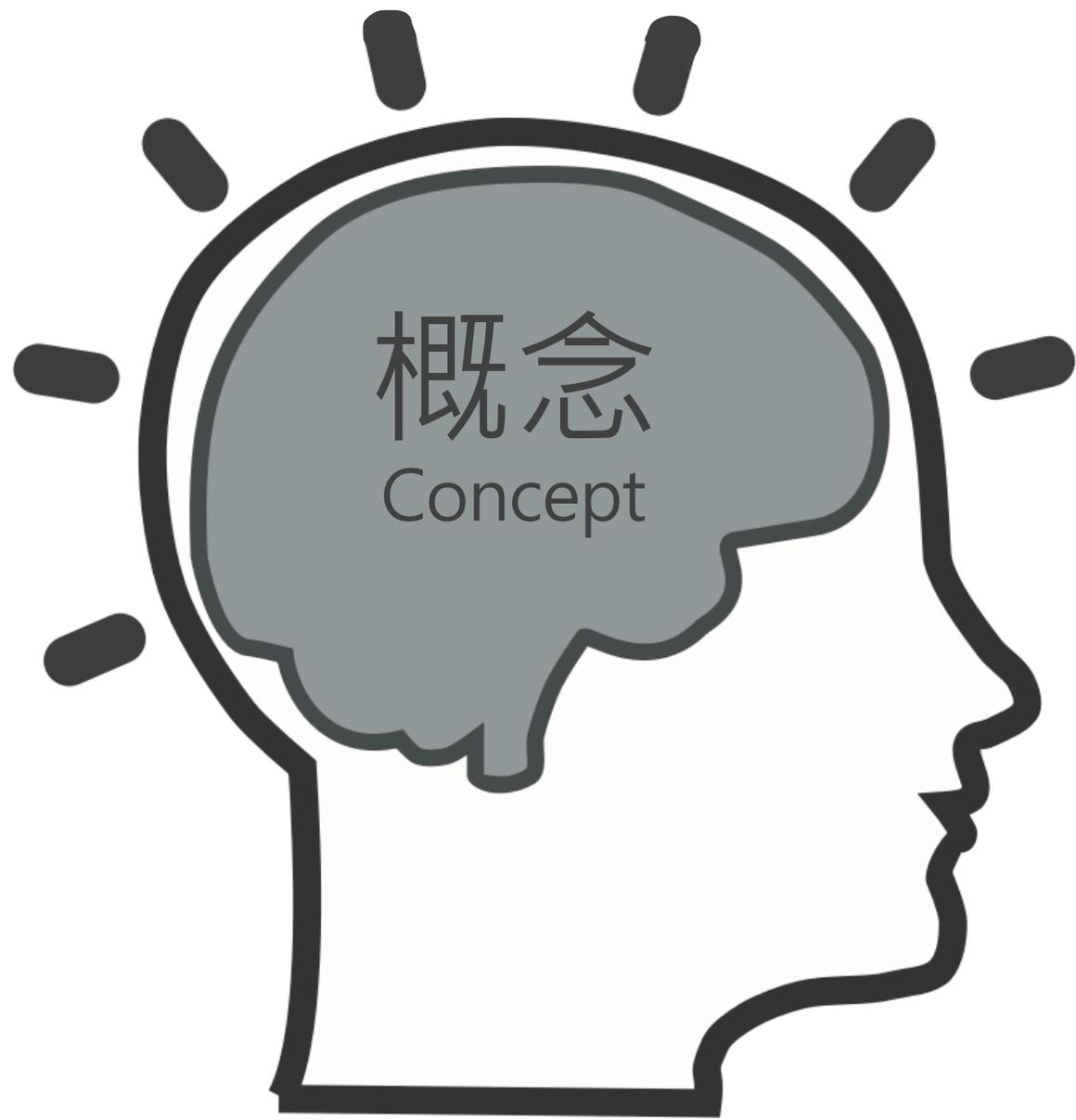


C 語言

輸出函式





概念

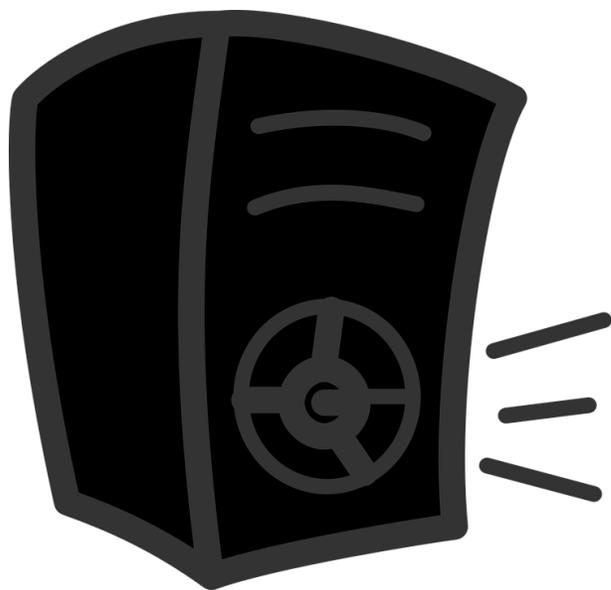
Concept

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



練習

請電腦印出
Hello, World!



先在編輯器上面給一個 樣式

```
#include <stdio.h>

int main( ){

    /*
    程式碼寫在這;
    */

    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



C 語言

程式碼的意義

需要一個樣式

- `printf("Hello, World!");`
只有這樣寫是印不出來的，所有的C語言都需要如右圖的樣式。
- 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
 ssp.h	2014/12/8 上午 06:32	C Header File	3 KB
 stdio.h	2014/12/8 上午 06:32	C Header File	4 KB
 string.h	2014/12/8 上午 06:32	C Header File	6 KB
 unistd.h	2014/12/8 上午 06:32	C Header File	3 KB

stdio.h内容

```
C stdio.h 6 x
C: > Program Files (x86) > Dev-Cpp > MinGW64 > x86_64-w64-mingw32 > include > C stdio.h > fprintf(FILE *, const char *, ...)
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     __builtin_va_list __local_argv; __builtin_va_start( __local_argv, __format );
300     __retval = __mingw_vprintf( __format, __local_argv );
301     __builtin_va_end( __local_argv );
302     return __retval;
303 }
```

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 <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

- 每一個行 (statement) ， 以分號 ; 作為結束

```
1 #include <stdio.h>
2
3 int main() {
4
5     printf("Hello C!\n")
6
7     return 0;
8 }
```

Compiler (3) Resources Compile Log Debug Find Results Close

Line	Col	File	Message
7	2		In function 'main': [Error] expected ';' before 'return'
28			recipe for target 'main.o' failed

沒有結束符號，是犯了語法錯誤，編譯不了的。

語法與語意

語法錯誤(syntax error)：

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。

語意的錯誤(sematic error)：

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。

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。

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) 。 它主要的目的是將便於人編寫、閱讀、維護的進階電腦語言所寫作的原始碼程式，翻譯為電腦能解讀、執行的低階機器語言的程式，也就是執行檔。

[編譯器- 維基百科，自由的百科全書](#)



C 語言

延伸的概念

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

- 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

脫序字元 (escape)的指令	描述 description
<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
<code>\"</code>	印雙引號"

空白並不是代表沒有，大小寫是有區別的

ASCII控制字元 (共33個)

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
0000 1011	11	0B	VT	VT	^K	垂直定位符號
0000 1100	12	0C	FF	FF	^L	換頁鍵