

5. while 文

- 条件が合っている間は、処理を繰り返す

```
while( 条件式 ){
    処理 1;
    処理 2;
}
```

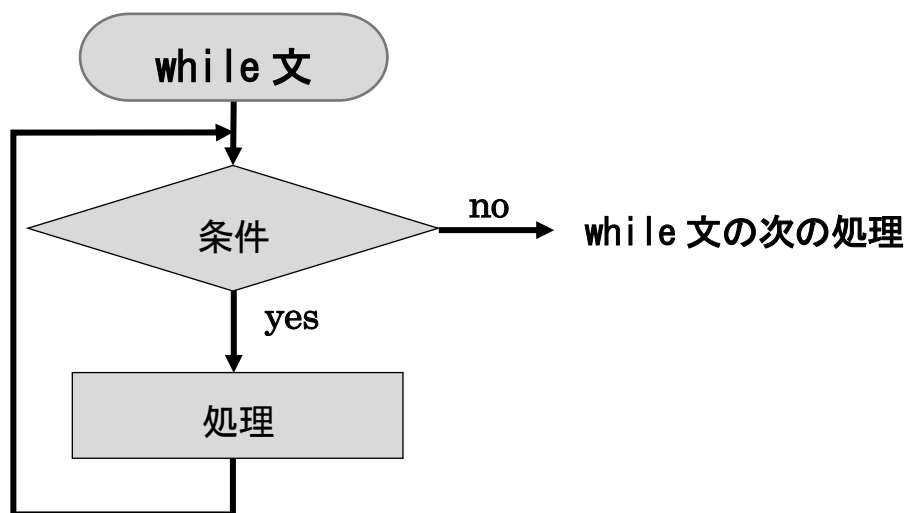
※処理が1つの場合、{ }は省略可

<図1 while 文の基本書式>

```
while( 条件式 );
```

※処理がない場合、セミコロン(;)を書く。
条件が合っている間、条件式を実行する。

<図2 while 文の書式 (条件式を繰り返す) >



<図3 while 文の処理の流れ>

```
while( i <= n ){
    sum = sum + i;
    i = i + 1;
}
```

Red callout boxes highlight '条件' (condition) pointing to `i <= n` and '処理' (processing) pointing to `sum = sum + i;` and `i = i + 1;`.

<図4 1~n の総和を求めるプログラムの一部>

- 無限ループ

```
while(1) 処理;
```

```
for(;;) 処理;
```

※処理を無限に繰り返して実行する

<図5 無限ループ>

■ 例題

```

#include<stdio.h>
int main(void){
    int n;           //1~nまでの総和を求める
    int i;           //項の数
    int sum;         //和の結果

    sum = 0;         //sumの初期化
    i = 1;           //iの初期化(1からの和なので)

    //nの入力
    printf("和を求めたい数を入力: ");
    scanf("%d", &n);

    //1~nまでの加算
    while(i<=n){
        sum = sum+i;           //加算
        printf("(%d) s= s + i = s + %d = %d¥n",i, i, sum); //途中結果表示
        i=i+1;                 //カウントアップ
    }
    printf("1 から n の総和=%d¥n", sum); //最終結果表示
}

```

<図6 while文を利用した1~nの総和を求めるプログラム>

- (1) 主に繰り返し回数が決まっていないときに用いる
- (2) while文の繰り返し処理を始める直前には、条件に使う変数などを初期化する必要がある

```

sum = 0;
i = 1;
printf("和を求めたい数を入力: ");
scanf("%d", &n);

while(i<=n){
    sum = sum+i;
    printf("(%d) s= s + i = s + %d = %d¥n",i, i, sum);
    i=i+1;
}

```

<図7 1~nの総和を求めるプログラムの一部>

(3) break文

- while{ }の外に抜け出して繰り返し処理を終了し、while{ }の次の処理に移る
- 繰り返し処理から抜け出るために、if文といっしょに使うことが多い

```

#include <stdio.h>
int main(void){

    int stop;
    int i=1;

    printf("stop loop number (1~10) ¥n");
    scanf("%d",&stop);

    while(i<=stop){
        print("%d 番目の処理¥n",i);
        if(i == stop)
            break;
        i=i+1;
    }
}

```

while文から抜け出して、
while文の次の処理へ移る

<図8 繰り返し処理回数表示プログラム(break)>

(4) continue 文

- While{ }の中で、次の繰り返し処理を飛ばし、その次の繰り返し処理に移る
- 飛ばす繰り返し処理を見つけるために、if 文と併用して使うことが多い

```
#include <stdio.h>
int main(void){

    int jump;
    int i=0;

    printf( "jump loop number (1~10) ♫n");
    scanf( "%d",&jump );

    while(i<10){
        i=i+1;
        if(i == jump)
            continue;
        print( "%d 番目の処理♫n",i);
    }
}
```

while 文から抜けず、while 文内の次の繰り返し処理へ移る

次の繰り返し処理に移る

<図 9 繰り返し処理回数表示プログラム (continue) >