

# C++基礎語法-2

---

還是鄭詠堯喔

# 今天的課程

---

迴圈、陣列，還有比上禮拜難滿多ㄉ練習題

你各位加油w

之後不是我教會更難喔喔喔w

# 迴圈

---

顧名思義就是會重複運作的概念，當某些特定條件成立時，便重複執行某段程式碼。

# while迴圈

---

基本上可以看做是重複運作的if，也就是當if的執行內容執行完畢後，再一次檢查條件式，如果仍然符合條件，便再執行一次執行 { } 內的程式碼。

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

int main(){
    int x; //宣告變數x

    cin >> x; //輸入x

    while (x < 10){ //當x<10時，執行{}內的程式
        cin >> x; //輸入x
    }

    return 0;
}
```

# for迴圈

---

跟while迴圈也很相似，但不同的地方在於多了起始值與變動的概念。

也就是語法變為了 `for` (宣告起始值; 判斷式; 變動) { 執行內容 }，而順序則是判斷式→執行內容→變動→判斷式的迴圈。

# for 實作

---

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

int main(){
    int x; //宣告變數x

    cin >> x; //輸入x
    for (int i=0;i<x;++i){//當i小於x時執行{}內的程式，而且每次執行完後執行++i;
        cout << i << ' ';
    }
    cout << endl;

    return 0;
}
```

另外要記得 `i` 是迴圈內的變數，因此在迴圈外就無法對 `i` 進行操作。

# break; continue;

---

除了因為條件式未通過跳離迴圈外，有時候迴圈內執行到一半觸發特定條件也需要跳離迴圈或略過某段code，因此便有`break;`跟`continue;`兩個函式可以使用。

`continue;` 也就是略過迴圈`{ }`內剩餘的程式碼，重新檢查條件式並執行

`break;` 不同於`continue`，使用`break`後就會直接跳離迴圈了。

而通常這兩個函式都會搭配`if else`使用，也就是觸發某些條件時才執行

# 遇到輸入至EOF的題目？

例如在zerojudge上禮拜有人先寫到的a004

就出現了輸入至EOF結束的測資輸入

輸入說明

輸入有若干行直到 EOF 結束，每行包含一個整數代表年份

而此時就可以用`while (cin >> n)`的語法去解決，雖然`cin >> n`看起來不像是個判斷式，但其實只要這個式子有回傳值就能作為判斷式使用，而`cin >> n`在有輸入的時候，便會有回傳非0的值，輸入結束 (EOF) 時便會回傳`false`。



# 陣列

---

陣列指的就是一塊連續的儲存空間，並且透過索引值去存取後面的資料。

另外陣列的大小必須要在宣告時就決定好，而且不能是用一個變數去宣告陣列大小，一定要用一個普通的數去宣告，例如一個大小為1000的整數陣列宣告時會像是 `int x[1000];`

而存取第0項時便是用 `x[0]`，第1項用 `x[1]`，以此類推，而大小為1000的陣列就只能存取到 `x[999]`。

# 陣列實作

---

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

int main(){
    int x[10]; //宣告大小為10的整數陣列

    for (int i=0;i<10;++i){
        x[i] = i;
    }

    for (int i=0;i<10;++i){
        cout << x[i] << '\n';
    }
}
```

# 二維陣列

---

前面教到的僅是一維的陣列，也就是由一個索引值去對應資料，但有些時候會用到不只一個索引值ex:火車票價表，所以便可以用2個索引值去形成一張表，如 `int x[100][100]` ; 而操作也是跟一維陣列一樣的方式。

# 練習

---

ckeisc oj 011

zj a015

//上完課發現這兩題太難ㄌ，你們先寫寫看下一頁的三角形就好，寫完再來試試其他題目

# 進階一點的練習

寫一個可以像右邊一樣輸入多少就輸出多大的三角形。

小提示:for迴圈裡面可以再包更多for迴圈喔

```
7
      0
      0 1
    0 1 2
  0 1 2 3
0 1 2 3 4
0 1 2 3 4 5
0 1 2 3 4 5 6

3
  0
 0 1
0 1 2

10
      0
      0 1
    0 1 2
  0 1 2 3
0 1 2 3 4
0 1 2 3 4 5
0 1 2 3 4 5 6
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7 8
0 1 2 3 4 5 6 7 8 9
```

# 再更難的練習題

---

寫出一個能夠將一個陣列由小到大排序的程式。(有涉及到演算法的範疇ㄌ，寫不出來沒關係，但只要用到你們目前學過的語法就能寫了)

(用bubble sort)

//直接放棄這邊吧，之後演算法小社再說ㄌ(如果你把前面題目都寫完ㄌ當然可以試試看啦)

# 教不完

---

以下東西是今天教不完的部分，可以自己讀讀看，看不懂也沒關係，下星期二還是會講ㄉ。

# char(字元)

---

顧名思義，也就是單一一個字出現，像是前面出現過的' '、'\n'，就分別是空白字元及換行字元，而你也可以用一個char型態的變數去儲存它，像是char nextLine = '\n';



# string(字串)

---

也就是由多個字元串在一起的東西，又稱為字元陣列，但在C++裡可以直接宣告string型態，而因為字串具有陣列的特性，因此可以直接用`str[i]`去存取第*i*個字元喔。

# 基本的字串操作

---

不同於前面介紹過的函式，字串有一些物件導向的函式可以使用，使用方法是 字串名稱.函式名稱()；，下面是一些常見運算子及函式。

`+`：如果對字串使用`+`的話，會把兩個字串接起來

`[]`：取字串的第`[]`內數字的項

`.size()` / `.length()`：功能一樣，都會回傳字串的長度

`.empty()`：如果字串是空的，則回傳`true`，字串內部有東西則回傳`false`

# 示範code喔喔

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

int main(){
    bool isEmpty;           //宣告布林值變數isEmpty
    int x;                   //宣告整數變數x
    string str1,str2,str3,str4; //宣告字串變數str1~4
    str2 = "xue di";         //將str2指定為學弟
    str3 = " lai cho yo";    //將str3指定為來秋遊
    str4 = "";               //將str4指定為空陣列

    str1 = str2 + str3;      //將str1指定為str2+str3
    x = str1.length();       //將x指定為str1的長度
    isEmpty = str4.empty();  //將isEmpty指定為str4是否為空

    cout << str1 << ' ' << x << ' ' << isEmpty << ' ' << str3[1];

    return 0;
}
```

# ASCII Code

---

自己去讀維基(X

好啦沒，簡單來說就是每個字元都有他的代碼，而多數英文字母、數字、符號都收錄於ASCII Code之中(中文沒有喔)，例如' a ' 就等於97這樣。

# 練習題

---

Zj a020

# 補充內容

---

上的完才講這邊，一些不一定要用的語法

上不完就自己讀喔

//很顯然的是上不完，自己讀讀看ㄟ

# do while

---

除了 `while() {}` 的語法外，也可以寫成 `do{}while()`，他跟普通的 `while` 不同的地方在於，他會先必定執行一次程式，然後才去檢查條件式是否進行下一次。

# 取值

---

如同在前面提到過的，只要有給予一個值就能作為判斷式，而其實多數的式子都具有值，例如`++i`，給予的值便是加1過後的`i`；而`x = y + z`給予的值便是賦予`x`的值喔。試試看下面哪些是合法的語法？`x`, `y`, `z`的值又分別是多少？

`x = y = z + 10;`                      `y`

`x = z * (y = z++);`                      `y`

`x = y + 10 = z;`                      `x`