



213-01.配列

配列

配列とは

配列インスタンスのイメージ

2次元配列のインスタンスイメージ

目的：

配列のデータ構造について学ぶ。

配列の使用方法について学ぶ。

ゴール：

同じ型の複数のデータを、ひと固まりで扱う仕組みを理解する。

複数のデータが、変数としてどのように扱われるのか(インスタンスとして扱われている)を理解する。

配列とは

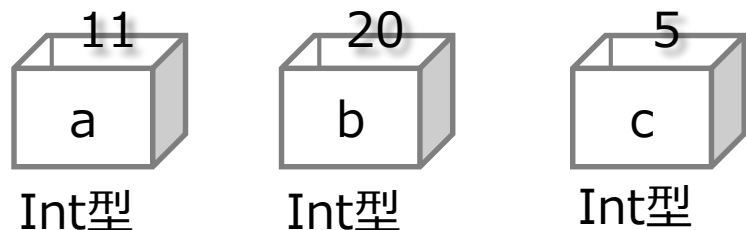
同じ種類（データ型）の複数のデータを格納するデータ構造。

基本型、参照型、どちらの型も代入できる。ただし、同じ配列には同じ型の値のみ代入できる。

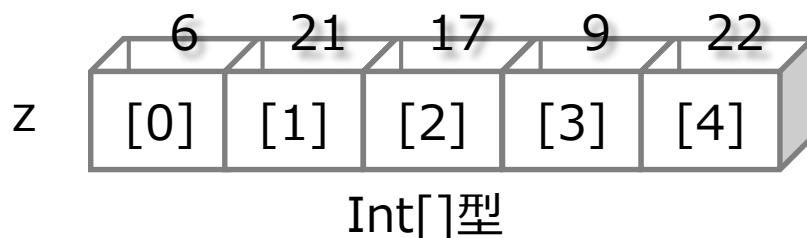
配列も参照型。

1つの配列には複数の箱があり、添字(インデックス)で保持されている場所を指定する。

変数



配列



※配列も参照型でのデータである点は後ほど説明

書き方：

```
要素のデータ型[] 変数名;  
変数名 = new 要素のデータ型[作成する要素数];
```

添字は0から始まる。

先頭の要素は0、最後の要素は要素数-1。

宣言と同時に要素数を指定することは出来ない。要素数の指定は作成する時。

各要素へのアクセス（参照・代入）は添字を指定して行う。

```
変数名[添え字];
```

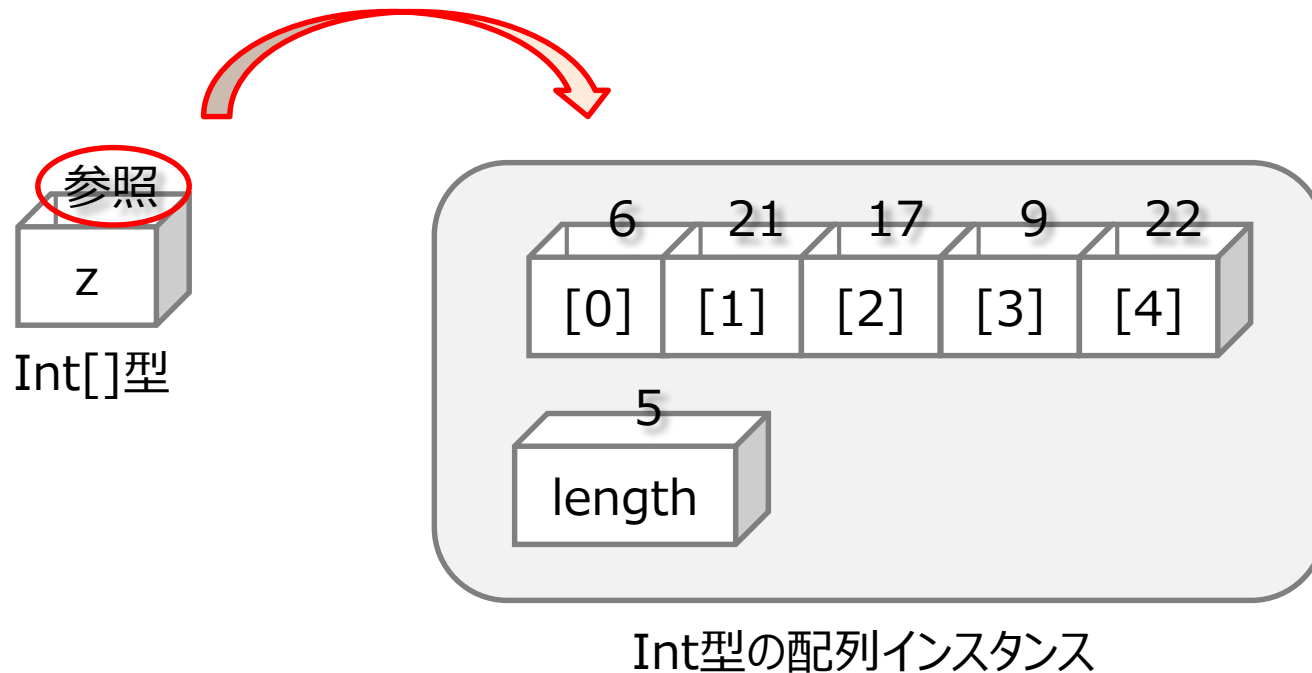
変数名.length で配列の要素数を取得できる。

```
変数名.length;
```

配列インスタンスのイメージ

変数 `z` には配列インスタンスへの参照が保持される。

配列



配列の宣言・作成・初期化をまとめて書くことができる。

書き方：

```
要素のデータ型[] 変数名 = new 要素のデータ型[] { 初期値, ... };
```

配列の要素に配列を代入する事もできる。

その事により配列を多次元化する事ができる。

通常は**1次元配列**（イメージ：横方向のみ）

使ったとしても**2次元**（イメージ：横方向×縦方向）

3次元（イメージ：横方向×縦方向×斜め方向）の場合は
処理の見直しも考えた方が良い。

2次元配列のインスタンスイメージ

