

2007年10月 3日

## 第3回 C言語の基本構造

# 情報処理論 (応用)

松山大学 経営学部

檀 裕也

<http://www.cc.matsuyama-u.ac.jp/~dan/education/application/>



# 今回の授業内容

---

- C言語の基本構造
  - 前回のサンプルの解説
  - 変数と算術演算子



# ソースコードの解説

- 画面に「Hello, world」と表示するプログラム

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    printf( "Hello, world\n" );
```

```
    return 0;
```

```
}
```

```
C:¥>hello.exe
```

```
Hello, world
```

```
C:¥>
```



# インクルード文

```
#include <stdio.h>
```

- 外部ファイルの取り込みを指定する
  - 標準入出力ライブラリのヘッダーファイルである `stdio.h` を取り込む
  - プログラムで `printf` 関数（出力）を使うため、ソースコードの冒頭に宣言する

```
#include <ファイル名>
```

```
#include "ファイル名"
```



# main 関数

```
int main()  
{  
  
}
```

- プログラムは、main 関数から実行を始める  
main 関数の中に処理を記述する
- main 関数の内容は { から } まで囲む  
C 言語では論理的なまとまりをブロック化



# return 文

```
return 0;
```

- 関数の呼び出し元に処理を戻す
  - プログラムを終了する (OSに処理を戻す)
  - 値0は正常に終了したことを示す

```
C:¥>hello.exe  
Hello, world
```

```
C:¥>
```

実行

戻る

```
#include <stdio.h>  
  
int main()  
{  
    printf( "Hello, world¥n" );  
  
    return 0;  
}
```



# printf 関数

```
printf( "Hello, world¥n" );
```

- 画面に文字列を表示する
  - 文字列は " と " で囲んで指定する
  - ¥n で改行を示す (エスケープシーケンス)

```
printf( "文字列" );
```



# エスケープシーケンス

- 特殊文字を表す
  - キーボードから入力できない文字
  - C言語の文法規則上、そのままでは使えない文字

<b>¥a</b>	警告音	<b>¥'</b>	'
<b>¥f</b>	改ページ	<b>¥"</b>	"
<b>¥n</b>	改行	<b>¥¥</b>	¥
<b>¥t</b>	水平タブ	<b>¥?</b>	?





# C言語の一般規則

---

- ソースコードの拡張子は `c` を使う (`hello.c`)
- 命令文には半角文字を使う
- `{` と `}` でブロックを構成する
- ひとつの命令文はセミコロン (`;`) で終了する
- スペースやタブ記号を自由に使うことができる



# ソースコードの基本構造

```
#include <stdio.h>
```

```
int main()  
{
```

処理内容

```
    return 0;
```

```
}
```

必要に応じて  
ヘッダーファイルを  
インクルードする



# 練習問題

- 自分の名前を画面に表示するプログラム `MyName.exe` を作成せよ。
  - ソースコードを `MyName.c` で作成する
  - `cl` コマンドでコンパイルし、実行ファイルを作る

```
C:¥>MyName.exe  
私の名前は檀裕也です。
```

```
C:¥>
```



# 変数

- 数値などのデータをメモリ上に格納するもの
  - 名前を付けて使用する
  - 変数  $a$  に値 10 を代入するとき  
 $a = 10;$   
と記述する
  - 宣言と同時に値を代入すること(初期化)ができる

<code>int a;</code>	<code>a</code> という名前の変数を宣言
<code>int b = 0;</code>	変数 <code>b</code> の初期値を 0 とする
<code>int x, y, z;</code>	同時に3つの変数を宣言する



# 算術演算子

- 四則演算（加減乗除）をする操作

演算子	意味	使用例	値 (a=10,b=3)
+	足し算	$a + b$	13
-	引き算	$a - b$	7
*	かけ算	$a * b$	30
/	割り算の商	$a / b$	3
%	割り算の剰余	$a \% b$	1



# 例題

- 2つの数値の和を計算するプログラム calc.c

```
#include <stdio.h>

int main()
{
    int a = 10, b = 3;
    int x;

    x = a + b;

    printf( "計算結果は %d です。¥n", x );

    return 0;
}
```



# 解説

```
printf( "計算結果は %d です。¥n", x );
```

- このように記述すると、printf 関数で表示する文字列のうち %d の部分に x の値が入る
  - 変換仕様 %d は整数の場合に使う
  - 複数の変換仕様を同時に使うことも可能 (その場合は、対応する値を用意する)



# 練習

- 和と積を計算するプログラム `calc2.exe` を作成せよ。ただし、出力結果に現れる4つの数字は、`printf` 関数の変換仕様を使うこと。

```
C:¥>calc2.exe
```

```
aの値が10、bの値が3のとき、
```

```
aとbの和は13です。
```

```
aとbの積は30です。
```

```
C:¥>
```





# 今回の課題

---

- 完成したソースコード `MyName.c` と `calc2.c` を添付してメールを送信せよ。
  - 宛先: `dan@cc.matsuyama-u.ac.jp`
  - 件名: プログラミング課題#02
  - 本文: 今回の授業に対する感想



# 次回の予定

---

- 日時： 10月 5日 (金)  
4 時限
- C 言語によるプログラミング
  - scanf による入力処理
  - if 文による条件分岐