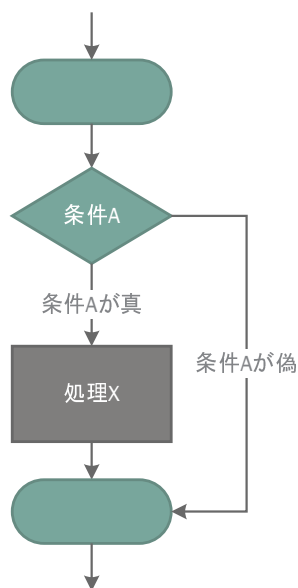


9 条件分岐

9.1 if 文

条件 (True か False) に応じて、処理を変化させるときに if 文という決められた文法を用います。if 文の手順は次のチャートの通りです：



これを Python では次の文法で書きます：

Python での if 文の構造

```

1 if 条件 A:
2     [ 処理 X ]
3     [ 処理 X ]
4     [ 処理 X ]
  
```

- 条件 A が True であれば処理 X が行われ、False であれば処理 X は行われません。
- if の行の行末のコロン『:』を忘れずに！
- 処理 X 数行にわたる場合は、上のようにインデントを揃える。

ここで処理 X の前のインデント（字下げ）が文法上重要な役割を果たします。処理 X が数行にわたるときに、どこまでが処理 X であるかはインデントによって判断されます。インデントは単なるスペースで見えないのですが、インデントが揃った部分が、処理されるプログラムの塊（ブロック）であると判断されます：

if 条件A:



if 文を使った例題をやってみましょう。次のアルゴリズムを Python プログラムにします。

- (1) 整数 a を入力させる。
- (2) a が偶数なら、『a is even』とプリントしてから、a を半分にする。
- (3) a が奇数なら何もしない。

● ファイル名 : if1.py

```

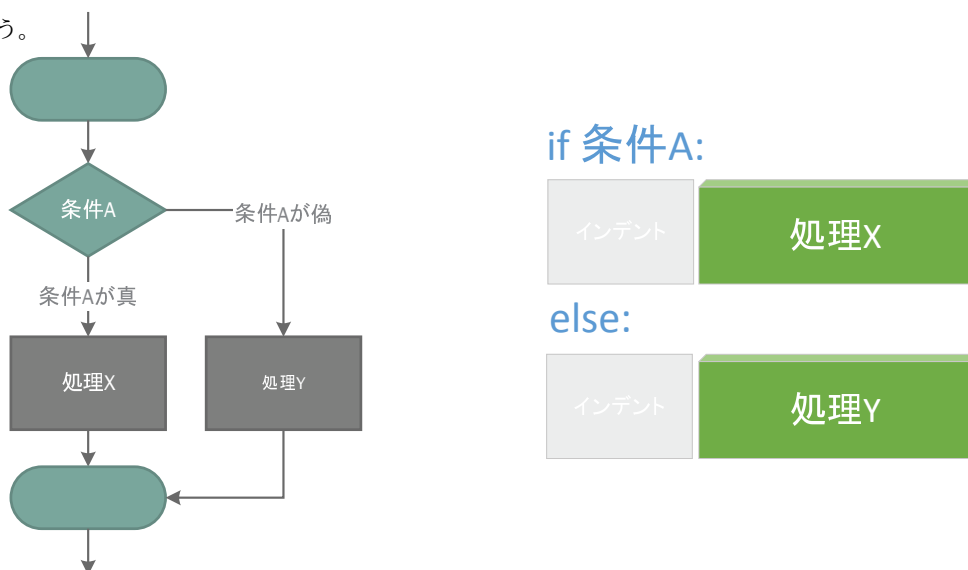
1 a = input('Input integer a: ')
2 a = int(a) # aを整数にする
3
4 if a%2 == 0: # もしaが偶数なら
5     print('a is even')
6     a = a//2 # aを半分にする。行頭のインデントに注意。
7
8 print(a) # aの値をプリント

```

上のプログラムの4~6行目が if 文です。実行していくつかの数値を入れて動作を試してみましょう。

9.2 if-else 文

次に、少し複雑な条件分岐を考えます。条件が真のときには処理 X、偽のときには別の処理 Y がしたいとしましょう。



もちろん、これは上で説明した if 文だけで書くことができます。つまり

```

1 if 条件A: # 条件Aが真なら
2     処理X # 処理Xを行い、偽なら行わない。
3 if not 条件A: # 条件Aが偽なら
4     処理Y # Yを行い、そうでなければ行わない。

```

と書くだけです。しかし、次に説明する if-else 文を使って、より明示的に書くこともできます。

Python での if-else 文の構造

```

1 | if 条件 A:
2 |     処理 X
3 | else:
4 |     処理 Y

```

- 条件 A が True であれば、処理 X が行い、False であれば処理 Y を行う。
- ここでも処理 X, 処理 Y はインデントによって判断されます。

if-else 文を使った例題をやってみましょう。次のアルゴリズムを考えます。

- (1) 整数 a を入力させる。
- (2) もし a が偶数ならば、a を半分にする。
- (3) もし a が奇数ならば、a を $3*a+1$ にする
- (4) a の値を表示する。

これをプログラムで書いてみましょう。

- ファイル名 : if2.py

```

1 | a = input('a?: ')
2 | a = int(a)      # 入力したものを整数に変換する。
3 |
4 | if a%2 == 0:   # もし a が偶数なら
5 |     a = a//2   # a を半分にする
6 | else:         # そうでなければ
7 |     a = 3*a + 1 # a を 3a+1 にする
8 |
9 | print(a)      # a の値をプリント

```

このプログラムを実行していくつかの数値を入れて動作を試してみましょう。

9.3 if-elif-else 文

さらに条件分岐を記述するためには if-elif-else 文を使います。elif は else if の略です。

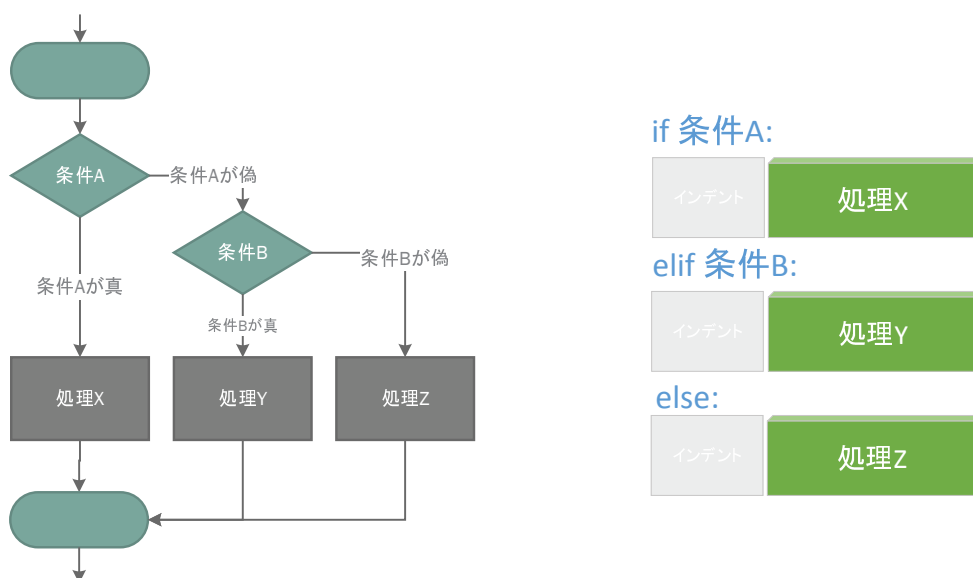
if-elif-else 文の構造

```

1 | if 条件 A:
2 |     処理 X
3 | elif 条件 B:
4 |     処理 Y
5 | else:
6 |     処理 Z

```

- 条件 A が真の場合に処理 X を行います。
- 条件 A が偽で条件 B が True のときに、処理 Y を行います。
- 条件 A も条件 B も成り立たないときに、処理 Z を行います。
- さらに条件 C, 条件 D, ... と条件が続く場合は elif で処理します。



さらに `elif` を追加して、`if-elif-elif-elif-else` のように条件を増やすことができます。

9.4 複雑な if 文の例

`if-elif-else` 文を使って、与えた西暦が閏年かどうかを判定するプログラムを作りたい。グレゴリウス暦では閏年は次のように決められている：

- 西暦年が 4 で割り切れる年は閏年とする。
- ただし西暦年が 4 で割り切れる年でも、100 で割り切れる年は閏年としない。
- ただし西暦年が 4 で割り切れ、100 でも割り切れる年でも 400 で割り切れる年は閏年とする。

上の閏年の定め方の文章には、『ただし』が多くあって、論理構造がわかりにくく、そのままではプログラムにはしづらい。そこで閏年の条件を次のように書き直します：

1. 西暦年が 400 で割り切れるならば、それは閏年である。
2. 上以外のとき、西暦年は 100 で割り切れるならば、それは閏年ではない。
3. 上以外のとき、西暦年が 4 で割り切れるならば、それは閏年である。
4. 上のどれにも当てはまらないとき、その年は閏年ではない。

これを実現する Python のプログラムは次のようになります。

- ファイル名：`uruuQ.py`

```

1 year = input('西暦を入力:') # 入力した数値を変数 year に代入する
2 year = int(year)
3
4 if year % 400 == 0: # もし year を 400 で割った余りが 0 なら
5     print(year, 'は閏年です。')
6 elif year % 100 == 0: # そうでないとき、もし year が 100 で割り切れたら
7     print(year, 'は閏年ではありません。')
8 elif year % 4 == 0: # そうでないとき、もし year が 4 で割り切れたら
9     print(year, 'は閏年です。')
10 else: # 上の全ての条件に当てはまらないとき
11     print(year, 'は閏年ではありません。')

```

10 練習問題

10.1 問題 : BMI 計算プログラム

名前, 身長, 体重を入力させ, そこから BMI を計算・表示し, その値に応じてやせ・肥満度の結果を出すプログラムを作りたい。プログラムは次の手順を行うものとする。

- (1) 名前を入力させ, 変数名 `namae` に代入
- (2) 身長・体重を入力させ, それぞれ `shintyo`, `weight` に代入
- (3) `bmi` を表示
- (4) もし `bmi < 18.5` なら, 「やせ気味です」と表示
- (5) そうでないとき, `bmi < 25.0` なら, 「ふつうです」と表示
- (6) そうでないとき, `bmi < 30` なら, 「太り気味です」と表示
- (7) うえのどちらでもないとき「太りすぎです」と表示

上の手順が実行されるように, つぎのプログラムの『*****』の部分を推測してプログラムを完成させなさい。

● ファイル名 : `bmi2.py`

```
1  namae = input('あなたの名前を入力してください:')
2  shintyo = input('あなたの身長は何センチですか?:')
3  shintyo = int(shintyo)
4  weight = input('あなたの体重は何キログラムですか?:')
5  weight = int(weight)
6
7  bmi = 10000.0*weight/(shintyo**2)
8  bmi = round(bmi,1)      # bmiの値を小数点以下2桁目を四捨五入する
9
10 print(namae, 'さんのBMIは', bmi, 'で', end='')
11
12 if *****:
13     print('やせ気味です。')
14 elif *****:
15     print('ふつうです。')
16 *****:
17     print('太り気味です。')
18 *****:
19     print('太りすぎです。')
```