

C++程式設計

單元三：迴圈

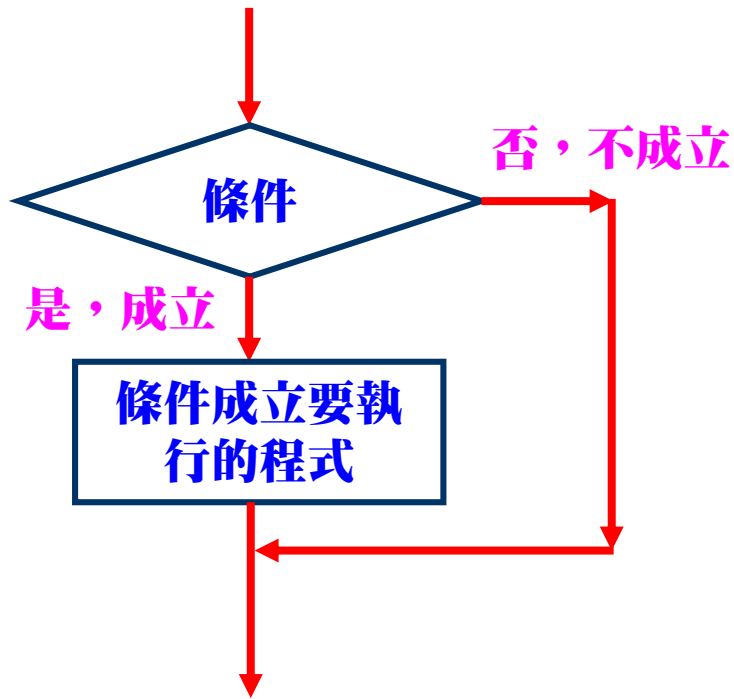
迴圈

所謂迴圈就是讓某一段程式重覆執行，直到條件被滿足為止。

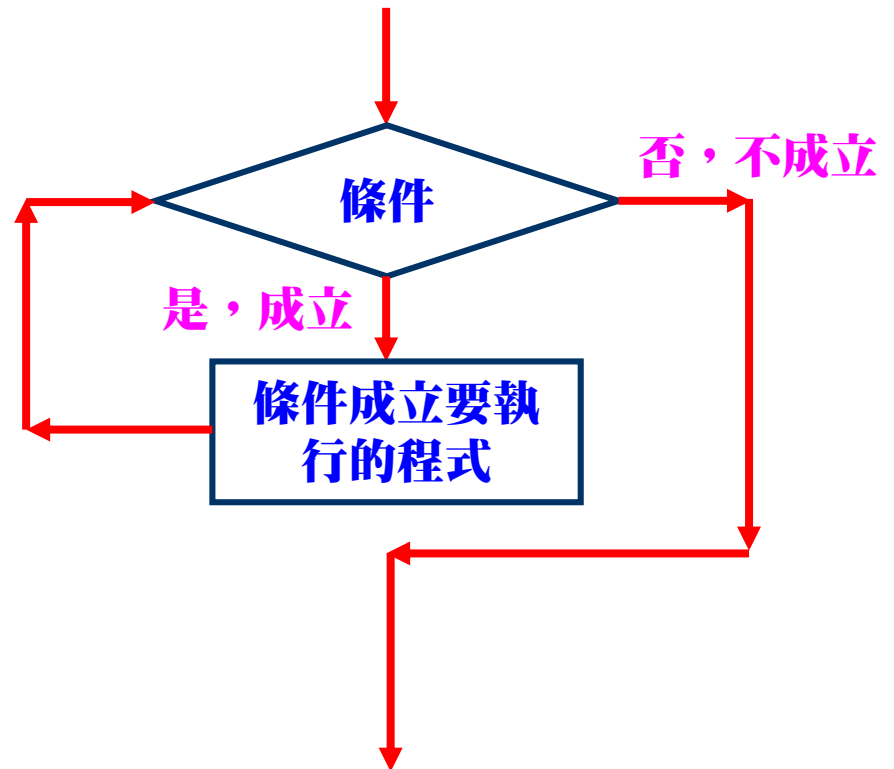
在某些特定的程式中，迴圈結構可以令程式變得更加簡潔。

判斷結構與迴圈結構

選擇結構



迴圈結構



迴圈結構

C++中可使用的迴圈指令如下：

- for 迴圈
- while 迴圈
- do...while 迴圈

迴圈其他配合指令如下：

- break
- continue

for迴圈語法

- 語法格式(一)

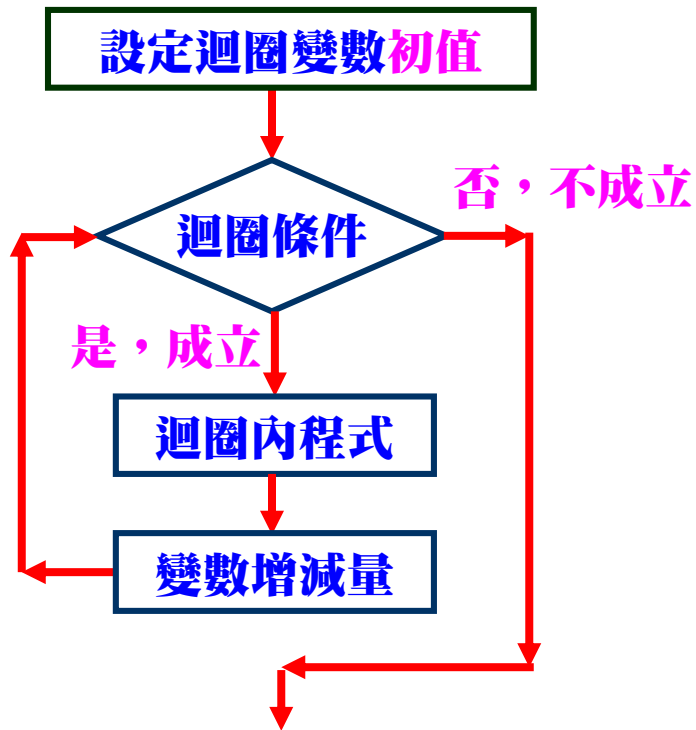
```
for(初值;進入迴圈條件;增減量) 迴圈內程式;
```

- 語法格式(二)

```
for(初值;進入迴圈條件;增減量) {  
    迴圈內程式;  
}
```

for迴圈流程圖

● 流程圖一

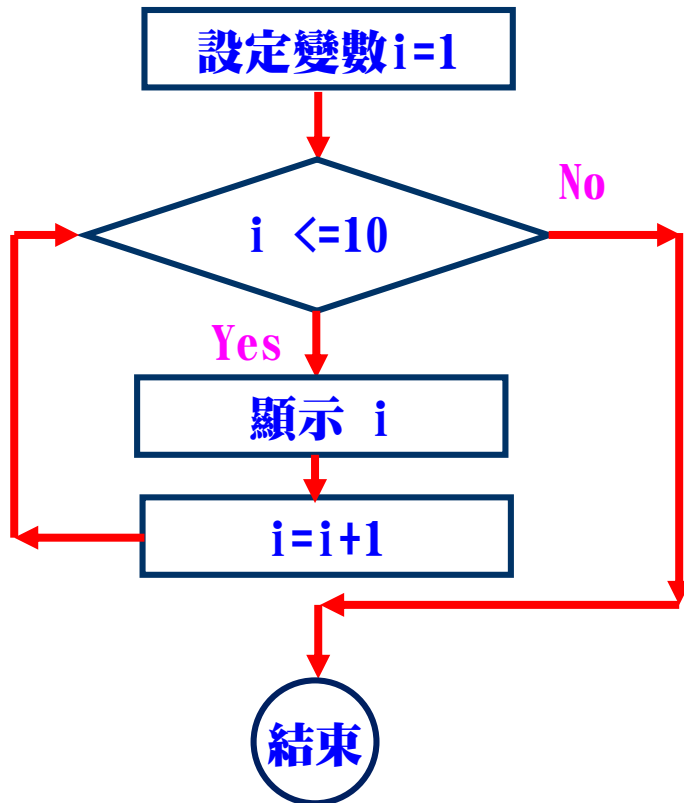


● C++程式一

```
for(變數初值;迴圈條件;變數調整)  
{  
    條件成立要執行的程式  
}
```

for 迴圈範例一

利用for迴圈印出 1 到 10 的數字



```
main(){
    int i;

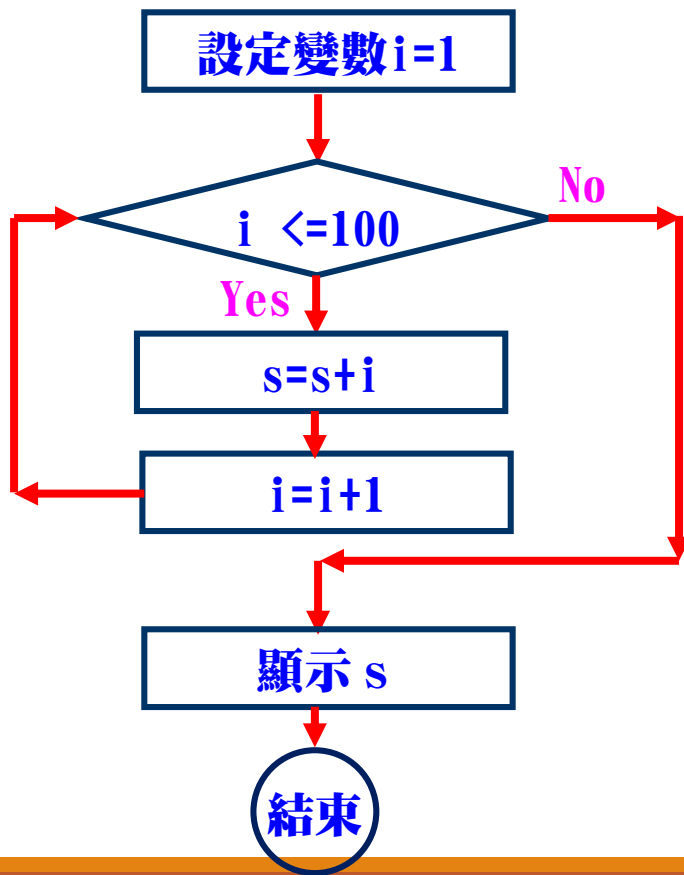
    for(i=1;i<=10;i++) {

        cout<<i<<"\t";

    }
    return 0;
}
```

for 迴圈範例二

利用for迴圈執行 1 加到 100 的整數和



```
main(){
    int i,s=0;

    for(i=1;i<=100;i++) {

        s+=i;
    }
    cout<<"1到100的整數和為："<<s;
    return 0;
}
```


for 迴圈範例三

利用for判斷所輸入的整數是否為質數（除了1及自己外，不能其它數整除）。

```
main(){
    int i,n=0;
    bool chk=true;
    cout<<"請輸入一個整數:";cin>>n;
    for(i=2;i<n;i++) {
        if(n%i==0) chk=false;
    }
    if(chk==true)
        cout<<n<<"是質數\n : ";
    else
        cout<<n<<"不是質數\n : ";
    return 0;
}
```

while迴圈語法

- 語法格式(一)

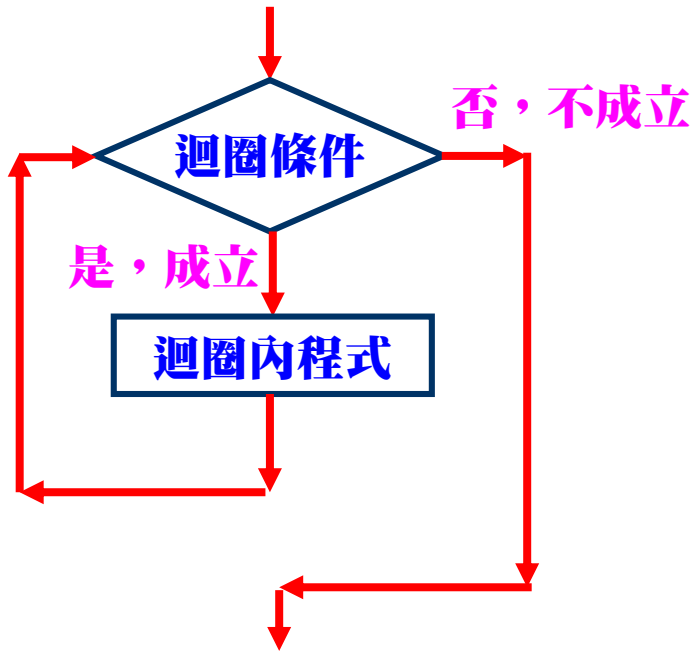
```
while(進入迴圈條件) 迴圈內程式;
```

- 語法格式(二)

```
while(進入迴圈條件 {  
    迴圈內程式;  
}
```

while迴圈流程圖

● 流程圖一

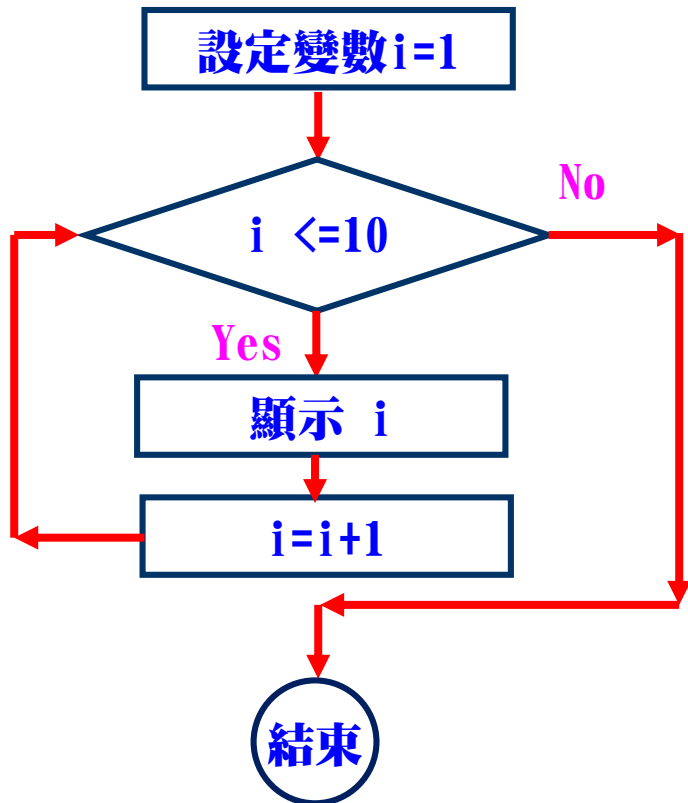


● C++程式一

```
while( 迴圈條件 )  
{  
    條件成立要執行的程式  
}
```

while 迴圈範例一

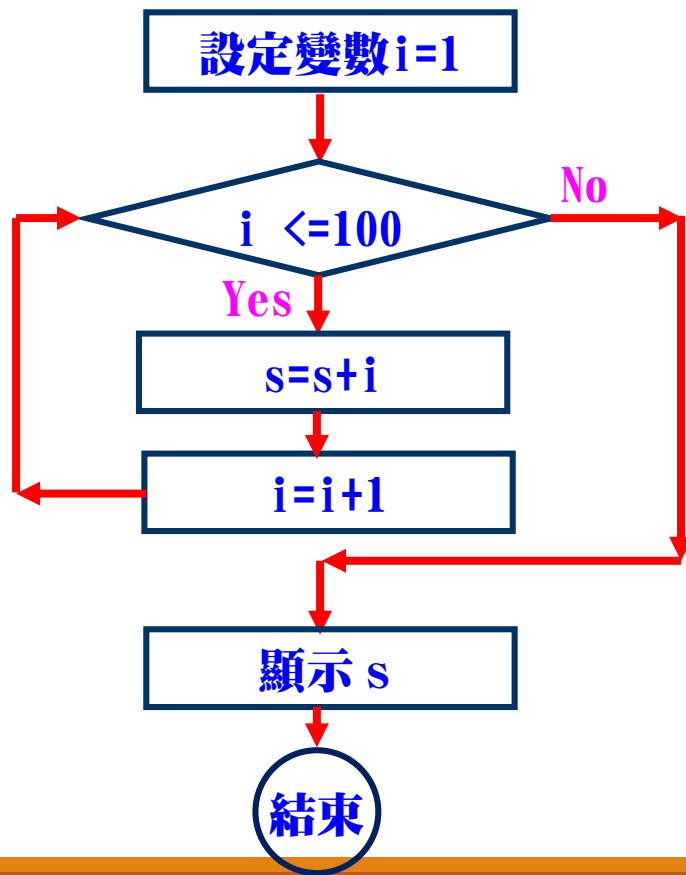
利用while迴圈印出 1 到 10 的數字



```
main(){  
    int i;  
    i=1;  
    while(i<=10) {  
  
        cout<<i<<"\t";  
        i++;  
  
    }  
    return 0;  
}
```

while 迴圈範例二

利用 while 迴圈執行 1 乘到 10 的積



```
main(){
    int i,s;
    i=1;s=1
    while(i<=10) {
        s*=i;
        i++;
    }
    cout<<"1 乘到 10 的積為："<<s;
    return 0;
}
```

while 迴圈範例三

利用 while 迴圈計算5的倍數中不小於輸入值整數的最小值

```
main(){
    int i,n;
    cout<<"請輸入任一整數：";cin>>n;
    i=0;
    while(i<n) {

        i+=5;

    }
    cout<<"不小於 "<<n<<" 的5的倍數最小為： "<<i;
    return 0;
}
```

while 迴圈範例四

利用 while 迴圈求輸入兩數的最大公因數 (gcd)

已知輾轉相除法求最大公因數的方法為：

$$\text{gcd}(a, b) = \begin{cases} b, & \text{if } a \bmod b = 0 \\ \text{gcd}(b, a \bmod b), & \text{other} \end{cases}$$

```
main(){
    int a,b,k;
    cout<< "請輸入兩個整數：";cin>>a>>b;
    while(a mod b !=0) {
        k=a mod b;
        a=b;
        b=k;
    }
    cout<<"兩數的最大公因數為 "<<b<<endl;
    return 0;
}
```

do while迴圈語法

- 語法格式

```
do {
```

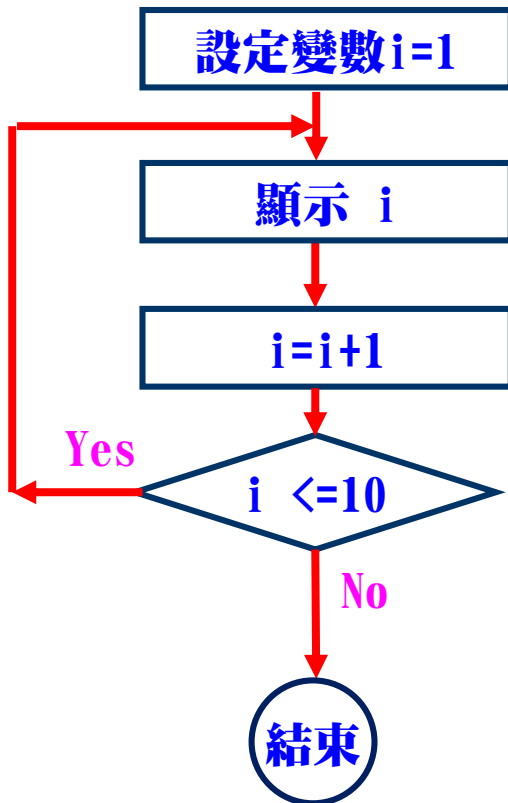
```
    迴圈內程式;
```

```
} while(繼續迴圈的條件);
```

此結構為後測式迴圈，因此迴圈內程式至少會執行一次

do while 迴圈範例一

利用while迴圈印出 1 到 10 的數字



```
main(){
    int i;
    i=1;
    do
    {

        cout<<i<<"\t";
        i++;

    } while(i<=10) ;
    return 0;
}
```

do while 迴圈範例二

利用do while迴圈寫一密碼判斷程式(假設密碼為1234)

```
main(){
    int pw;
    do
    {

        cout<< "請輸入密碼：";cin>>pw;

    } while(pw!=1234) ;

    cout<< "您輸入的密碼正確";

    return 0;
}
```

亂數函數

- 需引用之標頭檔

```
#include<cstdlib>
#include<ctime>
```

- 亂數種子—使每次所產生之亂數序皆不同

```
srand(time(NULL));
```

- 亂數函數—範圍為 0 ~ 3 2 7 6 7 間的整數亂數

```
rand();
```

- 指定產生介於 min~max 間的整數亂數

```
變數 = rand()%(max-min+1)+min;
```

do while 迴圈範例三

猜數字(所猜數字改由系統亂數產生)

```
#include<iostream>
#include<cstdlib>
#include<ctime>
using namespace std;
main(){
    int ans,pw;
    srand(time(NULL));
    ans=rand() % 100 +1;
    do {
        cout<<"輸入1~100間整數";
        cin>>pw;
    } while(pw!=ans) ;
    cout<< "猜對了!! ";
    return 0;
}
```

do while 迴圈範例三改良

猜數字(所猜數字由系統亂數產生)

```
#include<iostream>
#include<cstdlib>
#include<ctime>
using namespace std;

main(){
    int ans,pw;
    srand(time(NULL));
    ans=rand() % 100 +1;

    do {

        cout<<"輸入1~100間整數";
        cin>>pw;
```

```
        if (pw>100 || pw<0)
            cout <<"輸入有錯";
        else
            if(pw>ans)
                cout<<"太大\n";
            else if (pw<ans)
                cout<<"太小\n";
    } while(pw!=ans) ;

    cout<< "猜對了!! ";
    return 0;
}
```

C++的無窮迴圈寫法

- 以 for 迴圈

```
for(;;) {  
}
```

- 以 while 迴圈

```
while(1) {  
}
```

- 以 do ... while 迴圈

```
do {  
} while(1);
```

C++的break與continue

break

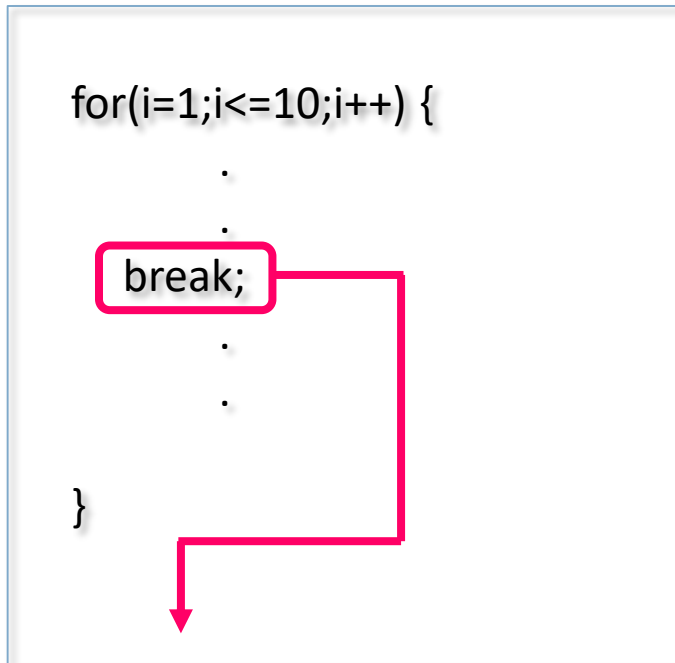
break不只限用於switch指令，亦可用於任何一個迴圈結構中，可強制跳離迴圈。

continue

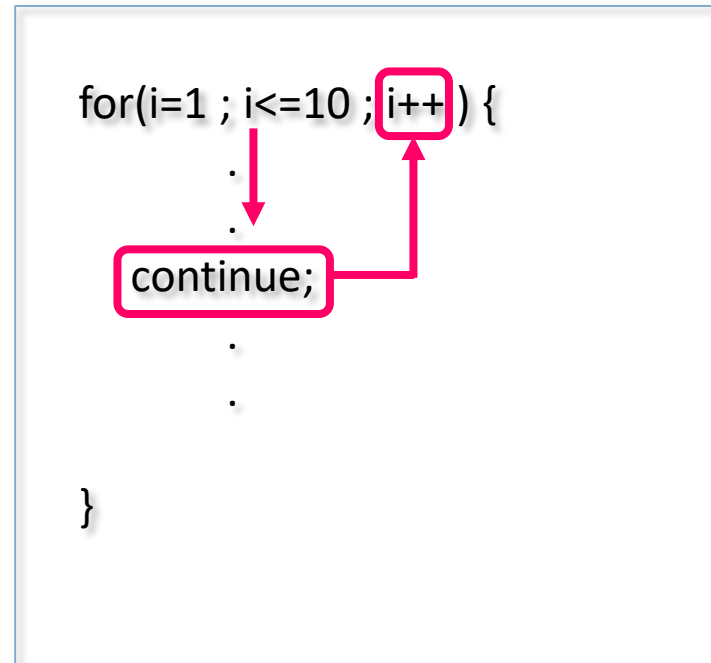
和**break**不同的是**continue**是將該次迴圈程序結束，然後返回迴圈開始的地方。

break與continue比較

break



continue



因為此兩個指令會強制改變迴圈流程，故一般會跟判斷指令配合使用!!

break應用

密碼判斷程式，最多只有三次機會

```
main(){
    int ans=123,pw,i;
    for(i=1;i<=3;i++) {
        cout<<"第"<<i<<"次輸入密碼:"<<cin>>pw;
        if(pw==ans)
            break;
        else
            if(i !=3) cout <<"輸入有錯，請重新輸入!!\n";
    }
    if(i>3)
        cout <<"你已連續輸入3次錯誤，無法登入!!\n";
    else
        cout<<"歡迎登入系統!!\n";
    return 0;
}
```

continue應用

計算1~100間某數的倍數之和

```
main(){
    int i,num,sum=0;
    cout<<"要計算什麼數的倍數和:";cin>>num;
    cout<<"1~100中 "<< num << "的倍數有:";

    for(i=1;i<=100;i++) {
        if(i%num!=0)
            continue;
        cout<<i<<"\t";
        sum+=i;
    }
    cout<<"\n"<< num <<"的倍數和為：" <<sum<<endl;
    return 0;
}
```

設定輸出數值欄寬(位數)

- 引用標檔

```
#include <iomanip>
```

- 語法

```
cout<<setw(位數) << 數值 ;
```

- 範例

```
cout<<setw(4) << 12 ; =====> _ _ 1 2
```