

# 大量のデータを1つのまとまりで扱える変数…「配列」

鹿取 祐二

abc [0]	
abc [1]	
abc [2]	
abc [3]	
abc [4]	

図1 配列の要素番号(添字)の付け方  
添字は0から始まることに注意

ここまでで紹介した基本型だけでは困る処理があります。それは、大量のデータの処理です。

例えば、1秒間隔で温度センシングを行って1分ごとに平均値を求めるとか、通信相手と1Kバイトのデータを送受信するなどの処理です。前者であれば1秒間隔のセンサ値を格納する60個の変数、後者であれば送信・受信ともに1024個の変数が必要です。それらをバラバラな変数として宣言したら大変なことになります。そんなときに利用するのが今から紹介する配列です。

## 1 配列の宣言

### ● 宣言の記述方法

大量のデータを1つのまとまりとして扱える配列は、次のように宣言します。

型 配列名 [要素数];

「型」は、基本型と同じく配列の各要素に格納するデータの属性を意味し、配列に付ける名前の「配列名」

abc [0]	3
abc [1]	1
abc [2]	4
abc [3]	1
abc [4]	5

図2 初期値付きの配列を宣言した結果  
`int abc[5] = { 3, 1, 4, 1, 5 };`と宣言した

を記述したら、[ ]の中に要素の個数を示す「要素数」を整数定数値で指定します。例えば次のような宣言です。

```
int abc[5]; // int型で5個の要素がある配列
```

これによりint型で5個の要素を持った配列abcを宣言したことになります。型、配列名、要素数は、ほとんどプログラマの自由なので、次のようにいろいろな配列を宣言できます。

```
unsigned char temp[60]; // 温度センサ用  
char buffer[1024]; // 送受信バッファ用
```

### ● 要素番号(添字)は0から始まる

配列の各要素は、配列名の後に[ ]を使い、abc[2]のように要素番号(添字と呼ぶ)を整数型で指定することで基本型の変数と同じように扱えます。ただし、添字には注意することが1つあります。図1に示すように添字は0から始まることです。配列の要素数は5個なので、添字は0から始まり、最後は4になります。宣言時に指定した要素数と同じ5番目の要素は存在しないことに注意を払う必要があります。

### ● 初期値を設定する方法

基本型の変数と同じく、配列も宣言と同時に初期化できます。ただし、配列は複数の要素があるので、次のように初期値は=の後ろの{ }の中にカンマで区切って記述します。

```
int abc[5] = { 3, 1, 4, 1, 5 };
```

これで図2に示すように、初期値付きで配列が準備されます。初期値の個数が要素数より多い場合はエラーになりますが、少ない場合はエラーになりません。図3のように、初期値の記載がない要素は0で準備されます。

```
int abc[5] = { 3, 1, 4 };  
// abc[3]とabc[4]は0となる
```

また、{ }の中に初期値を全く記載しないとエラーになります。もし全ての要素を0で準備したいときは、次のように初期値の0を1つだけ記述すれば良いことになります。