

第2回C言語講座 (By y.t)

```
    a += 6;               //a に 6 加算して代入
    printf("a = %d\n", a);
    a++;                 //a に1を加算
    printf("a = %d\n", a);
    return 0;
}
```

////////////////////////////////////

演算子はまだまだありますが、必要に応じて紹介していきます。

演算子のことをオペレータといいます。逆に演算対象をオペランドといい、演算子の左側を左オペランド、右側を右オペランドと呼んだりします。

2. if 文

プログラムではある条件でのみ動作する処理が必要になることがあります。例えばゲームの場合、A ボタンが押されたらジャンプする。というものです。二次方程式の解だと判別式の結果で解の表示を分けなければなりません。if 文はこのような条件分岐を実現する手段です。

////////////////////////////////////

```
/*
C言語講座サンプル2
if 文
by y.t
*/
#include <stdio.h> /*入出力のヘッダーファイル*/
/*メイン関数*/
int main()
{
    int a;               //変数の宣言
    printf("5を入力してください\n");
    scanf("%d", &a);
    if (a == 5)
    {
        printf("5が入力されました.\n");
    }
    return 0;
}
```

////////////////////////////////////

if 文中の () 内を条件式といいます。では、条件が複数ある場合の分岐はどうすればいいのでしょうか？実は、複数の条件は演算子を使って表すことができます。条件式で使う演算子を表にまとめました。

表2. 関係を表す演算子

| 演算子 | 機能 |
|-----|--|
| == | 左オペランドと右オペランドが等しい(等価演算子) |
| != | 左オペランドと右オペランドが等しくない(不等価演算子) |
| < | 左オペランドが右オペランドより小さい |
| <= | 左オペランドが右オペランド以下 |
| > | 左オペランドが右オペランドより大きい |
| >= | 左オペランドが右オペランドより以上 |
| && | 左オペランドと右オペランド共に真ならば(積結合演算子または条件AND演算子) |
| | 左オペランドか右オペランドが真ならば(和結合演算子または条件OR演算子) |
| ! | オペランドが否定されたなら(否定演算子またはNOT演算子) |

if 文では数学のような $3 < x < 6$ という記述は出来ません。このような記述はコンパイルエラーにはなりませんが常に真という結果になります。 $3 < x \ \&\& \ x < 6$ と書くのが正解となります。複数条件のサンプルを以下に示します。

//

```
/*
C言語講座サンプル3
if 文
by y.t
*/
#include <stdio.h> /*入出力のヘッダーファイル*/
/*メイン関数*/
```

```
int main()
{
    int a; //変数の宣言
    printf("数字を入力してください\n");
    scanf("%d", &a);
    if (0 <= a && a <= 10)
    {
        printf("a は 0~10 の間にあります。 \n");
    }
    if (a % 2 == 0 || a % 5 == 0)
    {
        if (a % 2 == 0 && a % 5 == 0)
        {
            printf("a は 2 の倍数でありかつ 5 の倍数です。 \n");
        }
        else
        {
```


第2回C言語講座 (By y.t)

```
int playerNumber;
srand(clock());
playerNumber = rand() % 10;
printf("コンピュータが選んだ数字を当ててね¥n");
scanf("%d", &playerNumber);
//ここに正誤判定などさまざまな処理を追加する
return 0;
}
```

5. 本日の関数

表5. 本日使った新しい関数

| 関数 | 引数 | 戻り値 | 処理内容 |
|---------------------------------------|---------------------|-------------------------|--|
| void srand(unsigned int seed); | seed 乱数の種 | | 乱数を初期化する。 必須ヘッダー stdlib.h |
| int rand(void); | | 乱数 | 乱数を生成する。 必須ヘッダー stdlib.h |
| clock_t clock(void); | | プログラム実行時から経過したク ロック数 | プログラム実行時から 経過したクロック数を 取得する。 必須ヘッダー time.h |