



10. 複習

2024資訊研究社語法班
Made with ❤️ by jheanlee



輸入輸出

使用 cin 取得輸入的數值

cin >> 變數;

cin 用空格或換行切開資料

使用 cout 印出訊息

cout << 資料;

```
/*-----變數宣告-----*/
```

```
int a, b, c;
```

```
/*-----輸入-----*/
```

```
cin >> a >> b >> c; // cin會依序把資料存入 a, b, c 中
```

```
/*-----輸出-----*/
```

```
cout << "You have entered the following values: " << endl;
```

```
cout << "a: " << a << endl;
```

```
cout << "b: " << b << endl;
```

```
cout << "c: " << c << endl;
```



變數 運算

常用資料的類別有以下幾種

bool - true 或 false

int - 整數 $2^{31}-1 \sim -2^{31}$

long long - 整數 $2^{63}-1 \sim -2^{63}$

float - 浮點數 (小數或極大的數) 7位有效位數

double - 浮點數 (小數或極大的數) 15位有效位數

char - 字元 (1個字母)

string - 字串 (很多字母)



變數 運算

用 = 改變變數的值

+ - * / 就是加減乘除
% 取餘數

a += b; (-= *= /= %= 同)
與 a = a + b; 一樣

a++; a--; 和 a += 1; a -= 1; 相同

```
int d = 0;  
d = d + 1;  
d += 1;
```



char string

char 儲存字元

要使用單引號'

string 儲存字串

要使用雙引號"

使用 變數.size() 取得字串長度

使用 變數[n - 1] 取得第n個字母

使用 + 號連接字串

```
string e = "abc";
cout << "e: " << e << endl; // e: abc
cout << "e.size(): " << e.size() << endl; // e.size(): 3
cout << "e[1]: " << e[1] << endl; // e[1]: b

e[1] = 'd';
e += "def";
cout << "e: " << e << endl; // e: adcdef
cout << "e.size(): " << e.size() << endl; // e.size(): 6
cout << "e[1]: " << e[1] << endl; // e[1]: d
```



邏輯

很直覺的大小比較

$a > b$ 大於

$a \geq b$ 大於等於

$a == b$ 等於

$a \leq b$ 小於等於

$a < b$ 小於

$a != b$ 不等於

邏輯運算子

$c \ \&\& \ d$ and

$c \ || \ d$ or

$!(c \ \&\& \ d)$ not

$\text{true} \ \&\& \ \text{true} \rightarrow \text{true}$

$\text{true} \ \&\& \ \text{false} \rightarrow \text{false}$

$\text{false} \ \&\& \ \text{false} \rightarrow \text{false}$

$\text{true} \ || \ \text{true} \rightarrow \text{true}$

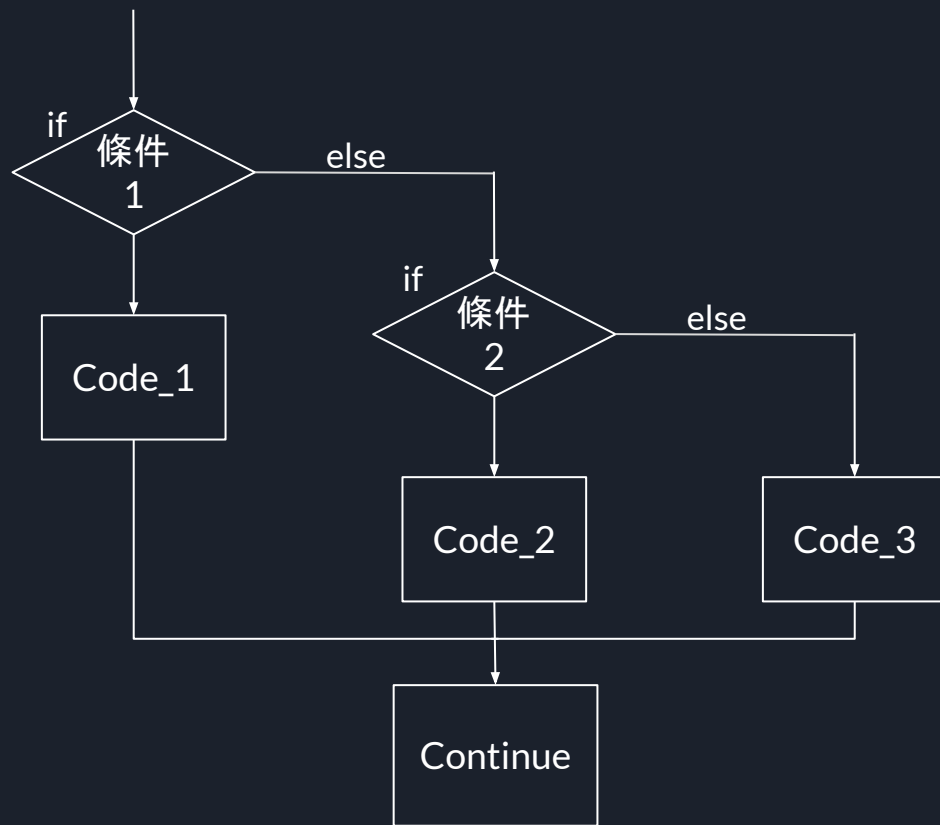
$\text{true} \ || \ \text{false} \rightarrow \text{true}$

$\text{false} \ || \ \text{false} \rightarrow \text{false}$

$!\text{true} \rightarrow \text{false}$

$!\text{false} \rightarrow \text{true}$

if else 的流程



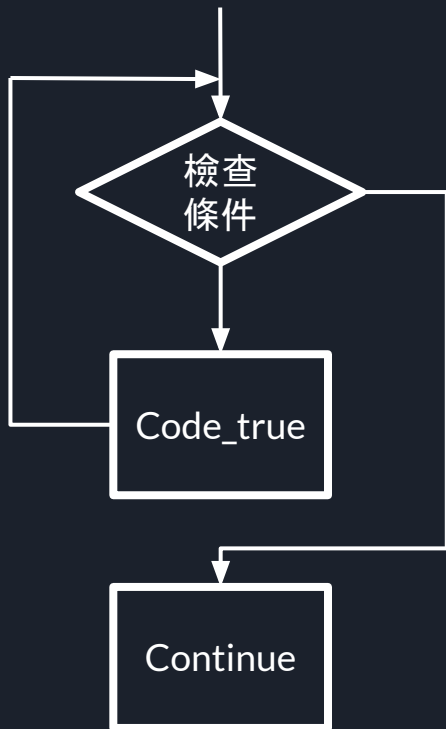


if, else if, else

```
if (條件1) {  
    //執行一些東西  
} else if (條件2) {  
    //執行一些另外的東西  
} else if (條件n) {  
    //執行另外一些東西  
} else {  
    //都沒有的話 就執行一些其他的東西  
}
```

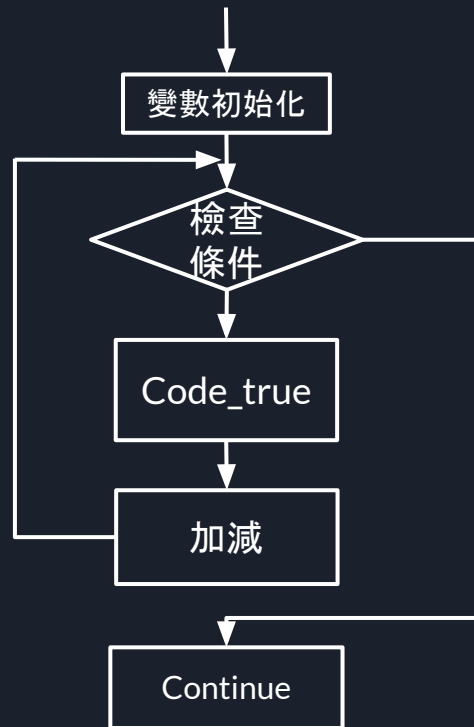
```
if (a < 0) {  
    cout << "a is negative" << endl;  
} else if (a == 0) {  
    cout << "a is zero" << endl;  
} else {  
    cout << "a is positive" << endl;  
}
```


while for



while 和 for 都是在設定條件下不斷重複執行

for 比 while 多了臨時變數和每次加減的功能





while for

```
while (執行條件) {  
    //只要條件是true 就會一直執行  
}
```

```
for (初始變數; 執行條件; 加減) {  
    //只要條件是true 就會一直執行  
}
```

```
{  
    int i = 0;  
    while (i < 3) {  
        cout << "i: " << i << endl;  
        /*  
        * i: 0  
        * i: 1  
        * i: 2  
        */  
        i++;  
    }  
}  
  
for (int i = 0; i < 3; i++) {  
    cout << "i: " << i << endl;  
    /*  
    * i: 0  
    * i: 1  
    * i: 2  
    */  
}
```

一維 array

0號位置	1號位置	2號位置	3號位置	4號位置	...
------	------	------	------	------	-----

array (陣列) 像是一個架子可以放很多變數也可以是空的

```
int arr[100];
```

內容物類型 變數名稱[最大數量];

```
int arr[100] = {初始內容物};
```

用 arr[n-1] 取得第n個資料

記得array 是從零開始算

~~記不起來的看這個影片~~

(有時候從零開始 有時候從一開始 從零開始的其中一個就是C++)

```
int arr[3] {0, 1, 2};
for (int i = 0; i < 3; i++) {
    cout << "arr[i]: " << arr[i] << endl;
    /*
     * arr[i]: 0
     * arr[i]: 1
     * arr[i]: 2
     */
}
arr[1] = -1;
for (int i = 0; i < 3; i++) {
    cout << "arr[i]: " << arr[i] << endl;
    /*
     * arr[i]: 0
     * arr[i]: -1
     * arr[i]: 2
     */
}
```



巢狀構造

這個看不懂

先把縮排排好

用數學上括號的方式理解

100(100(100(100 * 程式)))

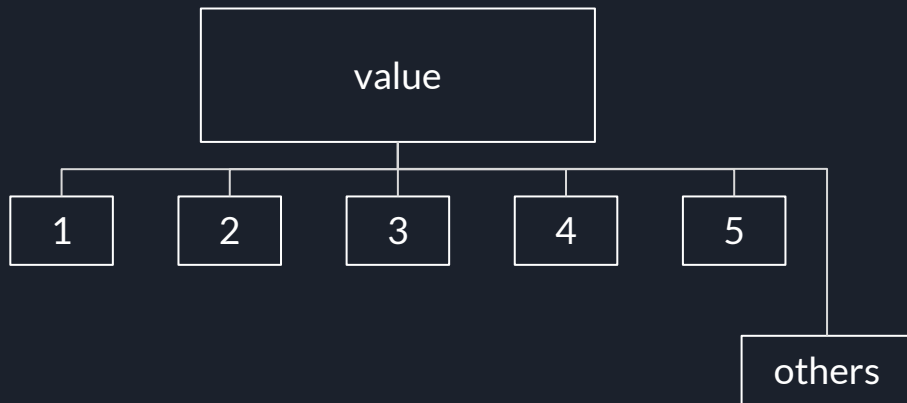
```
for (int i = 0; i < 100; i++) {  
    for (int j = 0; j < 100; j++) {  
        for (int k = 0; k < 100; k++) {  
            for (int l = 0; l < 100; l++) {  
                cout << i << ' ' << j << ' ' << k << ' ' << l << '\n';  
            }  
        }  
    }  
}
```

```
for (int i = 0; i < 2; i++) {  
    for (int j = 0; j < 3; j++) {  
        cout << "i: " << i << ", j: " << j << endl;  
        /*  
        * i: 0, j: 1  
        * i: 0, j: 2  
        * i: 1, j: 0  
        * i: 1, j: 1  
        * i: 1, j: 2  
        */  
    }  
}
```

switch

你今天有一個值，他可能是1到5或其他的數字，且
依據他的數值要有不同的動作

為了達成這個目的，我們要有系統性地去分類



switch的架構

前提:

- switch必須有2個值以上(包含其他類)
- switch各類別的資料類型(type)必須相同(或可轉換成相同)
- Switch不可以多次出現同樣的值

```
if (value == 1) {  
    cout << "one" << endl;  
} else if (value == 2) {  
    cout << "two" << endl;  
} else if (value == 3) {  
    cout << "three" << endl;  
} else if (value == 4) {  
    cout << "four" << endl;  
} else if (value == 5) {  
    cout << "five" << endl;  
} else {  
    cout << "none of above" << endl;  
}
```

```
switch (value) {  
    case 1:  
        cout << "one" << endl;  
        break;  
    case 2:  
        cout << "two" << endl;  
        break;  
    case 3:  
        cout << "three" << endl;  
        break;  
    case 4:  
        cout << "four" << endl;  
        break;  
    case 5:  
        cout << "five" << endl;  
        break;  
    default:  
        cout << "none of above" << endl;  
}
```