c++ environment for Mac

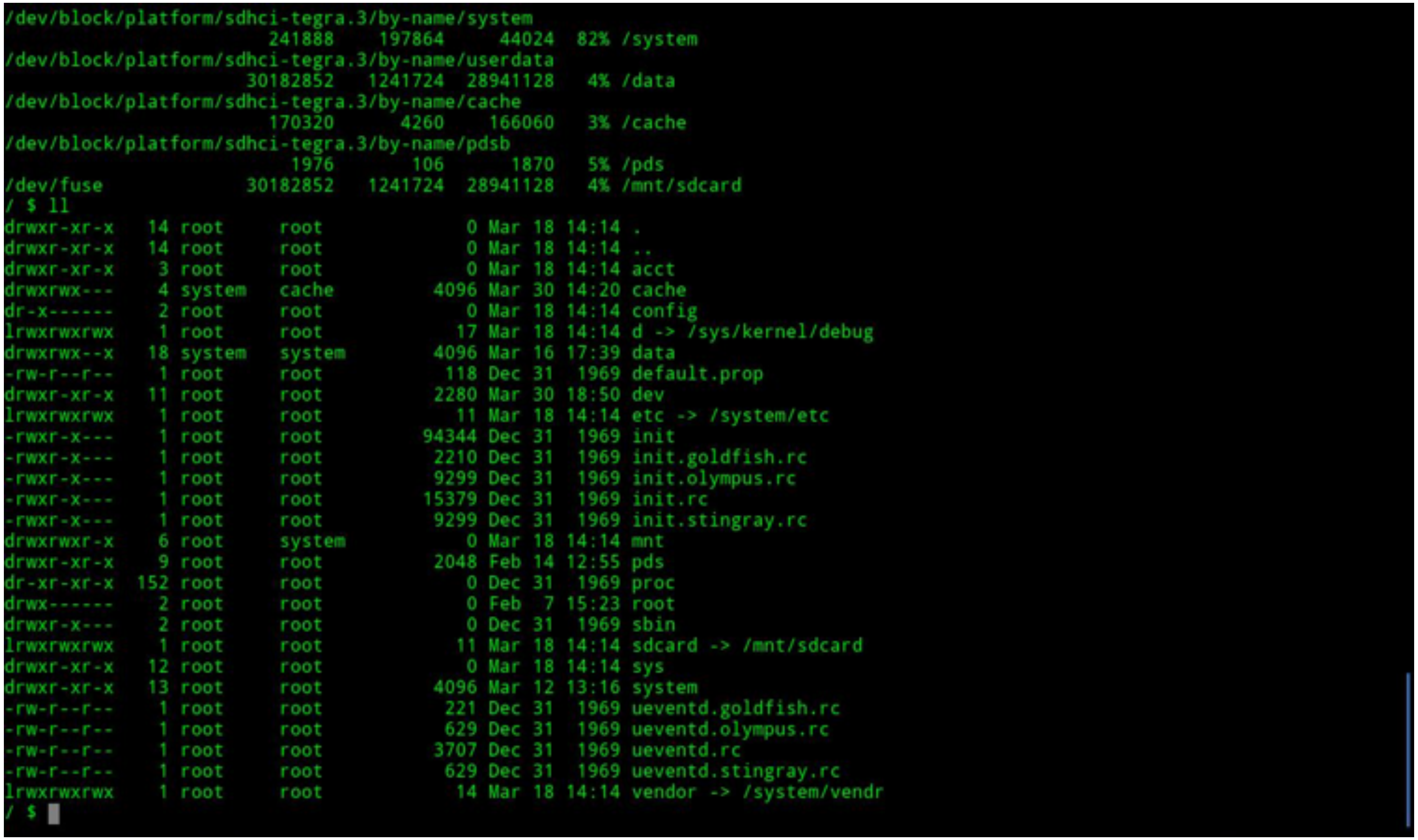
Why not Xcode?

- It tolerate too much errors
- It include lots of header by itself
- 所以實在不建議使用Xcode，因為程式很容易編譯不過，編譯不過就沒分數

contents

- Terminal — 操作界面
- Package manager — tool for installing compiler
- Install gcc — install the compiler for c++

Terminal



```
/dev/block/platform/sdhci-tegra.3/by-name/system
                        241888    197864     44024  82% /system
/dev/block/platform/sdhci-tegra.3/by-name/userdata
                      30182852   1241724  28941128   4% /data
/dev/block/platform/sdhci-tegra.3/by-name/cache
                        170320      4260    166060   3% /cache
/dev/block/platform/sdhci-tegra.3/by-name/pdsb
                          1976       106      1870   5% /pds
/dev/fuse             30182852   1241724  28941128   4% /mnt/sdcard
/ $ ll
drwxr-xr-x   14 root     root             0 Mar 18 14:14 .
drwxr-xr-x   14 root     root             0 Mar 18 14:14 ..
drwxr-xr-x    3 root     root             0 Mar 18 14:14 acct
drwxrwx---    4 system   cache         4096 Mar 30 14:20 cache
dr-x------    2 root     root             0 Mar 18 14:14 config
lrwxrwxrwx    1 root     root            17 Mar 18 14:14 d -> /sys/kernel/debug
drwxrwx--x   18 system   system        4096 Mar 16 17:39 data
-rw-r--r--    1 root     root           118 Dec 31  1969 default.prop
drwxr-xr-x   11 root     root          2280 Mar 30 18:50 dev
lrwxrwxrwx    1 root     root            11 Mar 18 14:14 etc -> /system/etc
-rwxr-x---    1 root     root         94344 Dec 31  1969 init
-rwxr-x---    1 root     root          2210 Dec 31  1969 init.goldfish.rc
-rwxr-x---    1 root     root          9299 Dec 31  1969 init.olympus.rc
-rwxr-x---    1 root     root         15379 Dec 31  1969 init.rc
-rwxr-x---    1 root     root          9299 Dec 31  1969 init.stingray.rc
drwxrwxr-x    6 root     system           0 Mar 18 14:14 mnt
drwxr-xr-x    9 root     root          2048 Feb 14 12:55 pds
dr-xr-xr-x  152 root     root             0 Dec 31  1969 proc
drwx------    2 root     root             0 Feb  7 15:23 root
drwxr-x---    2 root     root             0 Dec 31  1969 sbin
lrwxrwxrwx    1 root     root            11 Mar 18 14:14 sdcard -> /mnt/sdcard
drwxr-xr-x   12 root     root             0 Mar 18 14:14 sys
drwxr-xr-x   13 root     root          4096 Mar 12 13:16 system
-rw-r--r--    1 root     root           221 Dec 31  1969 ueventd.goldfish.rc
-rw-r--r--    1 root     root           629 Dec 31  1969 ueventd.olympus.rc
-rw-r--r--    1 root     root          3707 Dec 31  1969 ueventd.rc
-rw-r--r--    1 root     root           629 Dec 31  1969 ueventd.stingray.rc
lrwxrwxrwx    1 root     root            14 Mar 18 14:14 vendor -> /system/vendr
/ $ 
```

Terminal

- 終端機
- 使用指令來完成每一步動作
- 使用文字介面來操作電腦

Terminal tutorial

- 如果不會使用Terminal，可以先看網路上的教學
- LearnCode.academy — Basic Terminal Usage
https://www.youtube.com/watch?v=jDINUSK7rXE

Package manager

Package manager

用處：

- install package
- update package
- remove package
- and so on…

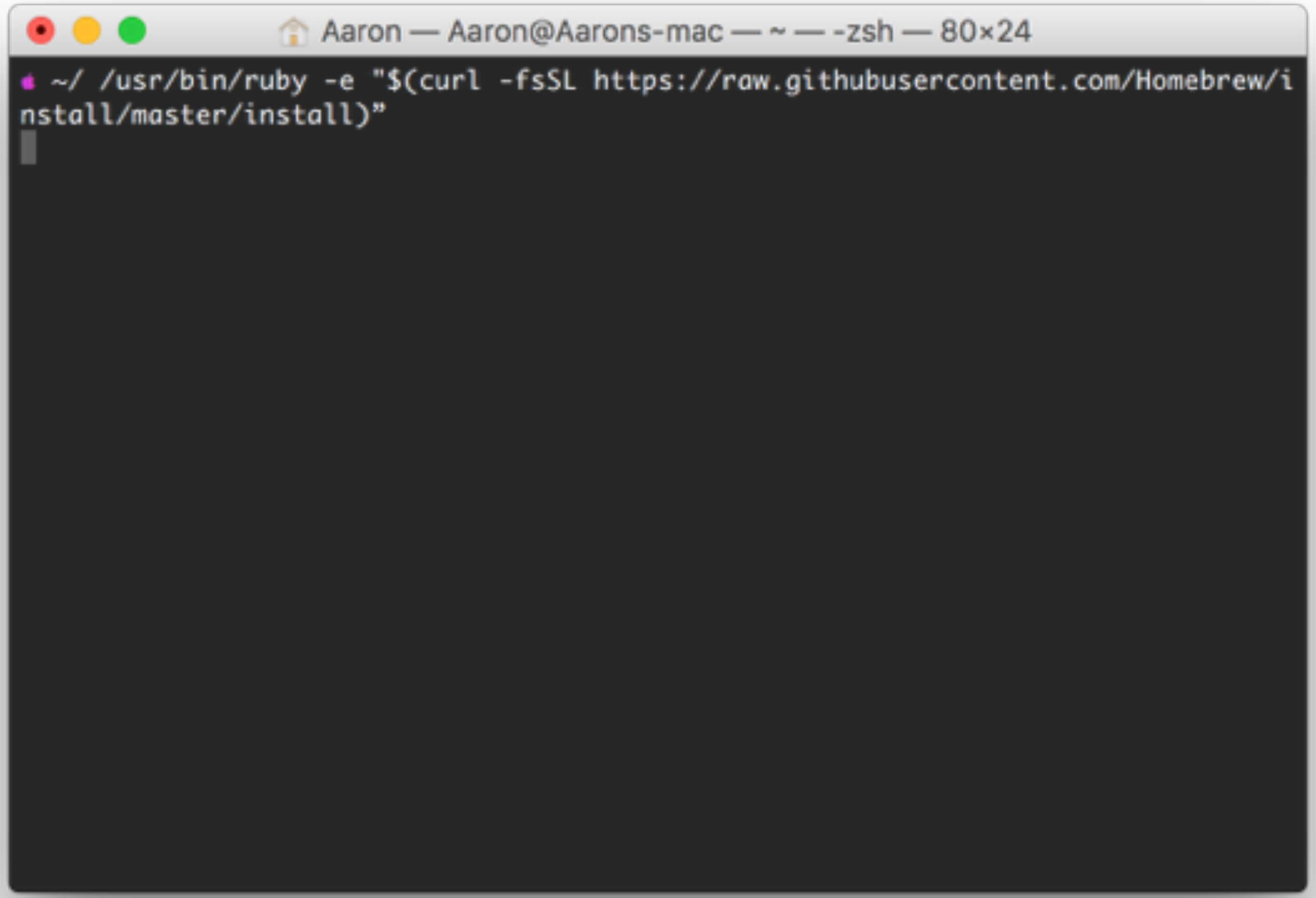
Package manager

- Mac上的其中一個Package manager就是HomeBrew

Install HomeBrew

- paste the following command to the terminal:

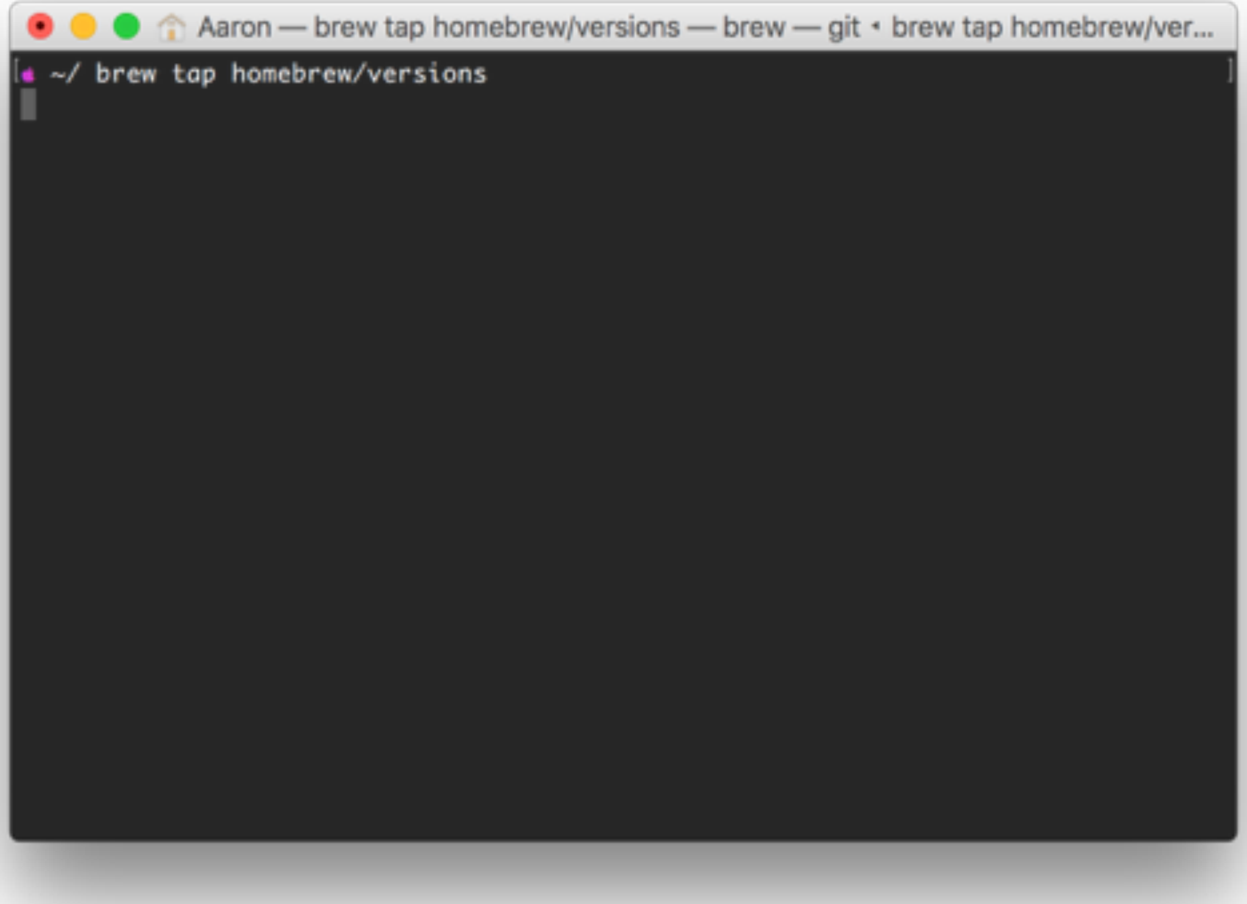
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"



```
~/ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/i
nstall/master/install)"
```

install gcc/g++

- brew tap homebrew/versions

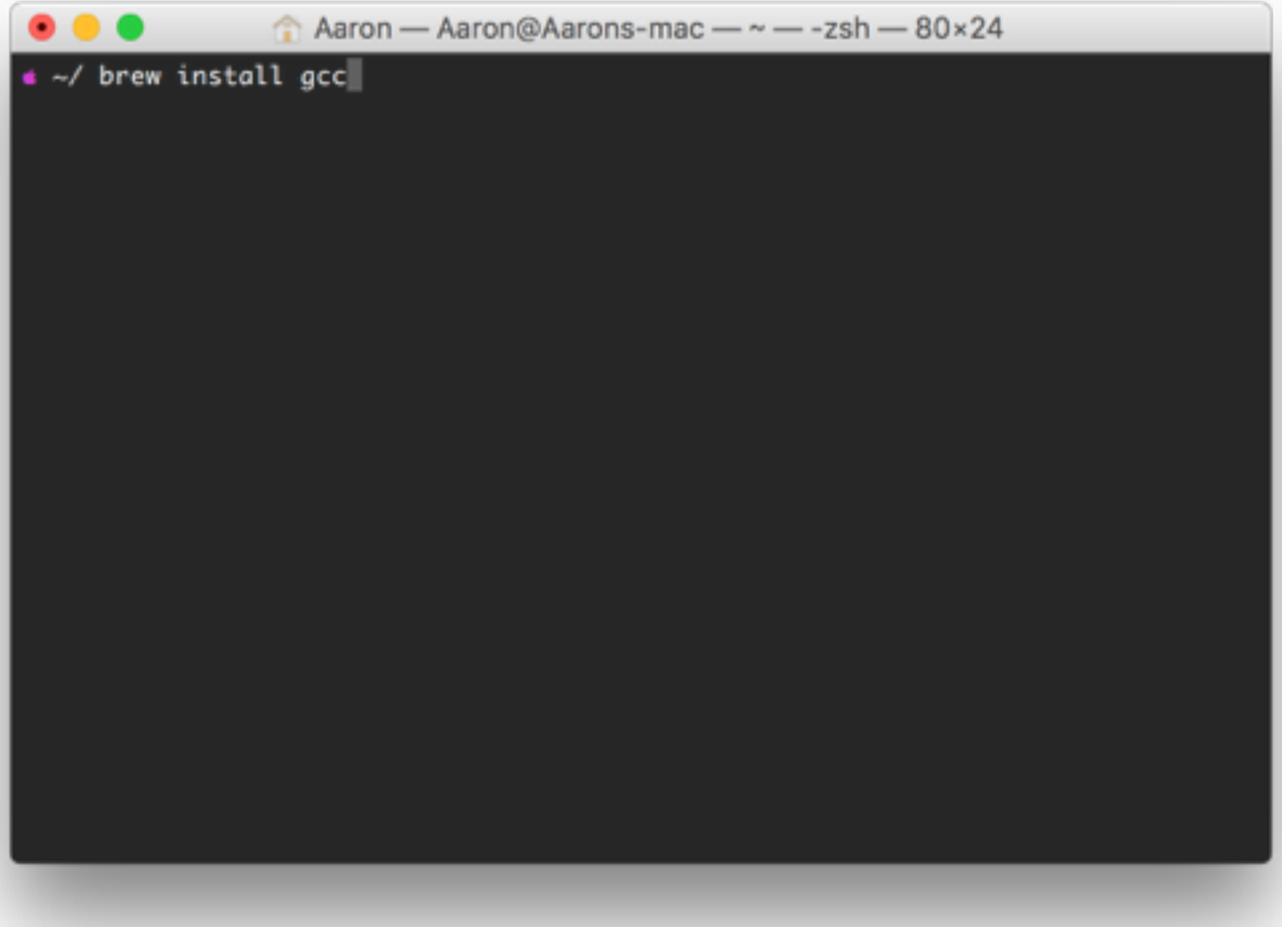


Aaron — brew tap homebrew/versions — brew — git ◂ brew tap homebrew/ver...

```
~/ brew tap homebrew/versions
```

install gcc/g++

- brew install gcc



```
~/ brew install gcc
```

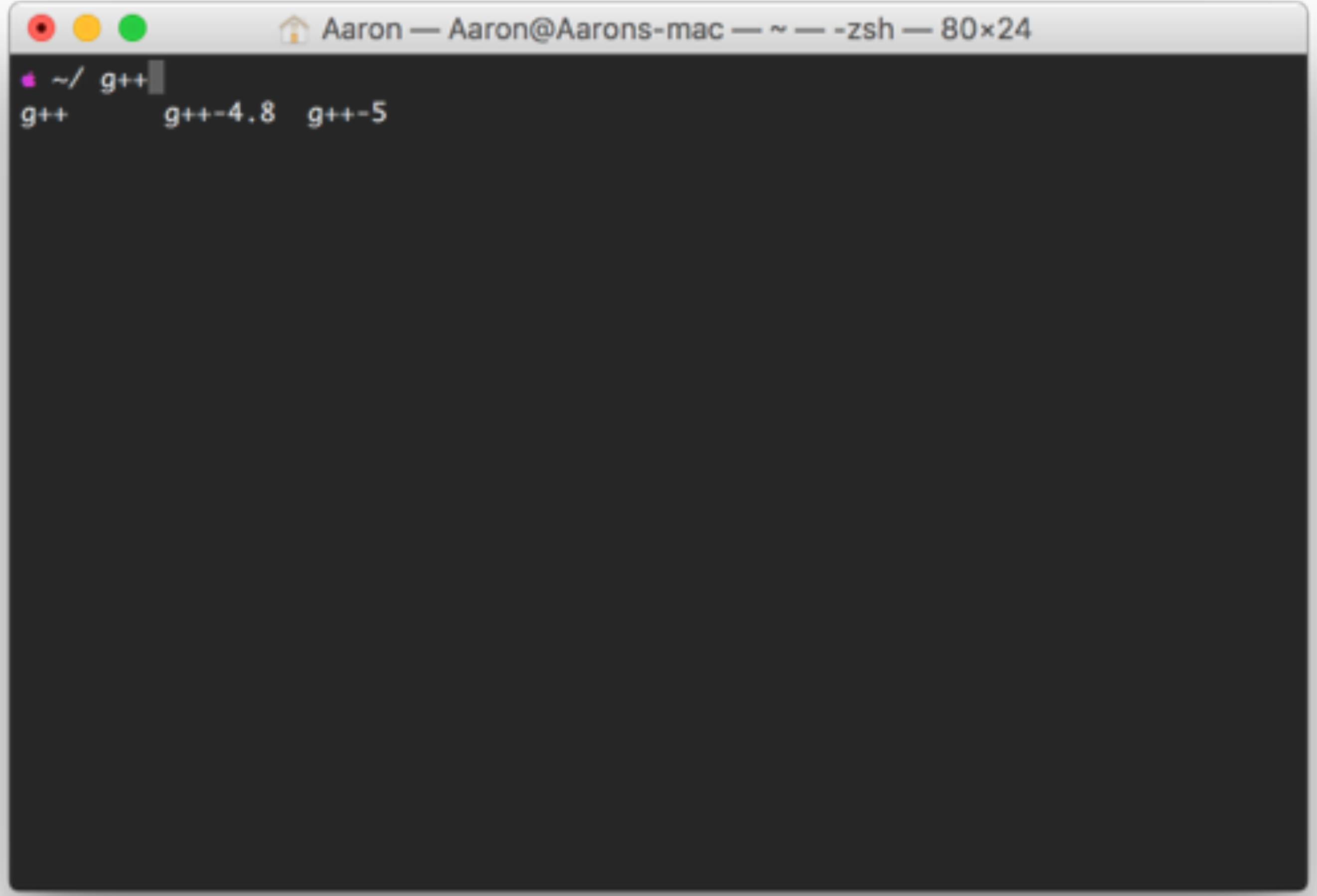
測試是否安裝成功

- 在Terminal上打 which g++
  顯示 /usr/local/bin/g++ 即成功

- 注意！
  如果顯示 /usr/bin/g++ 表示你的Terminal g++是指到mac內建的g++，用這個g++來compile還是會保有Xcode的缺點，因為Xcode就是用這個g++來compile你的程式

如果顯示 /usr/bin/g++

- 可以在Terminal打上 g++ 之後按幾下Tab，會出現不同版本的g++，選擇g++以外的來compile都可以



```
 ~/ g++
g++       g++-4.8  g++-5
```

Editor vs IDE

- editor :
  light weight,
  support lots of language,
  can’t debug (use terminal to find bug),
  need compiler & terminal to compile code
  often have extensions to support coding

- IDE：
  heavy
  code understanding
  debug
  integrated build
  …..

Editors

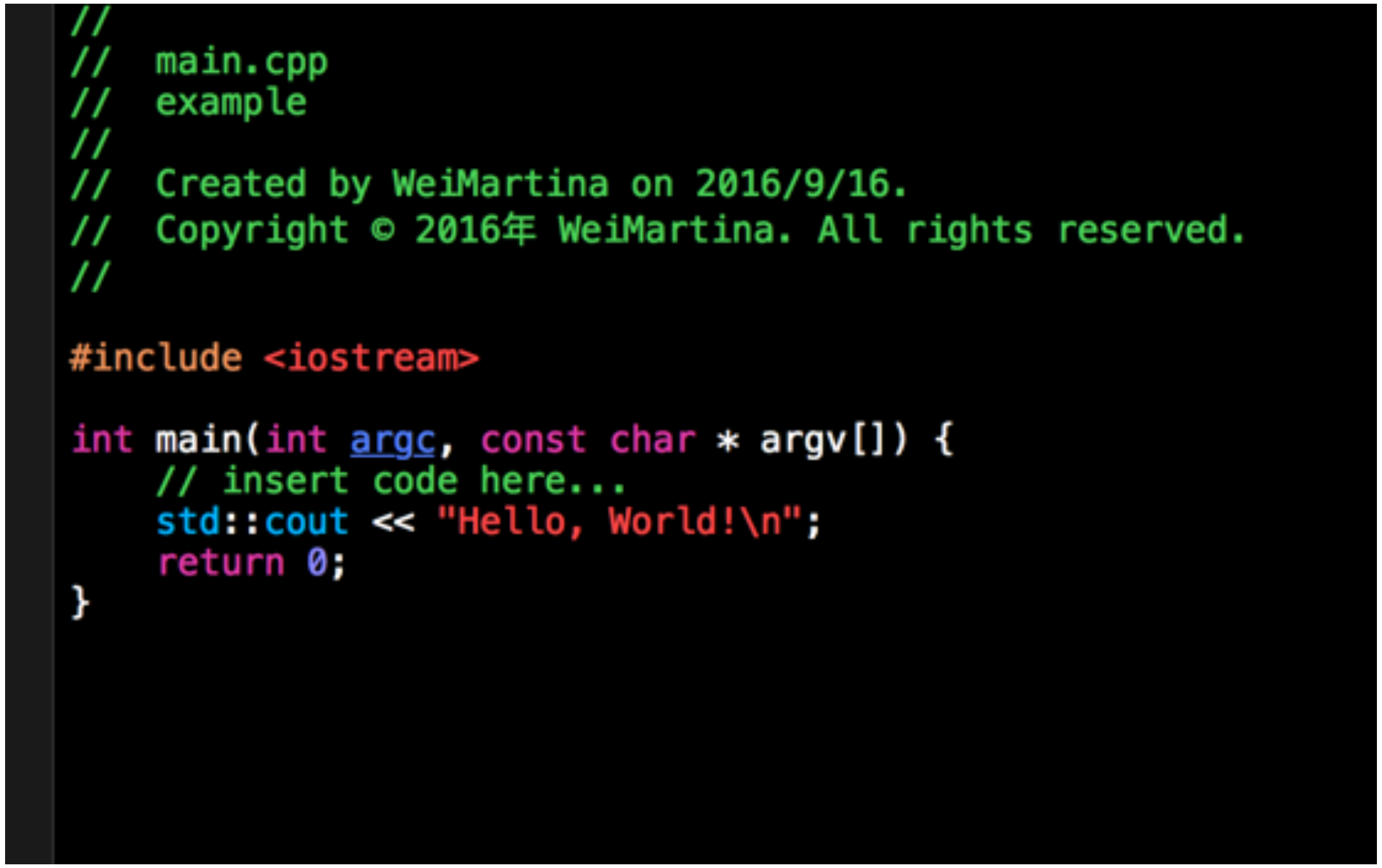
- 推薦使用 Sublime text 3
https://www.sublimetext.com/

- 建議安裝Sublime text的package management:
https://packagecontrol.io/installation

- 之後就可以google:Sublime text extension for c++，找到插件安裝使用看看，把Sublime text調整到自己最滿意的狀態

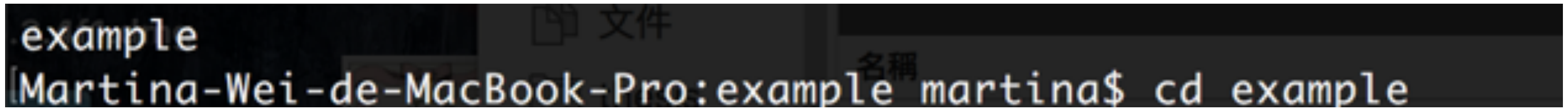
run C++ on terminal

- 怎麼在terminal上run *.cpp檔？



```
//
//  main.cpp
//  example
//
//  Created by WeiMartina on 2016/9/16.
//  Copyright © 2016年 WeiMartina. All rights reserved.
//

#include <iostream>

int main(int argc, const char * argv[]) {
    // insert code here...
    std::cout << "Hello, World!\n";
    return 0;
}
```

1. 打開terminal，變更路徑到*.cpp在的資料夾 (cd)



```
example
Martina-Wei-de-MacBook-Pro:example martina$ cd example
```

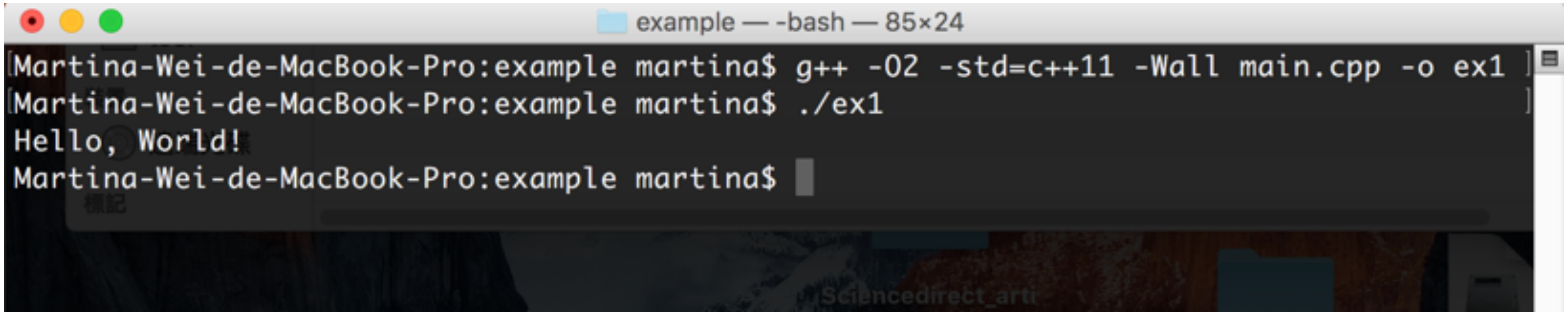
2. 要編譯的*.cpp只有一兩個的話可以直接在terminal輸入

g++ -O2 -std=c++11 -Wall **name_of_your.cpp** -o **name_of_your_output**

編譯完後，在terminal輸入 **./name_of_your_output**

3. 就可以看到輸出結果了

（補充）如果有*.h 放在*.cpp之後，以空白鍵分隔 **name_of_your.cpp name_of your.h**

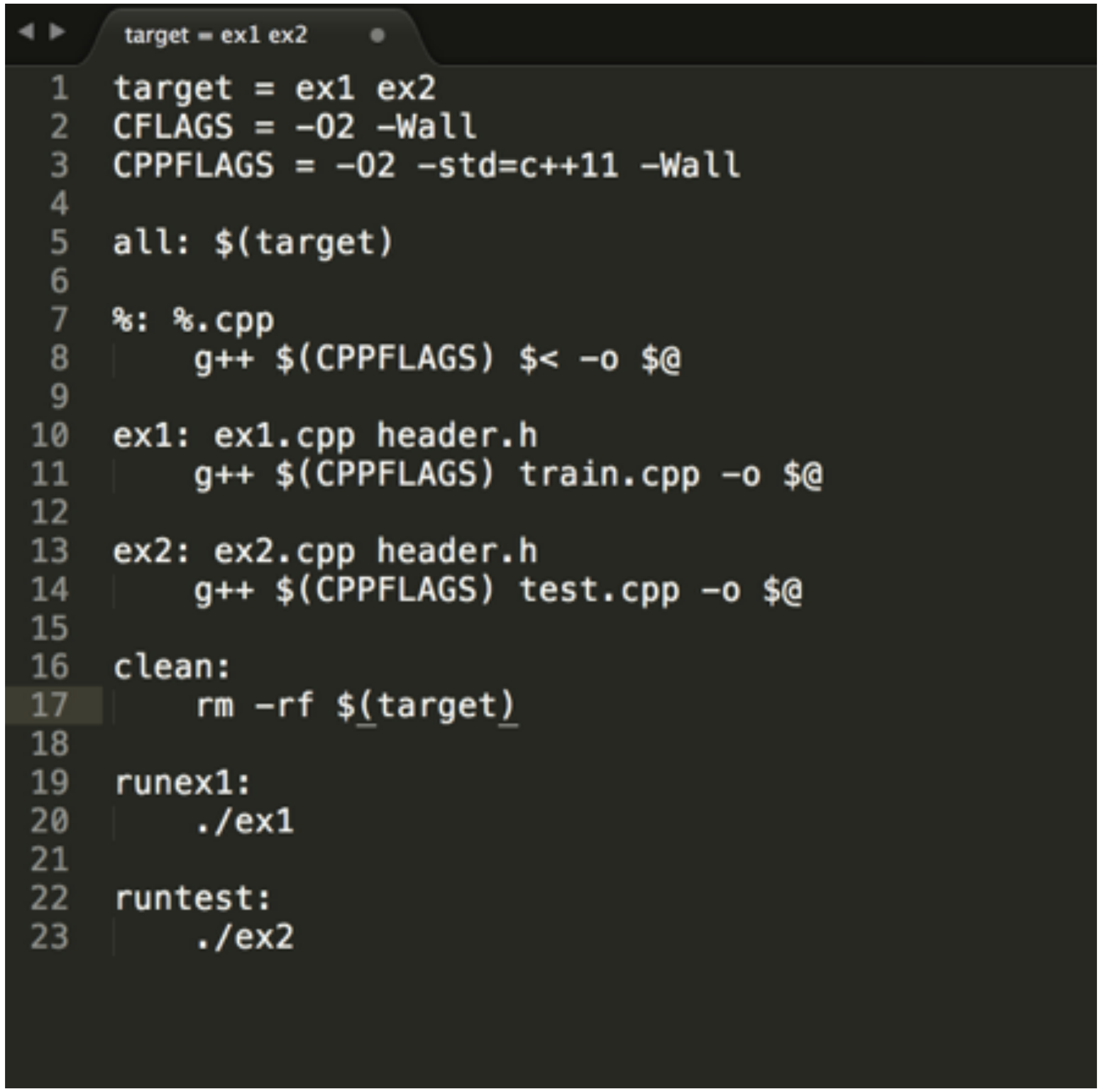


```
Martina-Wei-de-MacBook-Pro:example martina$ g++ -O2 -std=c++11 -Wall main.cpp -o ex1
Martina-Wei-de-MacBook-Pro:example martina$ ./ex1
Hello, World!
Martina-Wei-de-MacBook-Pro:example martina$ 
```

- 如果你同時有很多*.cpp需要編譯，可以使用makefile...

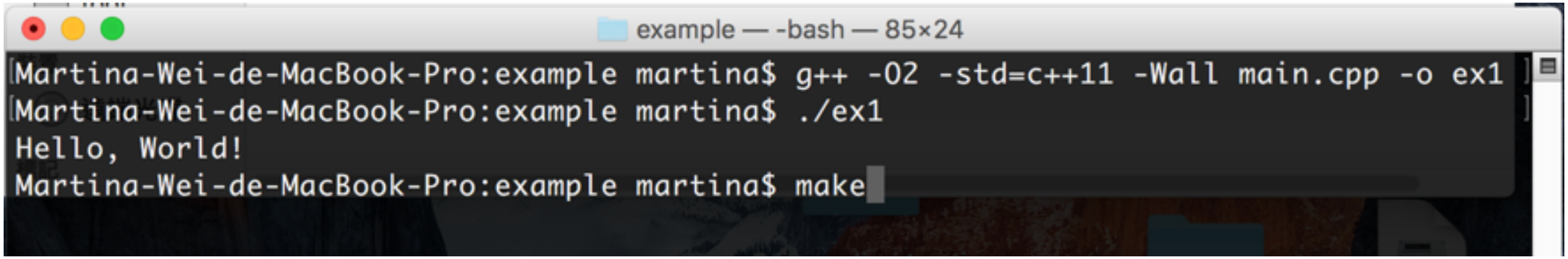
1. 用任何文字編輯器輸入下面的指令，存在cpp在的資料夾中，存成Makefile（沒有副檔名）

2. Makefile 裡，有“：”的下一行，前面必須是”tab”不能是個格空白鍵



```
target = ex1 ex2
CFLAGS = -O2 -Wall
CPPFLAGS = -O2 -std=c++11 -Wall

all: $(target)

%: %.cpp
	g++ $(CPPFLAGS) $< -o $@

ex1: ex1.cpp header.h
	g++ $(CPPFLAGS) train.cpp -o $@

ex2: ex2.cpp header.h
	g++ $(CPPFLAGS) test.cpp -o $@

clean:
	rm -rf $(target)

runex1:
	./ex1

runtest:
	./ex2
```

3. Terminal cd 到Makefile的位置，輸入make，就會依照Makefile進行編譯了



```
Martina-Wei-de-MacBook-Pro:example martina$ g++ -O2 -std=c++11 -Wall main.cpp -o ex1
Martina-Wei-de-MacBook-Pro:example martina$ ./ex1
Hello, World!
Martina-Wei-de-MacBook-Pro:example martina$ make
```

4. run 的方式一樣是在terminal輸入 **./name_of_your_output**