

# Java

配列 基礎

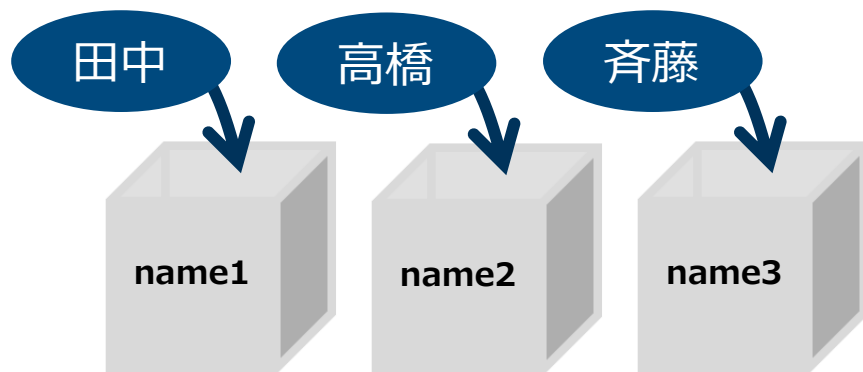
10 時間目

配列とは、  
同じ型の複数の数字や文字などを  
1つに格納して管理するもの  
一度サイズが決められると  
変更することはできない

# 配列のイメージ

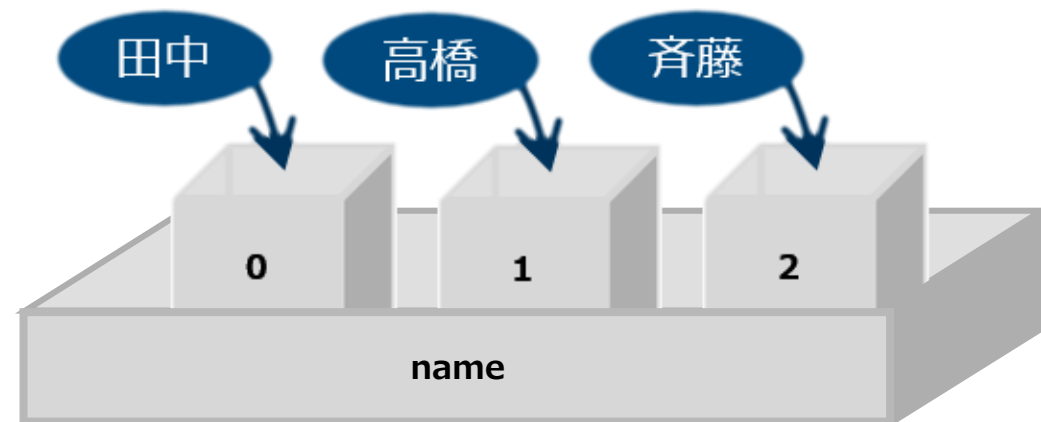
## 通常：別々の変数として管理

```
String name1="田中";  
String name2="高橋";  
String name3="斉藤";
```



## 配列として一括で管理

```
String[] name = new String[3];  
name[0]="田中";  
name[1]="高橋";  
name[2]="斉藤";
```



# 配列のイメージ

解説

格納する箱の番号は、プログラミングの性質上必ず0から始まります。

箱に入れられた「田中」「高橋」「斉藤」を

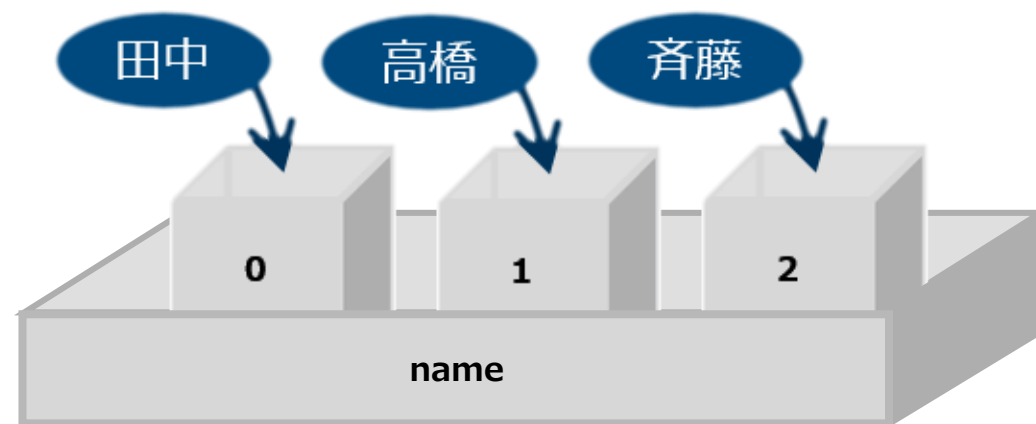
**value（又は値）** と呼びます。

「箱と値を合わせて」

**element(又は要素)** と呼びます。

箱の番号「0」「1」「2」を

**key（又は添え字）** と呼びます。

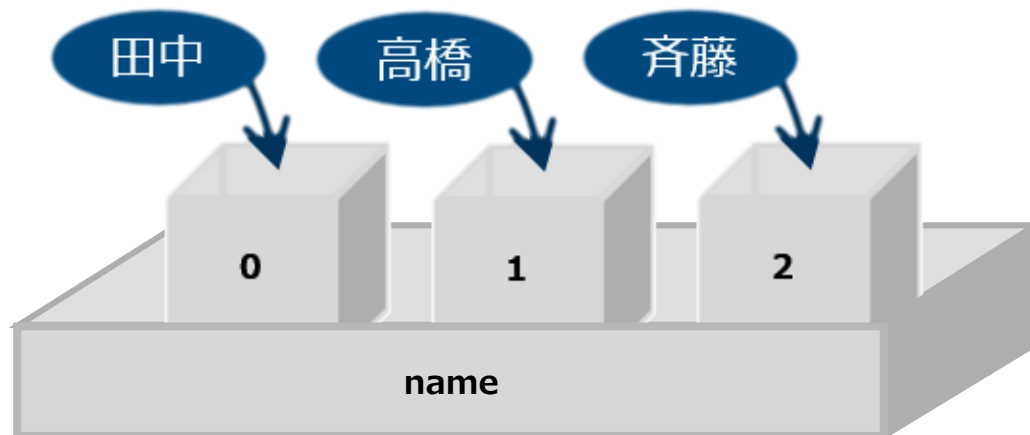


# 配列のイメージ

配列の各要素は、**配列変数[インデックス]** という形で記述します。

配列は、同じ変数で複数の要素を持つため、  
インデックスの数値で何番目の要素かを特定します。

**配列のインデックスは、0から始まります。**



# 配列の書き方

解説

配列の書き方には、2通りの方法があります。

Javaファイル

```
String[] name = new String[3];
```

```
name[0]="田中";  
name[1]="高橋";  
name[2]="斉藤";
```

Javaファイル

```
String[] name = {"田中", "高橋", "斉藤"};
```

# 配列の書き方

解説 配列に値を代入した後に、それぞれの値を使用することが出来る

Javaファイル

```
String[] name = new String[3];
```

```
name[0]="田中";
```

```
name[1]="高橋";
```

```
name[2]="斉藤";
```

```
System.out.println(name[0]);
```

田中

# 配列の書き方

解説 配列に値を代入した後に、それぞれの値を使用することが出来る

Javaファイル

```
String[] name = {"田中", "高橋", "斉藤"};  
System.out.println(name[0]);
```

田中



Javaファイル

```
char[] c = {'x', 'y', 'z'};
```

```
System.out.println(c[0]);
```

**X**

Javaファイル

```
char[] c = {'x', 'y', 'z'};
```

```
System.out.println(c[1]);
```

```
y
```

Javaファイル

```
char[] c = {'x', 'y', 'z'};
```

```
System.out.println(c[2]);
```

**z**

Javaファイル

```
int[] i = {1, 5, 10};
```

```
System.out.println(i[0]);
```

**1**

Javaファイル

```
int[] i = {1, 5, 10};
```

```
System.out.println(i[1]);
```

**5**

# データの更新

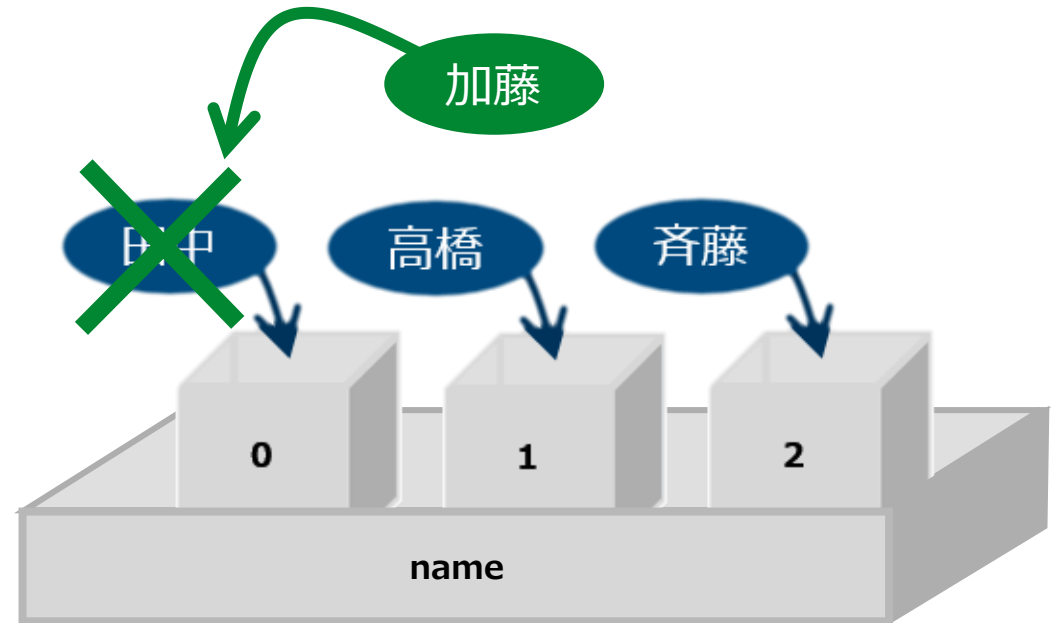
解説

一度、配列に格納した、データを更新することも可能。

Javaファイル

```
String[] name = {"田中", "高橋", "斉藤"};
```

```
name[0]="加藤";
```



## 更新した“加藤”を表示してみよう

Javaファイル

```
String[] name = {"田中", "高橋", "斉藤"};
```

```
name[0]="加藤";
```

```
System.out.println(name[0]);
```

**加藤**