

# 3.条件分岐

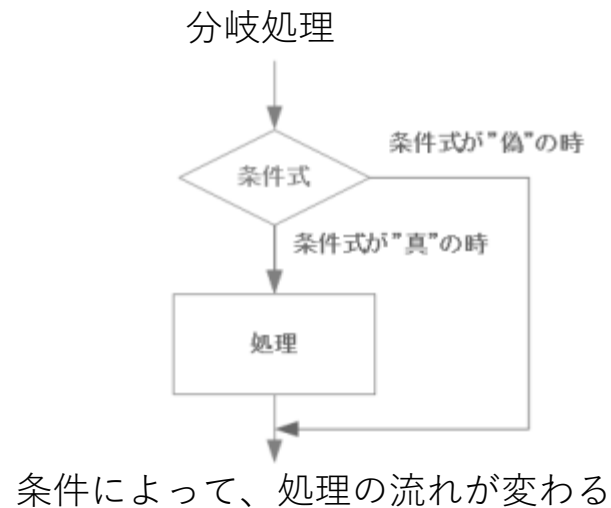
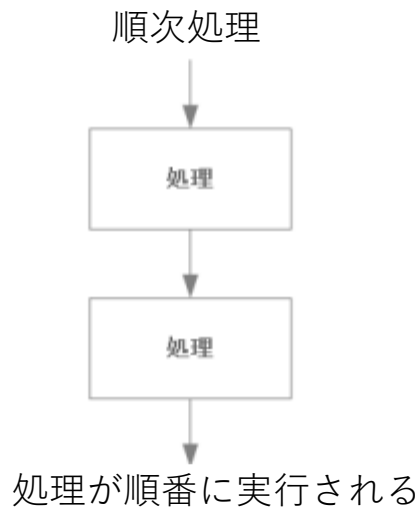
C言語入門①

# 順次処理

- プログラム中に記述された様々な処理を、その順番とおりに実行するだけのものを順次処理と言う

# 分岐処理

- ある条件で処理の流れが変わる処理を、**分岐処理**（ぶんきしゅり）と言う
- 分岐処理を記述するための命令として、**if(イフ)**と、**switch(スイッチ)**という命令が用意されている



# if文

- 「もし○○だったら、△△する」
- 条件が成り立った場合と成り立たなかった場合の2つ



# If文のサンプル

**sample3-1.c**

```
if(a > 0){  
    printf("入力した値は、正の数です。¥n");  
}
```

# 条件の例

演算子	意味	使用例
>	より大きい	$a > 0$
>=	以上	$a \geq 0$
<	より小さい	$a < 0$
<=	以下	$a \leq 0$
==	等しい	$a == 0$
!=	等しくない	$a != 0$

# if～else文

- 「もし○○だったら、△△する。そうでなければ××する」
- 条件が成り立つ場合のみ、処理 1 が実行される
- 条件が成り立たない場合、処理 2 が実行される

if(条件式)



# If文～elseのサンプル

sample3-2.c

```
if (a > 0) {  
    printf("入力した値は、正の数です。¥n");  
} else {  
    printf("入力した値は、正の数ではありません。¥n");  
}
```



# if～elseif～else①

- 複数の条件のどれにあてはまるかによってそれぞれ違う処理を行いたい時は、**elseif(エルスイフ)**を使う
- if文がなりたたなければ、elseifでさらに条件をしぼれる
- elseifはいくつつけてもかまわない

if～elseif～else②

if(条件式①)

条件式①真  
(true)

処理①

elseif(条件式②)

条件式①偽 (false)  
条件式②真 (true)

処理②

elseif(条件式③)

条件式①偽 (false)  
条件式②真 (false)  
条件式③真 (true)

処理③

else

処理④

条件式①～③偽 (false)

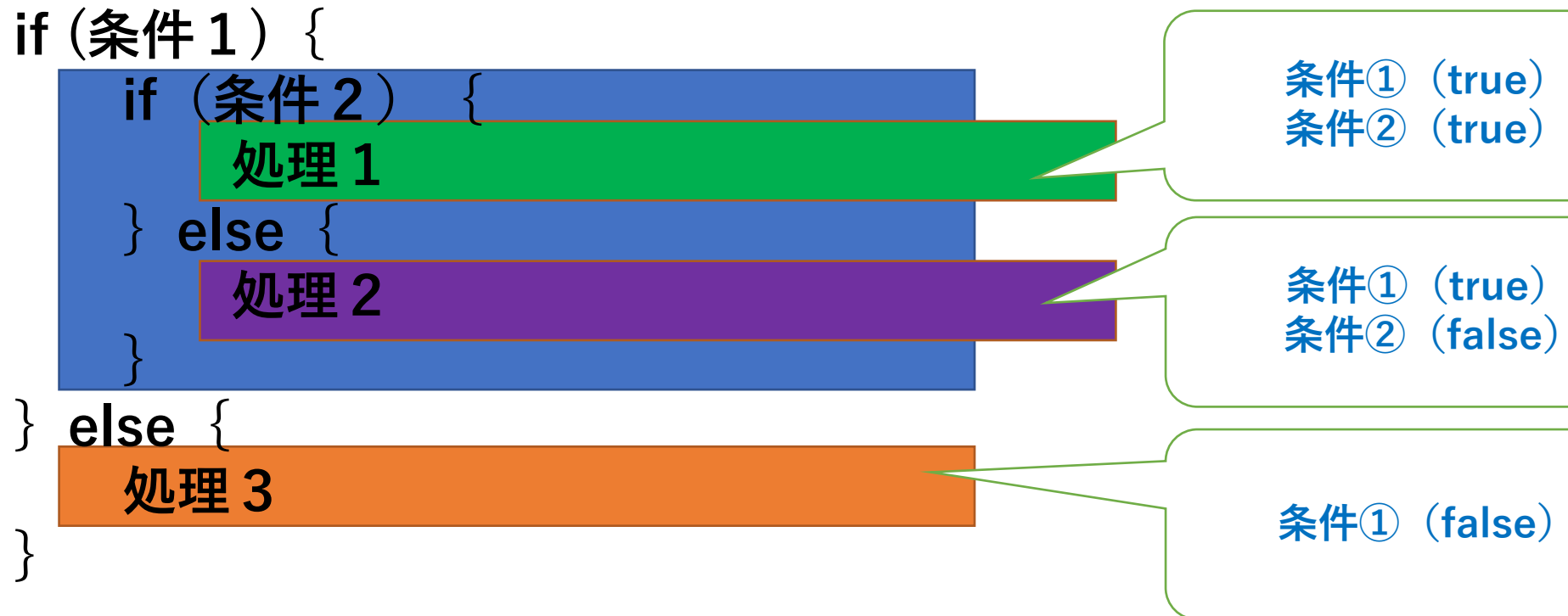
# If文～elseif～elseのサンプル

**sample3-3.c**

```
int num;
scanf("%d", &num);
if (num == 1) {
    printf("one¥n");
} elseif (num == 2) {
    printf("two¥n");
} elseif (num === 3) {
    printf("three¥n");
} else {
    printf("不適切な値です。¥n");
}
```

# if文のネスト

- If文の中に、更にif文を入れることが出来る。
- **ネスト**・**入れ子**と言う。



# 論理式との組み合わせ

- **&& (論理積 : AND)** . . . **A かつ B**
  - if( 条件式① && 条件式②) → 条件式①と②が両方成り立つ
- **|| (論理和 : OR)** . . . **A か B**
  - if( 条件式① || 条件式②) → 条件式①か②どちらかが成り立つ
- **!(否定:NOT)** . . . **A ではない**
  - if(!(条件式①)) → 条件式①が成り立たない

# if文のネストのサンプル

## sample3-4.c

```
if(1 <= dice && dice <= 6){  
    // さいころの目が、偶数か、奇数かで、処理を分ける。  
    if(dice == 2 || dice == 4 || dice == 6){  
        printf("丁（チョウ）です。¥n"); // 偶数ならば丁（チョウ）  
    }else{  
        printf("半（ハン）です。¥n"); // 奇数ならば半（ハン）  
    }  
}else{  
    printf("範囲外の数値です。¥n");  
}
```

# switch文①

**sample3-5.c**

- 式の値を条件とし、またそれを複数指定することができる。

## switch文②

```
switch(式) {
```

```
case 値①:
```

処理①

式が値①  
をとるとき

```
break;
```

```
case 値②:
```

処理②

式が値②  
をとるとき

```
break;
```

```
default:
```

処理④

式がそれ以外の値  
をとるとき

```
}
```