

配列の基本

宣言・サイズ確保・代入をおさえる

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

配列とは？

同じ型のデータを 番号（インデックス=添字）で管理する“連番の箱”

- 変数をたくさん作らずに、まとめて扱える
- インデックス（=添字 / そえじ）は「何番目か」を表す番号。0 から始まる
- 要素数が多くても for 文で一気に処理できる

3つのデータ → a[0], a[1], a[2] のように 1つの名前 + 番号でアクセスする。

```
// 変数を1つずつ作ると...
```

```
int a0 = 80;
```

```
int a1 = 95;
```

```
int a2 = 70;
```

```
// 配列なら1つにまとめられる
```

```
int[] a = {80, 95, 70};
```

1 宣言 — 入れ物の“種類”を決める

書き方: 型 [] 変数名 ;

```
int[] scores;    // int 型の配列を宣言  
float[] temps;    // float 型の配列  
String[] names;    // String 型の配列
```

[] は「配列である」という印。

この時点では 中身も大きさも まだ 無い（null の状態）。

ポイント：宣言だけでは箱はまだ用意されていない。次の「サイズ確保」とセットで初めて使える。

2 サイズ確保 — new で箱の“数”を決める

書き方: 変数名 = new 型 [個数] ;

```
scores = new int[5]; // 要素5 個ぶんの箱を用意
```

new した瞬間、中身は自動で初期値に。
int→0 / float→0.0 / boolean→false / String→null

宣言とサイズ確保は 1 行にまとめられる

```
int[] scores = new int[5]; // 宣言 + 確保
```

型 [] 名前 = new 型 [個数]; が定番の形。

3 代入・参照 — [] で場所を指定

書き方: 変数名 [番号] = 値 ;

```
scores[0] = 80;    // 先頭に代入  
scores[4] = 100;  // 5 番目 (最後) に代入  
  
println(scores[0]); // 80 を表示 (参照)
```

インデックス (添字) の範囲

0 ~ (要素数 - 1)

要素 5 個なら 0,1,2,3,4。

scores[5] は範囲外でエラー

エラー例: ArrayIndexOutOfBoundsException — 用意した箱の数を超えた番号にアクセスすると発生する。

初期化リストと length

宣言・確保・代入を一気に書く / 要素数を取り出す

```
int[] scores = {80, 95, 70, 60, 100};    // 宣言・確保・代入をまとめて
```

```
println(scores.length);    // 5 (要素数を取得)
```

```
for (int i = 0; i < scores.length; i++) {  
    println(scores[i]);    // 全要素を順に処理  
}
```

.length で要素数がわかるので、for 文の条件に使うと要素数が変わっても書き換え不要。

多次元配列（2次元配列）

配列の中に配列。行（たて）×列（よこ）で表やマス目を扱う

```
int[][] grid = new int[3][4];    // 3行×4列を確保
grid[1][2] = 80;    // 2行目・3列目に代入
println(grid[1][2]);    // → 80（参照）
```

```
int[][] a = {{1, 2, 3},    // 初期化リストでも
             {4, 5, 6}};    // 2行×3列
```

grid[行][列]

列 c →

行 r ↓

[0][0]	[0][1]	[0][2]	[0][3]
[1][0]	[1][1]	[1][2]	[1][3]
[2][0]	[2][1]	[2][2]	[2][3]

grid.length で行数、grid[r].length で列数。二重 for で全マス进行处理。画像のピクセルや盤面（マス目）などで活躍する。

3 ステップの まとめ

1

宣言

```
int[] a;
```

2

サイズ確保

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

3

代入

```
a[0] = 80;
```

覚える形: 型 [] 名前 = new 型 [個数] ;