

第7回 式と演算子 2

1. 評価の仕組み

式の評価 (evaluation)

3つの原則

2. 評価結果への置換

演算子はオペランドを使って結果を置き換える

複数ある式は段階的に置き換わる→テキスト p89

3. 評価の優先順位

異なる演算子がある場合、優先順位の高い演算子から順に評価される

→ テキスト p734 [<< サンプル >>](#)

4. 結合の規則

優先順位が同じ演算子が連続しているときの実行順番を決める規則

5. 演算子

6. 算術演算子

[<< サンプル >>](#)

<< 演算子の優先順位サンプル >>

7. 文字列型の決まり

テキスト p93

8. 代入演算子

[code0305.c](#)

[<< サンプル >>](#)

9. インクリメント・デクリメント演算子

テキスト p96

[<< サンプル >>](#)

10. 演習

サンプルソースコード : 07_sample.txt

結合規則

同じ順位の演算子が連続した場合でも、その実行順番を明確に定める必要がある。この規則が結合規則で多くの場合左から右に実行するが、代入演算子の様に逆の場合もある。

左から右 : 下記以外の 2 項演算子と条件演算子

$a*b/c$ は $(a*b)/c$ の順番

右から左 : 代入演算子、単項演算子、条件演算子

$a=b=c$ は $a=(b=c)$ の順番

参考 : <https://www.ibe.kagoshima-u.ac.jp/static/www1/edu/gengo0/p6.html>

算術演算子

算術演算子は数値の演算を行いその結果の値を返す。

演算子		書式	返す値
(単項の)+		+a	a の値を返す
(単項の)-	符号反転	-a	a の符号を逆にした値を返す
+	加算	a+b	a に b を加えた値を返す
-	減算	a-b	a から b を引いた値を返す
*	積算	a*b	a と b の積の値を返す
/	除算	a/b	a を b で割った商を返す a,b が共に整数なら商も整数を返す。5/2 は 2 a か b が実数なら実数の割り算結果を返す。5.0/2 は 2.5
%	剰余算	a%b	a を b で割った余りを返す (左右は整数) 5%2 は 1

代入演算子

1) C 言語では代入も演算子で行う。

演算子「=」は左辺の変数に右辺の値を代入する。「=」は値が等しいことを示す記号ではない。

a=10; は a に値 10 を代入する。

2) 代入演算子は代入した値を戻す。

a=b=10; の様な演算は b=10 が戻す値 10 を a に代入する。

3) 演算と組み合わせた代入演算子も用意されている。

total = total + n; の様な演算を伴う代入は

total += n; の様に簡略に表現できる(複合代入演算子)。

この書き方は変数 total に加算する事を明確に示す効果がある。

演算子		書式	代入する値(戻す値でもある)
=	代入	a=b	b の値を a に代入する
+=	加算と代入	a+=b	a に b を加えた値を a に代入する (a=a+b と同じ)
-=	減算と代入	a-=b	a から b を引いた値を a に代入する
=	積算と代入	a=b	a と b の値の積を a に代入する
/=	除算と代入	a/=b	a を b で割った商を a に代入する a が整数型の変数なら整数の割り算結果を戻す。 a が実数型の変数なら実数の割り算結果を戻す。
%=	剰余算と代入	a%=b	a を b で割った余りを a に代入する

インクリメント・デクリメント 演算子

変数の値を 1 増減する演算子、戻す値は演算子が前置か後置かで異なる。

演算子	副作用	書式	戻す値
++	変数の値を 1 増加	++a	前置演算子の場合は増加後の値を戻す
++	変数の値を 1 増加	a++	後置演算子の場合は増加前の値を戻す
--	変数の値を 1 減少	--a	前置演算子の場合は減少後の値を戻す
--	変数の値を 1 減少	a--	後置演算子の場合は減少前の値を戻す

a=a+1 ;

と書くと a の値と 1 の値を加算回路に入力し、出力を a に代入する形で a の値を 1 増やす。このような 1 増やすような処理は プログラムの中に多く現れる。そこで、CPU は汎用の加算処理とは別に、カウントアップ専用処理を用意していることが多い。機械語にカウントアップ命令があれば、加算する値 1 をメモリー上に用意する必要も無くなるので機械語も短くできる。

++a;

は a の値をカウントアップすることを明確にした書き方。a= b +1 や a=a+2 と書き間違える可能性も無くなる。

ビット演算子

現在の計算機はデータを bit 単位で保持している。C 言語ではこの bit データを操作する演算子も用意されている。

演算子		書式	戻す値
~	ビット反転	~a	ビットを反転した値を戻す
&	ビット単位の AND	a & b	a と b のビット毎の AND をとった値
	ビット単位の OR	a b	a と b のビット毎の OR をとった値
^	ビット単位の XOR	a ^ b	a と b のビット毎の XOR をとった値
<<	上位 bit 側へのシフト	a << n	a のビットを上位側に n 個シフトした値 下位ビットは 0 を補充
>>	下位 bit 側へのシフト	a >> n	a のビットを下位側に n 個シフトした値 符号無しの変数は上位に 0 を補充（論理シフト） 符号付の変数は最上位ビットの値を補充（算術シフト）

練習問題 10- 1

int 型の変数 x,y を任意の値で初期設定し、それらの値の和、差、積、商と余りを求めるプログラムを作成してください。(x、y は任意です。)

結果の表示 (x=21,y=10 にした例)

x = 21 y = 10

x+y = 31

x-y = 11

x*y = 210

x/y= 2 x%y = 1

<< サンプル >>

練習問題 10- 2

次のプログラムを実行して結果を確認しなさい。

```
#include <stdio.h>
int main(void){
    int a,b;
    a = 1;
    b = ++a;
    printf("a=%d¥tb=%d¥n", a, b);
    a = 1;
    b = a++;
    printf("a=%d¥tb=%d¥n", a, b);
    return 0;
}
```

インクリメント 演算子	b = ++a;	a = a + 1; b=a;	変数 a に 1 加算してから 変数 b に代入する。
	b = a++;	b = a; a = a + 1;	変数 b に代入してから 変数 a に 1 加算する。
デクリメント 演算子	b = --a;	a = a - 1; b = a;	変数 a に 1 減算してから 変数 b に代入する。
	b = a--;	b = a; a = a - 1;	変数 b に代入してから 変数 a に 1 減算する。

<< サンプル >>

練習問題 10- 3

次のプログラムが実行された時の結果はどのようなになりますか。

```
#include <stdio.h>

int main(void){
    int x, y, z;
    x = 2;
    y = 4;
    x = x + 1;
    y++;
    x -= y;
    z = x + y;
    printf("x = %d, y= %d, z = %d\n",x , y, z);
    return 0;
}
```

<< サンプル >>