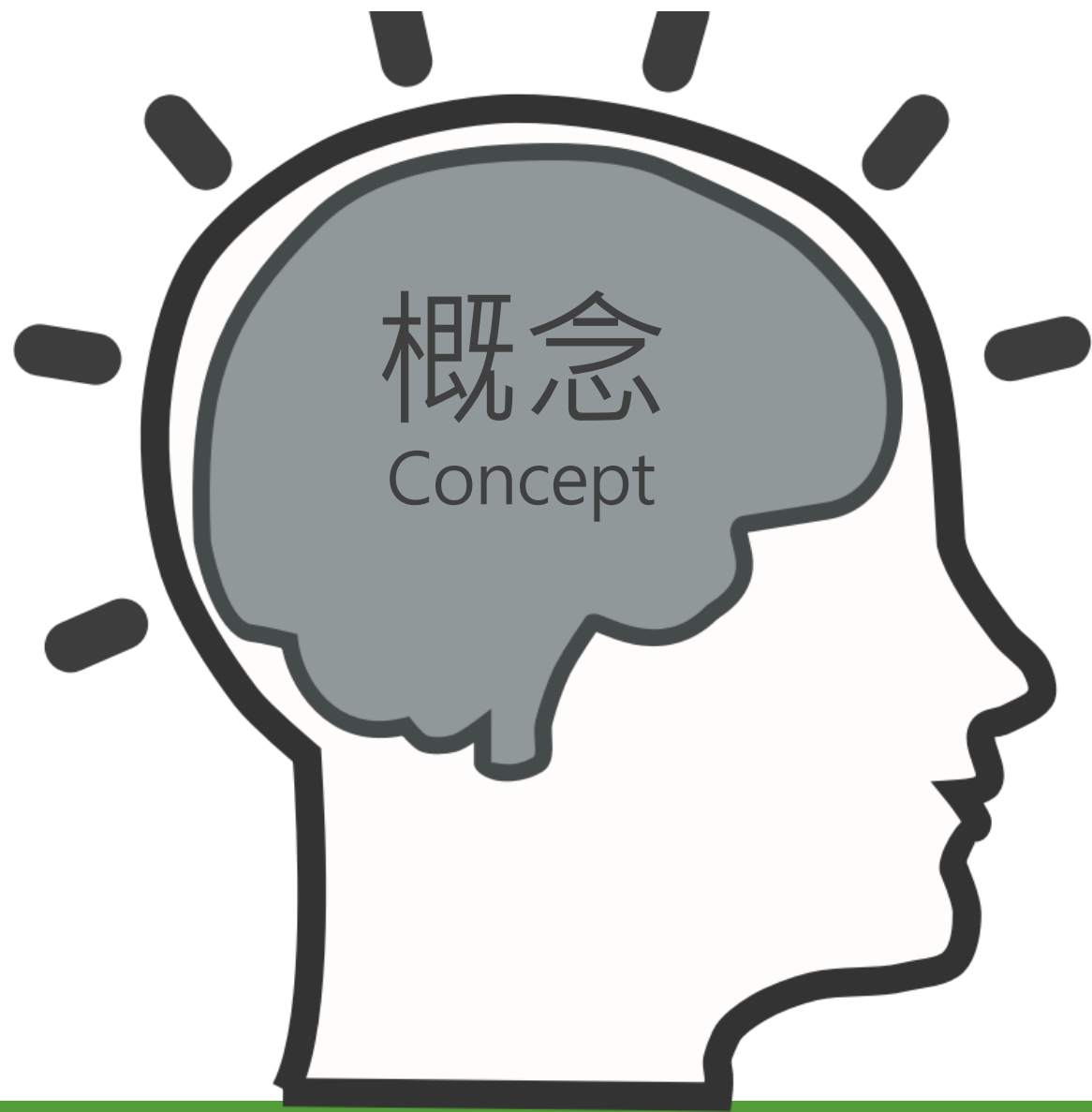




# 輸出函式



概念  
Concept

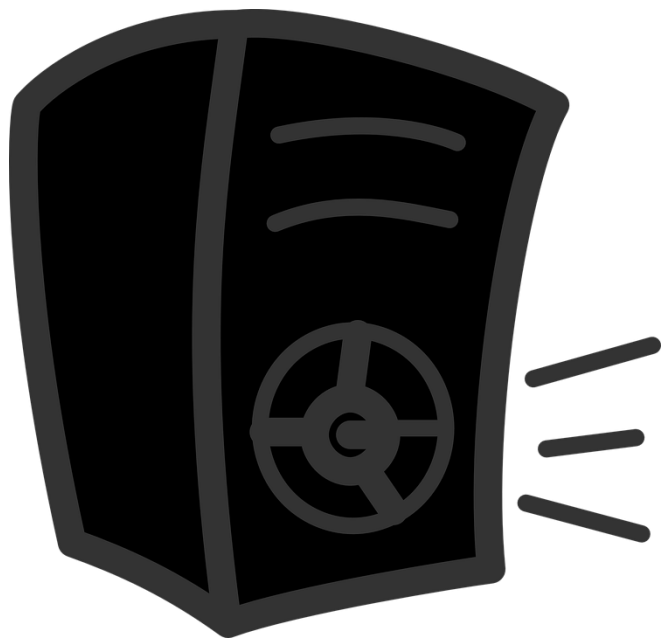
電腦是被製造出來幫忙人類的工具，  
讓電腦把它所知道的東西告訴我們是非常重要的。



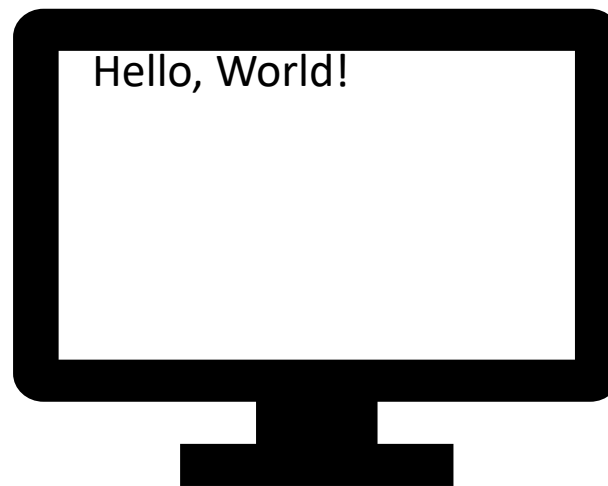
# 電腦把它想說的告訴我們，至少有兩種方法

---

讓電腦用喇叭發出聲音



用螢幕輸出文字



請電腦印出  
Hello, World!

練習



# 先在編輯器上面給一個樣式

---

```
#include <stdio.h>
int main()
{
    return 0;
}
```

# 在螢幕上印Hello, World!

---

```
#include <stdio.h>
int main()
{
    printf("Hello, World!\n");
    return 0;
}
```



將印出Hello, world!  
的程式碼, 寫在{ }裡  
The codes need to be  
written in the brace{ }



# 程式碼的意義



# 需要一個樣式

It needs a style

---

- `printf( "Hello, World!" );`  
只有這樣寫是印不出來的 所有的C語言都需要如右  
下圖黑字的樣式
- 所有類似`printf`的句子，  
都要寫在`{ }`裡才有效

```
#include <stdio.h>

int main()
{
    printf( "Hello, World!\n");
    return 0;
}
```

# #include <stdio.h>
















---

- include是包含的意思
- 用到的輸出功能printf()  
被定義在標準輸出入函式庫<stdio.h>
- 必須將stdio.h包含進來，寫法如：  
`#include <stdio.h>`

```
#include <stdio.h>
```

```
int main()  
{  
    printf( "你好\n");  
    return 0;  
}
```

# C的函式 庫範例

名稱	修改日期	類型	大小
 stdarg	2014/12/8 上午 01:48	C Header File	5 KB
 stddef	2014/12/8 上午 01:48	C Header File	13 KB
 stdexcept	2014/12/8 上午 01:48	C Header File	1 KB
 stdint	2014/12/8 上午 01:48	C Header File	7 KB
 stdio	2014/12/8 上午 01:48	C Header File	37 KB
 stdlib	2014/12/8 上午 01:48	C Header File	26 KB
 sti	2014/12/8 上午 01:48	C Header File	18 KB
 stierr	2014/12/8 上午 01:48	C Header File	3 KB
 stireg	2014/12/8 上午 01:48	C Header File	2 KB
 stdllock	2014/12/8 上午 01:48	C Header File	1 KB
 stm	2014/12/8 上午 01:48	C Header File	4 KB
 storage	2014/12/8 上午 01:48	C Header File	1 KB
 storduid	2014/12/8 上午 01:48	C Header File	1 KB
 storprop	2014/12/8 上午 01:48	C Header File	2 KB
 stralign	2014/12/8 上午 01:48	C Header File	7 KB

# stdio.h內容

```
C:\Program Files (x86)\Dev-Cpp\MinGW64\x86_64-w64-mingw32\include\stdio.h - Dev-C++ 5.11
檔案(F) 編輯(E) 搜尋(S) 檢視(V) 專案(P) 執行(Z) 工具(T) AStyle 視窗(W) 求助(H)
(globals)
專案 類別 除錯 main.c main.c main.c stdio.h
271 }
272
273 __mingw_ovr
274 __attribute__((__format__(gnu_scanf, 2, 0))) __MINGW_ATTRIB_NONNULL(2)
275 int vfscanf (FILE *_stream, const char *_format, __builtin_va_list __local_argv)
276 {
277     return __mingw_vfscanf( __stream, __format, __local_argv );
278 }
279 #endif /* __NO_ISOCEXT */
280
281
282
283 __mingw_ovr
284 __attribute__((__format__(gnu_printf, 2, 3))) __MINGW_ATTRIB_NONNULL(2)
285 int fprintf (FILE *_stream, const char *_format, ...)
286 {
287     register int __retval;
288     __builtin_va_list __local_argv; __builtin_va_start( __local_argv, __format
289     __retval = __mingw_vfprintf( __stream, __format, __local_argv );
290     __builtin_va_end( __local_argv );
291     return __retval;
292 }
293
294 __mingw_ovr
295 __attribute__((__format__(gnu_printf, 1, 2))) __MINGW_ATTRIB_NONNULL(1)
296 int printf (const char *_format, ...)
297 {
298     register int __retval;
299     return __mingw_printf( __format, ... );
300 }
```

中斷

Compiling single file...

行數： 285 列數： 1 反白字數： 6 總行數： 963 檔案長度： 37658 插入模式 完成解析 (花了 0.016 秒)

# 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()**

C programming contains more than one function, all programs start at main()

- 小括號的出現，代表一個函式 The appearance of parentheses represents a function
- **int**表示傳回一個整數值 int means to return an integer value
- 一個函式的程式碼，被一對大括號{ }所包圍  
The code of a function, surrounded by a pair of braces {}
- **return 0;**  
一種返回程式的方法，也代表程式正確的結束並返回  
It is a way to end and back

```
#include <stdio.h>
```

```
int main()  
{  
    printf( "你好\n");  
    return 0;  
}
```

```
printf("Hello, World!\n");
```

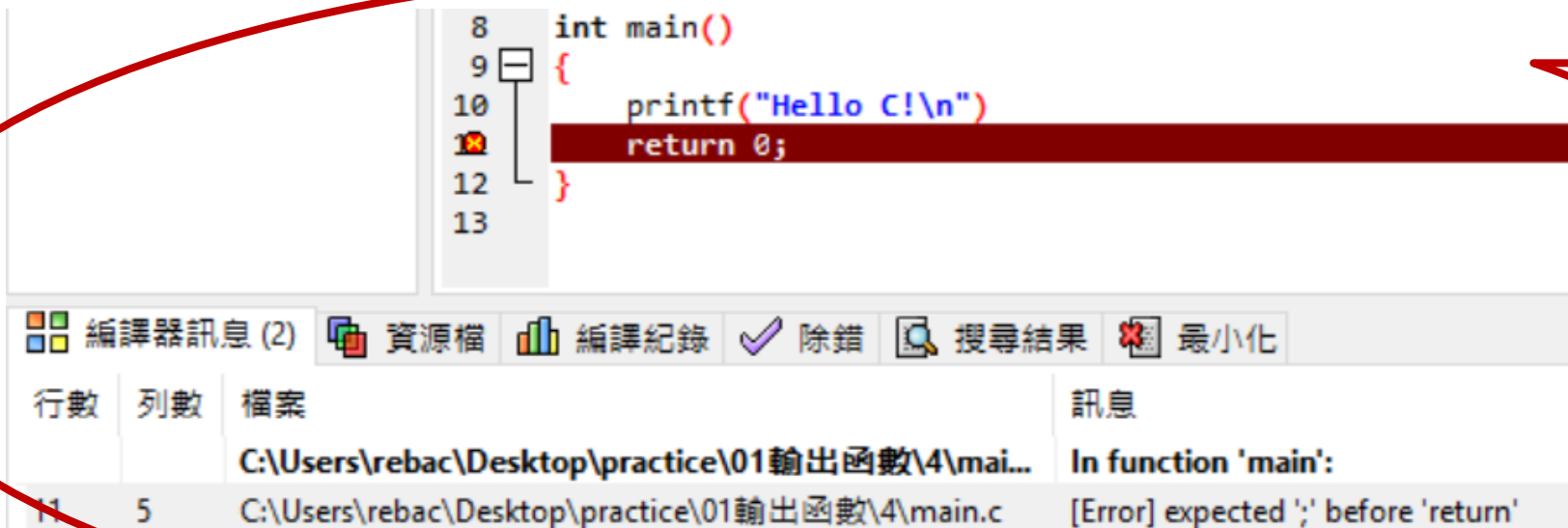
---

- printf()  
輸出函式，指示電腦去執行輸出指令
- 小跨號雙引號內的文字會被印出來
  - printf( "Hello, World!" )

```
#include <stdio.h>
int main()
{
    printf( "Hello World!\n");
    return 0;
}
```

# ; 是每一行指令的結束符號 ; It is the end symbol of each line

- 每一個行 ( statement ) , 以分號; 作為結束  
every statement is ended by semicolon;



```
8 int main()
9 {
10     printf("Hello C!\n")
11     return 0;
12 }
13
```

行數	列數	檔案	訊息
		C:\Users\rebac\Desktop\practice\01輸出函數\4\mai...	In function 'main':
11	5	C:\Users\rebac\Desktop\practice\01輸出函數\4\main.c	[Error] expected ';' before 'return'

沒有結束符號，  
是犯了語法錯誤，  
編譯不了的。  
It can't be  
compiled if  
no end  
symbol were  
used

# 語法與語意

## syntax and semantics

---

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。 If you make a syntax error, you will not be able to compile successfully.

而語意的錯誤，編譯器無法辨識出來，  
例如，輸出的結果與問題的需求是不同，

你好寫成你不好，就是語意上的問題，這些問題只能靠程式設計師小心偵錯。

The semantic error cannot be recognized by the compiler. For example, 你好 is written as 你不好, it is semantic problems, these problems can only rely on the programmer to carefully debug.



# 編譯器 Compiler

簡單的說編譯器，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。

Briefly speaking, a compiler means translating a human into a language understood by computers. There is a translator hidden in DICE, and the compiler can be executed by pressing execute.

## compiler

### 網路上的定義

編譯器，是一種電腦程序，它會將用某種編程語言寫成的源代碼，轉換成另一種編程語言。它主要的目的是將便於人編寫，閱讀，維護的高級計算機語言所寫作的源代碼程序，翻譯為計算機能解讀、運行的低階機器語言的程序，也就是可執行文件。編譯器將原始程序作為輸入，翻譯產生使用目標語言的等價程序。源代碼一般為高階語言，如 Pascal、C、C++、C#、Java 等，而目標語言則是彙編語言或目標機器的目標代碼，有時也稱作機器代碼。一個現代編譯器的主要工作流程如下：源代碼 → 預處理器 → 編譯器 → 彙編程序 → 目標代碼 → 鏈接器 → 可執行文件

<http://zh.wikipedia.org/zh-cn/Compiler>



# 延伸的概念

Extended concept

# 註解的用法

---

```
/*寫在這裡的文字，是給人看的。*/  
/*就是註解的意思*/
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    printf( "Hello, World!\n" );  
    //printf()是輸出函數
```

```
    return 0;
```

```
}
```

# 換行的用法 the way to get a new line

---

- 當印出兩個句子以上，會使用換行指令
- `printf( "Hello, \n" );`  
`printf("World!\n");`
- `\n`，表示換行 `\n`

# 一些脫序字元的用法

The usage of escape

脫序字元的指令 escape	描述 description
<code>\n</code>	換行 get a new line
<code>\t</code>	水平平移一個tab鍵距離 by one tab key distance
<code>\a</code>	發出警示值聲音 Make a warning sound
<code>\\</code>	印斜線\ to print slash \
<code>\"</code>	印雙引號" to print double quotes

# 空白並不是代表沒有，大小寫是有區別的

二進位	十進位	十六進位	縮寫	Unicode 表示法	脫出字元 表示法	名稱/意義
0000 0000	0	00	NUL	NUL	^@	空字元 (Null)
0000 0001	1	01	SOH	SOH	^A	標題開始
0000 0010	2	02	STX	STX	^B	本文開始
0000 0011	3	03	ETX	ETX	^C	本文結束
0000 0100	4	04	EOT	EOT	^D	傳輸結束
0000 0101	5	05	ENQ	ENQ	^E	請求
0000 0110	6	06	ACK	ACK	^F	確認回應
0000 0111	7	07	BEL	BEL	^G	響鈴
0000 1000	8	08	BS	BS	^H	退格
0000 1001	9	09	HT	HT	^I	水平定位符號
0000 1010	10	0A	LF	LF	^J	換行鍵