

## 6. 注釈・定数

### (1) 注釈

プログラムを書いて、後から見たら、何を書いているのか、分からなくなってしまうたり、他人が書いたプログラムを見ても、何をやっているのかがわからないケースがあります。

このようなことがないように、あらかじめ注釈(コメント)を書いておきましょう。

通常のプログラミング言語には、この注釈を記述する方法が必ず用意されています。

注釈は、プログラムの処理とは関係ありません。コンパイラに対して、「この部分は、コンパイラには関係ありません」という指定が必要です。

Cの標準では、この範囲を"/\*"と"\*/"で囲みます。

[例] /\* これは注釈です \*/

なお、C++では、"//"から、行の終わりまでを注釈とみなすのが標準です。

C++ではどちらでもいいよ!

```
a = B; // Bをaに代入
```

```
c = D; /* Dをcに代入 */
```



ただし、コメントが多すぎて煩わしいケースもあります。特に行間のコメントが多すぎて縦方向に長ったらしくなり、全体を見通すことができなくなることもあります。わたしの意見としては、次のようにすると良いと思います。

- ① 1文単位のコメントは文の後方に記述する。
- ② 言語の初心者向けのプログラム例でない限り、何(What)をやっているかのコメントを中心に記述し、どのように(How)処理しているかに関するコメントは必要最小限にとどめる。

### (2) 型宣言とサイズ

データの型は、その種類とサイズからなることは前述しましたね。サイズ指定は、たとえば

```
long int A; /* int は省略しても構わない */
short int A;
```

とします。単に int A; と書くと標準のサイズになります。

標準のサイズが short(16bits)か long(32bits)かは処理系によって異なります。マニュアル等で確認しましょう。

特に、整数、文字の場合、unsigned の指定があります。指定がない場合、

- 整数(short) -32,768～32,767
- 文字(ASCII) -128～127