

# 第5章 「配列」のまとめ

## 5.1 用語のまとめ

配列 同じ型のデータを集めて、番号（\_\_\_\_\_）でアクセスできるようにしたもの。C言語の配列の添字は\_から始まる。

教 p.88

```
int va[5];           /* 初期化しないとき */
int vb[5] = { 15, 20, 30 }; /* 配列の初期化は、式をコンマで区切って { } で囲む。
                           (残りの要素は 0で初期化される。) */
vb = { 15, 20, 30, 0, 0 }; /* 初期化子を代入することはできない。(教 p.92) */
vb = va;               /* 配列同士の代入はできない(教 p.93). */
```

教 p.90

配列と for 文 配列は for 文と相性が良い。配列の各要素に対して同じ操作を行なう時には for 文を使う。

```
for (____; ____; ____) {
    a[i] = ... ;
}
```

教 p.94

配列を逆順に並びかえる 2つの変数 x, yを入れ替えるのに、

```
x = y; y = x;
```

と書いてもダメで、別の変数を一つ用意して、

---

と書く必要がある。

教 p.96

オブジェクト形式マクロ（定数マクロ） プログラム中で繰り返し使う定数は名前をつける。

```
#define NUMBER 5
```

この前処理指令は NUMBER という \_\_\_\_\_ を定義する。マクロは他のコンパイル処理に先だって、一括して置換される。マクロを定義すると、

- 値の変更が容易になる。
- 定数の意味がわかり易くなる。（マジックナンバーを直接埋め込まないこと！）

マクロ名は通常 \_\_\_\_\_ にする。

教 p.98 代入演算子 代入 ( 变数 = 式 ) も式であり、値 ( \_\_\_\_\_ ) を持つ。代入演算子は \_\_\_\_\_ である。つまり、 $x = y = 0$  は \_\_\_\_\_ と解釈される。

教 p.102 多次元配列 2 つ以上の添字を持つ配列のこと。ただし、物理的には一次元に配置される。

```
int x[2][3] = {{ 1, 2, 3 }, { 4, 5, 6 }};
```

教 p.104 **break 文** ( もっとも内側の ) 繰返し文 ( do ~ while 文, while 文, for 文 ) を抜け出る。

**continue 文 ( 参考 )** ( もっとも内側の ) 繰返し文のはじめ ( do ~ while 文, while 文の場合は条件式、for 文の場合は第 3 式 ) にもどる。

break 文	continue 文

教 p.110 コンマ演算子

式<sub>1</sub>, 式<sub>2</sub>

は、式<sub>1</sub>、式<sub>2</sub>をこの順に評価し、式<sub>1</sub>の値を捨て、式<sub>2</sub>の値 ( と型 ) を持つ。

## 5.2 文法のまとめ

宣言 (declaration) に以下を追加する。

分類	一般形	補足説明
変数宣言	型 变数 = 式 , ... , 变数 = 式 ;	= 式 の部分は省略可能
配列宣言	型 变数 [ 定数 ] = { 式 , ... , 式 } ;	= 以降の灰色の部分は省略可能

式 (expression) に以下を追加する。

分類	一般形	補足説明
配列アクセス	式 [ 式 ]	a[1], b[2][3] など