshape(2,4)
      2 b = np.array([[1, 1], [2, 2]])
----> 3 print(a + b)
```

```
ValueError: operands could not be broadcast together with shapes (2,4) (2,2)
```

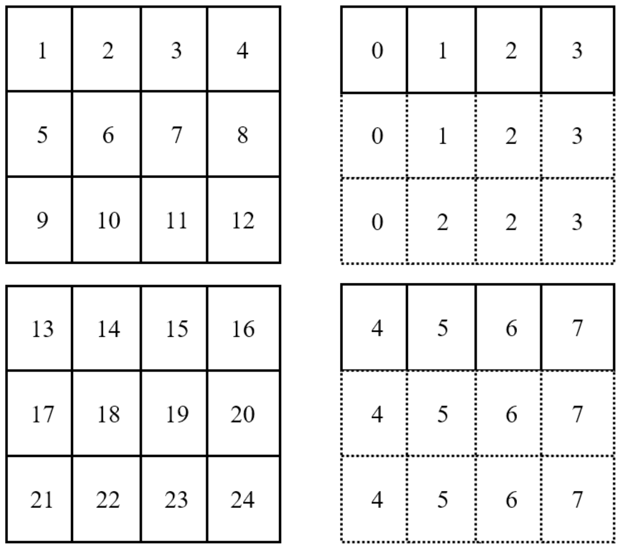
[32]:
```
a = np.arange(1,25).reshape(2,3,4)
b = np.array([1, 2, 1, 2])
print(a + b)
```

```
[[[ 2  4  4  6]
  [ 6  8  8 10]
  [10 12 12 14]]

 [[14 16 16 18]
  [18 20 20 22]
  [22 24 24 26]]]
```



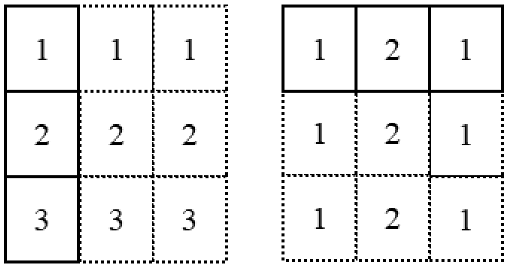
[33]:
```
a = np.arange(1, 25).reshape(2,3,4)
b = np.arange(8).reshape(2,1,4)
print(a)
print(b)
print(a + b)
```

```
[[[ 1  2  3  4]
  [ 5  6  7  8]
  [ 9 10 11 12]]

 [[13 14 15 16]
  [17 18 19 20]
  [21 22 23 24]]]
[[[0 1 2 3]]

 [[4 5 6 7]]]
[[[ 1  3  5  7]
  [ 5  7  9 11]
  [ 9 11 13 15]]

 [[17 19 21 23]
  [21 23 25 27]
  [25 27 29 31]]]
```



[34]:
```python
a = np.array([1,2,3]).reshape(3,1)
b = np.array([1,2,1]).reshape(1,3)
print(a)
print(b)
print(a + b)
```

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



1 1 1
2 2 2
3 3 3
1 2 1
1 2 1
1 2 1