

3章 データの扱い方を理解しよう

3章 データの扱い方を理解しよう

| Pythonで使われるデータの種類や簡単な計算などを学びます

本章の目標

- データ型の種類と特徴を理解する
- 数値の計算方法を学ぶ
- 文字列の結合方法を学ぶ

3章 データ型とは

| データ型とは、プログラムが扱うデータの種類を表すものです

- データには「数値」や「文字」など、さまざまな種類がある
- 種類ごとに扱い方が変わる
- 例えば数値は足し算ができるが、文字は結合になる

3章 よく使われるデータ型

| Pythonでよく使われるデータ型を4つ紹介します

| ① 文字列型 (str)

文字や文章を扱う

| ③ 浮動小数点型 (float)

小数を含む数値を扱う

| ② 整数型 (int)

整数の数値を扱う

| ④ 論理型 (bool)

TrueまたはFalseの値

3章 データ型① 文字列型 (str)

| 文字や文章のデータを表す型です

- strとはstring（ストリング）の略で「ひも」という意味
- 文字が連なっている状態をイメージするとわかりやすい
- シングルクオーテーション（'）またはダブルクオーテーション（"）で囲んで表現する

3章 データ型② 整数型 (int)

整数の数値データを表す型です

- intとはinteger（インテジャー）の略で「整数」という意味
- 小数点を含まない数値のこと
- 正の数、負の数、ゼロを含む

3章 データ型③ 浮動小数点型 (float)

| 小数を含む数値データを表す型です

- float（フロート）とは「浮動小数点数」という意味
- 小数点以下の値を持つ数値を扱うときに使う
- 科学的な計算や金銭の計算などで重要

3章 データ型④ 論理型 (bool)

| TrueまたはFalseの2つの値を持つ型です

- bool (布尔) とはboolean (ブーリアン) の略
- 条件分岐などで使われる
- Pythonでは先頭が大文字でTrueまたはFalseと書く

3章 数値を計算してみよう

| Pythonでは算術演算子を使って計算ができます

- 足し算、引き算、掛け算、割り算などの計算ができる
- 計算に使う記号を算術演算子という
- Pythonの算術演算子は数学の記号と似ているが、一部異なる

3章 算術演算子の一覧

| よく使われる算術演算子を紹介します

演算子	意味	例	結果
+	足し算	$5 + 3$	8
-	引き算	$5 - 3$	2
*	掛け算	$5 * 3$	15
/	割り算	$5 / 3$	1.666...
%	剰余（余り）	$5 \% 3$	2

3章 計算の実行例

Visual Studio Codeで計算を実行してみましょう

```
# 足し算  
print(5 + 3) # 結果: 8  
  
# 引き算  
print(5 - 3) # 結果: 2  
  
# 掛け算  
print(5 * 3) # 結果: 15  
  
# 割り算  
print(5 / 3) # 結果: 1.666666666666667  
  
# 剰余（余り）  
print(5 % 3) # 結果: 2
```

3章 文字列を結合してみよう

| 文字列同士を連結（結合）する方法を学びます

- 複数の文字列を1つにつなげることを結合という
- Pythonには文字列を結合する方法がいくつかある
- 状況に応じて使い分けると便利

3章 文字列結合の方法

| 文字列を結合する代表的な方法を紹介します

方法	説明
+演算子	文字列同士を+でつなげる
f文字列	f"..."の中に{変数}を埋め込む
format()	"{}"に.format()で値を埋め込む

3章 +演算子で結合

| +演算子を使って文字列を連結できます

```
first_name = "山田"  
last_name = "太郎"  
  
# +演算子で結合  
full_name = first_name + last_name  
print(full_name) # 結果: 山田太郎
```

- シンプルで直感的な方法
- 文字列同士の結合に使う

3章 f文字列で結合

| f文字列を使うと変数を埋め込んで結合できます

```
name = "山田"  
age = 25  
  
# f文字列で結合  
message = f"私の名前は{name}で、年齢は{age}歳です."  
print(message) # 結果: 私の名前は山田で、年齢は25歳です。
```

- Python 3.6以降で使える
- 変数を{}で囲んで埋め込む

3章 format()で結合

| format()メソッドを使う方法もあります

```
name = "山田"  
age = 25  
  
# format()で結合  
message = "私の名前は{}で、年齢は{}歳です。".format(name, age)  
print(message) # 結果: 私の名前は山田で、年齢は25歳です。
```

- {}がプレースホルダーになる
- format()の引数が順番に埋め込まれる

3章まとめ

| 本章では以下の内容を学習しました

データ型

- 文字列型 (str)、整数型 (int)、浮動小数点型 (float)、論理型 (bool) の4つ

数値の計算

- 算術演算子 (+、 -、 *、 /、 %) を使って計算ができる

文字列の結合

- +演算子、f文字列、format()の3つの方法がある