

データ抽出や並び替えにはPandasライブラリがよく使われています。Pandas<sup>(1)</sup>は、処理が高速で使いやすいデータ分析および操作ツールです。Pythonによるデータ分析やデータ操作の必須ライブラリですので使い方を覚えておいて損はないはずです。

Pandasが特に利用される分野として、センサ・データ活用、データ・サイエンス、科学技術計算、機械学習の前処理やデータ・クレンジングなどがあります。

リレーショナルまたはラベル付きのデータを簡単かつ直感的に操作できます。データの分析や操作の機能は数十種類あり、機能をグループに大別するとオブジェクト生成、データ表示、データ選択、欠損処理、データ操作、データ結合、グルーピング、再成形、時系列、カテゴリズ、プロット、データ入出力などがあります。

これらの機能を使ってPythonプログラムでデータ構造をチェックし、データの前処理やデータ抽出を行います。

## ● 準備

### ▶ インストール

Raspberry Pi OSにはPandasが標準インストールされていないので、下記のコマンドを実行してインストールします。

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

PandasはNumPyの機能を利用しており、依存関係にあります。動作にはNumPyのインストールが必要ですが、Raspberry Pi OSには標準インストールされています。

### ▶ ライブラリの呼び出し

プログラムで利用するときには、あらかじめimport文でライブラリを読み込んでおく必要があります。

ソースコードの冒頭でNumPyライブラリを読み込んだ後にPandasライブラリを読み込みます。

## 11-1 データ構造の変換を行う DataFrame 関数

### ● DataFrame関数の利用目的

表形式のデータ構造を持つデータフレームを扱う関数です。データフレームを作成したり、他のデータ形式をデータフレームに変換したりするのに使われます。

Pandasでは、基本的にデータフレームを使ってデータの分析や、操作、演算を行いますので、Pandasライブラリを活用するために必須の関数です。

データフレームの他にも、Series関数で作られる1次元配列にデータを格納して、分析や操作を行えます。

一般的には1次元配列よりもデータフレームの方がよく使われると思いますので、ここではDataFrame関数を解説します。

### ● DataFrame関数の書式

DataFrame関数の書式は次です。

```
DataFrame(data=None, index=None,
columns=None, dtype=None, copy=
False)
```

DataFrame関数の引数に指定できる要素は次の通りです。

#### • data

変換するデータを代入します。データ構造には、NumPy配列のndarray、イテラブル(range, リスト, タプル, 集合, 辞書, 文字列), データフレームを使用できます。

#### • index

入力データ(引数data)に対してインデックスを指定します。

#### • columns

列のラベルを指定します。

#### • dtype

データ・タイプを指定します。

#### • copy

データ入力時のデータのコピー可否を指定します。

DataFrame関数の出力データはデータ・フレームに変換されます。

### ● 実行例

ここではDataFrame関数を使ってデータを辞書からデータ・フレームに変換します。

変換後のデータ、列ごとのデータ・タイプ、列のラベルを表示します。

#### ▶ 使用例

図1の1行目のimport文でPandasライブラリを読み込んで、pdの別名を付けて呼び出せるようにします。

2～6行目で名前、体重、身長からなる辞書を作成して変数dataに代入します。

7行目はDataFrame関数に変数dataを渡し、オブジェクトをデータフレームに変換して変数dfに格納します。