

6-1 条件によって処理を分ける if 文

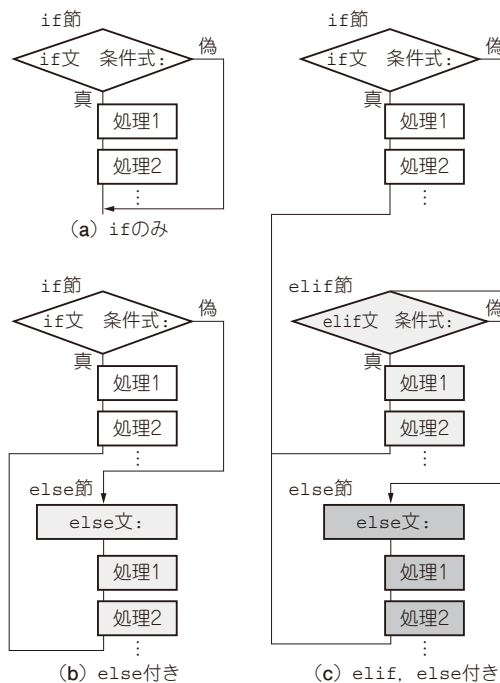


図1 条件式を評価してその後の処理を分岐する

リスト1 条件分岐の構文

```

if 条件式:
    条件式が真(True)のときに実行する処理
} ← 図1(a)

if 条件式:
    条件式が真(True)のときに実行する処理
else:
    条件式が偽(False)のときに実行する処理
} ← 図1(b)

if 条件式1:
    条件式1が真(True)のときに実行する処理
elif 条件式2:
    条件式2が真(True)のときに実行する処理
else:
    すべての条件式が偽(False)のときに実行する処理
} ← 図1(c)
    
```

この3つの文を使った条件分岐には、ifだけ、if…else、if…elif…elseの記述パターンがあります。処理は上から順に実行されるので複数の条件式を使ったプログラムを書くときには、アルゴリズムに注意が必要です。条件式の記述に重複や漏れがないか、評価の順序に問題や矛盾がないかを、よく確認することが重要です。

elif文が大量に続くプログラムは、多数の条件式を処理するため処理に時間がかかります。複雑な条件式の組み合わせにならないよう、分割してシンプルなソースコードに見直すと記述ミスを減らせると思います。

● if文の利用目的

if文は条件分岐の処理に使われます(図1)。単独で利用する以外にも、for文やwhile文のようなループ処理でよく利用されます。条件分岐を使ってループから抜け出したり、ループの中で分岐によって異なる処理を実行したりするのに使われます。

● if文の書き

条件分岐の構文をリスト1に示します。

● 条件分岐の構文は3種類

if文の条件分岐は条件式の中で真偽を評価します。真のときにif節に記述された処理が実行され、偽のときには実行されません。

複数の条件分岐を行うには、if文の後に、elif文を追加して条件式を記述します。elif文は複数記述でき、条件式の真偽の評価はif文の条件式と同じです。elif節には条件式が真のときに実行させたい処理を記述します。

if文やelif文の条件式のいずれも偽として評価されたときに処理するのにelse文を使います。else文はif文やelif文の後に記述し、else節に処理を記述できます。

● 実行例

▶ if節だけの場合

条件分岐で条件式の評価結果が真のときに処理する例です。if文の条件式で変数aの値と5を比較して、評価が真のときにprint関数が実行されるプログラムです。図2のプログラムでは、最初の行で変数aに5が代入されます。条件式の評価は真となり、画面にTrueが表示されます。もし、評価が偽ならif節の後に処理が接続され、if節の中にあるprint関数は実行されずに処理が終了します。

```

a = 5
if a == 5:
    print("True")
    
```

True

図2
if文1…if節だけ

▶ else節を使う場合

図3のプログラムは、条件式の評価が真と偽で異なる処理内容を実行したいときの例です。if文の条件