

# 基礎プログラミングI

---

## 第6回 コマンドライン引数とCSV読み込み処理

# コマンドライン引数

---

Rubyプログラムを実行するときに、コマンドラインから与えた引数をプログラムから利用できる。

# コマンドライン引数

例えば、

```
./test.rb a b c d
```

のように起動した場合、与えた引数 **a, b, c, d** は自動的に配列変数 **ARGV** に代入される。

# コマンドライン引数

具体的には、

```
ARGV[0] = "a"
```

```
ARGV[1] = "b"
```

```
ARGV[2] = "c"
```

```
ARGV[3] = "d"
```

のように値が格納される。これを利用するとプログラム起動時に与えた情報で動作するプログラムを作成できる。

# CSV読み込み処理

---

# 成績処理の例

配列を利用して、試験の得点の処理をするプログラムを作成する。例えば、以下のような試験結果ファイル `score.csv` があるとする。

山田太郎,50  
中町太郎,90  
飯森花子,91  
鶴岡一人,60  
酒田三吉,52  
三川一二三,12

このデータを読み込み、全員の平均点を計算した上で、一人ひとりの得点と平均点との差を出力するプログラムを作成する。

# プログラムの流れ

---

1. データファイルを全て読み込み、名前と得点を配列に格納する。
2. 名前用、得点用の2つの配列と合計点記録用の変数を用意する。
3. 1レコードずつデータを取り出し、2.で用意した配列へのデータの格納と合計点の計算を行う。
4. 配列のlengthメソッドを用いて要素数（人数）を算出し、平均点を計算する。
5. 全員の氏名、得点、平均点との差をループで出力する。

# 成績処理プログラム 手順 1

1. データファイルを全て読み込み、名前と得点を配列に格納する。

CSVデータの読み込み（ファイル名をプログラム中で指定する場合）：

```
seiseki = CSV.read("score.csv")
```

# 成績処理プログラム 手順 1

---

1. データファイルを全て読み込み、名前と得点を配列に格納する。

CSVデータの読み込み（コマンドラインからファイル名を指定する場合）：

```
seiseki = CSV.read(ARGV[0])
```

# 成績処理プログラム 手順 1

前ページの場合

```
./プログラム名.rb score.csv
```

とすると、プログラム名の直後に書いたファイル名がARGV[0]に入る。

# 成績処理プログラム 手順 1

---

score.csv ファイルをCSV.readで読み込むと、

1行目のレコードは[“山田太郎”, “50”]

2行目のレコードは[“中町太郎”, “90”], ...

のように配列化され、さらにそれぞれの行の配列を要素とした配列がseiseki変数に入る。

# 成績処理プログラム 手順 2

---

2. 名前用、得点用の2つの配列と合計点記録用の変数を用意する。

```
name = []
```

```
score = []
```

```
sum = 0
```

```
# 合計点計算用
```

# 成績処理プログラム 手順 3

3. 1レコードずつデータを取り出し、2.で用意した配列へのデータの格納と合計点の計算を行う。

```
seiseki.each do |row| # 1レコードずつ取り出したものが順次rowに入る
  name << row[0]      # 第0要素は氏名
  score << row[1].to_i # 第1要素は得点の文字列なので .to_i で整数化
  sum += row[1].to_i
end
```

# 成績処理プログラム 手順 4

---

4. 配列のlengthメソッドを用いて要素数（人数）を算出し、平均点を計算する。

```
average = sum.to_f/score.length
```

# 成績処理プログラム 手順 5

5. 全員の氏名、得点、平均点との差をループで出力する。

```
i = 0
print "--氏名-----+-得点-+-平均との差--\n"
while i < score.length
  printf("%-20s %5d %5.1f\n", name[i], score[i], score[i]-average)
  i += 1
end
```

# 成績処理プログラム

以上を組み合わせてプログラムが完成する。

```
#!/usr/koeki/bin/ruby
# -*- coding: utf-8 -*-

require 'csv'          # CSVライブラリを呼び出す
require './kprintf.rb' # printfの%Xsでの幅を合わせるため

score=[]
name =[]
sum = 0
seiseki = CSV.read(ARGV[0])
seiseki.each do |row|
  name << row[0]
  score << row[1].to_i
  sum += row[1].to_i
end
average = sum.to_f/score.length

i = 0
print "--氏名-----+得点+-平均との差--\n"
while i < score.length
  printf("%-20s %5d %5.1f\n", name[i], score[i], score[i]-average)
  i += 1
end
```

# まとめ

---

- コマンドラインに引数を与えると、与えた引数は配列変数**ARGV**に代入される。
- **ARGV**を利用すると、入力された文字列をプログラムの中で使うことができる。
- Rubyでは、CSVファイルを読み込んで**データ処理するプログラム**を作成できる。