

■ break 文による繰返しの中断と else 節

カウントダウンのプログラムも、1 から n までの和を求めるプログラムも、正でない数値を n に読み込んだ場合、おかしい結果を出力します。

キーボードから読み込む値を、《正の整数値》に限定するように、前のプログラムを変更しましょう。そのプログラムが **List 4-4** です。前半部が変更されています。

List 4-4

chap04/list0404.py

```
# 1からnまでの和を求める (nに正の整数値を読み込む)

print('1からnまでの和を求めます。')

while True:
    n = int(input('nの値: '))
    if n > 0:
        break

sum = 0
i = 1
while i <= n:
    sum += i # sumにiを加える
    i += 1 # iに1を加える
print('1から', n, 'までの和は', sum, 'です。')
```

実行例

1からnまでの和を求めます。
nの値: -5 ●
nの値: 0 ●
nの値: 5 ●
1から 5 までの和は 15 です。

0 以下であれば再読みみ

読み込んだ値を n に代入する文が、網かけ部の **while** 文の中に入っています。その **while** 文の判定式は、単なる **True** です。これは真ですから、**while** 文は無限に繰り返されます。このような繰返しは、**無限ループ**と呼ばれます。

■ break 文

キーボードから整数値を読み込んだ後は、 n が正であるかどうかを、**if** 文で判定します。判定が成立したときに実行しているのが、初登場の **break 文** (*break statement*) です。

Fig.4-4 に示すように、繰返し文中で break 文が実行されると、その繰返し文の実行が強制的に終了します。その結果、無限ループから脱出します。

- ▶ 4-3 節で学習する《多重ループ》の中で **break** 文が実行されると、その **break** 文を直接囲んでいる繰返し文の実行が中断されます。

なお、読み込んだ整数値 n が 0 以下であれば、**break** 文は実行されないため、**while** 文の実行が再び繰り返されます（「 n の値: 」と入力促して、読み込みを行います）。

■ else 節

図に示すように、**while** 文の末尾には **else** 節を置けます。**while** 文の繰返しが **break** 文で強制終了することなく（つつがなく）終了したときのみ、**else** 節内のスイートが実行されます。

- ▶ **else** 節付きの **while** 文の具体例は、**List 4-6** (p.96) で学習します。

continue 文による繰返し内の処理のスキップ

`break` 文と対照的な文が **continue 文** (*continue statement*) です。図に書かれているように、繰返し文中で `continue` 文が実行されると、ループ本体内の後続部の実行がスキップされて、プログラムの流れが判定式へと戻ります。

List 4-5 が、`continue` 文を用いたプログラム例です。

List 4-5

chap04/list0405.py

```

# 整数を次々と読み込んで正の整数値を加算

print('正の整数値を加算します (終了は-9999)。')

sum = 0
while True:
    n = int(input('整数値: '))
    if n == -9999:
        break
    if n <= 0:
        continue
    sum += n
print('正の整数の合計は', sum, 'です。')

```

実行例

正の整数値を加算します (終了は-9999)。
 整数値: 5 →
 整数値: 7 →
 整数値: -2 →
 整数値: 4 →
 整数値: -9999
 正の整数の合計は 16 です。

正のみが加算される

整数値 n を次々と読み込んでいき、加算を繰り返すことによって合計を求めます。その際、次のように処理を行います。

- n が -9999 であれば、読み込みを終了する。 … `break` 文を実行
- n が 0 以下であれば、加算しない (正の値のみを加算する)。 … `continue` 文を実行

n が 0 以下のときに実行される `continue` 文の働きによって、続く累算代入 `sum += n` の実行がスキップされます (そのため、負の値は `sum` に加算されません)。

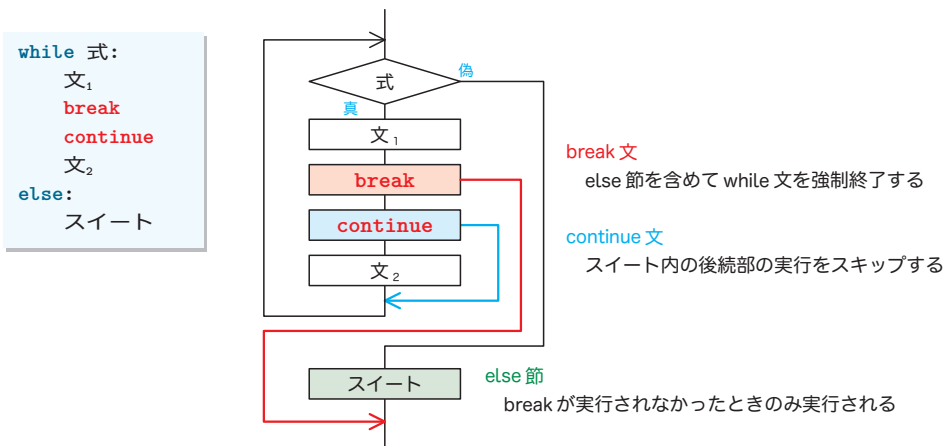


Fig.4-4 while 文と break 文と continue 文