

Numpy

Numpy

- Numpy とは Python で科学計算処理をするためのパッケージ
- データサイエンスパッケージの基盤

パッケージの利用

- `import` による利用：numpyパッケージの全機能を利用可能

```
import numpy # numpyパッケージを numpy という名前で読み込み
numpy.shape(A) # numpyパッケージのshape関数を利用
```

```
import numpy as np # numpy パッケージを np という別名で読み込み
np.shape(A) # numpyパッケージのshape関数を利用
```

- `from ... import ...` による利用：numpyパッケージの一部の機能を利用可能

```
from numpy import shape, sum # numpy パッケージの shape 関数と sum 関数を読み込み
shape(A) # numpyパッケージのshape関数を利用
```

型

- Pythonの基本型: bool, int, float, str
- Numpyの基本型:

型	numpy
論理値	bool
8, 16, 32, 64ビット符号付き整数	int8, int16, int32, int64
4, 8バイトの単精度浮動小数点実数	float32, float64

- 利用方法

```
from numpy import shape, int8, int16, int32, int64, float32, float64

a = int8(4) # import numpy as np とすると np.int8(4)
b = int16(5)
c = int32(10)
d = int64(10)
x = float32(1)
y = float64(2)
```

変数a, b, c, d, x, y はそれぞれ int8, int16, int32, int64, float32, float64 型

型変換

- 暗黙の型変換
 - int8 と floatの和 ➡ float64

```
a = int8(4)
b = float64(1.0)
a = a + b # int8とfloat64の和の結果はfloat64
```

- ルール：情報が落ちない様により表現範囲が広い型に合わせる

ビルトイン関数

- 数学で利用する関数があらかじめNumpyで定義
 - $\exp(x)$: 指数関数 (numpy)
 - $\log(x)$: 自然対数 (numpy)
 - $\sin(x)$, $\cos(x)$, $\tan(x)$: 三角関数 (numpy)
 - $\text{abs}(x)$: 絶対値 (標準, numpy)
 - $\text{sqrt}(x)$: 累乗根 (numpy)
 - $\text{min}(x, y, z, \dots)$, $\text{max}(x, y, z, \dots)$: 最大値, 最小値 (標準, numpy: 引数の挙動が少し違うので注意)

numpy配列

- リストより効率的にベクトル・行列を格納
- numpyでのベクトル・行列演算に利用
- リストと異なる点
 - コンストラクタ（numpy.array関数）で生成
 - 同じ型の要素しか格納できない
 - 二次元配列表現が可能
 - 行または列へのアクセスが可能

numpy配列 (ndarray) の作成

```
import numpy as np

x = np.array([1, 2, 3]) # リストからnumpy配列を生成する
print(type(x))
## 出力 <class 'numpy.ndarray'>

y = [[1, 2, 3], [4, 5, 6], [7, 8, 9], [-3, -1, -1]]
A = np.array(y) # リストのリストから二次元配列を生成する
print(type(A))
## 出力 <class 'numpy.ndarray'>
```

要素アクセスと代入

```
x[0] # 0 から始まる添字でアクセスできる
x[0] = 5 # 書き換え（代入）も可能
x
A[1,2] # 2 行 3 列目の要素にアクセス
# alist = [[1, 2, 3], [4, 5, 6], [7, 8, 9], [-3, -1, -1]]
# alist[1,2] # リストのリストは上記のようなアクセスはできない
A[:,0] # 1列目の要素を取り出す
A[:,0] = [5,6,7,8] # 列の要素をまとめて書き換え可能
```