

第5回 条件分岐

条件分岐の if 文について学ぶ。

【BMI 計算プログラム】

★前回作成した BMI 計算プログラムを改良して、次のプログラムを実行しましょう。最初の 2 行は前回保存したプログラムと同じなので、「一覧」「自分のコード」から前回保存したコードを呼び出すことで、続きを打ち込むことができる。今回は日本語入力を使うので、第3回で学習した「無効な文字」のエラーを出しやすくなるので注意すること。

```
a, b = map(float, input().split())
c = 10000 * b / (a ** 2)
if c < 18.5:
    d = "やせ型"
elif c < 25:
    d = "標準体型"
else:
    d = "肥満型"
print("BMIは{0:.1f}です。あなたは{1}です。".format(c, d))
```

実行例

入力：172 60

出力：BMI は 20.3 です。あなたは標準体型です。

あなたの身長と体重を入れて試してみましょう。

今回のプログラムでは、判定基準に基づいて、判定された結果が表示される。このプログラムの動作は、以下のようになる。

- (1) 入力を受け付けて、a と b にそれぞれ身長と体重が入る。
- (2) BMI 指数（身長と体重から肥満度を計算する指数）を計算し、c に格納する。ここまでは前回のプログラムと同じである。ログインして作業をしているのであれば、「一覧」「自分のコード」から前回のコードを呼び出すことができる。
- (3) **赤字の箇所**：BMI 指数が 18.5 未満であれば「やせ型」、18.5 以上 25 未満であれば「標準体型」、25 以上であれば「肥満型」という判定結果を変数 d に文字列として格納する。ここで条件分岐の if 文が使われているので、以下に詳しく解説する。
- (4) BMI 指数と判定結果を表示する。format 関数を使って表示形式を整えているので、後に詳しく解説する。

【インデントについて】

Python では、複合文¹をインデント（字下げ）によって表現する。たとえば

```
if c < 18.5:
    d = "やせ型"
```

¹ 複合文 https://docs.python.org/ja/3/reference/compound_stmts.html

という箇所について、if 文の意味については次に解説するが、まずは形式について着目する。複合文は1つ以上の節から成り、上の if 文は1つの節である。節は「ヘッダ」と「ブロック（スイート）」から成り、1行目の `if c < 18.5:` が「ヘッダ」であり、`d = "やせ型"` がこのヘッダに付随するブロックである。節に対するブロックはインデントが下がり、ブロックが複数の行にわたるときには同じレベルのインデントとなる。

Python では、このようにインデントがプログラムを解釈するための重要な意味を持つため、不要なインデントを入れてしまうとエラーとなる。インデントは「スペース」または「タブ」によって作られ、その数は同じブロック内で等しければいくつであっても良いが、「4個のスペース」を使うことが標準として推奨されている（「全角スペース」は使わないこと）。Paiza では、Tab キーを押すことで4個のスペースが入力される。また、if 文のステートメントブロックを入れて Enter キーを押すと、自動的にインデントがされる。さらには、インデントされた行で改行をすると、自動的にインデントが継続する。インデントを元に戻すにはバックスペース (BS) キーを使う。

【基本的な if 文】

if 文はプログラミング言語が最低限備えている必要がある制御構造である「選択」「ループ」の中で「選択」という制御構造である。

基本的な if 文の構造

```
if 条件式:
    条件が成り立つときの処理
```

「条件が成り立つときの処理」は、同じインデントブロックであれば何行でも記述できる。たとえば、次のように記述する。

```
if 条件式:
    条件が成り立つときの処理1
    条件が成り立つときの処理2
    条件が成り立つときの処理3
```

条件式の値は、条件を満たすときに True、満たさないときに False となる。条件式の値は、必ずこのどちらかになる。式の型は bool 型である。

たとえば、`c < 18.5` の時に `d` に「やせ型」という文字列を代入する処理は、次のようになる。

```
if c < 18.5:
    d = "やせ型"
```

ここで、`if c < 18.5:` という文の中で、`c < 18.5` が条件式である。条件式 `c < 18.5` は、`c` の値が 18.5 未満の時に True となり、18.5 以上の時に False となる。したがって、`c < 18.5` の時に `d = "やせ型"` が実行され、そうでない場合には実行されない。

数の大小比較をする比較演算子をまとめる。

Python	数学	意味
<code>c == 18.5</code>	<code>c = 18.5</code>	<code>c</code> が 18.5 に等しいなら True、等しくなければ False
<code>c != 18.5</code>	<code>c ≠ 18.5</code>	<code>c</code> が 18.5 に等しくないなら True、等しければ False

<code>c >= 18.5</code>	<code>c ≥ 18.5</code>	c が 18.5 以上なら True、それ以外なら False
<code>c <= 18.5</code>	<code>c ≤ 18.5</code>	c が 18.5 以下なら True、それ以外なら False
<code>c > 18.5</code>	<code>c > 18.5</code>	c が 18.5 より大きいなら True、それ以外なら False
<code>c < 18.5</code>	<code>c < 18.5</code>	c が 18.5 より小さいなら True、それ以外なら False

ここで、「等しい」を意味する比較演算子は、`=` を 2 個重ねていることに注意する。前回説明したように、Python では `=` は代入と解釈されるため、「等しい」を意味する場合には `==` と重ねることで区別する。if 文で `==` と書くべきところを `=` と書いてしまう間違いは、わかっているもうっかりと間違いやすいところである。

【if … else 文】

次に、条件式が True の場合と False の場合、それぞれ別の処理を書く if … else 文の構造を示す。

```
if 条件式:
    条件が成り立つ時の処理
else:
    条件が成り立たなかった時の処理
```

たとえば、`c < 18.5` の時に `d = "やせ型"`、そうでない時に `d = "やせ型ではない"` を実行するには、次のように書く。

```
if c < 18.5:
    d = "やせ型"
else:
    d = "やせ型ではない"
```

【elif による if 文の連鎖】

`else` と if を連結した `elif` を使って、次のように if 文を連鎖することができる。

```
if 条件式 1:
    条件式 1 を満たすとき
elif 条件式 2:
    条件式 1 は満たさず、条件式 2 を満たすとき
else:
    条件式 1 は満たさず、条件式 2 も満たさないとき
```

冒頭のプログラムの赤字の箇所では、この構造を使って、`c` の大きさによって 3 通りに `d` の値を変化させる処理を実現している。

【format 関数による表示形式の整え方】

プログラムの最後の行の

```
print("BMIは{0:.1f}です。あなたは{1}です.".format(c, d))
```

について解説する。print 文の引数には、

```
"BMI は{0:.1f}です。あなたは{1}です.".format(c, d)
```

と書かれているが、これは str 型に定義されている format メソッドである¹。ここでは、format メソッドに c と d という 2 つの変数が引数として渡されていて、その 2 つの変数が“BMI は{0:.1f} です。あなたは{1}です。”という文字列の{0:.1f}という箇所と{1}という「順序引数」に入って表示される。順序引数は、変数の順番に {0} {1} {2}, …となり、ここで 0 から始まることに注意する。{0}のところで{0:.1f}と指定することで、小数点以下 1 桁までを表示するようにしている。書式設定の仕様について詳しくは <https://docs.python.org/ja/3/library/string.html#formatstrings>

【最大値を求めるプログラム】

★3 つの整数の最大値を計算するプログラムを作成して実行をしましょう。

```
a, b, c = map(int, input().split())
m = a
if b > m:
    m = b
if c > m:
    m = c
print(m)
```

実行例

入力 : 4 5 7

出力 : 7

【★課題】

上記プログラムの赤字の箇所を修正して、3 つの整数値の「最小値」を出力するプログラムを作成して、実行をして動作を確認してから、ToyoNet ACE 経由で提出して下さい。赤字以外の場所は書き換えないこと。

実行例

入力 : 4 5 7

出力 : 4

入力 : 13 2 5

出力 : 2

【発展 : Python の禅】

「発展」では学生が自主的により深い学習をするためのきっかけとなるトピックを取り上げる。次のプログラム（イースターエッグ）を実行してみよう。

```
import this
```

¹ データ型（クラス）に対して定義される関数をメソッドと言う。詳しくは第 12 回「クラスの作成」で解説する。