Python基礎講座

ワーク集

2章 Pythonの基本的な書き方を理解しよう

ワーク

2章 ワーク1: print関数

問題

以下の出力結果になるようにコードを書いて実行してください。

1. `Hello, World!`
2. `私の名前は山田太郎です。`
3. `1 + 2 = 3`（計算結果を表示）
4. 複数行で「おはよう」「こんにちは」「こんばんは」を表示

2章 ワーク2: コメントとインデント

問題

以下のコードにはエラーがあります。正しく修正して実行してください。

```
# 年齢を判定するプログラム
age = 20
if age >= 18:
print("成人です")
else:
print("未成年です")
```

2章 ワーク3: チャレンジ - f文字列

問題

以下の変数を使って、指定された形式で出力するコードを書いてください。

```
name = "佐藤花子"
age = 25
city = "東京"
```

出力: 佐藤花子さん（25歳）は東京に住んでいます。

3章 データの扱い方を理解しよう

ワーク

3章 ワーク1: 演算子

問題

以下の計算を行うコードを書いて実行してください。

1. 15と7の足し算
2. 100から35を引く
3. 8と9の掛け算
4. 50を6で割った商（整数部分のみ）
5. 50を6で割った余り
6. 2の10乗

3章 ワーク2: 型変換

問題

以下のコードを実行するとエラーになります。エラーの原因を考え、正しく動作するように修正してください。

```
# エラーになるコード
price = "1500"
tax = 150
total = price + tax
print(total)
```

修正後の出力: `1650`

3章 ワーク3: チャレンジ - 消費税計算

問題

消費税（10%）を計算するプログラムを作成してください。

- 商品価格: 1980円
- 税込み価格を計算して表示
- 小数点以下は切り捨て（int関数を使用）

出力例: `税込み価格： 2178円`

4章 変数を理解しよう

ワーク

4章 ワーク1: BMI計算

問題

BMI（体格指数）を計算するプログラムを作成してください。

- BMI = 体重(kg) ÷ 身長(m)の2乗
- 身長: 170cm、体重: 65kg
- 結果を小数点第1位まで表示（round関数を使用）

出力例: `BMI: 22.5`

4章 ワーク2: チャレンジ - 単位変換プログラム

問題

以下の単位変換を行うプログラムを作成してください。

1. 摂氏温度を華氏温度に変換する
    - 華氏 = 摂氏 × 9 ÷ 5 + 32
    - 摂氏25度を華氏に変換して表示
2. キロメートルをマイルに変換する
    - マイル = キロメートル × 0.621371
    - 10kmをマイルに変換（小数点第2位まで）

5章 配列（リスト・タプル・セット）を理解しよう

ワーク

5章 ワーク1: リストの基本操作

問題

以下のリスト操作を行うコードを書いて実行してください。

```
fruits = ["apple", "banana", "orange"]
```

1. リストの要素数を取得する
2. 2番目の要素（banana）を取得する
3. リストの末尾に"grape"を追加する
4. "banana"を"melon"に変更する
5. 最初の要素を削除する

5章 ワーク2: リストの応用

問題

以下のリスト操作を行ってください。

```
numbers = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3]
```

1. リストを昇順にソートして表示
2. 最大値と最小値を取得して表示
3. 合計と平均を計算して表示

5章 ワーク3: チャレンジ - タプルとセット

問題1

以下のコードはエラーになります。なぜエラーになるか説明し、リストを使って同様の処理を実現してください。

```
coordinates = (10, 20)
coordinates[0] = 15  # エラー！
```

問題2

重複のあるリストから、重複を除いた新しいリストを作成してください。

```
numbers = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4]
# 出力: [1, 2, 3, 4]
```

6章 連想配列（ディクショナリ）を理解しよう

ワーク

6章 ワーク1: ディクショナリの基本

問題

以下のディクショナリ操作を行うコードを書いて実行してください。

```
person = {"name": "田中", "age": 30, "city": "東京"}
```

1. "name"の値を取得する
2. "email"キーに"tanaka@example.com"を追加する
3. "age"を31に更新する
4. "city"を削除する
5. すべてのキーを取得する

6章 ワーク2: チャレンジ - ディクショナリの応用

問題

商品管理のディクショナリを操作してください。

```
products = {
    "apple": {"price": 150, "stock": 20},
    "banana": {"price": 100, "stock": 30},
    "orange": {"price": 120, "stock": 15}
}
```

1. bananaの価格を取得して表示
2. appleの在庫を5減らす
3. 新商品"grape"（価格: 200, 在庫: 10）を追加
4. すべての商品の合計在庫数を計算して表示

7章 条件分岐のif文を理解しよう

ワーク

7章 ワーク1: if文の基本

問題

以下の条件分岐を実装して実行してください。

1. 変数 `score` が80以上なら「合格」、それ未満なら「不合格」と表示
2. 変数 `number` が正の数なら「正」、負の数なら「負」、0なら「ゼロ」と表示

```
score = 75
# ここにif文を書く

number = -5
# ここにif文を書く
```

7章 ワーク2: 成績判定プログラム

問題

点数に応じて成績を判定するプログラムを作成してください。

- 90点以上: A
- 80点以上90点未満: B
- 70点以上80点未満: C
- 60点以上70点未満: D
- 60点未満: F

```
score = 85
# ここに成績判定のif文を書く
```

7章 ワーク3: 論理演算子

問題

以下の条件を満たすif文を書いて実行してください。

1. `age` が18以上かつ65未満の場合に「働き盛り」と表示
2. `day` が「土曜日」または「日曜日」の場合に「週末」と表示

```
age = 30
# ここにif文を書く

day = "土曜日"
# ここにif文を書く
```

7章 ワーク4: チャレンジ - 入場料金計算

問題

映画館の入場料金を計算するプログラムを作成してください。

- 通常料金: 1800円
- 12歳以下: 1000円
- 65歳以上: 1200円

```
age = 10
# ここに料金計算のコードを書く
# 出力例: 入場料金: 1000円
```

8章 繰り返し処理のfor文を理解しよう

ワーク

8章 ワーク1: for文の基本

問題

以下の処理をfor文で実装して実行してください。

1. 1から10までの数を順番に表示する
2. リスト `["apple", "banana", "orange"]` の各要素を表示する
3. 1から100までの合計を計算する

8章 ワーク2: range関数

問題

range関数を使って以下の処理を実装してください。

1. 0から9までの数を表示する
2. 5から15までの数を表示する
3. 0から20までの偶数を表示する
4. 10から1までカウントダウンする

8章 ワーク3: 九九の表

問題

for文を使って九九の表（2の段から9の段まで）を作成してください。

出力例:

```
2 x 1 = 2
2 x 2 = 4
...
9 x 9 = 81
```

8章 ワーク4: チャレンジ - FizzBuzz

問題

1から30までの数について、以下のルールで出力するプログラムを作成してください。

- 3の倍数のときは「Fizz」
- 5の倍数のときは「Buzz」
- 3と5の両方の倍数のときは「FizzBuzz」
- それ以外はその数を出力

9章 繰り返し処理のwhile文を理解しよう

ワーク

9章 ワーク1: while文の基本

問題

以下の処理をwhile文で実装して実行してください。

1. 1から10までの数を表示する
2. 変数 `count` が0になるまで1ずつ減らしながら表示する（初期値5）

```
# 1. 1から10まで表示
num = 1
# ここにwhile文を書く

# 2. カウントダウン
count = 5
# ここにwhile文を書く
```

9章 ワーク2: break・continue

問題

以下の処理を実装してください。

1. 1から100まで繰り返し、50に到達したらループを抜ける
2. 1から20まで繰り返し、3の倍数はスキップして表示する

```
# 1. breakの使用
num = 1
while num <= 100:
    print(num)
    # 50になったらbreakする
    num += 1
# 2. continueの使用
for i in range(1, 21):
    # 3の倍数はcontinueでスキップ
    print(i)
```

9章 ワーク3: チャレンジ - 合計が100を超えるまで

問題

1から順に足していき、合計が100を超えたら終了するプログラムを作成してください。

```
total = 0
n = 1

# ここにwhile文を書く

print(f"{n}まで足すと{total}")  # 14まで足すと105
```

※早く終わった方向けの追加課題です

10章 関数を理解しよう

ワーク

10章 ワーク1: 関数の定義

問題

以下の関数を定義して実行してください。

1. 「Hello, World!」と表示する関数 `greet`
2. 2つの数を受け取り、合計を返す関数 `add`
3. 名前を受け取り、「こんにちは、○○さん」と表示する関数 `say_hello`

```
# 1. greet関数を定義
# 2. add関数を定義
# 3. say_hello関数を定義

# 動作確認
greet()
print(add(3, 5))
say_hello("田中")
```

10章 ワーク2: 戻り値のある関数

問題

以下の関数を作成してください。

半径を受け取り、円の面積を返す関数（$\pi = 3.14159$）

```
def circle_area(radius):
    # ここに処理を書く
    pass

# 動作確認
print(circle_area(5))  # 78.53975
```

10章 ワーク3: チャレンジ - デフォルト引数

問題

税込価格を計算する関数を作成してください。

- 関数名: `calculate_tax`
- 引数: `price`（税抜価格）、`tax_rate`（税率、デフォルト値0.1）
- 戻り値: 税込価格（整数）

```
def calculate_tax(price, tax_rate=0.1):
    # ここに処理を書く
    pass

# 動作確認
print(calculate_tax(1000))         # 1100
print(calculate_tax(1000, 0.08))  # 1080
```

11章 変数のスコープを理解しよう

ワーク

11章 ワーク1: ローカル変数とグローバル変数

以下のコードの出力結果を予測してください。予測後、実行して確認しましょう。

```
x = 10

def func1():
    x = 20
    print("func1内:", x)

def func2():
    print("func2内:", x)

func1()   # 出力予測: _____
func2()   # 出力予測: _____
print("グローバル:", x)  # 出力予測: _____
```

なぜfunc1内とグローバルで値が異なるのか説明できますか？

11章 ワーク2: チャレンジ - global文

問題

以下のコードを修正して、`counter`が正しく増加するようにしてください。

```
counter = 0

def increment():
    # ここを修正する
    counter = counter + 1

increment()
increment()
increment()
print(counter)  # 3と表示されるようにする
```

12章 クラスを理解しよう

ワーク

12章 ワーク1: クラスの基本

以下の仕様でクラスを作成して実行してください。

クラス名: `Dog`

属性:

- `name`（名前）
- `age`（年齢）

メソッド:

- `bark()`: 「ワンワン！」と表示
- `introduce()`: 「私は○○、○歳です」と表示

動作確認用コード

```
# ここにDogクラスを定義

# 動作確認
dog = Dog("ポチ", 3)
dog.bark()         # ワンワン！
dog.introduce()    # 私はポチ、3歳です
```

12章 ワーク2: チャレンジ - 銀行口座クラス

問題

以下の仕様で銀行口座クラスを作成してください。

クラス名: `BankAccount`

属性:

- `owner`（口座名義）
- `balance`（残高、初期値0）

メソッド:

- `deposit(amount)`: 入金（残高を増やす）
- `withdraw(amount)`: 出金（残高不足なら「残高不足」と表示）
- `get_balance()`: 現在の残高を返す

12章 ワーク2: チャレンジ - 銀行口座クラス（続き）

動作確認用コード

```
# BankAccountクラスを定義後、以下で動作確認
account = BankAccount("田中太郎")
account.deposit(10000)
print(account.get_balance())  # 10000

account.withdraw(3000)
print(account.get_balance())  # 7000

account.withdraw(10000)  # 残高不足
```

13章 日付・時刻の処理を理解しよう

ワーク

13章 ワーク1: datetimeモジュール

問題

以下の処理を実装して実行してください。

```
from datetime import datetime, timedelta

# 1. 現在の日時を取得して表示

# 2. 2025年12月31日のdatetimeオブジェクトを作成

# 3. 今日から100日後の日付を計算

# 4. 2つの日付の差（日数）を計算
```

13章 ワーク2: チャレンジ - 日付フォーマット

問題

strftime()を使って、現在の日時を以下の形式で表示してください。

1. `2025年12月05日`
2. `2025/12/05 14:30:00`

```
from datetime import datetime

now = datetime.now()
# 1. 「2025年12月05日」形式で表示

# 2. 「2025/12/05 14:30:00」形式で表示
```

ヒント: `%Y`=年、`%m`=月、`%d`=日、`%H`=時、`%M`=分、`%S`=秒

15章 実践課題

総合演習問題

実践課題1: 成績管理プログラム

課題

以下の機能を持つ成績管理プログラムを作成してください。

```
students = [
    {"name": "田中", "score": 85},
    {"name": "佐藤", "score": 72},
    {"name": "鈴木", "score": 90},
    {"name": "高橋", "score": 68},
    {"name": "伊藤", "score": 95}
]
```

1. 全生徒の平均点を計算して表示
2. 最高点の生徒名と点数を表示
3. 80点以上の生徒を一覧表示

実践課題2: 買い物カート

課題

以下のデータを使って、買い物カートの合計金額を計算してください。

```
cart = [
    {"name": "りんご", "price": 150, "quantity": 3},
    {"name": "バナナ", "price": 100, "quantity": 2},
    {"name": "オレンジ", "price": 120, "quantity": 5}
]
```

1. 各商品の小計を計算して表示
2. 合計金額を計算して表示
3. 消費税10%を加えた税込合計を表示

実践課題3: チャレンジ - 数当てゲーム

課題

1から100までのランダムな数を当てるゲームを作成してください。

- 正解より大きい場合は「もっと小さい」と表示
- 正解より小さい場合は「もっと大きい」と表示
- 正解したら「正解！」と試行回数を表示

```
import random

answer = random.randint(1, 100)
count = 0

# ここにゲームのロジックを書く
# while文とinput()を使用
```

実践課題4: チャレンジ - 簡易電卓

課題

2つの数と演算子を入力して計算する電卓を作成してください。

- 対応する演算: +, -, *, /
- 0で割った場合はエラーメッセージを表示

```
num1 = float(input("1つ目の数: "))
operator = input("演算子 (+, -, *, /): ")
num2 = float(input("2つ目の数: "))

# ここに計算ロジックを書く
```

実践課題5: チャレンジ - じゃんけんゲーム

課題

コンピュータとじゃんけんをするゲームを作成してください。

- グー: 0、チョキ: 1、パー: 2
- 勝敗を判定して結果を表示
- 3回勝負して、勝ち数を表示

```
import random

# ここにじゃんけんゲームを実装
```