7章 条件分岐のif文を理解しよう

7章 条件分岐のif文を理解しよう

条件分岐とは何か、概要を学び、実際にコードを書いてみます

本章の目標

- 条件分岐とは何か、概要をつかむこと
- if文の書き方を知り、実際にコードを書いてみること
- 比較演算子の種類を知ること

7章 本章について

プログラミングでは条件によって処理を分けたい場面がたくさんあります

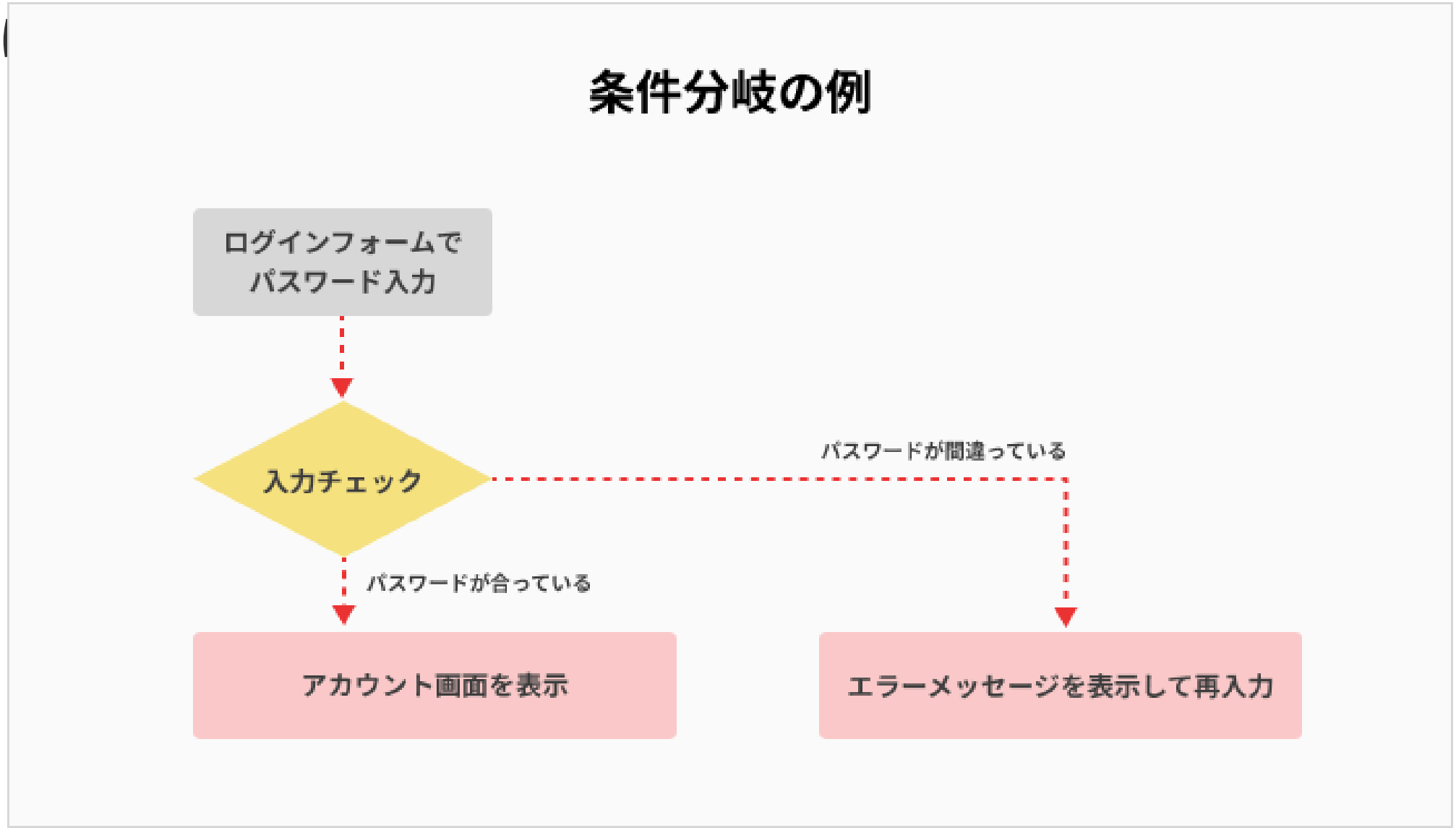
- 「ある条件に当てはまるときだけこの処理を実行したい」という場面で使う
- 条件分岐を使うことで様々な機能を実装できる

条件分岐の例

- もし金曜日なら商品を割引する
- 未ログインであればログイン画面を表示する

7章 条件分岐とは

条件


条件分岐の例
ログインフォームで
パスワード入力
入力チェック
パスワードが間違っている
パスワードが合っている
アカウント画面を表示
エラーメッセージを表示して再入力

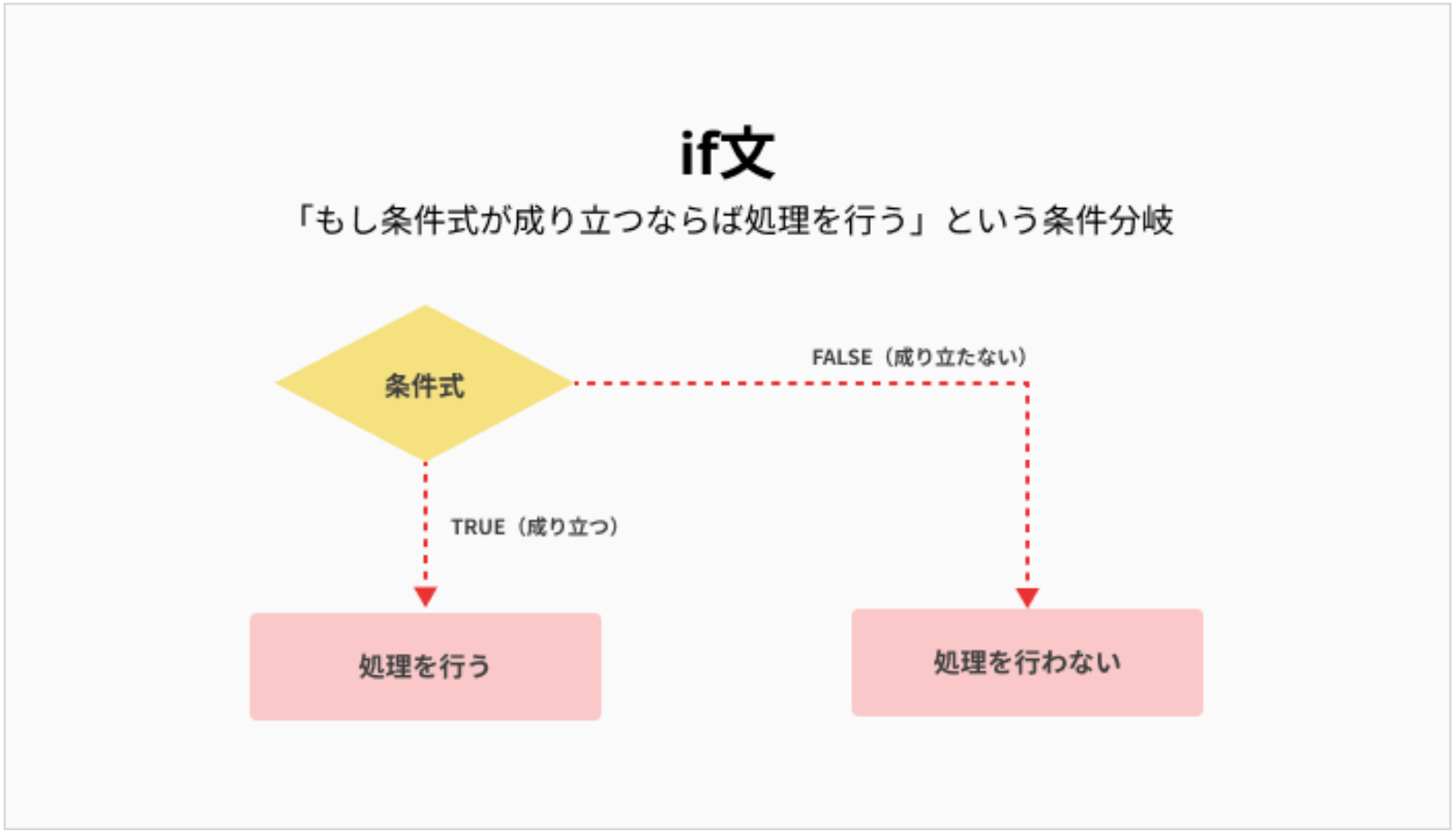
7章 条件分岐の使用例

> **「必須事項が入力されていたらフォーム内容を送信し、未入力であればエラーメッセージを表示する」**

- Pythonで条件分岐を行うときは、一般的に if文 を使う
- if文は最も基本的な条件分岐構文

7章 if文とは

「もし条件式が成り立つならば、処理を行う」という条件分岐構文です



7章 条件式とは

条件が成り立つかどうかを判定する式のことです

- 条件が成り立てば True を返す
- 成り立たなければ False を返す
- if文ではTrueが返されたときに処理が実行される

7章 if文の書き方

Pythonのif文は以下のように書きます

```
if 条件式:
    条件が成り立つときの処理
```

インデント

7章 if文の書き方のポイント

インデントが重要です

正しい書き方

```
if 条件式:
    処理1   # 条件成立時のみ
    処理2   # 条件成立時のみ
```

同じインデントの処理がまとめて実行される

間違った書き方

```
if 条件式:
    処理1   # 条件成立時のみ
処理2       # 常に実行される
```

インデントがないと別の処理になる

7章 if文の使用例

変数numの値によって処理を分岐させます

```
num = 50

# 変数numが10より大きいなら、文字列を出力する
if 10 < num:
    print("変数numは10より大きいです")

# 変数numが20より小さいなら、文字列を出力する
if num < 20:
    print("変数numは20より小さいです")
```

numは50なので、前者の条件のみ成り立つ

7章 戻り値とは

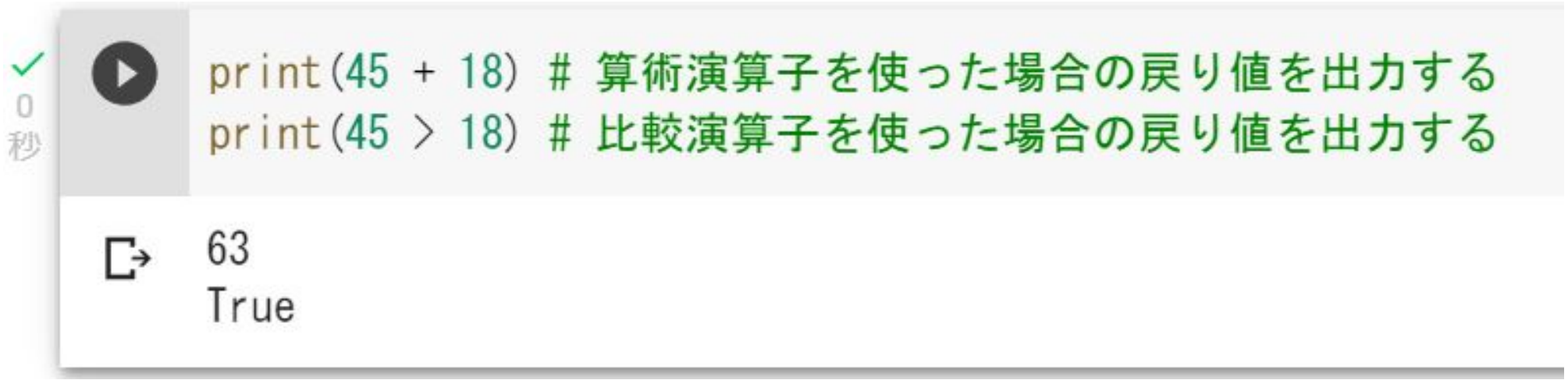
条件式から返ってきた値のことです

- 条件式は True（真） または False（偽） を返す
- この返ってきた値を 戻り値 または 返り値 という
- 四則演算の計算結果なども戻り値の1つ

算術演算子と比較演算子の戻り値を比較します

```
print(45 + 18) # 算術演算子を使った場合の戻り値を出力する
print(45 > 18) # 比較演算子を使った場合の戻り値を出力する
```

7章 戻り値を出力してみよう



7章 比較演算子とは

条件式に使う記号のことです

比較演算子	処理の内容
==	2つの値が等しい場合はTrueを返す
!=	2つの値が等しくない場合はTrueを返す
>	左辺が右辺より大きい場合はTrueを返す
>=	左辺が右辺以上の場合はTrueを返す
<	左辺が右辺より小さい場合はTrueを返す
<=	左辺が右辺以下の場合はTrueを返す

7章 等価演算子の注意点

データ型を合わせないと正しく比較できません

```
print("5" == 5)  # False（文字列と整数は等しくない）
```

int関数でデータ型を変換してから比較します

```
num1 = 5
num2 = "5"

if num1 == int(num2):
    print("num1とnum2は等しいです")
```

7章 if文を書いてみよう

3つのパターンで条件分岐を学びます

❶ ifのみ

もし○○であれば、●●する

❷ if + else

○○であれば●●し、それ以外は▲▲する

❸ if + elif + else

複数の条件で分岐する

「値が4であれば『大当たりです』と出力する」プログラム

```
import random

# 変数numに0〜4までのランダムな整数を代入する
num = random.randint(0, 4)

print(num)

# 変数numの値が4であれば、「大当たりです」と出力する
if num == 4:
    print("大当たりです")
```

elseで「それ以外のとき」の処理を追加します

```
if 条件式:
    条件が成り立つときの処理
else:
    条件が成り立たないときの処理
```

- elseを記述することで、条件式が成り立たないときにも処理を行える

「4なら大当たり、それ以外ならはずれ」と出力するプログラム

```
import random

num = random.randint(0, 4)
print(num)

if num == 4:
    print("大当たりです")
else:
    print("はずれです")
```

7章 パターン❸ if + elif + else

elifで条件式を複数作れます

```
if 条件式A:
    条件Aが成り立つときの処理
elif 条件式B:
    条件Bが成り立つときの処理
else:
    どの条件も成り立たないときの処理
```

- elifはいくつでも追加できる
- ifとelseは1つの条件分岐で一度しか記述できない

7章 パターン❸ if + elif + else

> 「4なら大当たり、3なら当たり、それ以外ならはずれ」

```
import random

num = random.randint(0, 4)
print(num)

if num == 4:
    print("大当たりです")
elif num == 3:
    print("当たりです")
else:
    print("はずれです")
```

7章 論理演算子とは

複数の条件式を組み合わせるときに使います

論理演算子	意味	説明
and	かつ	すべての条件が成り立つときにTrue
or	または	1つでも条件が成り立てばTrue

andとorを使った条件分岐

```
# すべての条件が成り立つ場合にのみ処理を行う
if 10 < num and num < 30:
    print("変数numは10より大きく、30より小さいです")

# 1つでも条件が成り立てば処理を行う
if num == 10 or num == 30:
    print("変数numは10または30です")
```

andとorの動作を確認するコード

```
import random

num = random.randint(0, 4)
print(num)

# and条件：numが1より大きく、かつ3より小さい（2のみ成立）
if 1 < num and num < 3:
    print("変数numは1より大きく、3より小さいです")
else:
    print("and条件が成り立ちませんでした")
```

or条件の例

```
# or条件：numが1または3（1か3のとき成立）
if num == 1 or num == 3:
    print("変数numは1または3です")
else:
    print("or条件が成り立ちませんでした")
```

7章 範囲判定の簡潔な書き方

1つの変数がある範囲内かどうか判定する場合

通常の書き方

```
if 1 < num and num < 3:
    処理
```

簡潔な書き方

```
if 1 < num < 3:
    処理
```

7章 まとめ

本章では以下の内容を学習しました

条件分岐とは

- 条件によって処理を分けること
- Pythonではif文を使う

if文の構文

- `if 条件式:` で条件分岐を開始
- `elif` で追加の条件を指定
- `else` でどの条件も成り立たないときの処理を指定

7章 まとめ（続き）

本章では以下の内容を学習しました

比較演算子

演算子	意味
`==`	等しい
`!=`	等しくない
`>` `>=`	より大きい / 以上
`<` `<=`	より小さい / 以下

7章 まとめ（続き）

本章では以下の内容を学習しました

論理演算子

- and（かつ）：すべての条件が成り立つときにTrue
- or（または）：1つでも条件が成り立てばTrue

条件分岐は様々な機能を実装するための基本です