10分でわかる

# NumPy配列のスライス

高 慎之助

# スライスのおさらい

In [1]:

```
import numpy as np
# 以下の配列を作成します
x = np.arange(1, 11)
x
```

Out[1]:

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

In [2]:

```
x[2]
```

スライスとは，

**配列から**[]を使って**任意の値を抽出する** 操作

スクエアブラケット

スライスの操作量

**NumPy配列** > Pythonのシーケンス

**リスト**, **タプル** など


## 今回話すこと

全8章のうち，

**初心者向けに書かれた 1,2,6章**

3, 4, 5, 7章は実務上必要になったら読むもの

※高度なスライスについて詳しく知りたい人は読み直し推奨

https://www.headboost.jp/numpy-array-slice/#index_id6

# 目次

- NumPy入門
  - 8.1NumPyを使う準備
  - 8.2多次元配列を定義する
  - 8.3多次元配列の要素を選択する
  - 8.4ndarrayのデータ型
  - 8.5多次元配列を用いた計算
  - 8.6ブロードキャスト
  - 8.7行列積
  - 8.8基本的な統計量の求め方
  - 8.9NumPyを用いた重回帰分析
- NumPy配列のスライス
  - 基本操作
    - 1次元配列のスライス
    - 多次元配列のスライス
    - 補足
  - 高度な操作
    - 配列によるスライス
    - 配列によるスライス　+基本のスライス
  - newaxisオブジェクト

# 基本操作　◆スライスの書き方

スライス対象のNumPy配列が

**1次元**

Pythonシーケンスと同じ

**2次元以上**

NumPy特有の書き方

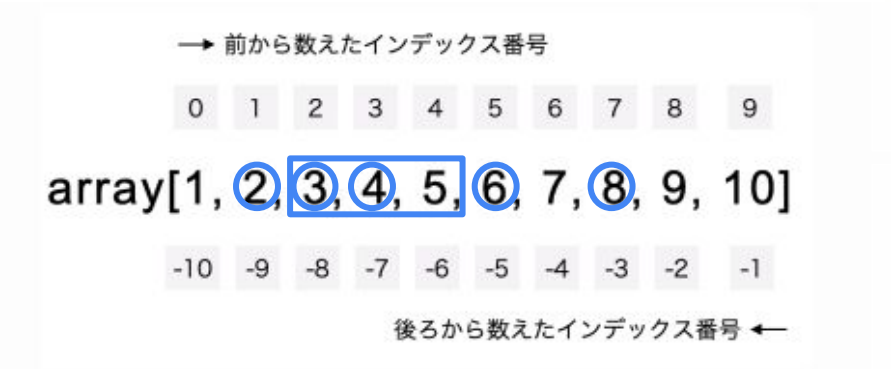
# 基本操作　◆1次元配列のスライス

In [1]:

```
import numpy as np
# 以下の配列を作成します
x = np.arange(1, 11)
x
```

Out[1]:

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




1要素のスライス

In [2]:

```
x[2]
```

複数要素のスライス

In [2]:

```
# インデックス番号2以上5未満の要素を抽出
x[2:5]
```

In [5]:

```
# インデックス番号1以上9未満の要素を2つ間隔で抽出
x[1:9:2]
```

# 基本操作　◆スライスの書き方

スライス対象のNumPy配列が

1次元

Pythonシーケンスと同じ

**2次元以上**

NumPy特有の書き方

# 基本操作　◆2次元配列のスライス

In [1]:

```
import numpy as np
x = np.arange(1, 11).reshape(2, 5)
x
```




原則

**x[2次元軸のインデックス，　1次元軸のインデックス ]**

In [2]:

```
# 2次元配列から2次元軸（行）のインデックス番号0、
# 1次元軸（列）のインデックス番号3をスライス
x[0, 3]
```

# 基本操作　◆2次元配列のスライス

In [1]:

```
import numpy as np
x = np.arange(35).reshape(5, 7)
x
```


In [5]:

```
# 2次元軸（行）のインデックス番号1以上4未満をスライス
# 続いて、1次元軸（列）のインデックス番号3をスライス
x[1:4, 3]
```

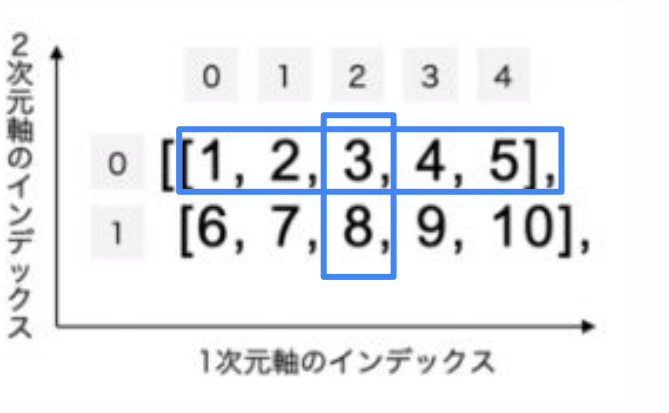
In [6]:

```
# 2次元軸（行）のインデックス番号1以上4未満をスライス
# 続いて、1次元軸（列）のインデックス番号3以上6未満をスライス
x[1:4, 3:6]
```

# 基本操作　◆2次元配列のスライス　◇特定の軸のみ指定

In [1]:

```
import numpy as np
x = np.arange(1, 11).reshape(2, 5)
x
```




2次元軸のみ指定

In [3]:

```
# 2次元軸のインデックスのみを指定した場合
x[0]
```

1次元軸のみ指定

In [4]:

```
x[:, 2]
```

# 基本操作　◆2次元配列のスライス　◇特定の軸のみ指定

In [1]:

```
import numpy as np
x = np.arange(35).reshape(5, 7)
x
```


2次元軸のみ指定

In [3]:

```
# 2次元軸（行）のインデックス番号1以上4未満をスライス
x[1:4]
```

In [4]:

```
# 2次元軸（行）のインデックス番号1以上を2つ間隔でスライス
x[1::2]  # x[1:5:2]と同じ
```

# 基本操作　◆2次元配列のスライス　◇特定の軸のみ指定

In [1]:

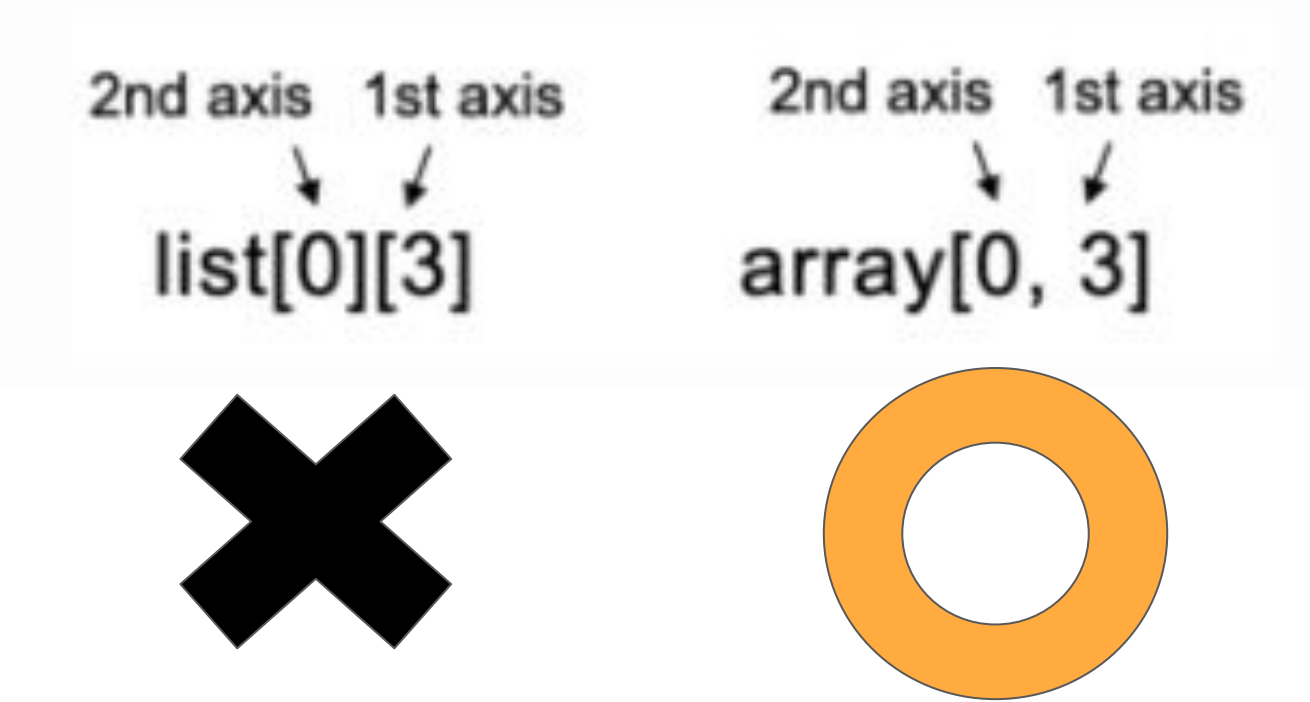
```
import numpy as np
x = np.arange(35).reshape(5, 7)
x
```


1次元軸のみ指定

In [8]:

```
# 1次元軸（列）のインデックス番号3以上6未満をスライス
x[:, 3:6]
```

# 基本操作　◆注意点



# 基本操作　◆補足

## 配列に対する値の代入

In [1]:

```
import numpy as np
x = np.arange(10)
x
```




In [2]:

```
x[2:7] = 10
x
```

## エリプシス (...)を使った次元軸の省略

◆例：4次元配列の1, 4次元軸のみ指定

```
[1]  1 import numpy as np
     2 z = np.arange(16).reshape(2,2,2,2)

     1 z

array([[[[ 0,  1],
         [ 2,  3]],

        [[ 4,  5],
         [ 6,  7]]],


       [[[ 8,  9],
         [10, 11]],

        [[12, 13],
         [14, 15]]]])
```

```
[7]  1 z[1,:,:,1]

array([[ 9, 11],
       [13, 15]])

     1 z[1,...,1]

array([[ 9, 11],
       [13, 15]])
```

【次元軸省略】上コード：コロン
下コード：エリプシス

# 高度な操作　◆配列によるスライス

**整数** or **bool値**

| 配列 | を | 配列 | でスライス |
|---|---|---|---|
| オリジナル配列 | | インデックス配列 | |

**インデックス配列の要素**
によってスライス結果が変わる

インデックス配列の要素が整数　＋　オリジナル配列が 1次元配列

In [1]:

```
import numpy as np
# オリジナル配列の作成
x = np.arange(10, 1, -1)
x
```

Out[1]:

```
array([10, 9, 8, 7, 6, 5, 4, 3, 2])
```

In [2]:

```
# スライス配列を使ってオリジナル配列をスライス
x[np.array([3, 3, 1, 8])]
```

Out[2]:

```
array([7, 7, 9, 2])
```

# 高度な操作　◆配列によるスライス ＋ 基本のスライス

In [1]:

```
import numpy as np
x = np.arange(35).reshape(5,7)
```


```
x[np.array([0, 2, 4]), 1:3]
```

# おまけ　◆newaxisオブジェクトについて

配列に対して要素数1の新しい次元を付与することが出来る技術

行列の演算で，配列の大きさを揃えたいときに便利

In [1]:

```
import numpy as np
x = np.arange(35).reshape(5,7)
x
```

Out[1]:

```
array([[ 0,  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]])
```

In [2]:

```
x.shape
```

Out[2]:

```
(5, 7)
```

In [3]:　**2次元軸に追加**

```
x[:,np.newaxis,:].shape
```

Out[3]:

```
(5, 1, 7)
```

# まとめ

- NumPy配列には，**様々なスライステクニック** がある
- スライス対象のNumPy配列が多次元

  **→配列が代入された変数名 [n次元軸，n - 1次元軸，・・・，1次元軸](n >= 2)**

  ex. `x[0, 3]`

  2次元軸のインデックス 0,
  1次元軸のインデックス 3

  `x[1:4, 3:6]`

  2次元軸のインデックス 1以上4未満,
  1次元軸のインデックス 3以上6未満

  **でスライス**

- インデックス配列の要素が整数 or boolで，スライスの結果が変わる

  ※高度なスライスについて詳しく知りたい人は，記事を読んで見てください