

4. フロー制御について学ぼう

4.1. if文による条件分岐

4.1.1. if文と条件式

Pythonのプログラムを実行すると、上から順にひとつひとつのステートメントが処理されていきます。しかし、実際にプログラムを作成するときには、以下のような条件によって、後の処理を分岐させたいことがあります。

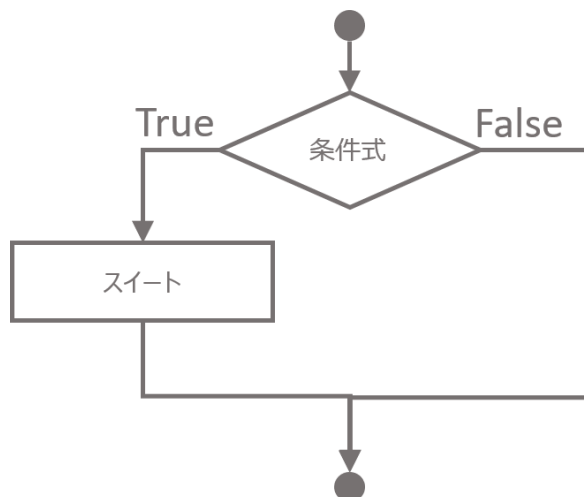
- 変数が指し示す値が50以上かどうか
- 今年はいくつ年かどうか
- フォルダ内に指定したExcelファイルが存在しているかどうか
- 指定したURLに正常にアクセスできるかどうか

このようなときに、多くのプログラミング言語では、ある条件によって以降の処理を分岐させる仕組みが用意されていて、それを「条件分岐」といいます。Pythonで条件分岐を実現する構文のひとつが「if文」で、以下のように記述します。

```
----- 構文 -----  
if 条件式:  
    スイート  
----- ここまで -----
```

if節の先頭行であるヘッダーにはキーワード「if」に続いて、条件式、そしてコロン記号「:」を記述します。条件式はブール値、すなわちTrueかFalseかいずれかの値を取るものです。条件式が成立している、つまりTrueの値であれば、if節に含まれるスイートを実行し、成立していないとき、つまりFalseであればスイートは実行されずに無視されます。この流れを図に表すと、図4-1のようになります。

図4-1 if文による条件分岐
(fig04-01.pngはいる)



if文の例として、sample04_01.pyを見てみましょう。

sample04_01.py if文による条件分岐

----- コード -----

```
x = 5
```

```
y = 3
```

```
if x > y:
```

```
    print('xはyより大きい')
```

----- ここまで -----

----- 実行結果 -----

```
xはyより大きい
```

----- ここまで -----

変数xの値のほうが、変数yの値よりも大きいので、条件式「 $x > y$ 」は成立している、つまりTrueですから、スイートの処理が実行されます。変数xの値を、変数y以下の値に設定すると、スイートは処理されずに、何も出力されなくなります。試してみてください。

なお、スイートは複数のステートメントで構成することもできますが、スイートに含まれるすべてのステートメントにはインデントを入れる必要があります。

4.1.2. if~else文

if文で条件式がfalseとなった際に、別のスイートを実行したいときがあります。その場合には、if文にelse（エルス）節を加えた、「if~else文」を使うことで実現することができます。if~else文は、以下のように、if文に加えてelse節を加えて記述します。

----- 構文 -----

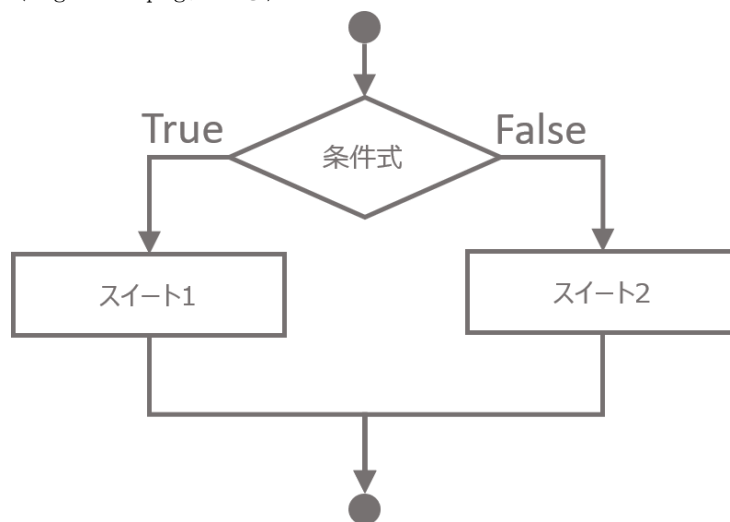
```

if 条件式:
    スイート1
else:
    スイート2
----- ここまで -----

```

else節のヘッダーはキーワード「else」とコロンのみで構成されます。if節のヘッダーで指定した条件式の値がTrueであればスイート1のみを実行し、Falseであればスイート2のみを実行します。この流れを表したものが図4-2です。

図4-2 if~else文による条件分岐
(fig04-02.pngはいる)



if~else文の例として、sample04-02.pyを実行してみましょう。ここで、if節のスイートだけでなく、else節のスイートもインデントする必要がありますので、よく注意してください。

```

sample04_02.py if~else文による条件分岐
----- コード -----
x = 5
y = 3

if x > y:
    print('xはyより大きい')
else:
    print('xはy以下')
----- ここまで -----

```

```
----- 実行結果 -----  
xはyより大きい  
----- ここまで -----
```

このまま実行すると、「 $x > y$ 」がTrueですから、if節のスイートが処理されます。変数xの値を、変数yの値以下に設定して実行すると、else節のスイートが処理されますので、実際に確認してみましょう。