

プログラミング基礎

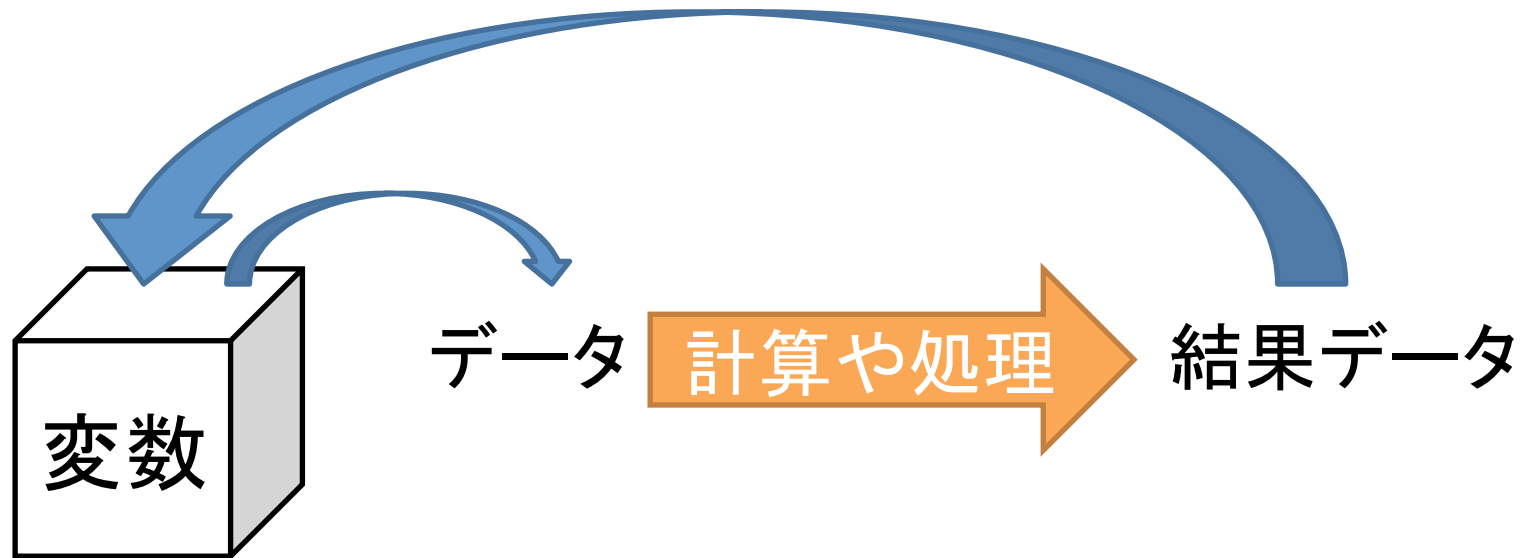
配列 (array)

今回の内容

- 配列
- 乱数 (random 関数)
- 変数の有効範囲 (scope)
- 課題: 棒グラフを描くプログラム

復習：変数(variable)

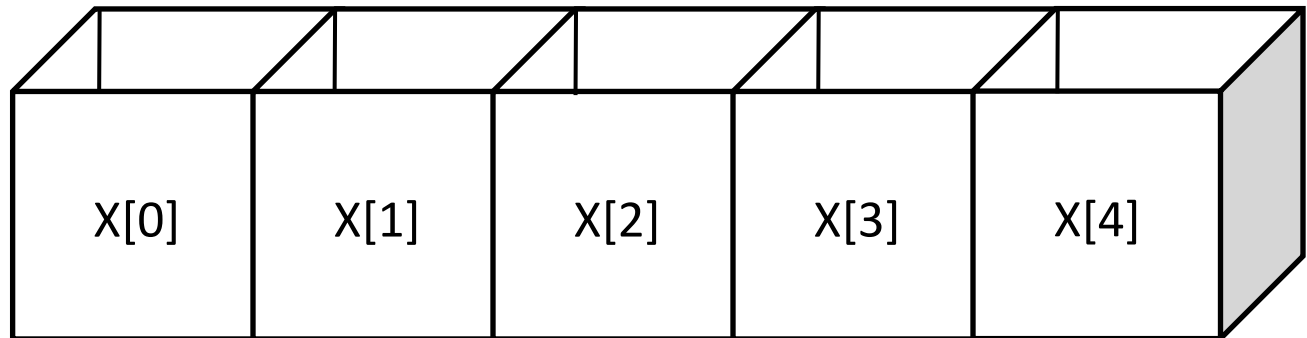
- データを記憶しておくための領域
 - 領域ごとに異なる名前を付けて区別する
- 変数を使うプログラムの基本的な流れ：



配列 (Array)

- たくさんの変数をひとまとめにしたもの
- 番号が振られている
- 配列にも型がある

配列X



配列の作り方・使い方

```
int[] scores = new int[3];
```

【書式】 型[] 配列名 = new 型[個数];

```
scores[0] = 88;  
scores[1] = 80;  
scores[2] = 76;
```

【書式】 配列名[番号]

最初の番号は 0
最後は 個数-1

```
for(int i = 0; i < scores.length; i = i + 1){  
    println(scores[i]);  
}
```

繰り返しと一緒に使うことが多い！

【書式】 配列名.length

```
int[] scores = {88, 80, 76};
```

と書いても良い

練習

- 練習1: int型の配列の最大値を求める関数
 - 引数の型は int[] 結果の型は？
 - 繰り返しを使います
 - 途中段階の最大値を覚えておく変数を使います
- 練習2: int型の配列の平均値を求める関数
 - 結果の型は float

random 関数(乱数)

- サイコロを振ったときのように適当な結果をランダムに返してくれる機能

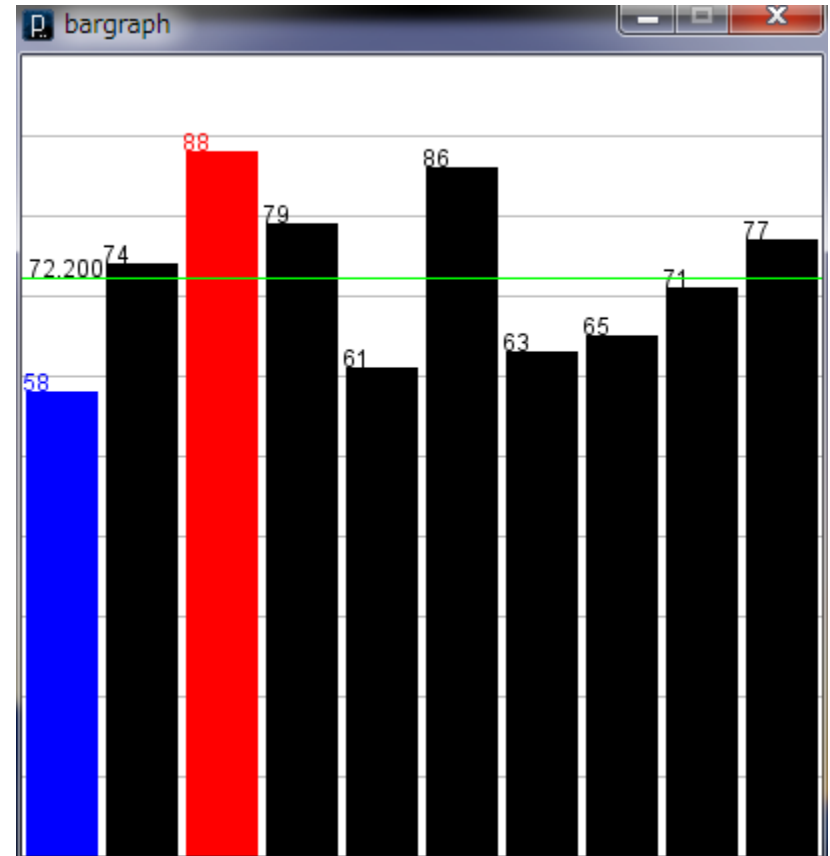
```
int[] scores = new int[5];

for(int i = 0; i < scores.length; i++){
    float f= random(6); // 0 以上 6 未満の乱数
    scores[i] = ceil(f); // f の小数点以下を切り上げ
}

for(int i = 0; i < scores.length; i++){
    println(scores[i]);
}
```

課題3：点数の棒グラフ

- 要件
 - 点数の配列は乱数で初期化する
 - 最高点：赤、最低点：青、その他：好きな色
 - 平均点の横線：緑



課題のヒント

- 最高・最低・平均点を計算する関数を作る
- setup関数
 - 点数の配列を乱数で初期化する
 - 最高・最低・平均点を計算して変数に入れておく
- draw関数
 - 繰り返しと条件分岐を使って描く