

工学研究部 24 年度 新入生講習

C 言語 - 23 みみ

第一回 変数・型・関数

これは何？

- C 言語をやります
- ハード講習を受ける人は最低第一回をやってね（できれば全部やることを推奨）
- C 言語は奥が深すぎてかつ難しいのでこの講習ではかるう〜く触ります
- ちゃんとやりたい人は物足りないと思いますが、初学者向けなので...ね？

環境を構築（その一）

最初に、C のコンパイラをインストールします。

Windows

~~このサイト~~

~~https://joho.g-edu.uec.ac.jp/joho/gcc_win/~~

~~の3までを終わらせればC言語を書くためのする準備は完了です~~

かとうが WSL を入れてくれたっぽい？です。ありがとう。

Mac

デフォルトで Clang というコンパイラが入っているので、とりあえずそれを使ってください...と言いたいところですが、できれば gcc を入れてください

https://joho.g-edu.uec.ac.jp/joho/gcc_mac/（1 まで）

環境構築（その二）

テキストエディタを用意しましょう。

おすすめは VSCode です。

（Ruby 講習でインストールしている人はそれで OK）

（参考）

<https://www.kikagaku.co.jp/kikagaku-blog/visual-studio-code-windows/>

とりあえず、Hello World

コンパイラが準備できたら、テキストエディタで以下のコードを書き、保存してください。

(ファイル名: c1-1.c)

```
#include <stdio.h>

int main(void){

    printf("Hello World.\n");

    return 0;

}
```

実行してみる

ターミナルを開き

(VSCode を使っている人は `ctrl + Shift + `` で開く)

```
$ gcc c1-1.c
```

でコンパイルし、

```
$ ./a.out
```

で実行。すると、

```
Hello World.
```

と表示されます。

変数と型

とりあえず、以下のコードを実行してみましょう。

(ファイル名: c1-2.c)

```
#include <stdio.h>

int main(void){
    int a = 0;
    printf("a = %d\n", a);
    return 0;
}
```

ちなみに、

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

は「おまじない」と言いたいところですが、かんたんに説明すると、標準入出力を扱うのに必要なものです。わけがわからないと思いますが、とりあえず書きましょう

変数と型

Ruby でも使った変数を C でも使うことができますが、Ruby よりも制限が厳しくなります。

詳しく見ていきましょう。

以下は、`a` という変数に 0 を代入するコードです。

```
int a = 0;
```

見慣れない `int` とかいうものがついていますね。

これが、「型」というもので、変数に入れられる値の種類を制限するものです。

変数と型

C 言語では、以下のような型を使うことができます（あくまで一例）。

種類	型
整数	int
小数	float, double （詳しい違いは今は略）
文字	char （第四回で詳しく...）

細かいツツコミはやめてね。

変数と型 - 例

(ファイル名: c1-3.c)

```
#include <stdio.h>

int main(void){
    int a = 1;
    int b = 2;
    int c = a + b;
    printf("a + b = %d\n", c);
    return 0;
}
```

printf については次回詳しくやるので今回はわからなくて OK。

変数と型 - 練習問題

- 例のプログラムでは足し算をやっていましたが、引き算、掛け算、割り算をやってみましょう
- `a` とか `b` とかの中身も色々変えてみましょう

変数と型 - 練習問題

答えあわせタイム

関数

関数には「引数」と「戻り値」があることは Ruby 講習でもやったと思います。

C 言語では引数と戻り値に、もちろん「型」が存在します。

(関数の例)

```
int func(int a){  
    // 関数の中身  
  
    // なにかしらの値を返す（戻り値）  
    return 0;  
}
```

関数の呼び出し方

Ruby と一緒です。

```
func(a);
```

のように呼び出します。

関数で遊んでみる

さっきの足し算のコードを書き換えてみます。

(ファイル名:c1-4.c)

```
#include <stdio.h>

int add(int a, int b){
    return a + b;
}

int main(void){
    printf("a + b = %d\n", add(a, b));
    return 0;
}
```

関数で遊んでみる - 練習問題

- 引き算（ `sub` ）、掛け算（ `mul` ）、割り算（ `div` ）をやる関数を作ってみましょう。
- 余裕がある人は（また今度やりますが）if 文を使ってゼロで割ったときの処理を書いてみましょう。

関数で遊んでみる - 練習問題

答え合わせタイム

第一回 おわり

次回予告：標準入出力