

配列の使い方（手順）

宣言 → 確保 → 代入 → 参照

宣言

型名 [] 配列名 ;

1. 配列用変数の宣言（準備）を行います。

```
int[] i;  
String[] str;
```

確保

配列名 = new 型名 [要素数] ;

2. 配列の要素数を定義します。

```
i = new int[5];  
str = new String[10];
```

代入

配列名 [添字] = 値 ;

3. 添字で指定した要素に値を代入します。

```
i[0] = 3;  
str[9] = “文字列”;
```

参照

配列名 [添字]

4. 添字で指定した要素の値を取得します。

```
System.out.println(i[0]+2); // 要素 [0]+2  
String txt = str[9];        // 変数 txt に代入
```

[添字] の
ポイント

配列の添字は [0] から始まります。
例えば配列の長さ（要素数）が [3] のときは
添字は [0]、[1]、[2] で、[3] は使えないので注意しましょう。

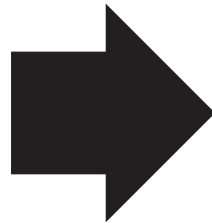
配列の初期化

型名 [] 配列名 = { 要素 0, 要素 1, ... , 要素 n };

変数の初期化と同様に配列も初期化（宣言時に初期値を代入）する事ができます。

まとめて初期値を設定する事でコードがスッキリとするので利用しましょう。

```
int[] array;  
array = new int[5];  
array[0] = 10;  
array[1] = 20;  
array[2] = 30;  
array[3] = 40;  
array[4] = 50;
```



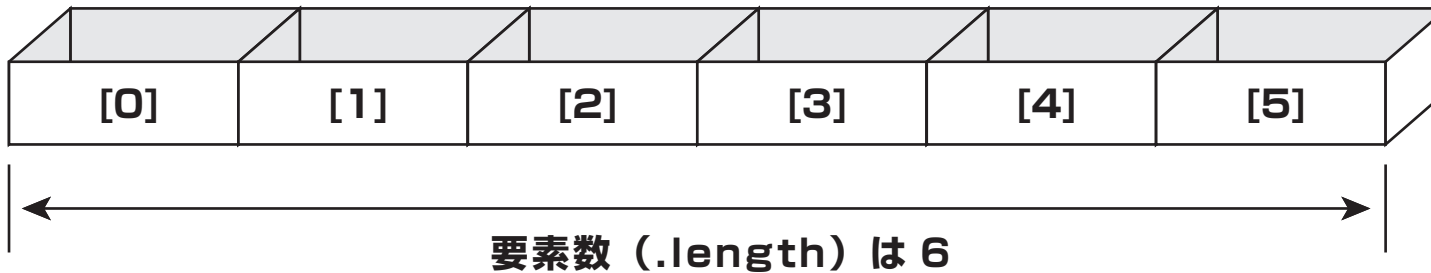
```
int[] array = { 10, 20, 30, 40, 50 };
```

初期化は配列の宣言時のみ行える処理になる為、プログラムの途中で、
array = { 10, 20, 30, 40, 50 };
と書く事は出来ないので注意しましょう。
プログラムの途中で一度に代入する場合は次のようにします。
array = new int[] { 10, 20, 30, 40, 50 };

配列の長さを取得する

変数名 (int 型) = 配列名 .length;

配列の長さ（要素数）を求めるには、配列名 .length を使用します。



(使用例)

```
int[] array = new int[6];  
int len = array.length;
```

「len」という int 型の変数に配列「array」の要素数6が代入されます。

配列と for 文の組み合わせ

配列の長さ (.length) を上手に使おう

配列と for 文は非常に相性が良い関係となっています。

配列は変数に番号を付けた様なものなので、

結果として配列の要素毎に同じ様な処理を繰り返す事が多くなります。

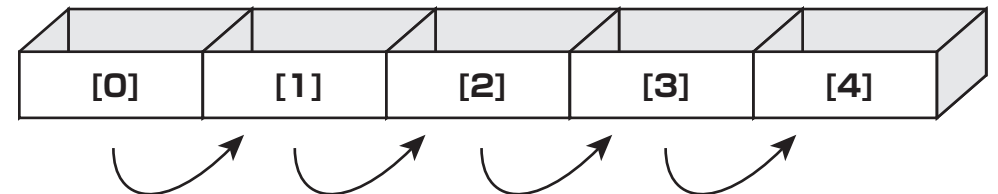
for 文の基本構文である

```
for (int i=0; i< 繰り返す回数 ; i++) {  
    繰り返し行いたい処理  
}
```

を次の様に置き換えると

ここが
ポイント

```
for (int i=0; i< 配列名.length; i++) {  
    配列名[i] に対して行いたい処理  
}
```



配列の全要素に対して順番に処理を行っていく
(int i=0; i< 配列名.length; i++)

配列のすべての要素に対して処理を行う事ができます。