

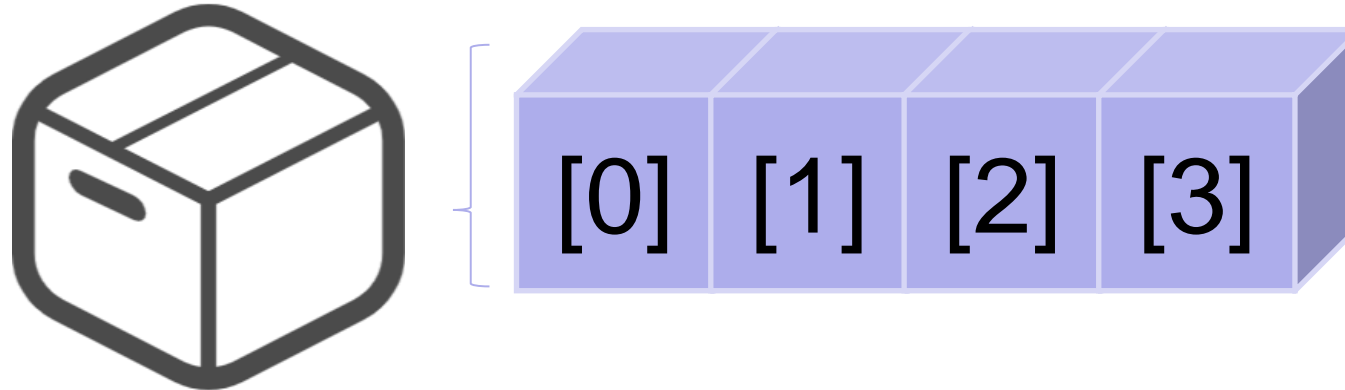
# 第7章 配列

## 目次

- 配列
- 配列の準備
- 配列の長さ
- 多次元配列

## 配列：配列とは

「複数個の値をまとめて保存できる変数」のようなもの。  
変数が1つの値のみ保存できるのに対し、  
配列は複数の値を保存できる。



配列とは、複数個の値をまとめて保存できる変数のようなものです。  
配列を利用することで、複数の値の管理がしやすくなります。

## 配列の準備

配列を使うためには、以下の2ステップが必要。

ステップ1: 配列変数を宣言する。

ステップ2: 配列を作成する。

型名[] 配列変数名; ← ステップ1

配列変数名 = new 型名[要素数]; ← ステップ2

## ステップ1: 配列変数を宣言する

配列変数を宣言する時は、  
変数の型の右側に[ ]を付けて型を指定する。

```
int[ ] height;
```

## ステップ2: 配列を作成する

- new演算子を使い、[]内で指定された個数の要素からなる配列を作成する。
- []の左側には型を記述する。

```
height = new int[4];
```

## 配列変数の宣言と配列の作成をまとめて1文で記述可能

```
型名[] 配列変数名 = new 型名[要素数];
```

```
int[] height = new int[4];
```

## 要素の呼び出し

[ ]内には添字を記述する。

添字を記述することで、配列中の特定の要素に値を代入できる。  
また、既に要素に保存された値を取得できる。

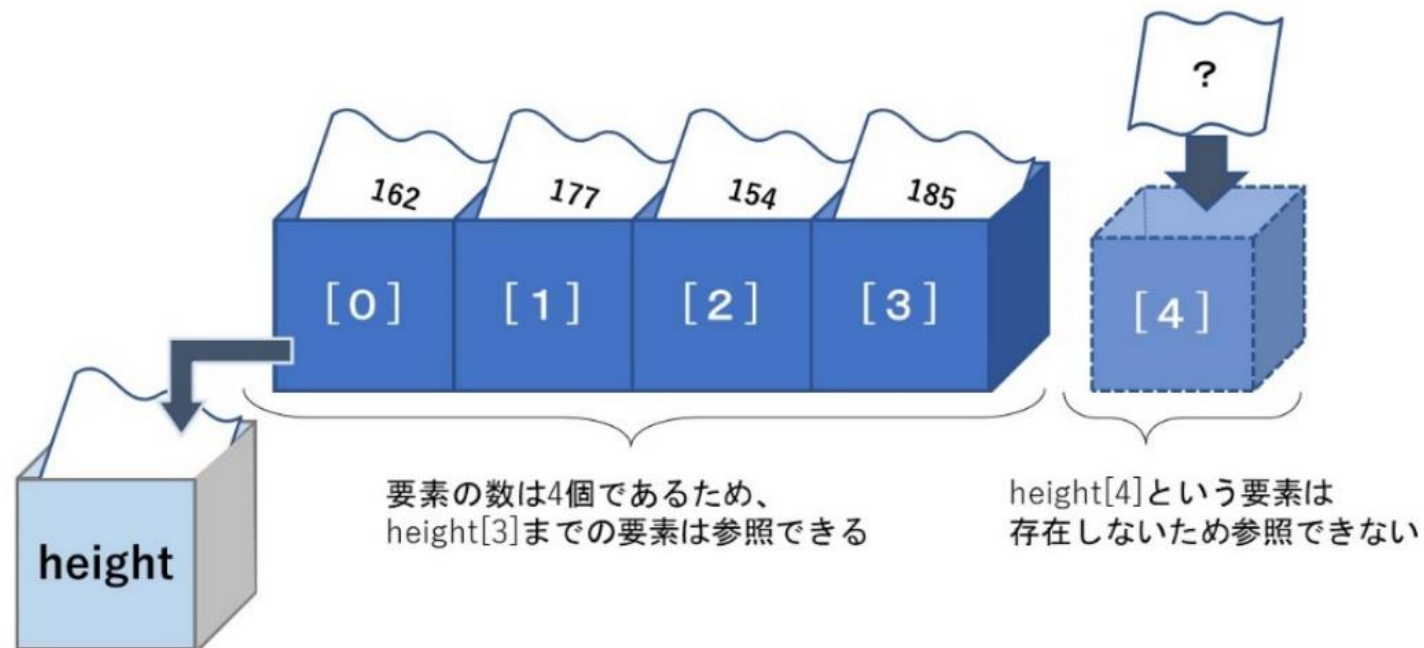
配列変数名[添字]

```
height[0]  
height[1]  
height[2]  
height[3]
```



## 添字を指定する際の注意点

- 配列に存在しない要素の参照はできない。
- 添え字には数字のみ利用できる。



## 添字を指定する際の注意点

配列を準備するときは、  
①「配列変数の宣言」②「配列の作成」を行きましょう。  
要素の添字は0から始まり、最大値は「要素数 -1」となります。  
存在しない要素は呼び出せません。

## 配列に値を代入

```
配列変数名[添字] = 式;
```

## 【Sample0701 配列に値を代入する】を作成しましょう

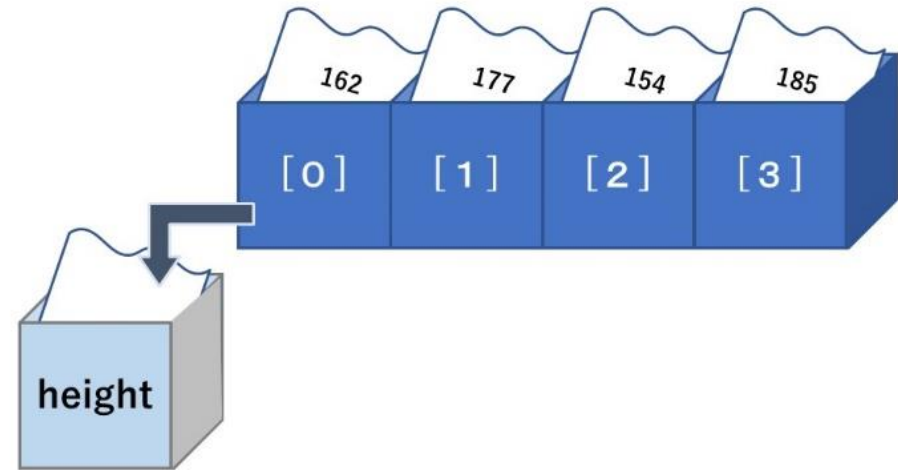
Let's try!



## Sample0701のポイント①

4つの要素に整数値を代入している。

```
height[0] = 162;  
height[1] = 177;  
height[2] = 154;  
height[3] = 185;
```



## 配列の初期化

「配列変数の宣言」と「配列の作成」に加えて、  
「値の代入」の処理も1行にまとめて記述可能。

```
型名[ ] 配列変数名 = {値 1, 値 2, 値 3, ……};
```

```
int[] height = {162, 177, 154, 185};
```

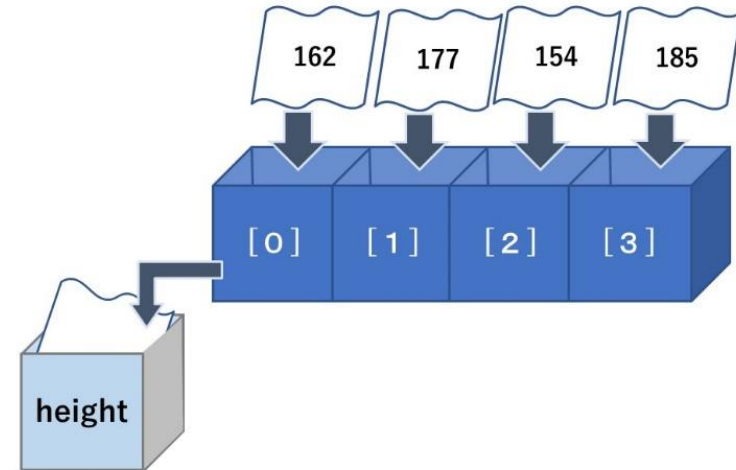
## 【Sample0702 配列を初期化する】を作成しましょう



## Sample0702のポイント①

要素数が4個の配列を作成し、各要素に値を代入している。

```
int[] height  
= { 162, 177, 154, 185 };
```



配列の初期化とは、「配列変数の宣言」、「配列の作成」、  
「値の代入」を1行でまとめて行うことです。



## 配列変数間での代入

配列変数には、配列そのものではなく、  
「参照」の情報が代入される。

参照には「データがメモリ中のどこに保存されているのか」  
という情報が含まれる。

## 【Sample0703 配列変数間で代入する】を作成しましょう

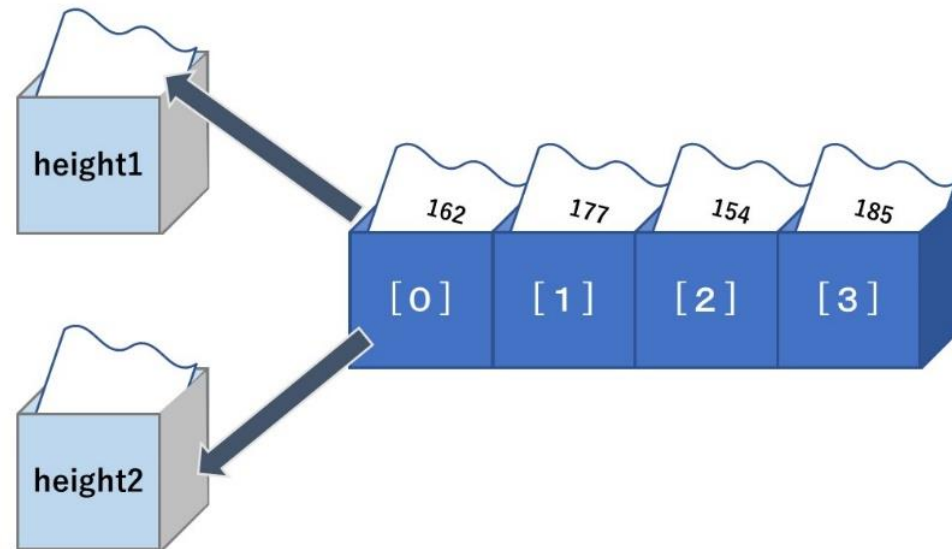
Let's try!



## Sample0703のポイント①

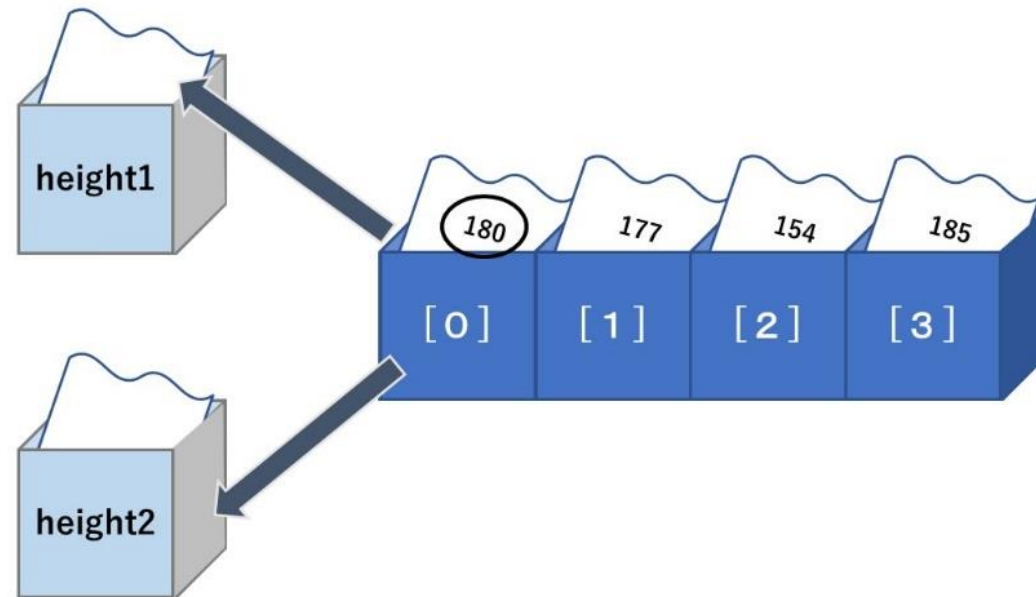
配列変数height1の値を配列変数height2に代入している。

```
height2 = height1;
```



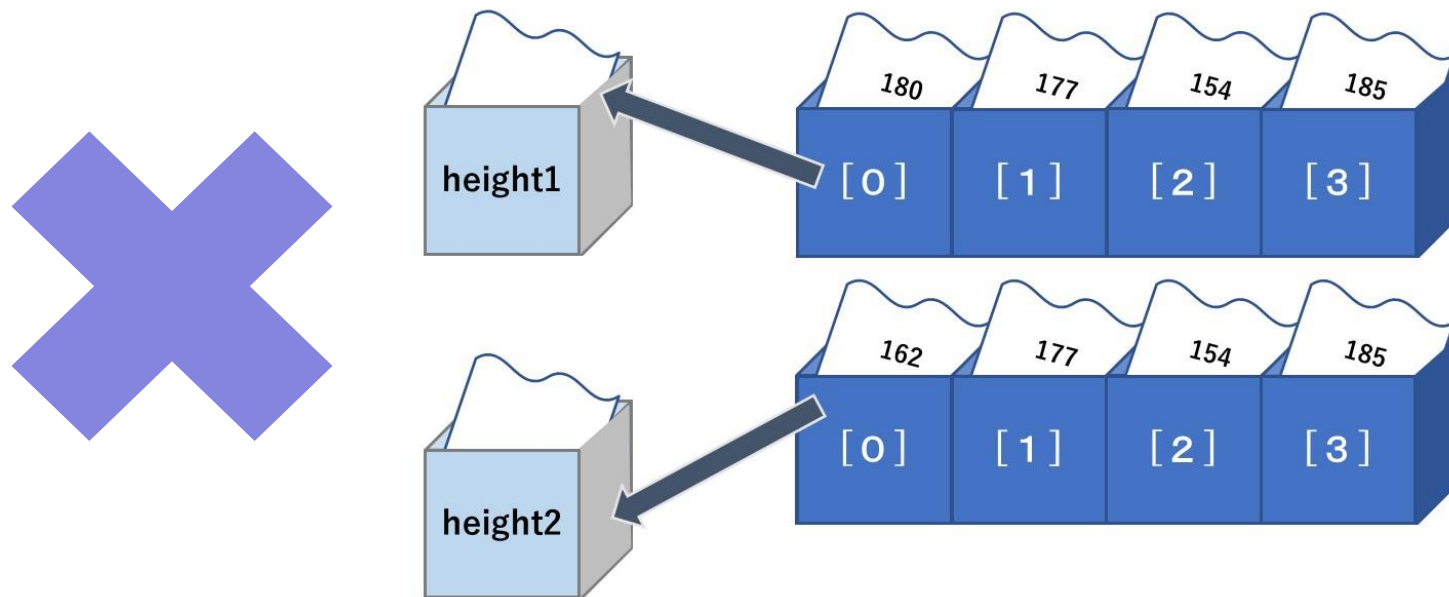
## Sample0703のポイント②

height1の添字0番の要素に新たに値を代入すると、height1とheight2の参照先の値が変更される。



## Sample0703のポイント②

配列変数間で値が代入されても、  
配列自体が複製されるわけではない点に注意。



## 配列の長さ

配列の要素数のことを配列の長さと呼ぶ。  
配列変数に続けて「.length」と記述することで、  
配列の長さを取得できる。

配列変数名.length

# 【Sample0704 配列変数の長さを求める】 を作成しましょう

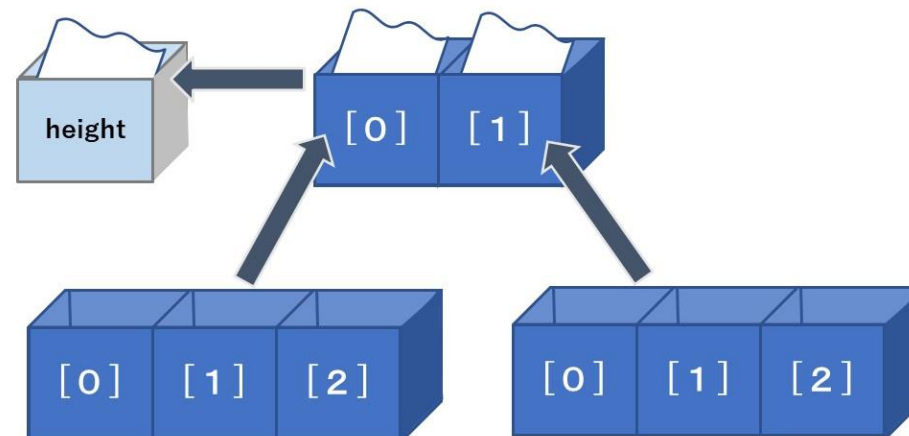
Let's try!



## 多次元配列

多段階構造の配列のこと。

1つの配列変数で複数種類(カテゴリ)、  
かつ複数の値をまとめて保存したい場合に利用する。





## 多次元配列の準備

2次元配列の場合、左の[]から順番に1次元目の要素数、2次元目の要素数を指定する。

```
型名 [ ] [ ] 配列変数名;  
配列変数名 = new 型名 [1次元目の要素数] [2次元目の要素数];
```

```
int [ ] [ ] height;  
height = new int [2] [3];
```

## 多次元配列の準備

1次元配列同様、多次元配列も「配列変数の宣言」と「配列の作成」を1行でまとめて記述することも可能。

```
int[] [] height = new int[2][3];
```

## 多次元配列の要素の呼出し

1つ目の「`[]`」内で、1次元目の配列の添字を記述して、その下の次元にあるどの配列を参照するかを指定する。

配列変数名 [1次元目の添字] [2次元目の添字]

```
height[0][1] = 177;
```

## 多次元配列の初期化

1次元配列と同様に、多次元配列でも初期化が可能。

```
int[][] height = {{162, 177, 154}, {148, 170, 145}};
```

## 多次元配列の長さ

1次元配列の長さを求めたい場合

配列変数名.length

2次元配列の長さを求めたい場合

配列変数名[1次元目の添字].length

## 章のまとめ

- 配列変数を宣言し、new演算子を使って配列を作成できます。
- 配列変数を使って要素を代入できます。
- 配列の添え字は、0から始まり、  
最大番号は「要素数 - 1」となります。
- { }を使って、配列を初期化できます。
- 配列変数間で代入すると、複数の配列変数が  
同じ配列を参照することになります。
- 「.length」を付けて、配列の長さを調べることができます。
- 多次元配列という多段階構造の配列を利用できます。