

5. 複数データの扱い

□ 概要

- Pythonでの複数データの扱い方について学びます。

□ 学習内容

- 5.1 リストの基本操作 宣言、追加、挿入、削除
- 5.2 リストの繰り返し
- 5.3 リスト内包表記
- 5.4 リストのソート
- 5.5 リストのインデックス
- 5.6 タプルの基本操作
- 5.7 辞書
- 5.8 辞書の基本操作
- 5.9 辞書の繰り返し
- 5.10 セットの基本操作
- 5.11 演習問題（別紙）



5.1 リストの基本操作 宣言、追加、挿入、削除

□ リストの宣言

- リストを使用すると、複数の値をまとめて管理することができます。
- リストを使用するには、[]で括った中に、カンマ(,)区切りで次のように宣言します。

変数名 = [要素1, 要素2・・・]



5.1 リストの基本操作 宣言、追加、挿入、削除

□ リストの追加

- 作成したリストに新しい要素を追加する場合は、リスト型の「append」メソッドを使用します。
- 新しい要素は、リストの最後に追加されます。

リスト.append(追加する要素)

5.1 リストの基本操作 宣言、追加、挿入、削除

□ リストの挿入

- 作成したリストに新しい要素を挿入する場合は、リスト型の「insert」メソッドを使用します。
- 1番目の引数に、要素を挿入するインデックス番号を指定し、2番目の引数に挿入する要素を指定します。

リスト.insert(挿入するインデックス番号, 要素)

5.1 リストの基本操作 宣言、追加、挿入、削除

□ リストの削除

– リストから要素を削除する際には、いくつかの方法があります。

1. 「del」文を使用する。
2. リスト型の「pop」メソッドを使用する。
3. リスト型の「remove」メソッドを使用する。

5.2 リストの繰り返し

- リストに対して繰り返し処理を行う一般的な方法としては、for文を使用します。

```
リスト = [要素1, 要素2...]
```

```
for [変数] in [リスト]:
```

```
    [実行する処理]
```

5.3 リスト内包表記

□ 内包表記とは

- リストなどのオブジェクトのループ処理を簡単に記述できる仕組みのこと。リストを作成する際などに使用します。

リスト = [式 for 変数名 in オブジェクト]



5.4 リストのソート

- リストの要素を昇順でソートさせたい場合は、`sort`関数を使用します。
- リストの要素が文字列の場合は、文字コードの昇順、リストの要素が数値であれば数値を小さい順からソートします。

ソートするリスト.`sort()`

5.5 リストのインデックス

- リストの要素を取得するには、単純に「リスト [インデックス番号]」で指定する方法がありますが、範囲を指定して要素を取得することもできます。

リスト[開始インデックス:終了インデックス:ステップ数]

5.6 タプルの基本操作

- リストと同じように、複数の要素を扱うことができるのがタプルです。
- タプルはリストと異なり、要素の変更ができません。
- そのため、変更したくない定数などを定義する際に使用されます。

変数名 = (要素1, 要素2・・・)

5.7 辞書

- Pythonには、キーと値を紐付ける辞書型 (dict型) というオブジェクトが存在します。
- 一つの要素はキーと値の組み合わせで構成されています。
- キーと値はコロンで区切り、データはカンマで区切ります。

```
{キー:値,...}
```

5.8 辞書の基本操作

□ 辞書の要素の取得

- 辞書型には、リストと違い要素の順番はありません。(3.6まで)
- インデックスを指定して要素を取得することはできません。
- 要素を取得するには、要素に割り当てたキーを指定します。

辞書[キー]

5.8 辞書の基本操作

□ 辞書への要素の追加

- 辞書へ新しい要素を追加する場合は、追加する辞書オブジェクトに存在していないキーを指定し、追加する値を指定します。

辞書[新しいキー] = 追加する値

setdefaultメソッドを使用して要素を追加することもできます。



5.8 辞書の基本操作

□ 辞書の削除

– 辞書から要素を削除する際には、いくつかの方法があります。

1. 「del」文を使用する。
2. 辞書型の「pop」メソッドを使用する。
3. 辞書型の「clear」メソッドを使用する。

5.9 辞書の繰り返し

- 辞書をfor文を使用して繰り返し処理を行うと、各要素のキーが取得できます。

```
辞書 = {キー:値, ...}
```

```
for [変数] in [辞書]:
```

```
    [実行する処理]
```

- 要素の値を取得したい場合、またはキーと値の両方を取得したい場合は、valuesメソッドとitemsメソッドを使用します。

5.10 セットの基本操作

- セットもリストと同様に、複数の要素を扱うことができます。
- セットはリストと異なり、要素の重複や順番の保持はされません。
- リストやタプルからセットを生成することができます。

```
変数名 = set([要素1, 要素2...])
```

```
変数名 = set((要素1, 要素2...))
```


5.10 セットの基本操作

□ セットへの要素の追加

- セットへ要素を追加するには、addメソッドを使用します。

セット.add(追加する要素)

□ セットの要素の削除

- セットから要素を削除するには、いくつかの方法があります。
 1. セット型の「remove」メソッドを使用する。
 2. セット型の「clear」メソッドを使用する。

5.11 演習問題（別紙）

□ 演習問題を解いてみましょう。

