

# プログラミング入門

## 第5回講義

分岐(その2) - switch-case -  
ループ(その1) - while -

◆ マークのあるサンプルプログラムは  
`/home/course/prog0/public_html/2006/lec/source/`  
下に置いてありますから、各自自分のディレクトリに  
コピーして、コンパイル・実行してみてください

# 良くあるミス(if編1)

## ■ {}のつけ忘れ

```
if (a != b)
  a = b;
printf("a,b were not equal\n");
```

このprintfはif文の外なので、必ず実行される！

## ■ 正しくは

```
if (a != b){
  a = b;
  printf("a,b were not equal\n");
}
```

## 良くあるミス(if編2)

### ■ if文に「;」をつけてしまう

```
if (a != b) ;  
    printf("a,b were not equal\n");
```

このprintfはif文の外なので、必ず実行される！

### ■ 正しくは

```
if (a != b)    
    printf("a,b were not equal\n");
```

# 良くあるミス(if編3)

## ■ =と==の違い

```
a = 1;
b = 0;
if (a = b) {
    printf("a,b were equal\n");
}
```

bをaに代入し式の値もbとなる

## ■ 正しくは

```
a = 1;
b = 0;
if (a == b) {
    printf("a,b were equal\n");
}
```

# インデント(段づけ)をしよう！

- 段づけをすると見やすく、間違いにくくなる
- 以下のスタイル1、2が標準的。

スタイル1(行数が少なく、コンパクトにまとまる)

```
if (a == b) {  
    printf("a,b were equal\n");  
}
```

自動的にインデントする方法は、  
Lec13-7,8で説明します

スタイル2(カッコの対応が見易い)

```
if (a == b)  
{  
    printf("a,b were equal\n");  
}
```

# ◆ 複雑なif文に騙されるな(1)

以下は日付を入力して、それが月の最後の日かどうかを表示するプログラム。(ただし閏年と、あり得ない日付(5月40日など)には対応していない)

```
#include <stdio.h>
main()
{
    int m,d;

    printf("月と日を空白で区切って入力してください : ");
    scanf("%d %d",&m, &d);

    if((m == 1) || (m == 3) || (m == 5) || (m == 7) ||
        (m == 8) || (m == 10) || (m == 12))
        if(d == 31) printf("%d月%d日は月の最後の日\n",m,d);
    else if((m == 4) || (m == 6) || (m == 9) || (m == 11))
        if(d == 30) printf("%d月%d日は月の最後の日\n",m,d);
    else if(m == 2)
        if(d == 28) printf("%d月%d日は月の最後の日\n",m,d);
    else printf("%d月%d日は月の最後の日ではない\n",m,d);
}
```

みかけは正しそう  
だが、実際はうまく  
動かない

/home/course/prog0/public\_html/2006/lec/source/lec05-1.c

## ◆ 複雑なif文に騙されるな(2)

前頁のプログラムをちゃんとインデント (if-elseの關係に注目して段づけする) してみると、以下のようになり、論理的におかしい事が分かる。

```
#include <stdio.h>
main()
{
    int m,d;

    printf("月と日を空白で区切って入力してください : ");
    scanf("%d %d",&m, &d);

    if((m == 1) || (m == 3) || (m == 5) || (m == 7) ||
        (m == 8) || (m == 10) || (m == 12))
        if(d == 31) printf("%d月%d日は月の最後の日\n",m,d);
    else if((m == 4) || (m == 6) || (m == 9) || (m == 11))
        if(d == 30) printf("%d月%d日は月の最後の日\n",m,d);
    else if(m == 2)
        if(d == 28) printf("%d月%d日は月の最後の日\n",m,d);
        else printf("%d月%d日は月の最後の日ではない\n",m,d);
}
```

[/home/course/prog0/public\\_html/2006/lec/source/lec05-2.c](/home/course/prog0/public_html/2006/lec/source/lec05-2.c)

## ◆ 複雑なif文に騙されるな(3)

これが正しいプログラム(インデントも正しくされている)

```
#include <stdio.h>
main()
{
    int m,d;

    printf("月と日を空白で区切って入力してください : ");
    scanf("%d %d",&m, &d);

    if(((m == 1) || (m == 3) || (m == 5) || (m == 7) ||
        (m == 8) || (m == 10) || (m == 12))&&(d == 31))
        printf("%d月%d日は月の最後の日\n",m,d);
    else if(((m == 4) || (m == 6) || (m == 9) || (m == 11))&&(d == 30))
        printf("%d月%d日は月の最後の日\n",m,d);
    else if((m == 2)&&(d == 28)) printf("%d月%d日は月の最後の日\n",m,d);
    else printf("%d月%d日は月の最後の日ではない\n",m,d);
}
```

lec05-3c.cは  
更に閏年にも  
対応した



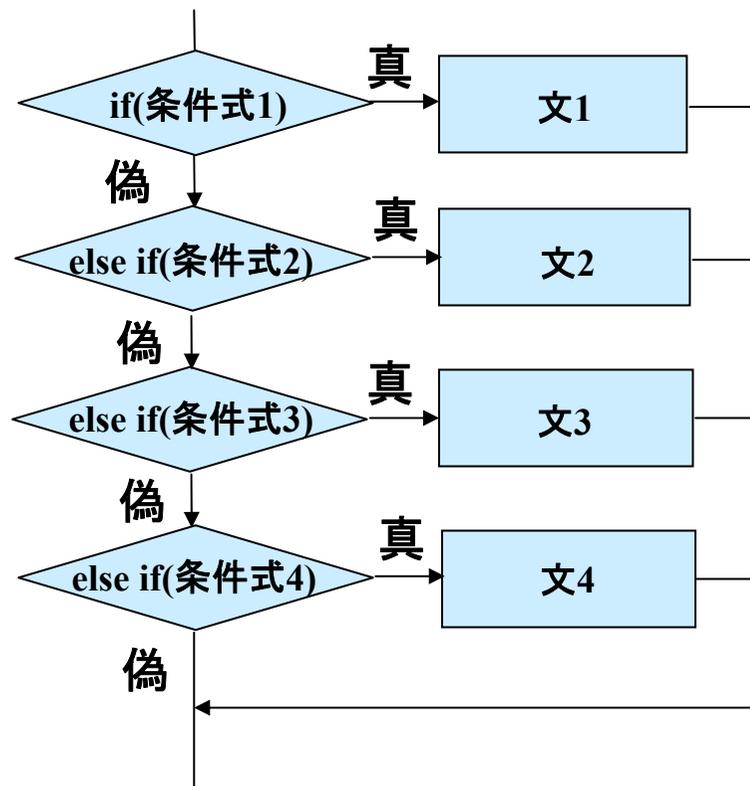
## 交通信号機：先週の復習

```
#include <stdio.h>
main() {
    int signal;
    printf("0:red, 1:green, 2:yellow : ");
    scanf("%d",&signal);
    if(signal == 0) printf("Stop\n");
    else if(signal == 1) printf("Go\n");
    else if(signal == 2) printf("Be careful\n");
    else printf("Look at the traffic signal\n");
}
```

[/home/course/prog0/public\\_html/2006/lec/source/lec04-4.c](/home/course/prog0/public_html/2006/lec/source/lec04-4.c)

# if～else if～else文の図的表現

フローチャート

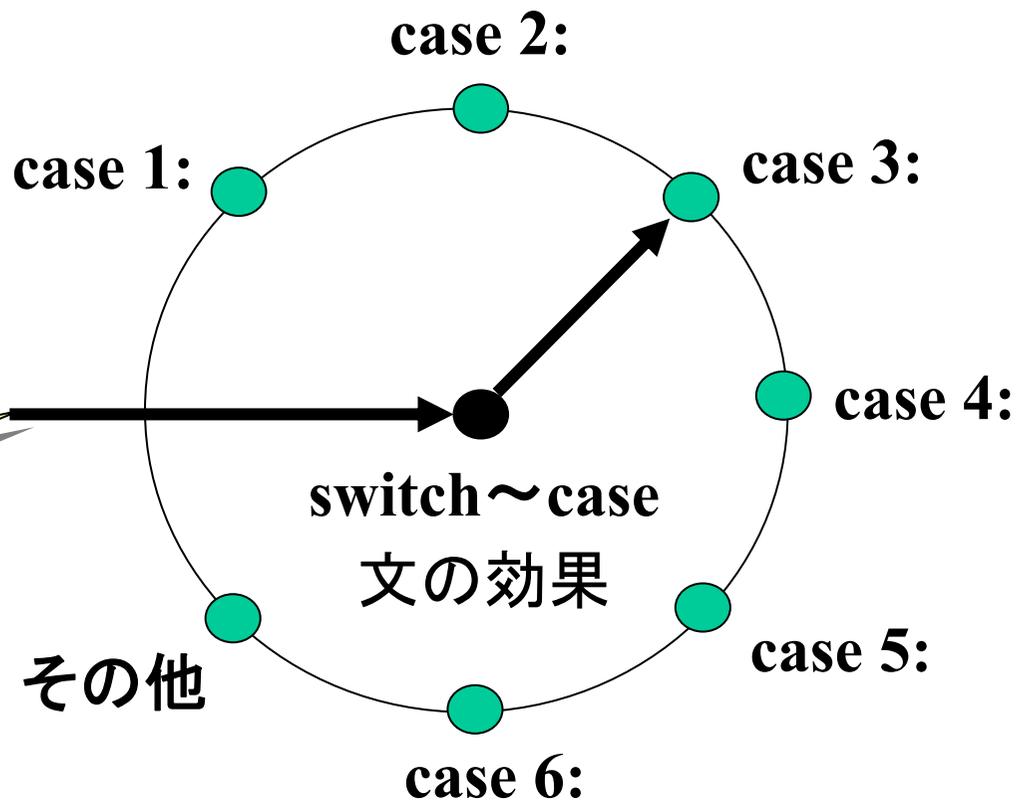


# 多岐選択のもう一つの方法-switch文-(p.117)

条件式の値が、**定数式**で表せるとき!!

switch文  
が便利

式の数 = 3 のとき、  
case 3: の文を選択



# switch-case文の書式

```
switch (式) {  
  case 定数式1:  
    文1;  
    文2;  
    ...  
    break;  
  case 定数式2:  
    ...  
    break;  
  default:  
    ...  
    break;  
}
```

式の値と定数式の値が等しいとき、「:」に続く文を選択

複数の文を書けるが、{}は必要ない!

switch文から抜ける

どのcaseとも値が不一致のとき選択

default中のbreakはなくても良い

# 入賞者の判別プログラム (p.118)

```
scanf ("%d", &jyuni);  
switch(jyuni) {  
  case 1:  
    printf("一位\n");  
    break;  
  case 2:  
    printf("二位\n");  
    break;  
  case 3:  
    printf("三位\n");  
    break;  
  default:  
    printf("残念\n");  
}
```

変数 jyuni の値によって  
1なら「一位」と表示  
2なら「二位」と表示  
3なら「三位」と表示  
1, 2, 3位以外なら  
「残念」と表示



# switch~case文のポイント(1)

- 式の値は、整数(又は文字)
- 定数式の値は、整数(又は文字)
- **default**は、必ず **case** 文の最後に記述
- 一つのswitch-case文の中で、**定数式の値が等しい複数のcase文が存在するのは誤り**

```
case 1:  
    ...  
    break;  
 case 1:  
    ...  
    break;
```



## switch~case文のポイント(2)

通常は**必ず**break文をつける。但し、意図してつけない場合は**必ず**コメントにてその旨を説明しておく

```
switch(month) {  
    /* 以下の7つの月は同じ処理なのでbreak省略 */  
    case 1:  
    case 3:  
    case 5:  
    case 7:  
    case 8:  
    case 10:  
    case 12:  
        printf("31日まである月\n");  
        break;  
    default:  
        printf("短い月\n");  
}
```

[/home/course/prog0/public\\_html/2006/lec/source/lec05-4.c](/home/course/prog0/public_html/2006/lec/source/lec05-4.c)



## 交通信号機: switch版(p.125)

```
#include <stdio.h>
main() {
    int signal;
    printf("0:red, 1:green, 2:yellow : ");
    scanf("%d",&signal);
    switch(signal) {
        case 0:
            printf("Stop\n");
            break;
        case 1:
            printf("Go\n");
            break;
        case 2:
            printf("Be careful\n");
            break;
        default:
            printf("Look at the traffic signal\n");
    }
}
```

見かけはif-else-if版より  
長いが、読み易い

[/home/course/prog0/public\\_html/2006/lec/source/lec05-5.c](/home/course/prog0/public_html/2006/lec/source/lec05-5.c)

# 良くあるミス(switch編1)

## ■ switch文にbreakを付け忘れる

```
switch(a) {  
  case 1:  
    printf("a=1\n");  
    /* break つけ忘れ */  
  case 2:  
    printf("a=2\n");  
    break;  
  ..  
}
```



aが1の時、2つとも表示される。  
aが2の時は正常に動作する。

# 良くあるミス(switch編2)

## ■ case文に書けるのは定数だけ

```
switch(a) {  
  case >5:  
    printf("a>5\n");  
    break;  
  case 2:  
    printf("a=2\n");  
    break;  
  ..  
}
```



「case label does not reduce to an integer constant」  
と言うメッセージが出てコンパイルエラーになる

# 制御の流れのまとめ

- プログラム実行の流れを把握
  - 接続
  - 選択(前回・今回のテーマ)
  - 繰り返し(今回・次回のテーマ)
- 選択のプログラミング
  - if文,if～else文,if～else if文
  - switch～case文,break文



# ループとは何か(p.132)

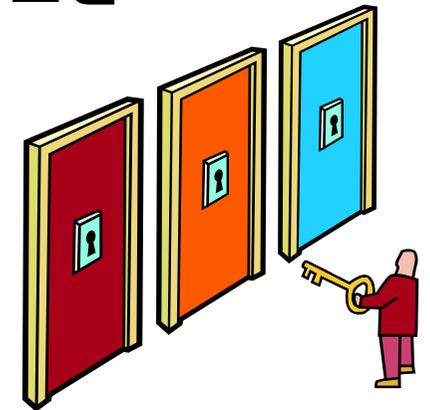
- 同じことを、ある条件が成立している間何度も繰り返す
  - お金がある**限り**宝くじを買う
  - グラウンドを**20周**する
- Cの繰り返しの命令(文)
  - `while, for, (do-while)`



# ループの種類(p.132)

## ■ 見張り方式

- ある特定の条件を満たしている間は、処理を繰り返す(例:  $a$ が0より大きい間)
- 一般に繰り返し回数はあらかじめ決まっていない



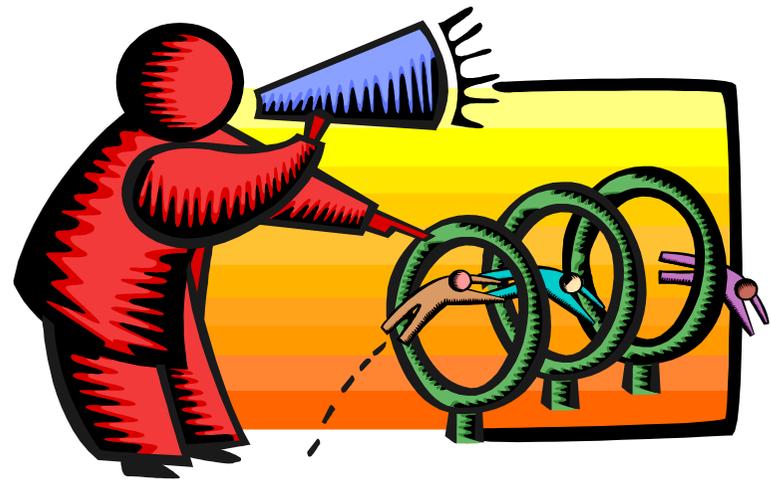
## ■ カウンタ方式

- ループの回数を数える変数(カウンタ変数などと呼ぶ)を用いて繰り返し処理を決定する(例: 10回)
- 一般的に繰り返し回数は固定

# while 文 (見張り方式に良く使われる) (p.132)

条件式が真の間はループの中身を実行し続け、  
条件式が偽になれば終了 (ループの次の文を  
実行) する

```
while (条件式) {  
    文;  
    ...  
}
```



# ◆ サンプルプログラム

```
#include <stdio.h>
main()
{
    int i, total;
    i = 1;
    total = 0;
    while( i <= 10 ){
        total = total + i;
        i = i + 1;
    }
    printf("1から10までの和は %d です\n", total);
}
```

1から10までの和を求める

変数「i」、「total」を用意する  
「i」に1を代入  
「total」に0を代入  
「i」が10以下の間ループする  
「total」にiを加える  
「i」に1を加える  
ループここまで  
「total」を表示する

/home/course/prog0/public\_html/2006/lec/source/lec05-6.c

# 実行結果

```
std1dc1{s1000000}1: gcc lec05-6.c  
std1dc1{s1000000}2: ./a.out  
1から10までの和は 55 です  
std1dc1{s1000000}3:
```