

## 2-4 : 配列

配列は、同じ型の変数をたくさん作りたい時に使います。

例えば、int 型変数を 10 個用意したいとき、一つの方法として

```
int a, b, c, d, e, f, g, h, i, j;
```

とできますが、非常に面倒な上見栄えが悪くなります。そこで配列を使うと楽に宣言することができます。

### 使い方の例

```
int a[10];
```

とすると a[0] a[1] a[2] a[3] a[4] a[5] a[6] a[7] a[8] a[9] の 10 個の int 型変数を宣言したことになります。

ここで、変数の [ ] 内の数値は 0 から始まることに注意しましょう！

```
a[0] = 10;
```

```
a[1] = a[2] + a[3];
```

```
a[4]++;
```

宣言した変数は普通の変数と同様に扱うことができます。

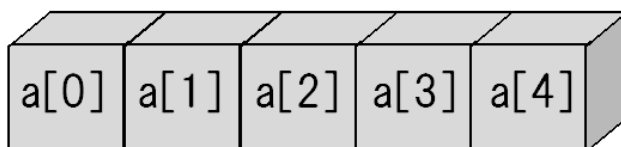
また、for 文と併用するときに力を発揮します。

for 文での配列の応用は 2-7 を参照してください。

## 次元配列のイメージ

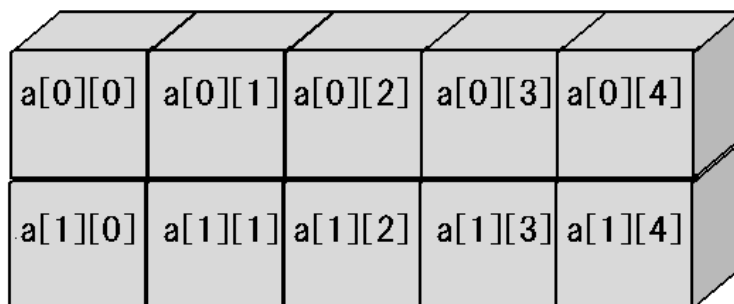
1 次元配列

a[5];



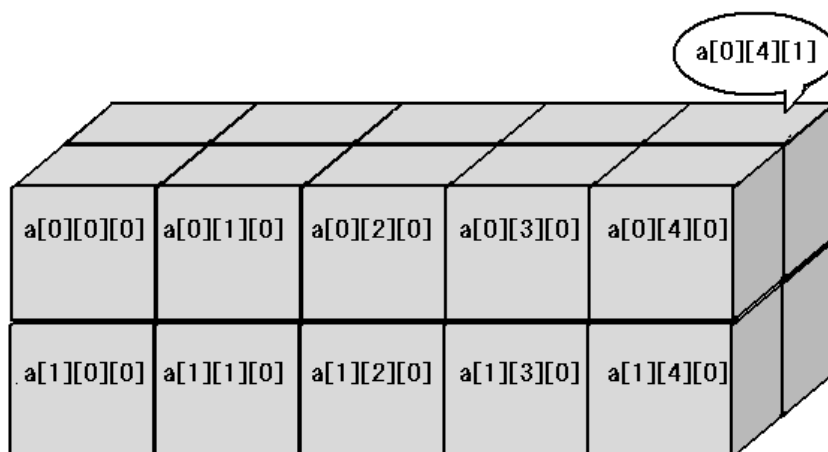
2 次元配列

a[2][5];



3 次元配列

a[2][5][2];



### 例題 1

配列を利用して int 型変数 a を 2010 個作ってください。

### 例題 2

3 次元配列 X[10][2][2] はいくつの変数が見えるようになりますか。

### 例題 3

int 型変数 A を 10 個、B を 5 個、C を 30 個作りたい。まとめて書くにはどうすればいいですか。

### 例題 1・解答

```
int a[2010];
```

これで a[0] ~ a[2009] の 2010 個の int 型変数ができました。

### 例題 2・解答

$$10 \times 2 \times 2 = 40$$

全部で 40 個の変数が見えるようになりました。

### 例題 3・解答

```
int A[10], B[5], C[30];
```

通常の変数のように、コンマで続けて書くこともできます。