

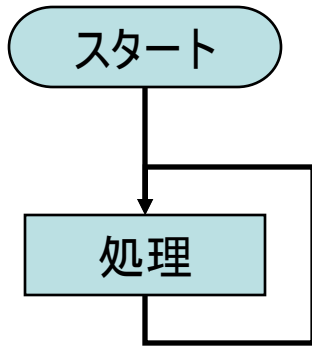
# プログラミング論I

## (4) for (繰り返し処理)

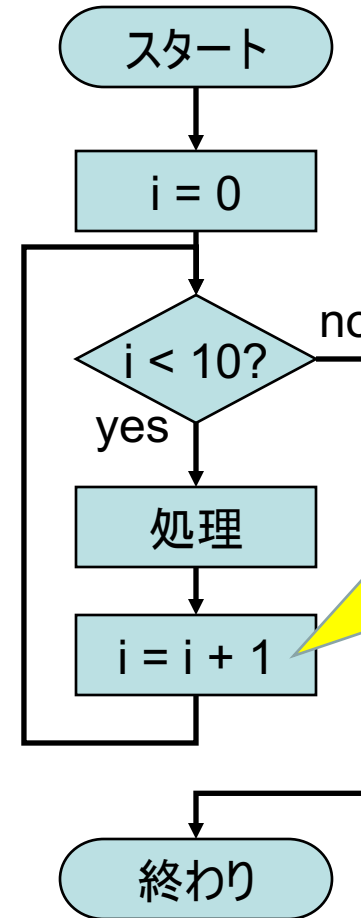
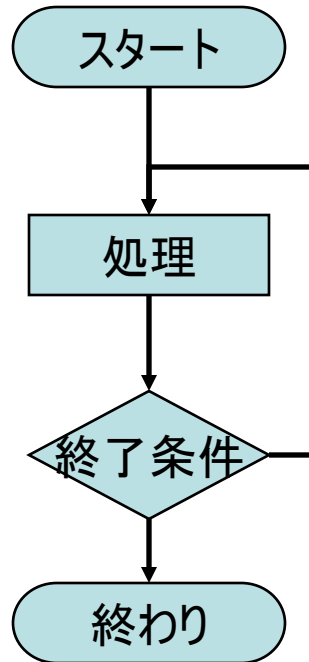
電子情報工学専攻 日浦 慎作

座席は自由です

# 繰り返し



・・・永遠に終わらない！  
(無限ループ)



数学ではありえない数式だが、C言語の  $=$  は代入なので  $i$  を1増やす処理である

- 繰り返しには「終了」のメカニズムが必要

10回、繰り返し  
(変数  $i$  で繰り返し回数を管理)

# 繰り返しとは

- プログラムの同じ部分を何度も実行する
  - 同じことを何回も書かなくていい
  - プログラムが見やすくなる
  - 実行する回数を変更しやすい
- 繰り返しのポイントは
  - 何回繰り返すか
  - 繰り返し回数をどのように管理するか
    - ここでも変数を用いる

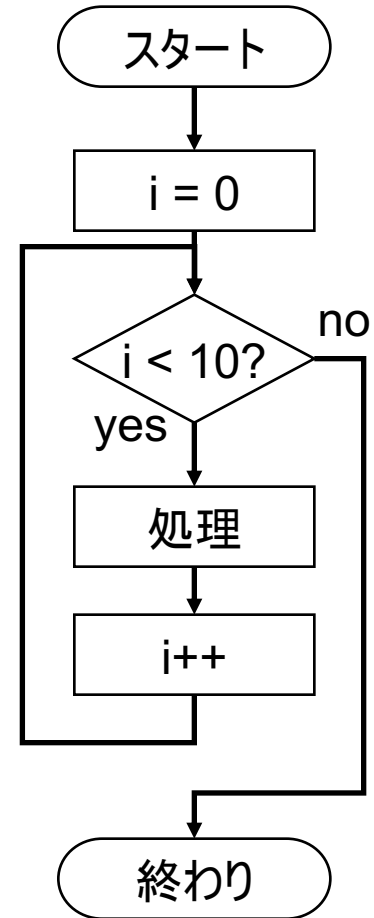
プログラミングが上手な人は、「うまく手間を省く」人です！

# 繰り返しの書き方

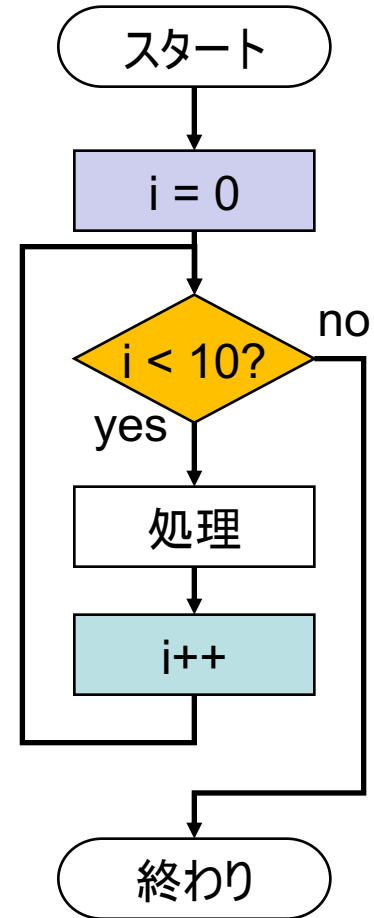
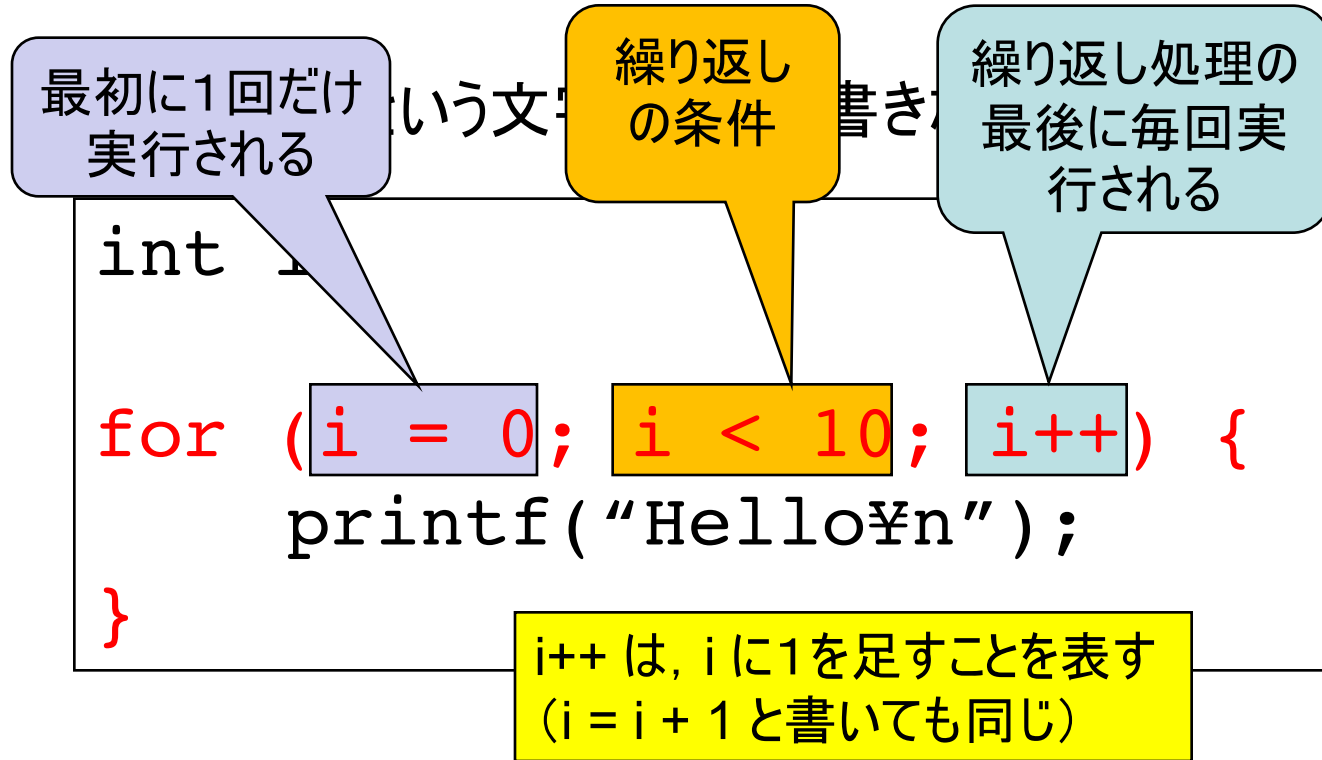
Hello という文字を10行書きたい

```
int i;  
  
for (i = 0; i < 10; i++) {  
    printf("Hello¥n");  
}
```

1. 最初  $i$  に 0 を入れる
2.  $i$  が 10 より小さい場合はブロックを実行  
そうでない場合は下へ抜ける
3. ブロックが終わると  $i$  に 1 を足す



# 繰り返しの書き方



1. 最初 i に 0 を入れる
2. i が 10 より小さい場合はブロックを実行  
そうでない場合は下へ抜ける
3. ブロックが終わると i に1を足す

# 繰り返しの書き方

0から9まで順に表示したい

```
int i;  
  
for (i = 0; i < 10; i++) {  
    printf("a:%d¥n", i);  
}  
printf("b:%d¥n", i);
```

iが10になると条件を満たさないで、ブロックに入るのは  
iは0から9まで。  
繰り返し回数は10回

実行結果

```
a:0  
a:1  
<中略>  
a:8  
a:9  
b:10
```

forを抜けてきたときには  
iは10になっている

- 繰り返し回数を管理する i は、処理で使っても良い
- i は 増える順番でなくてもよい
  - 例1: 1ずつ減っていく順に処理をする
  - 例2: 2倍ずつしていく

# 繰り返しの書き方

10から0まで順に表示したい

```
int i;  
  
for (i = 10; i >= 0; i--) {  
    printf("%d\n", i);  
}
```

2の累乗を表示したい

```
int i;  
  
for (i = 2; i < 10000; i *= 2) {  
    printf("%d\n", i);  
}
```

# 値を変える演算子

複合代入演算子	意味	他の書き方
<code>i++</code>	<code>i</code> に 1 を加える(インクリメント)	<code>i = i + 1</code>
<code>i--</code>	<code>i</code> を 1 減らす(デクリメント)	<code>i = i - 1</code>
<code>i += 2</code>	<code>i</code> を 2 増やす	<code>i = i + 2</code>
<code>i -= 2</code>	<code>i</code> を 2 減らす	<code>i = i - 2</code>
<code>i *= 2</code>	<code>i</code> を2倍にする	<code>i = i * 2</code>
<code>i /= 2</code>	<code>i</code> を $\frac{1}{2}$ にする	<code>i = i / 2</code>
<code>i %= 2</code>	<code>i</code> を2で割ったあまりを代入する	<code>i = i % 2</code>

- なくても良い演算子だが、ちょっと便利
  - for文の中では ; をつけない(最後なので)  
`for(i = 0; i < 10; i += 2) { ... }`
  - プログラムの実行文では ; をつける  
`i *= 2;`

偶数を順に



# 二重ループ

```
int i, j;

for (i = 0; i < 5; i++) {
    for (j = 0; j < 5; j++) {
        printf("i = %d, j = %d ¥n", i, j);
    }
}
```

- 繰り返しの繰り返し
  - 上の例では, printf は25回実行される.

# 練習問題に向けての参考情報

printf での、表示桁数指定

```
printf("%d\n", 1);  
printf("%d\n", 10);
```

結果

1
10

数値の桁数に応じて  
文字数が増減

```
printf("%2d\n", 1);  
printf("%2d\n", 10);
```

結果

1
10

最低でも2桁を  
使って表示

- 他に、以下の機能がある

- 空白で埋める代わりに 0 で埋める `%02d` → 01
- 左寄せで表示 `%-2d` → 1□ (□は空白を表す)
- 小数点以下の桁数を指定 `%8.3f` → □□□3.141
  - 8 が全体の文字数, 3 が小数点以下の桁数

# プログラム例(1)

```
#include <stdio.h>

int main(void) {
    int i, j;

    for(i = 0; i < 10; i++) {
        for(j = 0; j <= i; j++) {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

結果

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
```

# プログラム例(2)

```
#include <stdio.h>

int main(void) {
    int i, j;

    for(i = 9; i >= 0; i--) {
        for(j = 0; j <= i; j++) {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

結果

```
*****
*****
*****
*****
*****
*****
*****
*****
**
*
```