

## ● 科学技術計算と言えばNumPy

NumPy<sup>①</sup>は、Pythonで科学技術計算を行うためのライブラリです。量子コンピューティング、統計計算、信号処理、画像処理、グラフとネットワーク、天文学のプロセス、認知心理学、バイオ・インフォマティクス、ベイズ推定、数学的分析、シミュレーション・モデリング、多変量解析、地理的处理、インタラクティブ・コンピューティングなどに利用されます。

データ・サイエンス分野で抽出、変換、読み込み、探索的分析、モデル化と評価、ダッシュボード<sup>注1</sup>でのレポートなどのワークフローに活用されています。

数値演算がPython標準の関数やライブラリよりも高速に処理できるので、科学技術計算を行わない場合

注1：各種データをグラフィカルに表示し、1つの画面にまとめたものがダッシュボードです。データ可視化アプリやウェブ・アプリなどでダッシュボードが使われます。PythonではNumPyでデータの収集・集計して、Dashライブラリでダッシュボード付きのウェブ・アプリを作る例があります。

もメリットがあります。

特に配列の要素の並び替え、追加、削除などは、処理時間を短縮できるので、ラズベリー・パイのようにコンピュータ性能が低いデバイスにも向いています。

## ● 最新版のインストール方法

NumPyは、Raspberry Pi OSに標準インストールされています。

もし最新版をインストールする場合には、以下のコマンドを実行してインストールが可能です。

```
# sudo apt-get update
# sudo apt-get -y upgrade
# sudo apt-get install -y python3-numpy
```

apt-getコマンドでインストールされるのは、必ずしもリリース済みNumPyの最新版ではなく、Raspberry Pi OSのリポジトリに登録されている最新版になります。

## 9-1 NumPy配列を作るarray関数

### ● 標準ライブラリより処理が速い

Pythonには同種類の複数データを扱うために、C言語などの配列に似たデータ構造としてリストがあります。リストは利便性は高いのですが、高速な処理には不向きなため、NumPyでは異なるデータ構造を利用できるようになっています。

このデータ構造(配列)を使うと、NumPyライブラリの機能を使って科学技術計算を高速に処理できるようになります。

以下で標準のリストとNumPy配列の違いを説明します。

#### ▶ 標準のリスト

Pythonに組み込まれているlist関数は、外部ライブラリをimportしなくても使え、1つのリストの要素として異なる型を格納することもできます。list関数で作るリストによって、多次元配列を表現することも可能ですが、狭義の配列<sup>注2</sup>とは異なる場合があります。

配列ライクな簡単な処理を行うのであればlist関数で作るリストで十分な場合が多いです。この場合、

文字列、整数、小数、ブール値を1つのリストに格納できます。

#### ▶ NumPy配列

NumPy配列は数値を扱うことに最適化されている配列です。同じ型のオブジェクトしか格納できません。その代わりほとんどの場合、list関数で作ったリストよりも処理が速いです。

数値以外のブール型、ユニコード文字列、Pythonオブジェクト型も扱えます。

Pythonで標準のリストを作成したあとで、NumPyのarray関数を使うことでNumPy配列に変換できます。

これ以外にも、NumPy配列の作成には、arange関数、ones関数、zeros関数などが使えます。

### ● PythonのオブジェクトをNumPy配列に変換するarray関数の書式

array関数の書式は次の通りです。

```
numpy.array(object, dtype=None, *, copy=True, order='K', subok=False, ndmin=0, like=None)
```

### ● array関数の引数

よく使うarray関数の引数を次に示します。

#### ▶ object

NumPy配列に変換するリスト、タプルなどのシーケンスを代入します。

注2：狭義の配列とは、コンピュータ・プログラムにおける配列という意味です。これは同一型のデータ(オブジェクト)をメモリ上に1列に間隔を空けずに並べた構造です。しかし、list関数では異なる型のデータが混在して1列に並べられた構造も許されるため、狭義の配列と同一とは認められず、このようなデータ構造をリストと呼んでいます。