

# Python基礎講座

---

## ワーク集 解答例

## 2章 Pythonの基本的な書き方を理解しよう

---

ワーク 解答例

## 2章 ワーク1: print関数 解答例

```
print("Hello, World!")
print("私の名前は山田太郎です。")
print("1 + 2 =", 1 + 2)
print(f"1 + 2 = {1 + 2}")          # 1. Hello, World!
                                         # 2. 名前の表示
                                         # 3. 計算結果を表示
                                         # 3. f文字列でも可

print("おはよう")
print("こんにちは")
print("こんばんは")                 # 4. 複数行で表示
```

## 2章 ワーク2: コメントとインデント 解答例

### 問題のエラー原因

if文やelse文の中の処理にはインデント（字下げ）が必要です。

```
age = 20
if age >= 18:
    print("成人です")      # ← インデントを追加
else:
    print("未成年です")      # ← インデントを追加
```

出力結果: 成人です

## 2章 ワーク3: チャレンジ - f文字列 解答例

```
name = "佐藤花子"  
age = 25  
city = "東京"  
print(f"{name}さん ({age}歳) は{city}に住んでいます。")
```

### 出力結果

```
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 解答例

# 3章 ワーク1: 演算子 解答例

```
print(15 + 7)      # 1. 足し算 → 22
print(100 - 35)    # 2. 引き算 → 65
print(8 * 9)       # 3. 掛け算 → 72
print(50 // 6)     # 4. 商 (整数部分) → 8
print(50 % 6)      # 5. 余り → 2
print(2 ** 10)     # 6. 2の10乗 → 1024
```

## 3章 ワーク2: 型変換 解答例

### エラーの原因

price が文字列型、 tax が整数型のため、 + 演算子でエラーになります。

```
price = "1500"
tax = 150
total = int(price) + tax    # priceを整数型に変換
print(total)                # 出力: 1650
```

### 3章 ワーク3: チャレンジ - 消費税計算 解答例

```
price = 1980
tax_rate = 0.1
total = int(price * (1 + tax_rate))    # 小数点以下は切り捨て
print(f"税込み価格: {total}円")
```

#### 出力結果

```
税込み価格: 2178円
```

## 4章 変数を理解しよう

---

### ワーク 解答例

## 4章 ワーク1: BMI計算 解答例

```
height_cm = 170
weight = 65
height_m = height_cm / 100      # メートルに変換
bmi = weight / (height_m ** 2)  # BMI計算
bmi_rounded = round(bmi, 1)    # 小数点第1位まで
print(f"BMI: {bmi_rounded}")
```

出力結果: **BMI: 22.5**

## 4章 ワーク2: チャレンジ - 単位変換プログラム 解答例

```
# 1. 摂氏→華氏
celsius = 25
fahrenheit = celsius * 9 / 5 + 32
print(f"摄氏{celsius}度 = 華氏{fahrenheit}度") # 華氏77.0度

# 2. キロ→マイル
km = 10
miles = round(km * 0.621371, 2)
print(f"{km}km = {miles}マイル") # 6.21マイル
```

# 5章 配列（リスト・タプル・セット）を理解しよう

---

## ワーク 解答例

# 5章 ワーク1: リストの基本操作 解答例

```
fruits = ["apple", "banana", "orange"]
print(len(fruits))          # 1. 要素数 → 3
print(fruits[1])            # 2. 2番目の要素 → banana
fruits.append("grape")      # 3. 末尾に追加
print(fruits)                # ['apple', 'banana', 'orange', 'grape']
fruits[1] = "melon"          # 4. "banana"を"melon"に変更
print(fruits)                # ['apple', 'melon', 'orange', 'grape']
del fruits[0]                # 5. 最初の要素を削除
print(fruits)                # ['melon', 'orange', 'grape']
```

## 5章 ワーク2: リストの応用 解答例

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

sorted_numbers = sorted(numbers)      # 1. 昇順ソート
print(sorted_numbers)                # [1, 1, 2, 3, 3, 4, 5, 5, 6, 9]
print(f"最大値: {max(numbers)}")     # 2. 最大値 → 9
print(f"最小値: {min(numbers)}")     # 2. 最小値 → 1

total = sum(numbers)                # 3. 合計と平均
average = total / len(numbers)
print(f"合計: {total}, 平均: {average}") # 合計: 39, 平均: 3.9
```

# 5章 ワーク3: チャレンジ - タプルとセット 解答例

## 問題1: タプルのエラー

タプルは変更不可（イミュータブル）なため、要素の変更ができません。

```
coordinates = [10, 20]    # リストに変更すれば変更可能
coordinates[0] = 15
print(coordinates)        # [15, 20]
```

## 問題2: 重複を除いたリスト

```
numbers = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4]
unique = sorted(list(set(numbers)))  # [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 解答例

# 6章 ワーク1: ディクショナリの基本 解答例

```
person = {"name": "田中", "age": 30, "city": "東京"}  
print(person["name"])                      # 1. 値取得 → 田中  
person["email"] = "tanaka@example.com"      # 2. キー追加  
person["age"] = 31                           # 3. 値更新  
del person["city"]                         # 4. キー削除  
print(person.keys())                       # 5. 全キー取得  
  
# dict_keys(['name', 'age', 'email'])
```

## 6章 ワーク2: チャレンジ - ディクショナリの応用 解答例

```
products = {  
    "apple": {"price": 150, "stock": 20},  
    "banana": {"price": 100, "stock": 30},  
    "orange": {"price": 120, "stock": 15}  
}  
print(products['banana']['price'])          # 1. 価格取得 → 100  
products["apple"]["stock"] -= 5             # 2. 在庫を5減らす  
products["grape"] = {"price": 200, "stock": 10} # 3. 新商品追加  
total_stock = sum(p["stock"] for p in products.values())  
print(f"合計在庫数: {total_stock}")         # 4. → 60
```

## 7章 条件分岐のif文を理解しよう

---

ワーク 解答例

## 7章 ワーク1: if文の基本 解答例

```
# 1. 合格/不合格判定
score = 75
if score >= 80:
    print("合格")
else:
    print("不合格")    # ← こちらが出力される

# 2. 正/負/ゼロ判定
number = -5
if number > 0:
    print("正")
elif number < 0:
    print("負")        # ← こちらが出力される
else:
    print("ゼロ")
```

## 7章 ワーク2: 成績判定プログラム 解答例

```
score = 85
if score >= 90:
    grade = "A"
elif score >= 80:
    grade = "B"
elif score >= 70:
    grade = "C"
elif score >= 60:
    grade = "D"
else:
    grade = "F"
print(f"点数: {score}点 → 成績: {grade}")
```

出力結果: 点数: 85点 → 成績: B

## 7章 ワーク3: 論理演算子 解答例

```
# 1. 18以上かつ65未満の場合に「働き盛り」  
age = 30  
if age >= 18 and age < 65:  
    print("働き盛り")  
  
# 2. 「土曜日」または「日曜日」の場合に「週末」  
day = "土曜日"  
if day == "土曜日" or day == "日曜日":  
    print("週末")
```

## 7章 ワーク4: チャレンジ - 入場料金計算 解答例

```
age = 10

if age <= 12:
    price = 1000
elif age >= 65:
    price = 1200
else:
    price = 1800

print(f"入場料金: {price}円")
```

出力結果: 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 解答例

# 8章 ワーク1: for文の基本 解答例

```
# 1. 1から10までの数を順番に表示
for i in range(1, 11):
    print(i)

# 2. リストの各要素を表示
fruits = ["apple", "banana", "orange"]
for fruit in fruits:
    print(fruit)

# 3. 1から100までの合計を計算
total = 0
for i in range(1, 101):
    total += i
print(f"合計: {total}") # 合計: 5050
```

## 8章 ワーク2: range関数 解答例

# 1. 0から9までの数を表示

```
for i in range(10):  
    print(i)
```

# 2. 5から15までの数を表示

```
for i in range(5, 16):  
    print(i)
```

# 3. 0から20までの偶数を表示

```
for i in range(0, 21, 2):  
    print(i)
```

# 4. 10から1までカウントダウン

```
for i in range(10, 0, -1):  
    print(i)
```

## 8章 ワーク3: 九九の表 解答例

```
# 九九の表 (2の段から9の段まで)
for i in range(2, 10):
    for j in range(1, 10):
        print(f"{i} x {j} = {i * j}")
print() # 段の区切りに空行
```

### 出力結果 (一部)

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

## 8章 ワーク4: チャレンジ - FizzBuzz 解答例

```
for i in range(1, 31):
    if i % 3 == 0 and i % 5 == 0:
        print("FizzBuzz")
    elif i % 3 == 0:
        print("Fizz")
    elif i % 5 == 0:
        print("Buzz")
    else:
        print(i)
```

出力例: 1, 2, Fizz, 4, Buzz, Fizz, 7, 8, Fizz, Buzz, 11, Fizz,  
13, 14, FizzBuzz...

# 9章 繰り返し処理のwhile文を理解しよう

---

## ワーク 解答例

## 9章 ワーク1: while文の基本 解答例

```
# 1. 1から10まで表示
num = 1
while num <= 10:
    print(num)
    num += 1

# 2. カウントダウン
count = 5
while count > 0:
    print(count)
    count -= 1
print("発射!")
```

出力: 1, 2, ..., 10 / 5, 4, 3, 2, 1, 発射 !

# 9章 ワーク2: break・continue 解答例

```
# 1. breakの使用 (50で停止)
num = 1
while num <= 100:
    print(num)
    if num == 50:
        break
    num += 1

# 2. continueの使用 (3の倍数をスキップ)
for i in range(1, 21):
    if i % 3 == 0:
        continue
    print(i)
# 出力: 1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 20
```

## 9章 ワーク3: チャレンジ - 合計が100を超えるまで 解答例

```
total = 0
n = 1
while total <= 100:
    total += n
    n += 1
n -= 1 # ループを抜けた時点でnは1つ進んでいるので-1
print(f"{n}まで足すと{total}")
```

出力: 14まで足すと105

※  $1+2+3+\dots+14 = 105$

# 10章 関数を理解しよう

---

## ワーク 解答例

# 10章 ワーク1: 関数の定義 解答例

```
def greet():                      # 1. greet関数
    print("Hello, World!")

def add(a, b):                    # 2. add関数
    return a + b

def say_hello(name):              # 3. say_hello関数
    print(f"こんにちは、{name}さん")

# 動作確認
greet()                          # Hello, World!
print(add(3, 5))                 # 8
say_hello("田中")                # こんにちは、田中さん
```

# 10章 ワーク2: 戻り値のある関数 解答例

```
def circle_area(radius):
    pi = 3.14159
    return pi * radius ** 2

print(circle_area(5)) # 78.53975
```

## 解説

$$\text{円の面積} = \pi r^2 \rightarrow 3.14159 \times 5^2 = 78.53975$$

# 10章 ワーク3: チャレンジ - デフォルト引数 解答例

```
def calculate_tax(price, tax_rate=0.1):
    return int(price * (1 + tax_rate))

print(calculate_tax(1000))      # 1100 (税率10%)
print(calculate_tax(1000, 0.08)) # 1080 (税率8%)
```

## 解説

税率を省略すると0.1 (10%) が適用されます。

# 11章 変数のスコープを理解しよう

---

## ワーク 解答例

# 11章 ワーク1: ローカル変数とグローバル変数 解答例

```
x = 10

def func1():
    x = 20                  # ローカル変数 (新しいx)
    print("func1内:", x)

def func2():
    print("func2内:", x)    # グローバル変数を参照

func1()                      # func1内: 20
func2()                      # func2内: 10
print("グローバル:", x)      # グローバル: 10
```

# 11章 ワーク2: チャレンジ - global文 解答例

```
counter = 0

def increment():
    global counter  # グローバル変数を使用することを宣言
    counter = counter + 1

increment()
increment()
increment()
print(counter)  # 3
```

# 12章 クラスを理解しよう

---

## ワーク 解答例

# 12章 ワーク1: クラスの基本 解答例

```
class Dog:  
    def __init__(self, name, age):  
        self.name = name  
        self.age = age  
  
    def bark(self):  
        print("ワンワン！")  
  
    def introduce(self):  
        print(f"私は{self.name}、{self.age}歳です")  
  
# 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: 銀行口座クラス 解答例 (1/2)

```
class BankAccount:  
    def __init__(self, owner):  
        self.owner = owner  
        self.balance = 0  
  
    def deposit(self, amount):  
        self.balance += amount  
        print(f"{amount}円入金しました")  
  
    def withdraw(self, amount):  
        if amount > self.balance:  
            print("残高不足")  
        else:  
            self.balance -= amount  
            print(f"{amount}円出金しました")  
  
    def get_balance(self):  
        return self.balance
```

## 12章 ワーク2: 銀行口座クラス 解答例 (2/2)

```
# 動作確認
account = BankAccount("田中太郎")
account.deposit(10000)          # 10000円入金しました
print(account.get_balance())    # 10000
account.withdraw(3000)          # 3000円出金しました
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 解答例

# 13章 ワーク1: datetimeモジュール 解答例

```
from datetime import datetime, timedelta

now = datetime.now()
print(f"現在: {now}")                                # 1. 現在日時

new_year = datetime(2025, 12, 31)
print(f"指定日: {new_year}")                          # 2. 指定日

future = now + timedelta(days=100)
print(f"100日後: {future}")                          # 3. 100日後

diff = new_year - now
print(f"日数差: {diff.days}日")                      # 4. 日数差
```

# 13章 ワーク2: チャレンジ - 日付フォーマット 解答例

```
from datetime import datetime  
  
now = datetime.now()  
  
formatted1 = now.strftime("%Y年%m月%d日")  
print(formatted1)    # 2025年12月07日  
  
formatted2 = now.strftime("%Y/%m/%d %H:%M:%S")  
print(formatted2)    # 2025/12/07 14:30:00
```

記号: %Y 年, %m 月, %d 日, %H 時, %M 分, %S 秒

# 15章 実践課題

---

## 解答例

# 実践課題1：成績管理プログラム 解答例

```
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}
]

avg = sum(s["score"] for s in students) / len(students)
print(f"平均点: {avg}点") # 1. 平均点 → 82.0点

top = max(students, key=lambda s: s["score"])
print(f"最高点: {top['name']}さん ({top['score']}点)") # 2. 最高点

for s in students: # 3. 80点以上の生徒一覧
    if s["score"] >= 80:
        print(f" {s['name']}: {s['score']}点")
```

## 実践課題2: 買い物カート 解答例

```
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}
]

total = 0
for item in cart:
    subtotal = item["price"] * item["quantity"]
    print(f"{item['name']}: {subtotal}円") # 1. 小計
    total += subtotal

print(f"合計: {total}円") # 2. 合計 → 1250円
tax_total = int(total * 1.1)
print(f"税込合計: {tax_total}円") # 3. 税込 → 1375円
```

# 実践課題3: 数当てゲーム 解答例

```
import random

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

while True:
    guess = int(input("1~100の数を入力: "))
    count += 1

    if guess == answer:
        print(f"正解！{count}回で当たりました！")
        break
    elif guess > answer:
        print("もっと小さい")
    else:
        print("もっと大きい")
```

## 実践課題4: 簡易電卓 解答例 (1/2)

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

if operator == "+":
    result = num1 + num2
elif operator == "-":
    result = num1 - num2
elif operator == "*":
    result = num1 * num2
elif operator == "/":
    if num2 == 0:
        print("エラー: 0で割ることはできません")
        result = None
    else:
        result = num1 / num2
```

## 実践課題4: 簡易電卓 解答例 (2/2)

```
else:  
    print("無効な演算子です")  
    result = None  
  
if result is not None:  
    print(f"結果: {result}")
```

# 実践課題5: ジャンケンゲーム 解答例 (1/2)

```
import random

hands = ["グー", "チョキ", "パー"]
wins = 0

for i in range(3):
    print(f"\n--- {i+1}回目 ---")
    player = int(input("グー(0), チョキ(1), パー(2): "))
    computer = random.randint(0, 2)
    print(f"あなた: {hands[player]} vs PC: {hands[computer]}")
```

## 実践課題5: ジャンケンゲーム 解答例 (2/2)

```
if player == computer:  
    print("あいこ")  
elif (player - computer) % 3 == 2:  
    print("あなたの勝ち！")  
    wins += 1  
else:  
    print("あなたの負け...")  
  
print(f"\n結果: {wins}勝/{3-wins}敗")
```