


# 繰り返し処理

Python プログラミング入門

for文 / while文 / break・continue

 for文

 while文

 break / continue

 使い分け

# この知識があると何ができる？

繰り返し処理は、「使えるプログラム」を作るための必須スキルです



## 同じ処理を自動化

100行の手作業を  
たった3行で実現  
効率が劇的に向上



## データの一括処理

リストの全要素に  
同じ処理を適用  
集計・変換が簡単



## 入力の検証

正しい値が入るまで  
繰り返し入力を求める  
エラーに強いプログラム

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

# 繰り返し処理とは？

同じ処理を何度も実行する仕組み

## ループなし

```
5行必要...  
# 1から5まで表示  
print(1)  
print(2)  
print(3)  
print(4)  
print(5)
```



## ループあり

```
たった2行！  
# 1から5まで表示  
for i in range(1, 6):  
    print(i)  
  
→ 1, 2, 3, 4, 5
```

100回繰り返したい場合も、ループならrange(1, 101)に変えるだけ

# for文の基本

決まった回数だけ繰り返す

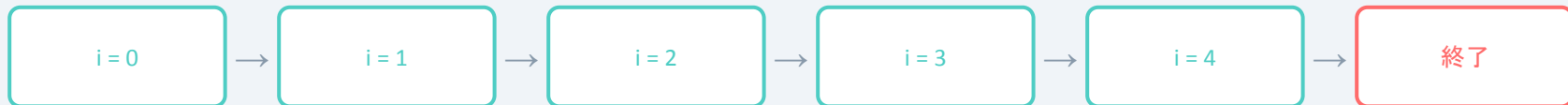
## 構文

```
for 変数 in range(回数):  
    繰り返す処理
```

## 実行例

```
for i in range(5):  
    print(i)
```

## 動作の流れ



プログラムは0から数え始める。range(5) は 0, 1, 2, 3, 4 の5回

# range() の3つの使い方

## 1 range(終了値)

0から始まる

```
for i in range(5):  
    print(i)
```

→ 0, 1, 2, 3, 4

## 2 range(開始, 終了)

開始値を指定

```
for i in range(1, 6):  
    print(i)
```

→ 1, 2, 3, 4, 5

## 3 range(開始, 終了, ステップ)

増加量を指定

```
for i in range(0, 10, 2):  
    print(i)
```

→ 0, 2, 4, 6, 8



終了値そのものは含まれない。range(5) は 0~4、range(1, 6) は 1~5

# for文の実用例

## 1~10の合計を計算

CODE

```
total = 0

for i in range(1, 11):
    total += i

print(total)
→ 55
```

## 九九の7の段

CODE

```
number = 7

for i in range(1, 10):
    result = number * i
    print(result)

→ 7, 14, 21 ... 63
```

## for文 + if文の組み合わせ

```
for i in range(1, 21):
    if i % 2 == 0: print(i) → 2, 4, 6 ... 20
```



ループの中でif文を使えば  
条件に合うデータだけ処理で  
きる

# while文の基本

条件がTrueの間、繰り返す

## 構文

```
while 条件式:  
    繰り返す処理
```

## 実行例: カウントダウン

```
CODE  
time = 5  
  
while time > 0:  
    print(time)  
    time -= 1
```

→ 5, 4, 3, 2, 1

## 動作の流れ



for文と違い、繰り返す回数が事前にわからない場合に使う

# 無限ループに注意！

条件が永遠にTrueのままだと、プログラムが止まらなくなる

 NG: 無限ループ

DANGER

```
count = 1
```

```
while count <= 5:  
    print(count)  
    # count += 1 を忘れた！
```

 OK: 正しいループ

OK

```
count = 1
```

```
while count <= 5:  
    print(count)  
    count += 1
```

## 回避のポイント

1. ループ内で条件変数を必ず更新

2. 終了条件が正しいか確認

3. 意図的な無限ループには break を用意

# break文とcontinue文

ループの流れを制御する2つの命令



**break: ループ全体を終了**

CODE

```
for i in range(1, 11):  
    if i == 5:  
        break  
    print(i)
```

→ 1, 2, 3, 4  
# 5で止まる



**continue: その回だけスキップ**

CODE

```
for i in range(1, 6):  
    if i == 3:  
        continue  
    print(i)
```

→ 1, 2, 4, 5  
# 3だけスキップ

break = 「ループ自体をやめる」 / continue = 「今回だけ飛ばして次へ」

# for文 と while文 の使い分け

## for文

回数がわかっている繰り返し

```
# 10回繰り返す
for i in range(10):
    print(i)
```

- ✓ 回数が決まっている処理
- ✓ リストの各要素を処理
- ✓ カウンターが必要な場合

## while文

条件を満たすまで繰り返し

```
# 100を超えるまで
while total < 100:
    total += n
```

- ✓ 回数が事前にわからない
- ✓ ユーザー入力待ち
- ✓ 条件を満たすまで処理

# まとめ

## for文

決まった回数だけ繰り返す  
range()で範囲を指定  
range(終了), range(開始, 終了), range(開始, 終了, ステップ)

## while文

条件がTrueの間繰り返す  
無限ループに注意  
条件変数を必ず更新する

## break / continue

break → ループ全体を終了  
continue → その回だけスキップ  
用途に応じて使い分ける

## 使い分け

回数が決まっている → for文  
条件を満たすまで → while文  
迷ったらfor文から試す



次のステップ → リストとループを組み合わせ、より実践的なプログラミングを学ぶ