

# 情報科教諭向け Pythonプログラミング基礎

## ～ 演習課題集 ～

# 演習課題 目次

## print

- ☐ 演習課題 1
- ☐ 演習課題 2

## 選択処理

- ☐ 演習課題 3
- ☐ 演習課題 4
- ☐ 演習課題 5
- ☐ 演習課題 6

## 繰返し処理

- ☐ 演習課題 7
- ☐ 演習課題 8
- ☐ 演習課題 9
- ☐ 演習課題 1 0

## 選択処理と繰返し処理

- ☐ 演習課題 1 1

## リスト

- ☐ 演習課題 1 2
- ☐ 演習課題 1 3
- ☐ 演習課題 1 4

## リストと繰返し処理

- ☐ 演習課題 1 5

## 発展課題【難】

- ☐ 発展課題 1 選択処理の応用 1
- ☐ 発展課題 2 選択処理の応用 2
- ☐ 発展課題 3 繰返し処理の応用 1
- ☐ 発展課題 4 繰返し処理の応用 2
- ☐ 発展課題 5 リストの応用

# 演習課題 1

---

printを使った表示

# 演習課題 1（問題）

---

1. 変数 name に 自分の名前を代入する
2. 「名前は」を表示した後に  
変数 name の値を表示する

# 演習課題 1 （実行例）

---

[実行結果例]（ name に 'KCS 太郎' を代入 ）

Print output (drag lower right corner to resize)

名前は KCS 太郎

# 演習課題 1（プログラム）

---

作成したプログラムを以下に転記しましょう。

# 演習課題 2

---

printを使った改変

## 演習課題 2 （問題）

---

1. 変数 name に 自分の名前を代入する
2. 変数 age に 年齢を代入する
3. 「名前：」を表示した後に変数 name の値を表示し、  
「年齢：」を表示してから変数 age の値を表示する



## 演習課題 2 （実行例）

---

[実行結果例]（ name に 'KCS 太郎'、 age に 20 を代入 ）

Print output (drag lower right corner to resize)

名前： KCS 太郎 年齢： 20

# 演習課題 2 （プログラム）

---

作成したプログラムを以下に転記しましょう。

# 演習課題 3

---

選択処理 1

# 演習課題 3 （問題）

---

1. 変数 name に 自分の名前を代入する
2. 変数 age に 年齢を代入する
3. 「名前：」を表示した後に変数 name の値を表示し、  
「年齢：」を表示してから変数 age の値を表示する
4. 年齢が20以上なら「成人です」を表示する  
(20未満なら何もしない)

## 演習課題 3 （実行例）

---

[実行結果例]（ name に 'KCS 太郎'、 age に 20 を代入 ）

Print output (drag lower right corner to resize)

```
名前： KCS 太郎 年齢： 20  
成人です
```

[実行結果例]（ name に 'KCS 太郎'、 age に 19 を代入 ）

Print output (drag lower right corner to resize)

```
名前： KCS 太郎 年齢： 19
```

# 演習課題 3 （プログラム）

---

作成したプログラムを以下に転記しましょう。

# 演習課題 4

---

選択処理 2

## 演習課題 4 （問題）

---

1. 変数 name に 自分の名前を代入する
2. 変数 age に 年齢を代入する
3. 「名前：」を表示した後に変数 name の値を表示し、  
「年齢：」を表示してから変数 age の値を表示する
4. 年齢が20以上なら「飲酒はほどほどに」を表示する  
そうでなければ「飲酒はいけません」を表示する



## 演習課題 4 （実行例）

---

[実行結果例]（ name に 'KCS 太郎'、 age に 20 を代入 ）

Print output (drag lower right corner to resize)

```
名前： KCS 太郎 年齢： 20  
飲酒はほどほどに
```

[実行結果例]（ name に 'KCS 太郎'、 age に 19 を代入 ）

Print output (drag lower right corner to resize)

```
名前： KCS 太郎 年齢： 19  
飲酒はいけません
```

# 演習課題 4 （プログラム）

---

作成したプログラムを以下に転記しましょう。

# 演習課題 5

---

選択処理 3

# 演習課題 5 （問題）

---

1. 変数 name に 自分の名前、 age に 年齢を代入する
2. 「名前：」を表示した後に変数 name の値を表示し、  
「年齢：」を表示してから変数 age の値を表示する
3. 年齢が 6以下なら「園児」を表示する。そうでなくて  
年齢が12以下なら「児童」を表示する。そうでなくて  
年齢が18以下なら「生徒」を表示する。そうでない場合  
は「学生」を表示する。

# 演習課題 5（実行例）

## [実行結果例 1]

Print output (drag lower right corner to resize)

名前： KOS 太郎 年齢： 6  
園児

## [実行結果例 2]

Print output (drag lower right corner to resize)

名前： KOS 太郎 年齢： 12  
児童

## [実行結果例 3]

Print output (drag lower right corner to resize)

名前： KOS 太郎 年齢： 18  
生徒

## [実行結果例 4]

Print output (drag lower right corner to resize)

名前： KOS 太郎 年齢： 19  
学生

# 演習課題 5 （プログラム）

---

作成したプログラムを以下に転記しましょう。

# 演習課題 6

---

選択処理の応用

# 演習課題 6 (問題)

---

1. 変数 age に 年齢、gender に TrueかFalse を代入する
2. gender が True なら「女性」を表示した後、age を -5 する。  
そうでなければ「男性」を表示する。
3. 年齢が0未満なら「エラー」を表示する。そうでなくて年齢が18  
以下なら「未成年」を表示する。それ以外は「成人」を表示する。
4. 「年齢：」を表示してから変数 age の値を表示する。
5. gender が True なら「(実際の年齢は秘密)」を表示する。



# 演習課題 6（実行例）

## [実行結果例 1]

Print output (drag lower right corner to resize)

```
女性  
未成年  
年齢： 18  
(実際の年齢は秘密)
```

## [実行結果例 3]

Print output (drag lower right corner to resize)

```
男性  
成人  
年齢： 23
```

## [実行結果例 5]

Print output (drag lower right corner to resize)

```
女性  
エラー  
年齢： -1  
(実際の年齢は秘密)
```

## [実行結果例 2]

Print output (drag lower right corner to resize)

```
女性  
成人  
年齢： 19  
(実際の年齢は秘密)
```

## [実行結果例 4]

Print output (drag lower right corner to resize)

```
男性  
成人  
年齢： 24
```

## [実行結果例 6]

Print output (drag lower right corner to resize)

```
男性  
エラー  
年齢： -1
```

# 演習課題 6（プログラム）

---

作成したプログラムを以下に転記しましょう。

# 演習課題 7

---

繰返し処理 1

# 演習課題 7 (問題)

---

1. 変数 cnt に 1 を代入する
2. 繰返し(while)を使い、  
1～10までの数値を表示する。

# 演習課題 7 （実行例）

---

## [実行結果例]

Print output (drag lower right corner to resize)

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

# 演習課題 7（プログラム）

---

作成したプログラムを以下に転記しましょう。

# 演習課題 8

---

繰返し処理 2

## 演習課題 8 （問題）

---

1. 変数 sum に 0、cnt に 1 を代入する
2. 繰返し(while)を使い、  
1～100までの和を変数sumに求める
3. 「1 から 100 までの合計：」を表示した後に  
変数sumの値を表示する。



# 演習課題 8（実行例）

---

## [実行結果例]

Print output (drag lower right corner to resize)

```
1 から 100 までの合計： 5050
```

# 演習課題 8（プログラム）

---

作成したプログラムを以下に転記しましょう。

# 演習課題 9

---

繰返し処理 3

## 演習課題 9 （問題）

---

1. 変数 `sum` に 0 を代入する。
2. 繰返し(for)を使い、変数 `cnt` の値を1～100まで変化させて、1～100までの和を変数`sum`に求める。
3. 「1 から 100 までの合計：」を表示した後に  
変数`sum`の値を表示する。

# 演習課題 9（実行例）

---

## [実行結果例]

Print output (drag lower right corner to resize)

```
1 から 100 までの合計： 5050
```

# 演習課題 9 （プログラム）

---

作成したプログラムを以下に転記しましょう。

# 演習課題 1 0

---

繰返し処理 4

# 演習課題 1 0 （問題）

---

1. 変数 sum に 0 を代入する。
2. 繰返し(for)を使い、変数 cnt の値を2～100まで2ずつ変化させて、偶数の和を変数sumに求める。
3. 「100 までの偶数の合計：」を表示した後に変数sumの値を表示する。



# 演習課題 1 0 （実行例）

---

## [実行結果例]

Print output (drag lower right corner to resize)

```
100 までの偶数の合計： 2550
```

# 演習課題 10（プログラム）

---

作成したプログラムを以下に転記しましょう。

# 演習課題 1 1

---

繰返し処理と選択処理

# 演習課題 1 1 (問題)

---

1. 変数 sum に 0 を代入する。
2. 繰返し(for)を使い、変数 cnt の値を0～100まで1ずつ変化させて、奇数の和を変数sumに求める。  
(0 は 偶数でも奇数でもないものとする)
3. 「100 までの奇数の合計：」を表示した後に  
変数sumの値を表示する。

# 演習課題 1 1 （実行例）

---

## [実行結果例]

Print output (drag lower right corner to resize)

```
100 までの奇数の合計： 2500
```

# 演習課題 1 1（プログラム）

---

作成したプログラムを以下に転記しましょう。

# 演習課題 1 2

---

リストを利用 1

# 演習課題 1 2 (問題)

---

1. リスト coins に 1, 5, 10, 50, 100, 500 を代入する。
2. メッセージ表示後 coins の 全ての値を表示する
3. メッセージ表示後 coins の 1番地の値を表示する
4. メッセージ表示後 coins の 2～4番地の値を表示する
5. メッセージ表示後 coins の 3番地以降の値を表示する
6. メッセージ表示後 coins の 3番地未満の値を表示する



# 演習課題 1 2 （実行例）

## [実行結果例]

Print output (drag lower right corner to resize)

```
全ての値： [1, 5, 10, 50, 100, 500]  
1番地の値： 5  
2～4番地の値： [10, 50, 100]  
3番地以降の値： [50, 100, 500]  
3番地未満の値： [1, 5, 10]
```

# 演習課題 1 2（プログラム）

---

作成したプログラムを以下に転記しましょう。

# 演習課題 1 3

---

リストを利用 2

# 演習課題 1 3 (問題)

---

1. リスト coins に 1, 5, 10, 50, 100, 500 を代入する。
2. coinsの末尾に 2000 を追加して全要素を表示する
3. coinsの6番地に 1000 を挿入して全要素を表示する
4. coinsの6番地の要素を削除した後、coinsの 2000 の要素を削除する。
5. 「現在のcoinsは」を表示してcoinsの全要素を表示する

# 演習課題 1 3 (実行例)

## [実行結果例]

Print output (drag lower right corner to resize)

```
[1, 5, 10, 50, 100, 500, 2000]  
[1, 5, 10, 50, 100, 500, 1000, 2000]  
現在のcoinsは [1, 5, 10, 50, 100, 500]
```

# 演習課題 1 3 （プログラム）

---

作成したプログラムを以下に転記しましょう。

# 演習課題 1 4

---

リストの並び替え

# 演習課題 1 4 （問題）

---

1. リスト values に 4, 7, 3, 2, 9, 5 を代入する
2. 現在のvaluesの全要素をメッセージと共に表示する
3. valuesを昇順に並び替え、valuesの全要素を表示する
4. valuesの3番目に小さい値をメッセージと共に表示する
5. valuesを降順に並び替え、valuesの全要素を表示する
6. valuesの3番目に大きい値をメッセージと共に表示する



# 演習課題 1 4 (実行例)

[実行結果例]

Print output (drag lower right corner to resize)

```
valuesの初期値 : [4, 7, 3, 2, 9, 5]  
[2, 3, 4, 5, 7, 9]  
3番目に小さい値 : 4  
[9, 7, 5, 4, 3, 2]  
3番目に大きい値 : 5
```

# 演習課題 1 4 （プログラム）

---

作成したプログラムを以下に転記しましょう。

# 演習課題 1 5

---

リストと繰返しを利用

# 演習課題 1 5 (問題)

---

1. リスト even に 0, 2 を代入する。
2. 現在の全要素をメッセージと共に表示する
3. 繰返し(for)を使い、変数 cnt の値を1～9まで変化させ、  
2の倍数の値をリストの末尾に追加していく
4. 現在の全要素をメッセージと共に表示する

# 演習課題 1 5 （実行例）

---

[実行結果例]

Print output (drag lower right corner to resize)

```
処理前： [0, 2]
```

```
処理後： [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

# 演習課題 1 5 （プログラム）

---

作成したプログラムを以下に転記しましょう。

# 発展課題 1

---

選択処理の応用 1

(基本情報科目Bサンプル問題改変 - 年齢と金額による入場料計算)

# 発展課題 1（問題）

---

1. 変数 gate に 入場料、変数 age に年齢を代入してメッセージとともに表示する。  
(入場料は1円単位が発生しないものとする)
2. gate が 1000以上5000未満なら gate を100円引き、  
gate が 5000以上10000未満なら gate を500円引き、  
gate が 10000以上なら gate を1000円引きする。
3. age が 6以下 なら gate の半額、それ以外はgateの金額  
を「支払いは XXXX 円」という形式になるように表示する。



# 発展課題 1（実行例）

## [実行結果例 1]

Print output (drag lower right corner to resize)

入場料： 900 年齢： 7  
支払いは 900 円

## [実行結果例 3]

Print output (drag lower right corner to resize)

入場料： 1000 年齢： 7  
支払いは 900 円

## [実行結果例 5]

Print output (drag lower right corner to resize)

入場料： 5000 年齢： 7  
支払いは 4500 円

## [実行結果例 7]

Print output (drag lower right corner to resize)

入場料： 10000 年齢： 7  
支払いは 9000 円

## [実行結果例 2]

Print output (drag lower right corner to resize)

入場料： 900 年齢： 6  
支払いは 450 円

## [実行結果例 4]

Print output (drag lower right corner to resize)

入場料： 1000 年齢： 6  
支払いは 450 円

## [実行結果例 6]

Print output (drag lower right corner to resize)

入場料： 5000 年齢： 6  
支払いは 2250 円

## [実行結果例 8]

Print output (drag lower right corner to resize)

入場料： 10000 年齢： 6  
支払いは 4500 円

# 発展課題 1（プログラム）

---

作成したプログラムを以下に転記しましょう。

# 発展課題 2

---

選択処理の応用 2

(うるう年の判断)

## 発展課題 2（問題）

---

1. 変数 `year` に 西暦 を代入し、西暦を表示する
2. うるう年の場合は「うるう年です」を表示し、うるう年ではない場合は「うるう年ではありません」を表示する。

[うるう年]

西暦年号が4で割り切れる年をうるう年とする。ただし、西暦年号が100で割り切れて400で割り切れない年は平年とする。

## 発展課題 2（実行例）

### [実行結果例 1]

Print output (drag lower right corner to resize)

2024  
うるう年です

### [実行結果例 2]

Print output (drag lower right corner to resize)

2025  
うるう年ではありません

### [実行結果例 3]

Print output (drag lower right corner to resize)

2100  
うるう年ではありません

### [実行結果例 4]

Print output (drag lower right corner to resize)

2000  
うるう年です

# 発展課題 2（プログラム）

---

作成したプログラムを以下に転記しましょう。

# 発展課題 3

---

繰返し処理の応用 1

(2の累乗の計算[足し算のみ利用])

## 発展課題 3 （問題）

1. 変数 `sum` に 1 を代入する。
2. 繰返し(`while`もしくは`for`)を使い、変数 `cnt` の値を1～10まで1ずつ変化させて、2の1乗から2の10乗までの数値をメッセージと共に表示する。ただし、べき乗や乗算を利用せず、足し算のみで算出すること。

※ 2の1乗は (2の0乗 + 2の0乗)、2の2乗は (2の1乗 + 2の1乗)、  
2の3乗は (2の2乗 + 2の2乗)、2の4乗は (2の3乗 + 2の3乗)  
...



# 発展課題 3 （実行例）

## [実行結果例]

Print output (drag lower right corner to resize)

```
2 の 1 乗 : 2
2 の 2 乗 : 4
2 の 3 乗 : 8
2 の 4 乗 : 16
2 の 5 乗 : 32
2 の 6 乗 : 64
2 の 7 乗 : 128
2 の 8 乗 : 256
2 の 9 乗 : 512
2 の 10 乗 : 1024
```

# 発展課題 3 （プログラム）

---

作成したプログラムを以下に転記しましょう。

# 発展課題 4

---

繰返し処理の応用 2

(基本情報科目Bサンプル問題改変 - 最大公約数)

## 発展課題 4 （問題）

1. 変数 num1 と num2 に 正の整数 を代入して表示する

2. num1 と num2 の最大公約数を求めて表示する。

最大公約数は次の(1)～(2)の性質を利用して求める。

(1) num1とnum2が等しいとき,num1の値が最大公約数といえる。

(2) num1とnum2が等しくないとき,num1がnum2より大きい場合はnum1を(num1 – num2)とする。num2がnum1より大きい場合はnum2を(num1 – num2)として,等しくなるまで続ける。

※ num1やnum2の値は処理中に変更されて良いものとし、最大公約数が存在しない数は「1」を表示する

# 発展課題 4 （実行例）

## [実行結果例 1]

Print output (drag lower right corner to resize)

```
num1 = 21 , num2 = 14  
最大公約数は 7
```

## [実行結果例 2]

Print output (drag lower right corner to resize)

```
num1 = 100 , num2 = 40  
最大公約数は 20
```

## [実行結果例 3]

Print output (drag lower right corner to resize)

```
num1 = 90 , num2 = 90  
最大公約数は 90
```

## [実行結果例 4]

Print output (drag lower right corner to resize)

```
num1 = 131 , num2 = 17  
最大公約数は 1
```

# 発展課題 4 （プログラム）

---

作成したプログラムを以下に転記しましょう。

# 発展課題 5

---

リストの応用  
(お釣りの算出)

## 発展課題 5（問題）

---

1. リスト coins に 500, 100, 50, 10, 5, 1 を代入する。
2. 変数 price に 適当な金額を代入する  
(1以上1000未満の値しか入らないものとする)
3. 変数 change に  $1000 - \text{price}$  の結果を求めて表示する。
4. 繰返しや選択、変数 yen(coinsの値) と cnt(硬貨の枚数) を利用して、changeで必要な硬貨の枚数を表示する。  
(0枚の硬貨は表示しない)



# 発展課題 5（実行例）

## [実行結果例 1]

Print output (drag lower right corner to resize)

```
お釣りは 666 円  
500 円硬貨が 1 枚  
100 円硬貨が 1 枚  
50 円硬貨が 1 枚  
10 円硬貨が 1 枚  
5 円硬貨が 1 枚  
1 円硬貨が 1 枚
```

## [実行結果例 2]

Print output (drag lower right corner to resize)

```
お釣りは 312 円  
100 円硬貨が 3 枚  
10 円硬貨が 1 枚  
1 円硬貨が 2 枚
```

# 発展課題 5（プログラム）

---

作成したプログラムを以下に転記しましょう。