

C言語講座

第1日

- 変数と演算、型変換
- 標準入出力関数
 - 関数
- プリプロセッサ

変数

- ◆ コンピュータに確保される作業エリア
- ◆ 数を入れる箱とも言われる
- ◆ $f(x)$ の x にあたるもの

変数の宣言

◆ 変数を使うには箱を用意する必要がある→宣言

◆ 宣言は
データ型 変数名;
で行える。

例) `int x; double y;`

演算子

◆ 計算符号と同じようなもの

演算	演算子	例	例の意味
代入	=	$a=2$	a に2を代入
加算	+	$a+b$	a と b を足す
減算	-	$a-b$	a から b を引く
乗算	*	$a*b$	a と b をかける
除算	/	a/b	a を b で割る
剰余算	%	$a\%b$	a を b で割る余りを求める
インクリメント	++	$a++$	a に1加算(後置演算)
		$++a$	a に1加算(前置演算)
デクリメント	--	$a--$	a を1減算(後置演算)
		$--a$	a を1減算(前置演算)

前置演算と後置演算について

◆ $n = a++$ と $n = ++a$ の違い

$$n = a++$$

代入が先にされる
ので $a \neq n$ に

$$n = ++a$$

$++$ が先にされるの
で $a == n$ に

コメント

コメントは、単なるメモスペースになり、プログラムに影響を与えなくなる。

例)

```
int main() {  
    int x;  
    /* int y; */  
    x=x+y;  
}
```



エラー

関数

$f(x)$ のこの $f()$ のこと

関数 → 関数

ブラックボックス

→ 処理の一まとまりのこと

引数と返値

- ◆ 引数
関数を呼び出すときに、その関数に与える値 ($f(x)$ の x)
ex) $a+b$ をさせる関数に渡す a と b
- ◆ 返値
関数を呼び出した元に返す処理結果
ex) 上の例で言う $a+b$ の値

関数の宣言・定義

- ◆ 関数は宣言・定義をするとき次の形をとる

返値のデータ型 関数名(引数) {

処理内容

}

ライブラリ

関数の中身が書いてある

代表的なライブラリ (のヘッダ)

`stdio.h`

`stdlib.h`

`string.h`

`math.h` , `time.h`

プリプロセッサ

- ◆ プリプロセッサ
コンパイルをされる前に処理をさせる命令…先頭に#
- ◆ #include
.hを読み込み、引っ付けてコンパイルする
- ◆ #define A B
コンパイル前にAをBに置き換える

*printf*関数

文字列を表示させる関数

`printf(“中身” , 変数)`

*scanf*関数

文字列を受け取る関数

`scanf(“中身”, &変数)`

「&変数」で変数のアドレス(格納されている場所)を表す