

条件分岐

Python プログラミング入門

プログラムに「判断」をさせる——状況に応じて処理を変える技術を視覚的に学ぶ



if文の基本



if-else文



条件式の書き方

この知識があると何ができる？

条件分岐は、プログラムに「状況に応じた判断」をさせるための必須スキルです



ログイン判定

パスワードが正しいか判定して、結果に応じたメッセージを表示



購入可否の判定

予算や在庫を確認して購入できるかどうかを自動で判断



合否判定・スコア分け

テストの点数や条件で合格/不合格を自動判定



生成AIにコードを書かせる場合でも、条件分岐を理解していれば「指示の精度」が格段に上がります

条件分岐とは？

プログラムに「判断」をさせる仕組み

日常生活のたとえ

「雨が降っているなら、傘を持っていく」

「残高が足りていれば、商品を購入する」

「年齢が18歳以上なら、入場を許可する」

→ 条件によって実行する処理を変えること
これが「条件分岐」

条件分岐の流れ

1. 条件を判定する

True か False か？

2. True なら

→ ある処理を実行する

3. False なら

→ 別の処理を実行(または何もしない)

if文の基本

条件がTrueのときだけ処理を実行する

構文

```
if 条件式:  
    処理(インデント必須)
```

- ✓ ifの後に条件式を書く
- ✓ 条件式の最後に:(コロン)をつける
- ✓ 実行する処理はインデント(字下げ)する

CODE

```
# 年齢を判定する  
age = 20  
  
if age >= 18:  
    print("成人です")  
    print("ようこそ!")  
  
print("プログラム終了")  
  
→ 成人です  
→ ようこそ!  
→ プログラム終了
```

インデントの重要性

Pythonではインデントがプログラムの構造を決定する

✔️ 正しい例

```
age = 20
if age >= 18:
    print("成人です")
    print("ようこそ")
```

```
print("終了")
```

↑ インデントあり → if文の中

↑ インデントなし → if文の外

❌ 間違った例

```
age = 20
if age >= 18:
print("成人です")
# ↑ エラー！インデントがない
```

💡 インデントのルール

スペース4つ or タブ1つ
深さは必ず揃える

様々な条件式

if文では比較演算子や論理演算子を自由に使える

比較演算子を使った条件

```
# 等しいかどうか
password = "abc123"
if password == "abc123":
    print("ログイン成功")

# 大小の比較
temperature = 28
if temperature >= 25:
    print("夏日です")

# 等しくないことをチェック
if status != "completed":
    print("処理中です")
```

論理演算子を使った条件

```
# and: 両方Trueのとき実行
age = 25
has_license = True

if age >= 18 and has_license:
    print("運転できます")

# or: どれかTrueのとき実行
day = "土曜日"
if day == "土曜日" or
   day == "日曜日":
    print("休日料金")
```

if-else文

条件がTrueなら処理A、Falseなら処理B — 2つの選択肢

構文

```
if 条件式:  
    Trueのときの処理  
else:  
    Falseのときの処理
```

CODE

```
# 年齢で成人かどうか判定  
age = 15  
  
if age >= 18:  
    print("成人です")  
    print("入場できます")  
else:  
    print("未成年です")  
    print("保護者の同意が必要")
```

- 未成年です
- 保護者の同意が必要

age = 15 なので age >= 18 は False → else の中が実行される

if-elseの実用例

ログイン判定

```
pw = "python123"  
correct = "python123"  
  
if pw == correct:  
    print("ログイン成功")  
else:  
    print("パスワードが違います")
```

在庫チェック

```
stock = 0  
  
if stock > 0:  
    print("在庫あり")  
    print("購入できます")  
else:  
    print("在庫切れ")
```

合否判定

```
score = 75  
pass_line = 60  
  
if score >= pass_line:  
    print("合格です!")  
else:  
    print("不合格です")
```

条件式の書き方のコツ

💡 わかりやすい変数名を使う

```
# わかりにくい
if x >= 18: print("OK")

# わかりやすい
if age >= 18: print("成人です")
```

💡 複雑な条件は変数に入れる

```
# 条件を変数に入れる(推奨)
is_adult = age >= 20
has_income = income >= 200

if is_adult and has_income:
```

✔️ 否定形より肯定形

```
# 否定形(避ける)
if not is_not_available:

# 肯定形(推奨)
if is_available:
```

✔️ 範囲指定を簡潔に

```
# 通常の見かけ
if score >= 60 and score < 80:

# Python の簡潔な書き方
if 60 <= score < 80:
```

まとめ



if文の基本

条件がTrueのとき処理を実行
if 条件式: で条件を書く
コロン: を忘れずに
処理は必ずインデントする



if-else文

2つの選択肢を用意できる
Trueのとき / Falseのとき
で異なる処理を実行
条件の結果を変数にも格納可能



条件式の書き方

わかりやすい変数名を使う
複雑な条件は変数に入れる
肯定形で書く
範囲指定を簡潔に書く



次のステップ → elif で3つ以上の選択肢、ネストした条件分岐を学ぶ