

## ■ 最初のプログラミング作業

非常に簡単なプログラムを作成、実行してみましょ。これで一連のプログラム作成と実行、修正の手続きを学びます。(これをプログラミング=プログラムすること、と言います。)

まず非常に簡単な C 言語によるプログラムのサンプルを示します。実行すると一行、画面に

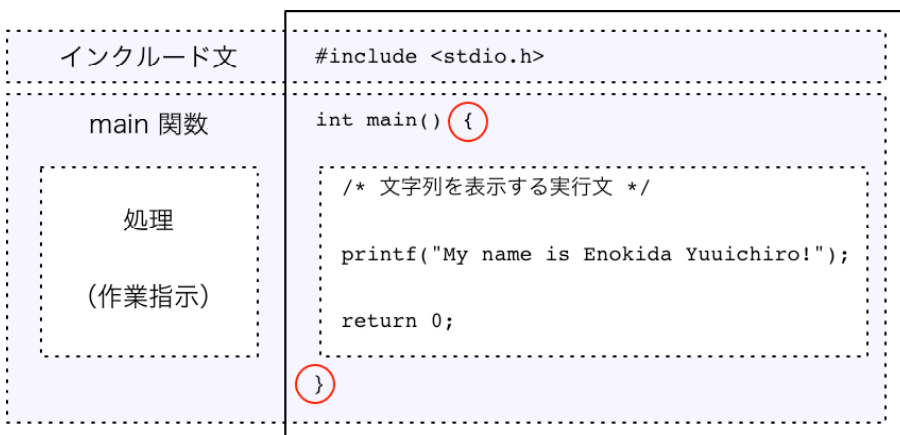
My name is Enokida Yuuichiro!

と表示するだけのプログラムです。

```
#include <stdio.h>
int main(){
    /* 文字列を表示する実行文 */
    printf("My name is Enokida Yuuichiro!");
    return 0;
}
```

このプログラムは非常に短いものですが C プログラムの基本的な構造をきちんと備えています。以下の図のを見ながら、プログラムが下記のパートからなることに注目してください。

- 先頭は `#include` で始まるインクルード文
- 次に `main` 関数があり、それは `{` で始まり `}` (図中の○部分) で終わる
- その間に処理の記述 (作業指示)、つまりプログラムの本体部分がある

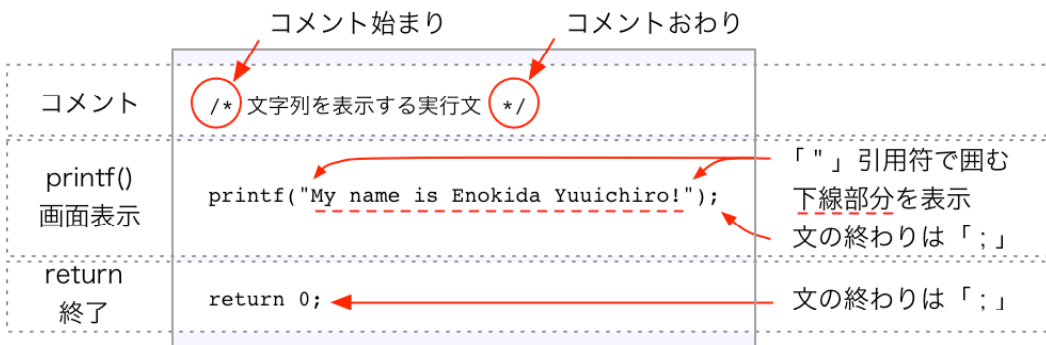


`{ }` で囲まれた部分をブロックと呼びます。プログラムは `main` 関数のブロックの中を上から順に一文ずつ実行します。この実行される文 (行) のことを実行文と呼ぶ時があります。

今回のプログラムではブロック内に三行の記述があります。一行目はコメント、二行目は `printf()` による画面表示、三行目は処理の終了を示す `return` 文です。こうした一行一行の記述にも一定の構造と規則があります。

一文ずつ詳しく見ていきましょう。

- /\* と \*/ で囲まれているのはコメント文（メモ・注釈であって処理として実行されない）
- 実行文の終わりは「;」（セミコロン）
- 処理の最後（main 関数の最後）は return 文で終わる





### ● 課題 1 : 簡単なプログラムの実行

先に示したサンプルプログラムのうち、表示されるメッセージの部分の何か別のもの（自分のモットー、または単に名前の部分を自分の名前に入れ替える）にして、実行し、結果を講師に見せて下さい。

以下に手順をざっと説明しますので追いかけて作業すると良いでしょう。

#### □ プログラムの入力

プログラムへの入力には以下のテキストエディタを利用してください。

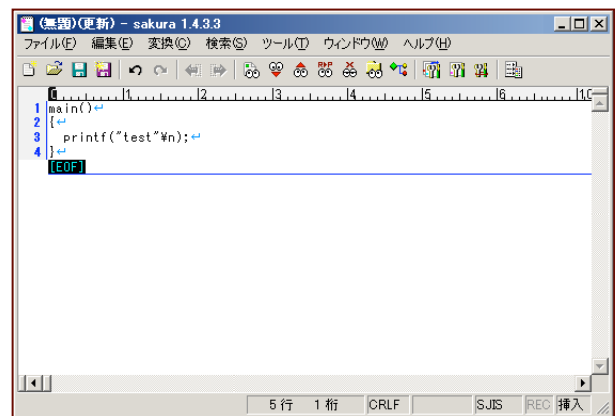
「共通アプリケーション」フォルダ  を開き、「テキストエディタ」アイコン  をダブルクリックして起動してください。以下のような画面が表示されるでしょう。

ここにプログラムを入力し、ファイルメニューの上書き保存（または名前をつけて保存）を使って、保存してください。保存するときの名前は必ず「hello.c」のように、ファイル名の最後に「.c（ピリオドと c）」とつくようにしてください(\*1)。

#### □ プログラムの実行

easy2doit を使って入力したプログラムを実行し、結果を確認してください。

画面上に「My name is Enokida Yuuichiro!」と表示されるはずですが。



\*1 そうすることで「このファイルは C 言語プログラムである」という目印を付けているわけです。拡張子と呼んでいます。

## ■ 制御文字

以下の二つのプログラムの実行結果はそれぞれ異なったものになります。

プログラム：

```
#include <stdio.h>
int main(){

    printf("My name ");
    printf("is ");
    printf("Enokida Yuuichiro!");

    return 0;
}
```

```
#include <stdio.h>
int main(){

    printf("My name \n");
    printf("is \n");
    printf("Enokida Yuuichiro!\n");

    return 0;
}
```

実行結果：

```
My name is Enokida Yuuichiro!
```

```
My name
is
Enokida Yuuichiro!
```

注目してほしいポイント：

殆ど同じプログラムなのに、結果が一行になるか、改行をしているかが違っていています。これは右のプログラムでは制御文字（今回は改行文字）が追加されたためです。

## □ 制御文字の種類

printf() 文などに現れる引用符の中で使える制御文字のうち代表的なものを示します。

上の二つのプログラムの実行結果が違うのは、この改行を意味する制御文字「\n」が printf() の表示文字列のなかに含まれているためです。

¥n	改行する（次の行の先頭に移動）
¥"	「"」を表示する
¥'	「'」を表示する
¥t	タブコードを入れる
¥¥	「¥」記号を表示する

printf() などでは引用符が重要な意味をもちますから、引用符そのものを表示させることが普通にはできません。また制御文字であることを表すために ¥ を使ったため、今度は ¥ そのものを表示されうことも普通にはできなくなりました。制御文字はこうした場合にも使われます。

## ● 課題 2：制御文字

下記のような結果を出すプログラムを作成してください。

（メッセージや好きな食べ物などは適当に自分に合わせて変更して下さい。）

```
My name is
Enokida Yuuichiro!
I like "Sushi" much.
```