

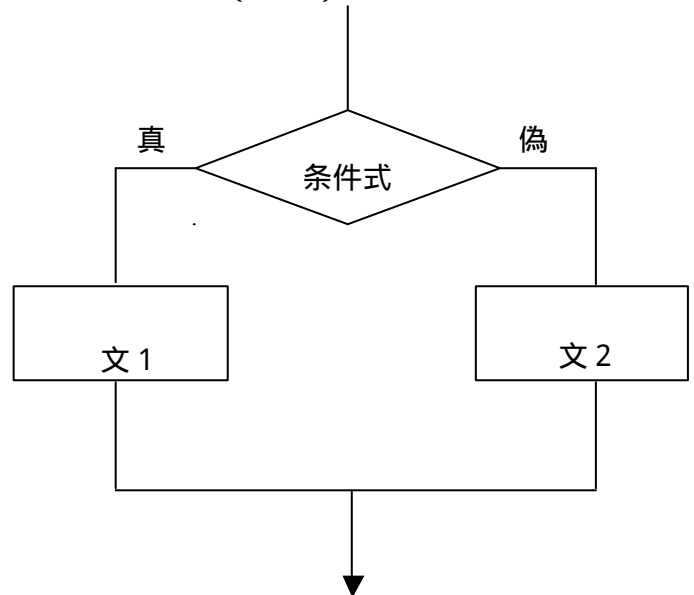
制御の流れ

1. 分岐

1.1 if 文による分岐

構文 1	if (条件式) 文 ;
構文 2	if (条件式) 文 1 ; else 文 2 ;
構文 3	if (条件式 1) 文 1 ; else if (条件式 2) 文 2 ; : : else if (条件式 n) 文 n ; else 文 ;

if 文の流れ図 (構文 2)

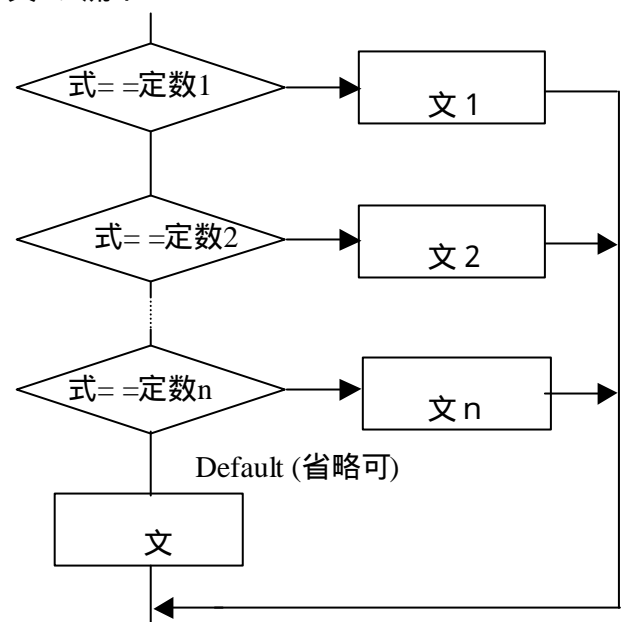


- ・複数の文を実行するときは、{ }でくくって複文にする。

1.2 switch-case 文による多方向分岐

switch (式){	
case 定数 1:	文 1 ; break ;
case 定数 2:	文 2 ; break ;
:	:
:	:
case 定数 n:	文 n ; break ;
default :	文 ;
}	

switch-case 文の流れ



- ・「式」の値と一致する定数の case 部にジャンプし、break 文に出会うか switch case 文の終端の } に出会うと処理を終了する。case 部にbreak文がなければ、以降のcase部の文を実行する。
- ・一致する定数がないときは、default 部を実行する。default 部は省略可。

2. 繰り返し

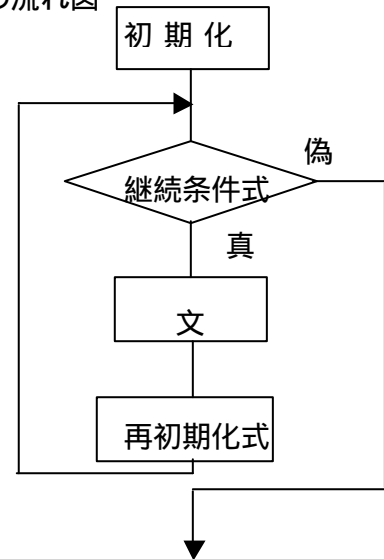
2.1 for 文

```
for(初期化式 ; 継続条件式 ; 再初期化式)
    文 ;
```

```
例 1 ) int i, sum=0; /* int 型変数 sum を 0 に初期設定 */
        for(i=1; i<=10; i++) /* i++ は i=i+1 と同じ */
            sum=sum+i;
```

```
例 2 ) int i, sum=0;
        for(i=1; i<=10; i++){ /* 複数の文は { } でくる */
            sum=sum+i;
            printf("i=%d, sum=%d\n", i, sum);
        }
```

for 文の流れ図



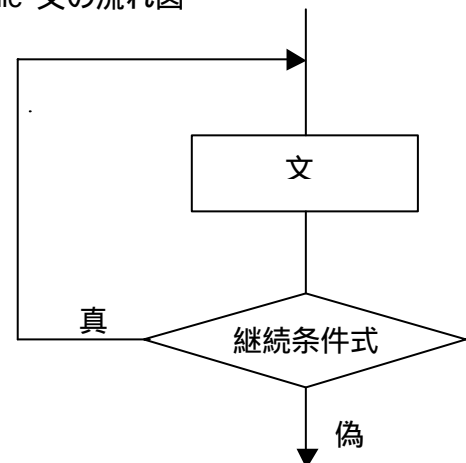
2.2 do-while 文

```
do
    文 ;
while(継続条件式);
```

```
例 1 ) int i=10, sum=0; /* int 型変数 i と sum の初期設定 */
        do
            sum=sum+i;
        while(--i); /* i に i-1 を代入してから、 */
                    /* i が真(0 以外)ならば繰り返す */
```

```
例 2 ) 上記 do while 文は下記と同じ
        do{ /* 複数の文は { } でくる */
            sum=sum+i; i=i-1; /* 1 行に複数の文を書ける */
        }while(i!=0); /* i が 0 以外ならば繰り返す */
```

do - while 文の流れ図



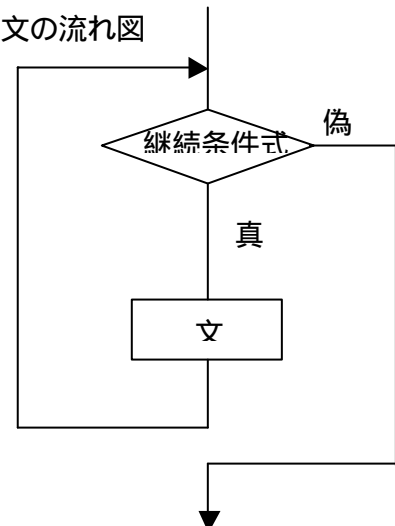
2.3 while 文

```
while(継続条件式)
    文 ;
```

```
例 1 ) int i=11, sum=0;
        while(--i) /* i に i-1 を代入してから */
            sum=sum+i; /* i が真(0 以外)ならば繰り返す */
```

```
例 2 ) int i=10, sum=0; /* i の初期設定値の変化に注意 */
        while(i>0){ /* i が正ならば繰り返す */
            sum=sum+i; i--; /* i-- は i=i-1 と同じ */
        } /* 複数の文は { } でくる */
```

while 文の流れ図



注：本頁中の例はすべて 1 から 10 までの和を計算するプログラムである。

3. 中断

3.1 break 文

- ・ switch 文の中での使用：switch 文を終了させる。
- ・ for 文，do while 文，while 文の中での使用：繰り返し処理を終了させる。

3.2 continue 文

- ・ for 文，do while 文，while 文の中での使用：その回の繰り返し処理をスキップさせる。

例 1) break 文と continue 文の機能の相違

```
#include <stdio.h>

void main(void)
{
    int i;

    for(i=1; i<=10; i++){
        printf("%n");           /* 改行する */
        printf("i=%d ", i);
        if(i==4) break;         /* i が 4 の時ループを抜けて次の文へジャンプする */
        printf("i*i=%d ", i*i);
        if(i==2) continue;     /* i が 2 の時のみループ内の以降の処理をスキップする */
        printf("i*i*i=%d ", i*i*i);
    }
    printf("%n break と continue の違いは解かりましたか？%n");
}

```

-----実行開始-----

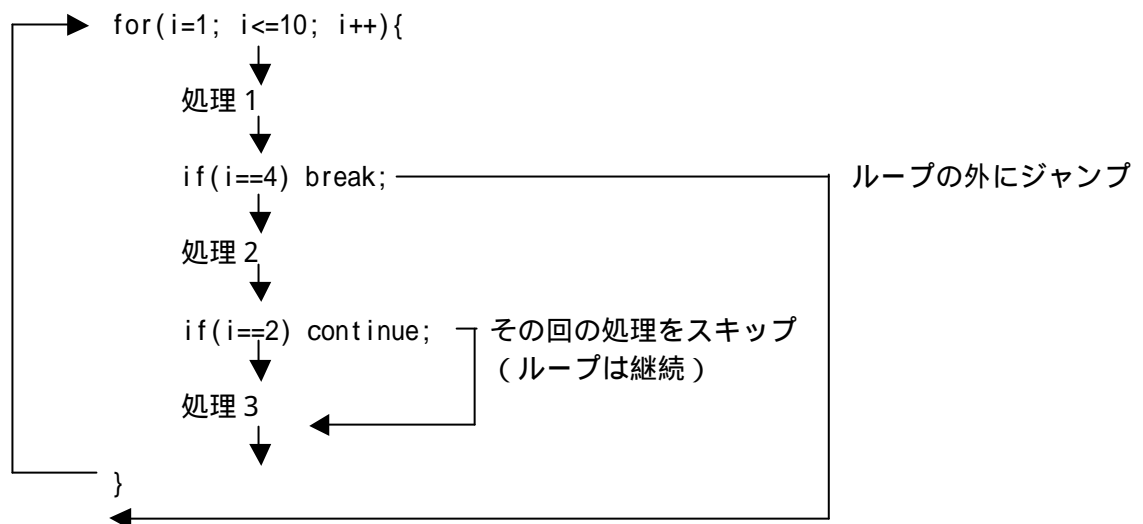
```
i=1 i*i=1 i*i*i=1
i=2 i*i=4
i=3 i*i=9 i*i*i=27
i=4

```

break と continue の違いは解かりましたか？

-----おしまい-----

流れ図



例 2) 無限ループを break 文により終了する

```
#include <stdio.h>

void main(void)
{
    int n;
    while(1){
        printf("input n ==>"); scanf("%d",&n);
        if(n==0) break;
        printf("10 進数 : %d    8 進数 : %o    16 進数 : %x ¥n",n,n,n);
    }
}
```

```
-----実行開始-----
input n ==>7
10 進数 : 7    8 進数 : 7    16 進数 : 7
input n ==>8
10 進数 : 8    8 進数 : 10    16 進数 : 8
input n ==>13
10 進数 : 13   8 進数 : 15    16 進数 : d
input n ==>16
10 進数 : 16   8 進数 : 20    16 進数 : 10
input n ==>177
10 進数 : 177  8 進数 : 261    16 進数 : b1
input n ==>0
-----おしまい-----
```

制御の流れ（問題）

1. 「year=」と表示し、変数 n に西暦の年を入力し、その年がうるう年かどうかを表示する。これを 0 または負の数字が年として入力されるまで繰り返すプログラムを作成せよ。

うるう年とは以下のいずれかにあてはまる年のことである。

- ・ 4 で割り切れ、かつ 100 で割り切れない年
- ・ 400 で割り切れる年

実行例

```
year=2000
2000 年はうるう年です。
year=1999
1999 年はうるう年ではありません。
year=5000
5000 年はうるう年ではありません。
year=0
```

2. 西暦 1800 年から 2000 年までの間のうるう年を全部表示（10 個毎に改行すること）してから、うるう年がその間に何回あるか表示するプログラムを作成せよ。

実行例

```
西暦1800年から2000年までの間のうるう年：
1804 1808 1812 1816 1820 1824 1828 1832 1836 1840
1844 1848 1852 1856 1860 1864 1868 1872 1876 1880
1884 1888 1892 1896 1904 1908 1912 1916 1920 1924
1928 1932 1936 1940 1944 1948 1952 1956 1960 1964
1968 1972 1976 1980 1984 1988 1992 1996 2000
西暦1800年から2000年までの間にうるう年は49回あります。
```