

# プログラミング入門

## 第7回

# switch文(p.133)

## ◆ switch文

- 式を評価した値によって処理を分けるときに使用する.
- if～else if～elseよりも簡単に記述できる.

```
switch (式) {  
    case 定数1:  
        文1; // 式の評価が定数1の場合に処理  
        break;  
    case 定数2:  
        文2; // 式の評価が定数2の場合に処理  
        break;  
    ...  
    default:  
        文D; // 式の評価がいずれでもなかった場合  
        break;  
}
```

# switch文(p.133)

## ◆ラベル

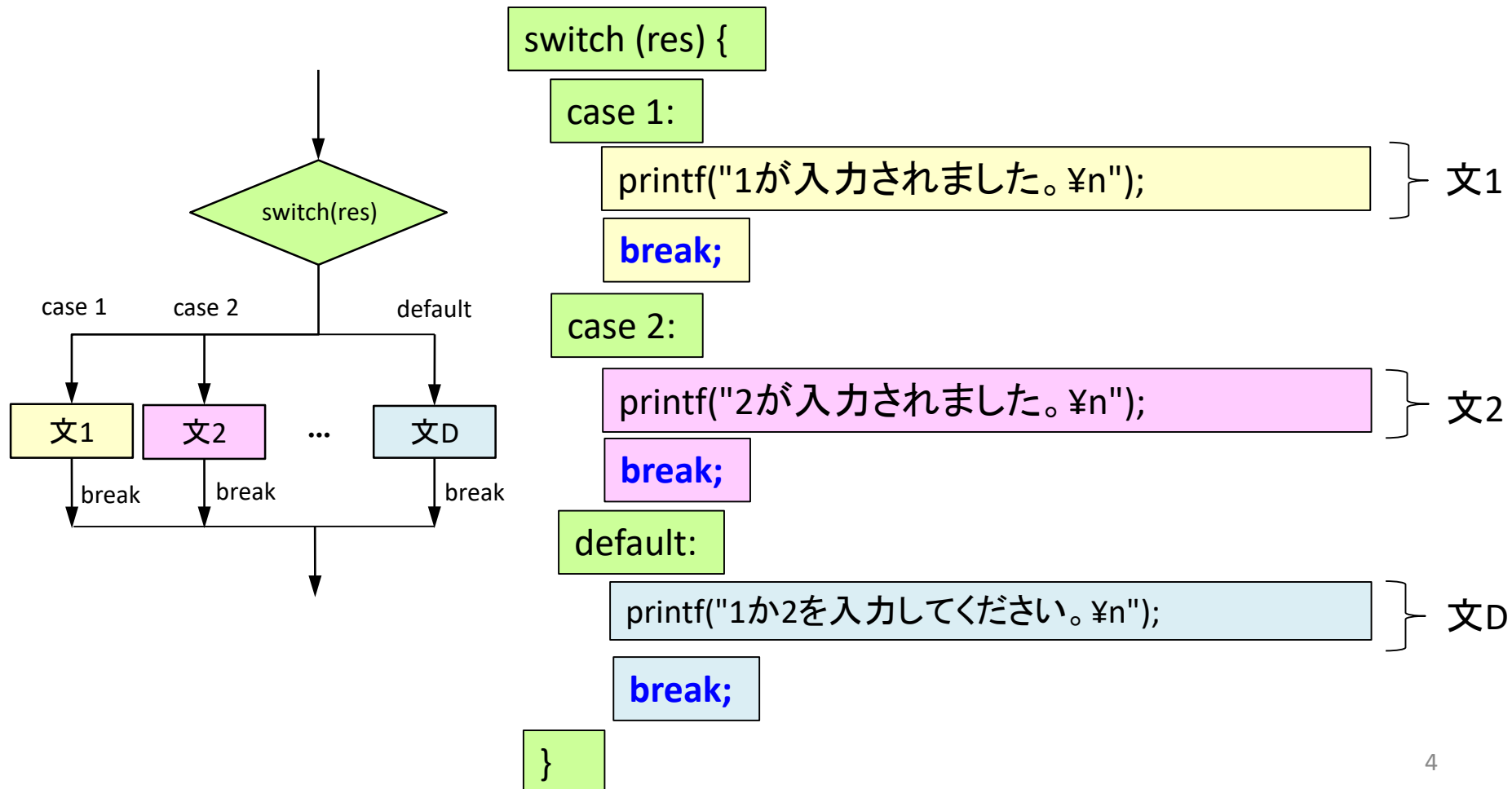
- case 定数1:やcase定数2:は処理の移動先を表すものであり、**ラベル**と呼ぶ
- どのcaseラベルにも該当しない場合には、defaultラベルに移動する

```
switch (res) {  
    case 9:      // resの値がcaseラベルの定数9と一致したときの移動先  
        ...  
    case 5:      // resの値がcaseラベルの定数5と一致したときの移動先  
        ...  
    default:     // どのcaseラベルにも該当しなかった時の移動先  
        ...  
}
```

# switch文(p.134-135)

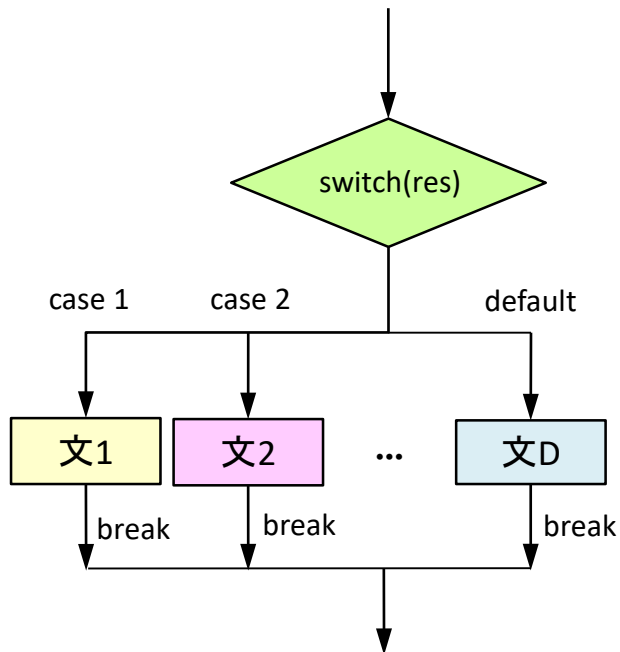
## ◆すべてのcaseのあとにbreak文がある場合

- break文によりswitch文のブロックから抜け出る

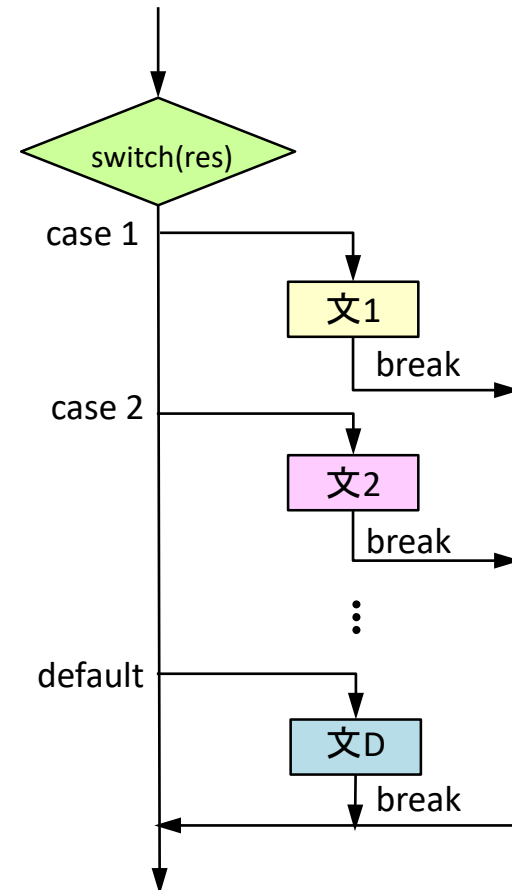


# switch文のフローチャート

switch 文に対するフローチャートで分岐を縦に伸ばす方向に描くと右図のようになる

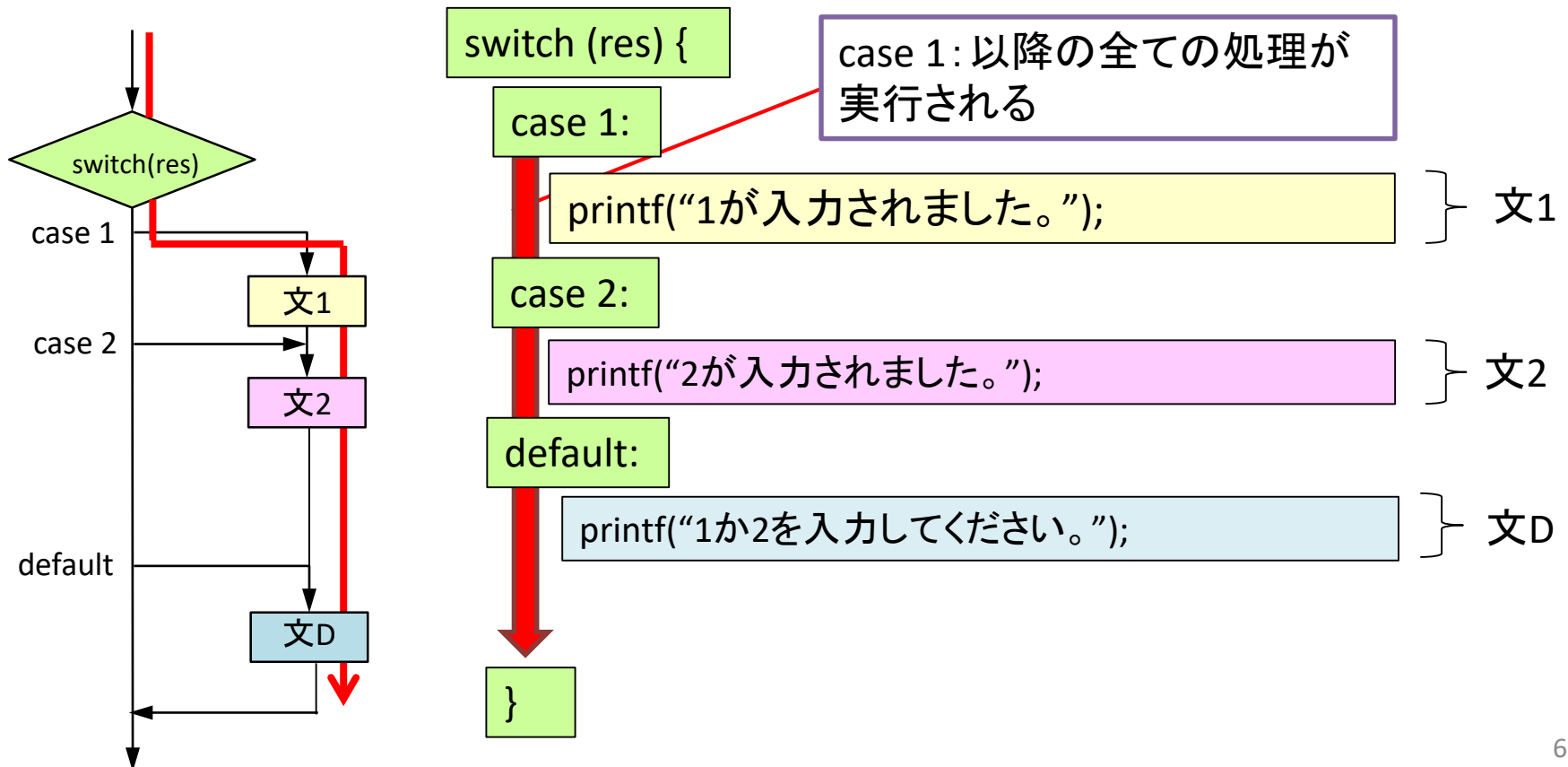


=



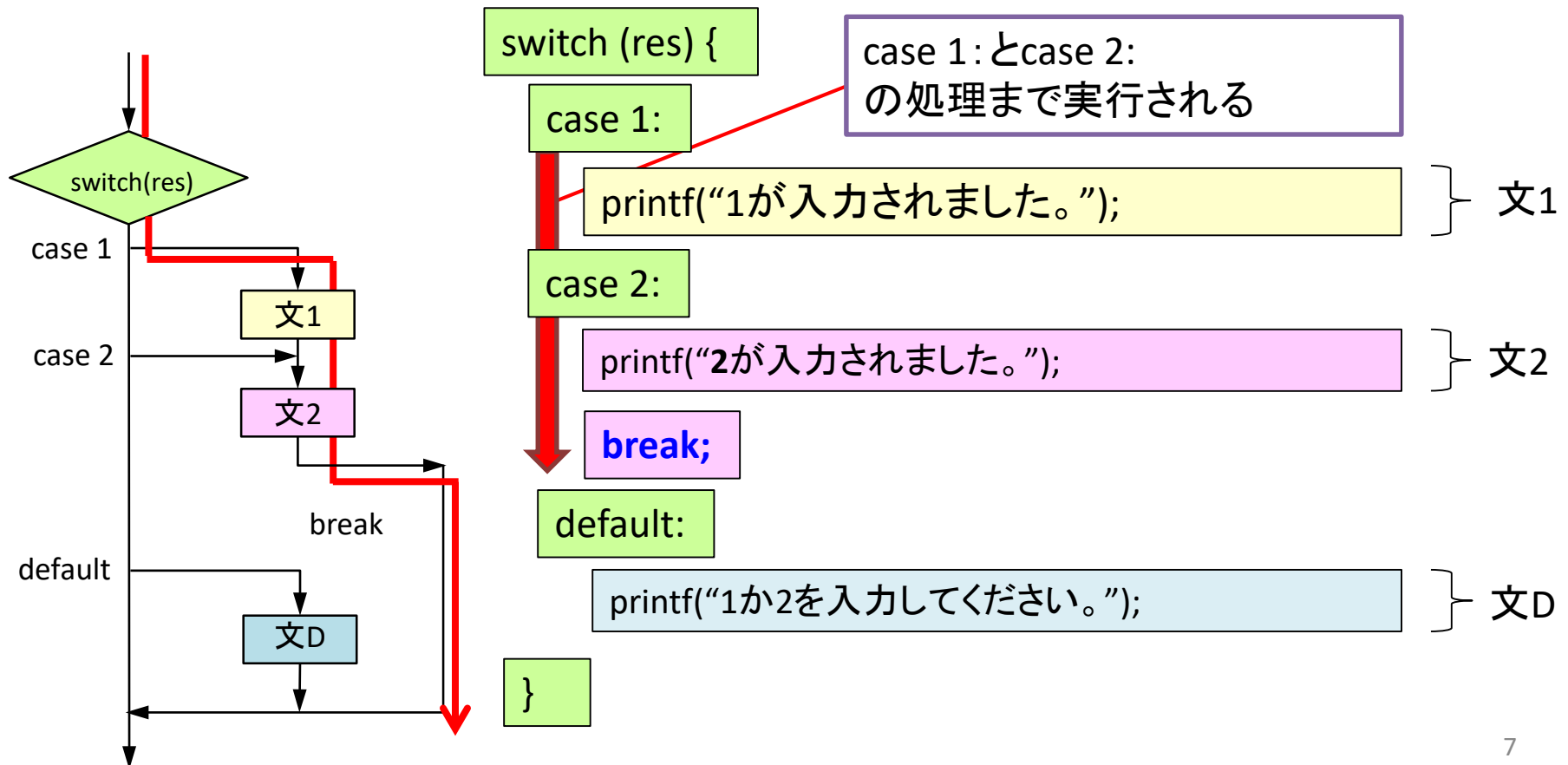
# break文が抜けていると？ (p.136)

- ◆ switch文は処理の移動先をcaseラベルで指定しているだけなので、 break文が無いと次のcaseの処理も実行されてしまう



# break文が抜けていると？(補足)

◆途中でbreakがある場合はswitchの処理を抜け出る



# 論理演算子(p.138)

## ◆論理演算子(logical operator)

- 複雑な条件を記述するために用いる

成績が「優」であり, **かつ**, お金がある



「**かつ**」を表す論理演算子「**&&**」

(成績が「優」である) **&&** (お金がある)

条件式  
(論理式ともいう)

- ◆ **&&** の左オペランドと右オペランドの両方が 真 の場合,  
式**全体**の値が 真 となる. それ以外の場合は 偽 となる.



# 論理演算子(p.139)

## ◆ 論理積(&&) 「左オペランド && 右オペランド」

- 左オペランドと右左ペランドがともに 真 の場合のみ, 式全体が 真 になる. それ以外は 偽 となる.
- 日本語では「～、かつ～」

## ◆ 論理和(||) 「左オペランド || 右オペランド」

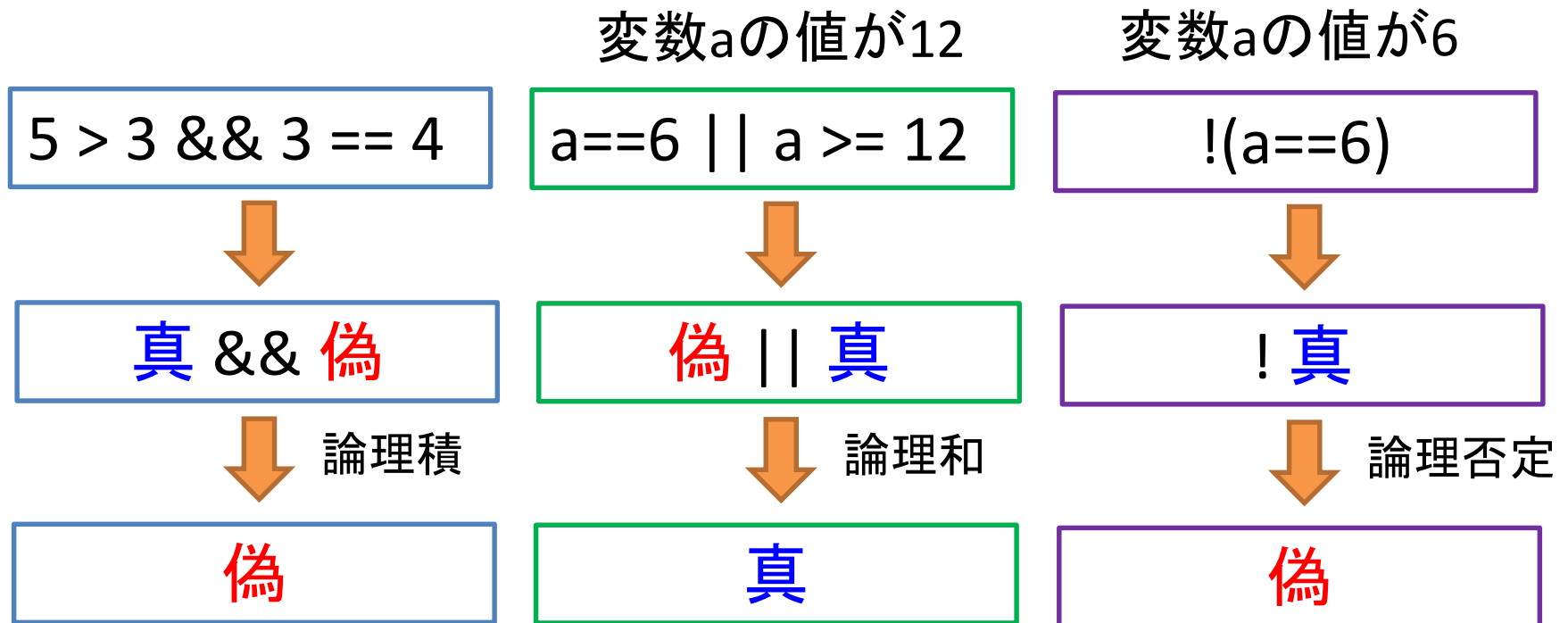
- 左オペランドと右オペランドの少なくとも一方が 真 の場合, 式全体は 真 になる. それ以外は 偽 となる.
- 日本語では「～、または～」

## ◆ 論理否定(!) 「!オペランド」

- オペランドが 偽 の場合, 式全体は 真 になる. それ以外は 偽 になる.
- 日本語では「～ではない」

# 論理演算子(p.139-140)

## ◆ 論理演算子を使ったコード



# 論理演算子(p.141)

## ◆Sample6.cのソースコード

```
res = getchar(); // キーボード入力して1文字読み込んで変数resに格納
if (res == 'Y' || res == 'y'){ // resの値が'Y'または'y'のとき条件が真
    printf("あなたは男性ですね。¥n");
}
else if(res == 'N' || res == 'n'){ // resの値が'N'または'n'のとき条件が真
    printf("あなたは女性ですね。¥n");
}
else{
    printf("YかNを入力してください。¥n");
}
```

# 条件演算子のしくみ(p.142-143)

- ◆ 条件演算子(conditional operator)

第1オペランド ? 第2オペランド : 第3オペランド

- ◆ 第1オペランドは, 「(条件式)」の形式
- ◆ 第2オペランドと第3オペランドは, 同じ型の値を持つ式
- ◆ 式全体の値は, 第1オペランドの条件式が 真 ならば, 第2オペランドの値になり, そうでなければ, 第3オペランドの値になる.
- ◆ 典型的な使用方法是次のようになる.

変数 = (条件式) ? 式T : 式F;

例:     char ans = (res == 1) ? 'A' : 'B';  
          // res が 1 ならば, ans には 'A' が代入され,  
          // res が 1 でないならば, ans には 'B' が代入される.

# 条件演算子のしくみ(p.142-143)

- ◆ 簡単な条件判断の場合は, if 文の代わりに条件演算子を利用すると便利である.

