

C++

輸出函數

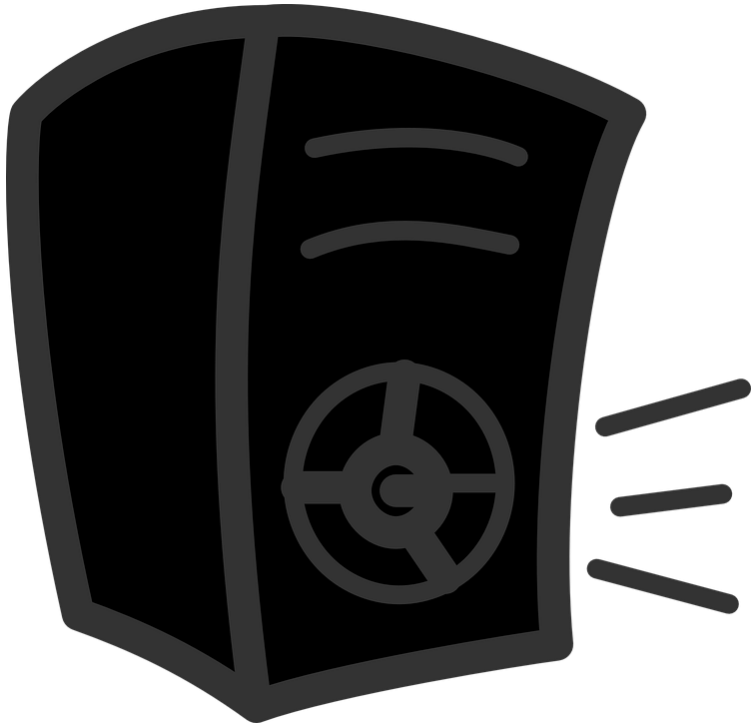
概念

電腦是被製造出來幫忙人類的工具，  
讓電腦把它所知道的东西告訴我們是非常重要的。

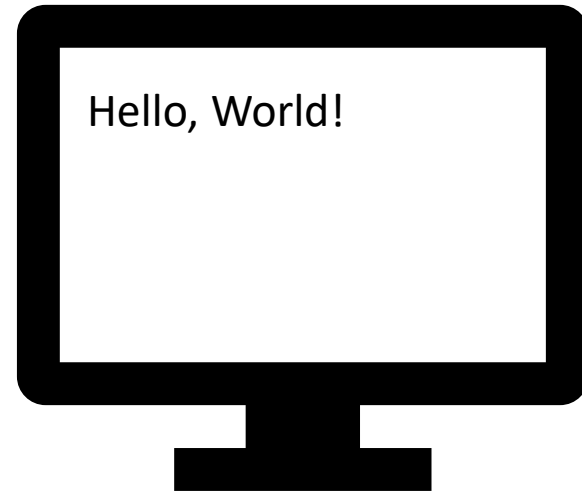


電腦把它想說的告訴我們，  
至少有兩種方法

讓電腦用喇叭發出聲音



用螢幕輸出文字



請在螢幕上顯示  
Hello, World!

問題

用螢幕輸出文字



# 先在編輯器上面給一個 樣式

```
#include <iostream>
using namespace std;
int main()
{
    return 0;
}
```

# 加上Hello, World!程式碼

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"Hello, World!"<<endl;
    return 0;
}
```



將顯示Hello, world!  
的程式碼, 寫在{}裡

# 程式碼的意義



# 需要一個樣式

- `cout<<"Hello, World!"<<endl;`  
只有這樣寫，是印不出來的
- 所有的C++語言都需要如右圖紅字的樣式
- 所有類似`cout<< <<endl;`的句子，  
都要寫在樣式中的`{ }`才有效

```
cout<<"Hello, World!"<<endl;
```

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"Hello, World!"<<endl;
    return 0;
}
```

# #include <iostream>

- include是包含的意思。
- 告訴編譯程序的預處理器，將輸入輸出的標準文件(iostream)包含在本程式中，這個標頭檔包含了C++中定義的基本標準輸入輸出函式(cin/cout)的聲明。

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"Hello, World!"<<endl;
    return 0;
}
```

# 還有一個管輸出入的函數：cstdio

- 都是管輸入輸出的函式庫
- `iostream`管  
`cin/cout`
- `cstdio`管  
`scanf/printf`
- 功能一樣，各有特色
- 本節先介紹`cin/cout`

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    cout<<"Hello, World!"<<endl;
    return 0;
}
```

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    printf( "Hello, World!\n");
    return 0;
}
```

標準函式庫

以C++  
為例


Reference	
C library:	
<cassert> (assert.h)	
<cctype> (ctype.h)	
<cerrno> (errno.h)	
<cfenv> (fenv.h)	C++11
<cfloat> (float.h)	
< cinttypes> (inttypes.h)	C++11
<ciso646> (iso646.h)	
<climits> (limits.h)	
<locale> (locale.h)	
<cmath> (math.h)	
< csetjmp> (setjmp.h)	
<csignal> (signal.h)	
<cstdarg> (stdarg.h)	
<cstdbool> (stdbool.h)	C++11
<cstddef> (stddef.h)	
<cstdint> (stdint.h)	C++11
<cstdio> (stdio.h)	
<cstdlib> (stdlib.h)	
<cstring> (string.h)	
<ctgmath> (tgmath.h)	C++11
<ctime> (time.h)	
<cuchar> (uchar.h)	C++11
<wchar> (wchar.h)	
<cwctype> (wctype.h)	
Containers:	
<array>	C++11
<deque>	
<forward_list>	C++11
<list>	
<map>	
<queue>	
<set>	
<stack>	
<unordered_map>	C++11
<unordered_set>	C++11
<vector>	
Input/Output:	
Multi-threading:	
Other:	

Containers:	
Input/Output:	
<fstream>	
<iomanip>	
<ios>	
<iosfwd>	
<iostream>	
<istream>	
<ostream>	
<sstream>	
<streambuf>	
Multi-threading:	
<atomic>	C++11
<condition_variable>	C++11
<future>	C++11
<mutex>	C++11
<thread>	C++11
Other:	
<algorithm>	
<bitset>	
<chrono>	C++11
<codecvt>	C++11
<complex>	
<exception>	
<functional>	
<initializer_list>	C++11
<iterator>	
<limits>	
<locale>	
<memory>	
<new>	
<numeric>	
<random>	C++11
<ratio>	C++11
<regex>	C++11
<stdexcept>	
<string>	
<system_error>	C++11
<tuple>	C++11
<typeindex>	C++11
<typeinfo>	
<type_traits>	C++11
<utility>	
<valarray>	

# using namespace std;

使用std(標準)命名空間的意思。

所謂命名空間是標準C++中一種機制，用在控制不同函式庫的衝突問題，使用它可以在不同空間內使用相同名字的函式庫或者函式。



就先打上這一行吧

# int main()

- 一個C++程式包含一個或以上的函式，所有程式開始於 **main()**
  - 小括號的出現，代表一個函式
- **int**表示傳回一個整數值
- 一個函式的程式碼，被一對大括號**{ }**所包圍
- **return 0;**  
一種返回程式的方法，也代表程式正確的結束並返回

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"Hello, World!"<<endl;
    return 0;
}
```

# cout<<"Hello, World!"<<endl;

- cout()  
輸出函式，告訴計算器將" " 之間的字符串送到標準輸出設備（如螢幕）上
- 雙引號內的文字會被顯示出來
  - cout<<"Hello, World!"<<endl;
  - <<endl有換行的功能

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"Hello, World!"<<endl;
    return 0;
}
```

# ;是每一行指令的結束符號

- 每一行（或者說每一個statement），以分號;作為結束。

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout<<"Hello, DICE!"<<endl
6     return 0;
7 }
8
9
```

沒有結束符號，  
就犯了語法錯誤，  
無法編譯



# 語法與語意

- 如果犯了語法上的錯誤，將會無法編譯成功，而且在編譯的過程會提示語法上的錯誤。
- 而語意的錯誤，編譯程式就無法辨識出來，例如，輸出的結果與問題的需求是不同的，**你好**寫成**你不好**，就是語意上的問題，這些問題只能靠工程師小心檢查錯誤。

# 編譯器與編譯

- 簡單來說，編譯器，就是將人類懂的語言翻譯成電腦懂的語言，DICE裡頭藏有一個翻譯官，按下執行就可以執行編譯的工作。

編譯器（compiler）是一種電腦程式，它會將某種程式語言寫成的原始碼（原始語言）轉換成另一種程式語言（目標語言）。

一個現代編譯器的主要工作流程如下：

原始碼（source code）→ 預處理器（preprocessor）→ 編譯器（compiler）→ 組譯程式（assembler）→ 目的碼（object code）→ 連結器（linker）→ 執行檔（executables），最後打包好的檔案就可以給電腦去判讀執行了。

# 延伸學習- cout( )的格式問題

# 一些轉換符號的用法

跳脫字符的程式	描述
<code>\n</code>	換行
<code>\t</code>	水平平移一個tab鍵距離
<code>\a</code>	發出警示值聲音
<code>\\</code>	印斜線\ 印雙引號"
<code>\"</code>	

```
cout<<"Hello, World!"<<endl;
```

endl已包含換行的功能

# 空白並不是代表沒有，大小寫是有區別的

先暫且將它視為注意事項，小心進行，詳細說明在字元數字本一家

Bin ( 二進位 )	Oct ( 八進位 )	Dec ( 十進位 )	Hex ( 十六進位 )	縮寫/字符	解釋
0000 0000	00	0	0x00	NUL(null)	空字元符
0000 0001	01	1	0x01	SOH(start of headline)	標題開始
0000 0010	02	2	0x02	STX(start of text)	正文開始
0000 0011	03	3	0x03	ETX(end of text)	正文結束
0000 0100	04	4	0x04	EOT(end of transmission)	傳輸結束
0000 0101	05	5	0x05	ENQ(enquiry)	請求
0000 0110	06	6	0x06	ACK(acknowledge)	收到通知
0000 0111	07	7	0x07	BEL(bell)	響鈴
0000 1000	010	8	0x08	BS(backspace)	退格
0000 1001	011	9	0x09	HT(horizontal tab)	水平制表符
0000 1010	012	10	0x0A	LF(NL line feed, new line)	換行鍵