

知らなきや  
ヤバイ

# 基礎ほど奥深〜い! Cプログラミングの当たり前!?

## 第1回 意外と制約が多い! 識別子の名前の付け方

邑中 雅樹

先月号(本誌2012年12月号)まで全12回でお届けした連載「教科書には載っていない! プログラミングちょい技」は、ご好評のうちに幕を閉じることができました。読者の皆様に厚く御礼申し上げます。

「Interfaceへの声」へ寄せられる感想を毎号確認していましたが、その中でもさらに初学者に向けたCプログラミングの連載への期待が大きいと感じました。前号までの連載は、初学者向けとは言いつつも、実践的でややトリッキーな技法の紹介が主でした。

そこで、今月号からは、初心者向けに、もう少し教科書的な基本部分について、実践的なプログラミングのちょい技を伝授するべく、新連載を開始します。

本連載で取り上げる内容は、MISRA-Cなどの組み込み向けのコーディング・ガイドライン(コラム1参照)を意識したものです。読者の皆様が、既存の組み込みソフトウェア向けコーディング・ガイドラインの意図をくみ取りやすくなることを、本連載の目標とします。

第1回目の今回は、識別子の命名規則について解説します。名前なんて付いていけば良い!なんて、甘く考えていると痛い目にあうかもしれません(図1)。

### 変数名/関数名/マクロ名... 識別子の名前の付け方には制約がある

C言語には、識別子(identifier)という概念が存在します。このように改めて書くとなんだか難しそうですね。識別子とは、変数名や関数名やマクロ名やラベルなど、C言語でのプログラミングを行う際に使っている英数字の羅列(名前)のことです。C言語の仕様書にもある言葉なので、丸暗記しましょう。

識別子の命名には、そのシステム仕様(コンパイラやOS)上や、コーディング・ルール上において、いくつかの制約があります。本稿では、それらの制約や対応について整理します。

識別子に関しては、C言語の歴史の中で細かい変遷があります。執筆時点の最新版である、C言語仕様のISO C99に準じて解説をし、必要に応じてほかの標準仕様や処理系についても言及していきたいと思います。

### 制限1: 使用できる文字種

あらゆるプログラミング言語で同じことが言えますが、何かを操作するには名前が必要です。自由に名前を付けられれば良いですが、付けられない名前もあります。例えば、文法を曖昧にするようなものだったり、過去や将来の言語仕様と非互換になるようなものだったり、ほか環境への移植性を損なうようなものが該当します。

識別子の先頭として使用できる文字には制約があります。使える文字は、

- 英大文字(A, B, C, ..., Z)

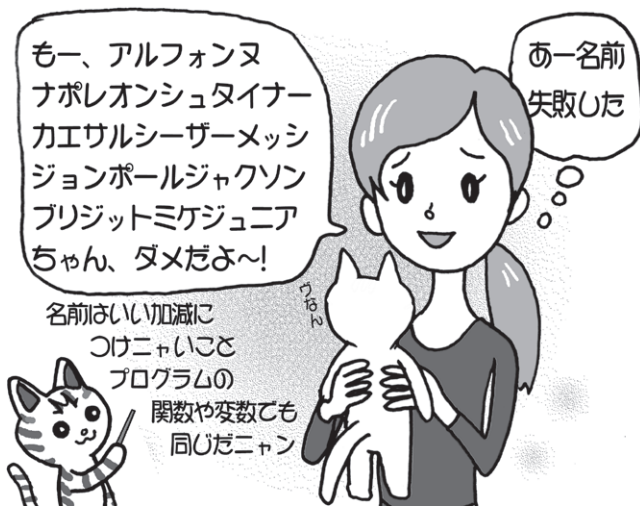


図1 名前は大切! 後のことも考えてちゃんと付けよう