

+
•
○

プログラミング基礎 総論

基本的な処理や考え方



変数とデータ型

- **変数**：データを一時的に保存するための名前付きの領域です。箱のように考えるとわかりやすいでしょう。変数名は、英字で始まる英数字やアンダースコアの組み合わせで構成されます。例えば、x や total_sum などです。
- **データ型**：変数が保持するデータの種類を示します。主なデータ型には、整数 (int)、浮動小数点数 (float)、文字列 (string) などがあります。整数 (int) 型の変数には整数しか入らない
例：整数型の変数 a があるとき
 - a = 10
 - ✗ a = “りんご” 、 a = 1.2

演算

- **算術演算**：加算（+）、減算（-）、乗算（*）、除算（/）など基本的な数学演算が可能です。整数の除算では、余りを求める演算子（%）などがあります。
- **比較演算**：値同士を比較し、等しい（==）、異なる（!=）、大きい（>）、小さい（<）、以上（>=）、以下（<=）などの結果を得ることができます。
- **論理演算**：複数の条件を組み合わせる際に使用し、論理積（AND）、論理和（OR）、否定（NOT）などがあります。



制御構造

条件分岐：特定の条件に応じて異なる処理を実行します。
一般的にはif（もし～ならば）、else（そうでなければ）、else ifもしくはelif（ifの条件に一致せず、もし～ならば）などの構文が使用されます。

繰り返し（ループ）：
同じ処理を複数回実行する際に使用します。
forループやwhileループが一般的です。

関数

定義：特定の処理をまとめ、再利用可能なコードブロックとして定義します。関数には名前があり、必要に応じて引数を受け取り、処理結果を返すことができます。

呼び出し：定義された関数は、その名前を呼び出すことで実行されます。例えば、`print("hello")`は`print`関数を呼び出して文字列を表示します。

配列（リスト）

- **定義**：複数の値を一つの変数で管理するためのデータ構造です。要素にはインデックス（添字）を使用してアクセスします。多くの言語では、インデックスは0から始まります。
- **操作**：要素の追加、削除、更新、検索などの操作が可能です。例えば、pythonではappend()メソッドでリストに要素を追加できます。