

# 函數呼叫

Function call

by 多拉 A 夢

今天不是要教函數撰寫

~~請輕輕為自己嘆一口氣....~~

今天只要『用（叫）』函數

先想想高中（國中？）數學

先想想高中（國中？）數學

$$f(x) = y = ax^2 + bx + c$$

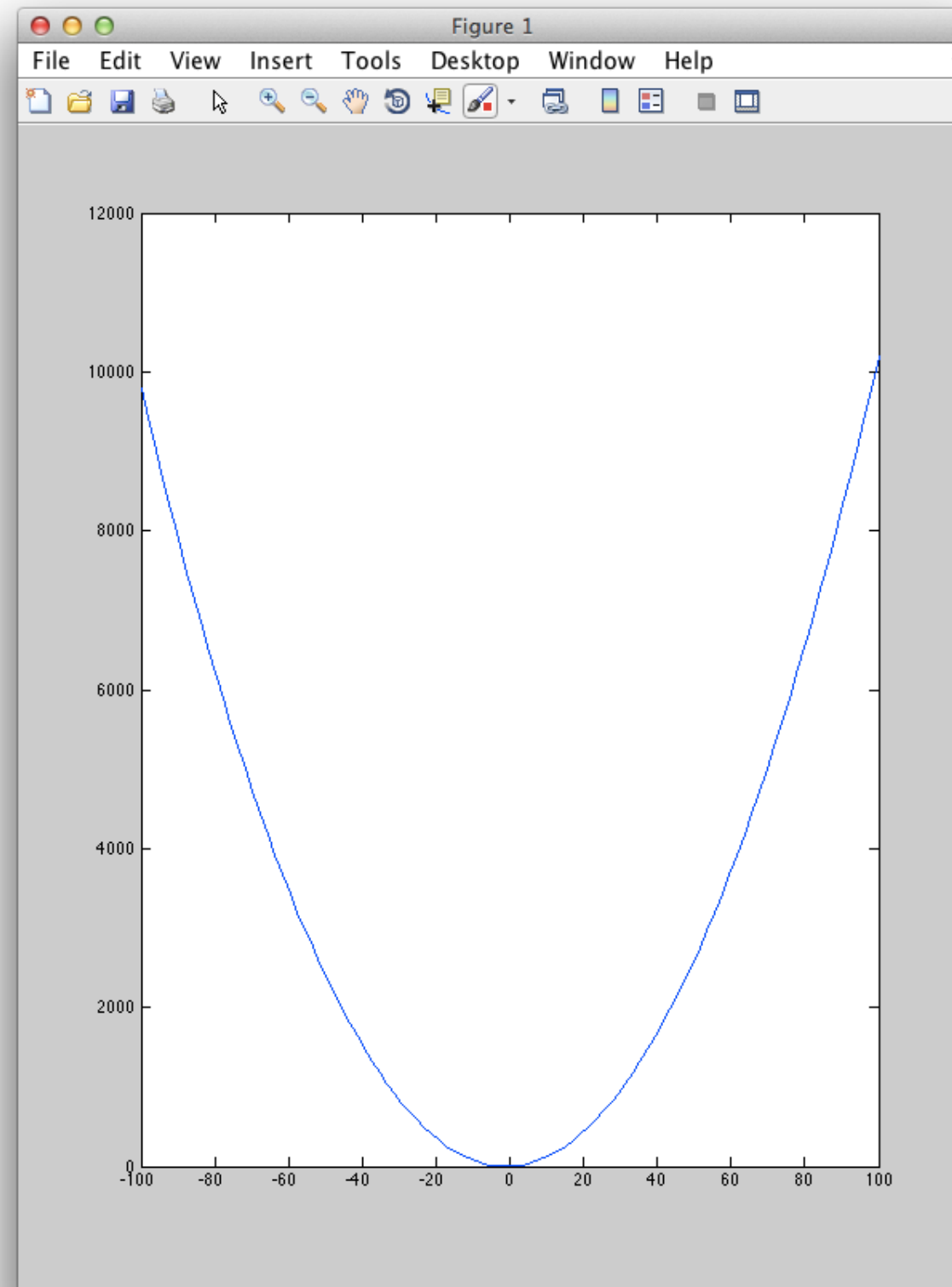
(註：  $ax^2$  讀作『ax平方』)

$$f(x) = y = ax^2 + bx + c$$

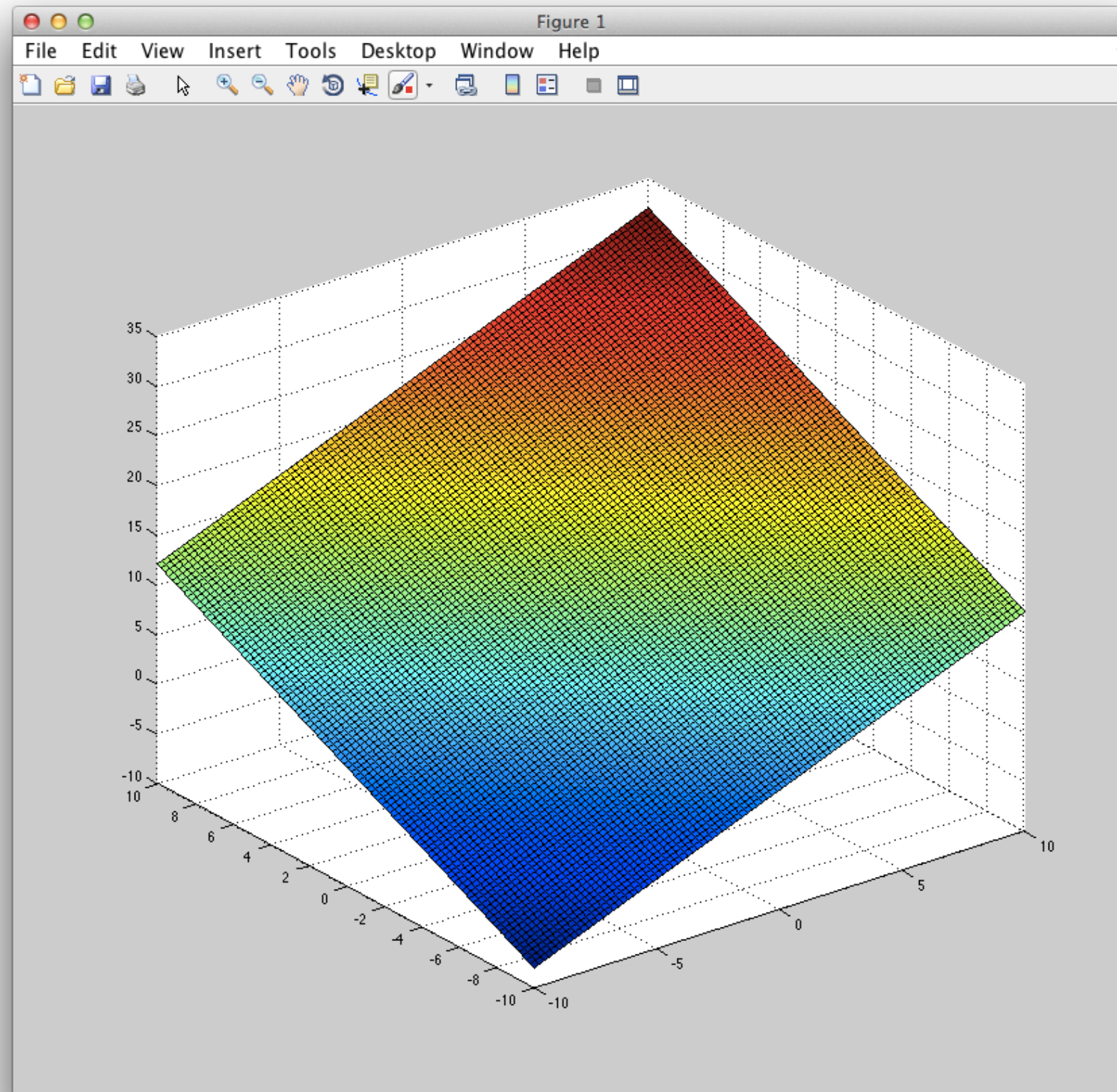
給一個  $x$

我們可以得到一個對應的  $y$

$$f(x) = y = x^2 + 2x + 1$$



$$f(x, y) = z = x + y + 12$$

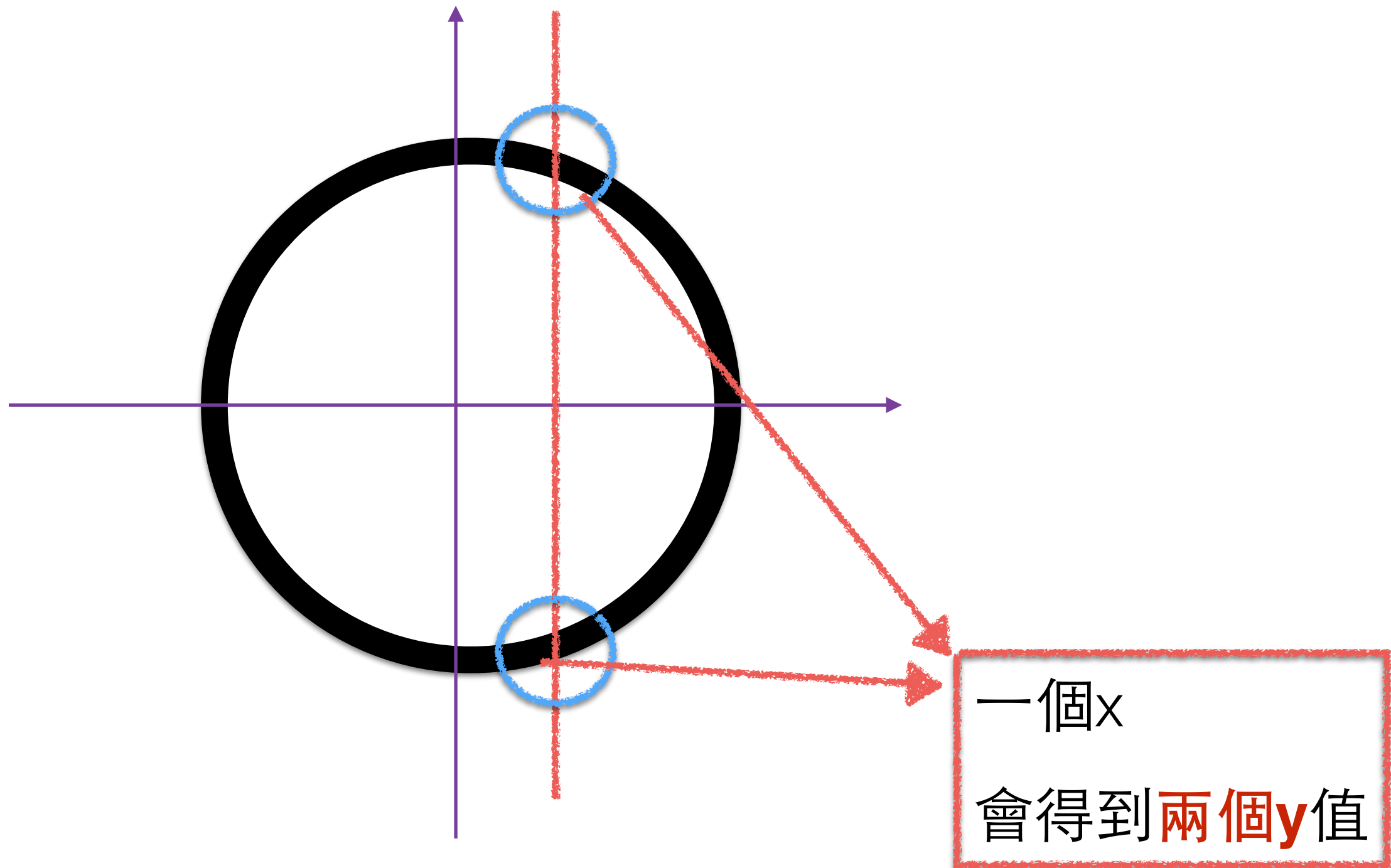




# 函數合法條件

- 一多一：合法，Ex.  $y = x + 2$ 
  - $(x=1, y=3), (x=2, y=4) \dots\dots$
- 多對一：合法
  - $y = x^2 (x = 2, y = 4), (x = -2, y = 4) \dots$
  - $z = y + x + 3 (x=1, y=0, z=4), (x=0, y=1, z=4)$
- 一對多：不合法  $\Rightarrow$  給一個 $x$ ，會得到多個 $y$  ....
  - $x^2 + y^2 = 4$  (圓形) if  $x = 0$ , then  $y = +2$  or  $(-2)$
  - $x^2 + y^2 + z^2 = 16$  (球體)

# 圓形（不是函數）



這裡不是數學補習班...

資訊之芽なのです

What is 函數呼叫？

What is 函數呼叫？

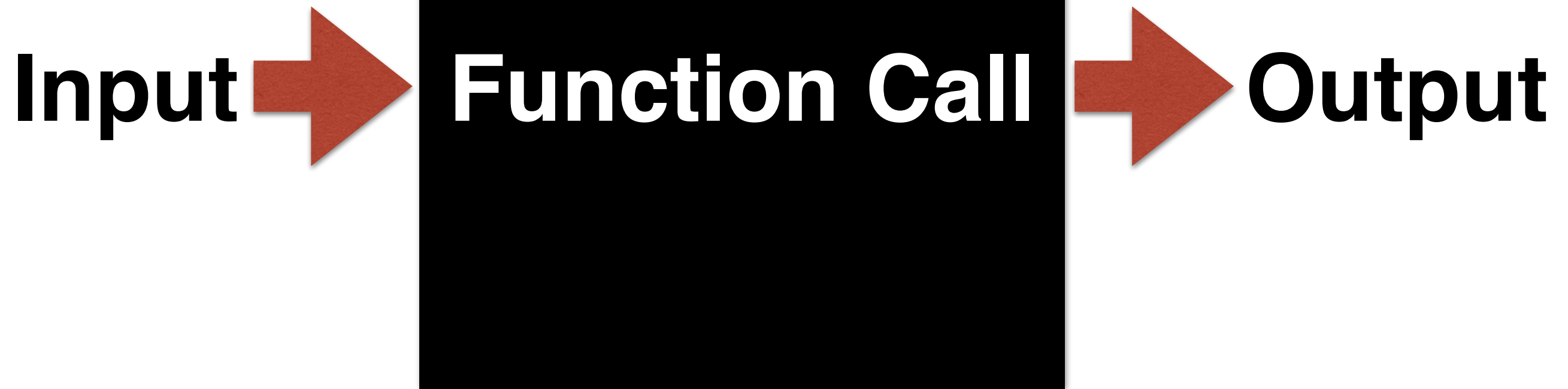
~~function call~~

簡單來說...

丟東西進去.....

會有對應的東西跑出來： D

可以想成一個黑盒子



# 今天要來介紹....

- `#include <cmath>`
- `#include <algorithm>`



# # include <cmath>

- 一個函式庫
- 用來做數學的計算  $\sin(x)$ ,  $\cos(x)$ ,  $\tan(x)$  .....
- 也可以拿來開根號  $\text{sqrt}(x)$  .....
- 請使用 **double** data type , **不要使用int**
- **如果要直接放整數請記得打 .0 , Ex. 30.0, 4.0....**

# 開根號(square root = sqrt)

```
1 #include <iostream>
2 #include <cmath>
3
4 int main(){
5
6     std::cout << sqrt(16.0) << std::endl;
7
8     double x, y;
9     x = 20;
10    y = sqrt(x);
11    std::cout << y << std::endl;
12
13    double a = 5, b = 7, c;
14    c = sqrt( a * a + b * b );
15    std::cout << c << std::endl;
16
17    return 0;
18 }
```

$\sin()$ ,  $\cos()$ ,  $\tan()$ ...

~~小時候最常聽到的一句話~~

~~你去小七買東西會用到三角函數嗎~~

# $\sin$ (這裡要放弧度)

- $\pi = 3.1415926\dots = 180^\circ$
- $\sin(\pi) = \sin(180^\circ)$
- $\sin(\pi * (\text{你想要的角度} / 180))$

# 練習：sin(30度)

```
1 #include <iostream>
2 #include <cmath>
3
4 int main(){
5     double pi = 3.1415926;
6     std::cout << sin( pi * ( 30.0 / 180.0 ) ) << std::endl;
7
8     double x = 30;
9     std::cout << sin( pi * ( x / 180 ) ) << std::endl;
10
11     return 0;
12 }
```

同理  $\cos()$ ,  $\tan()$  .....

# 其他的請自己查：D

The screenshot shows a web browser window displaying the cplusplus.com website. The page is titled "<cmath> (math.h)" and is part of the "Reference" section. The left sidebar contains a navigation menu with links to "C++", "Information", "Tutorials", "Reference", "Articles", and "Forum". The "Reference" section is expanded, showing a list of C library headers, with "<cmath> (math.h)" selected. The main content area displays the header information for "<cmath> (math.h)", including a search bar, a "Not logged in" status, and buttons for "register" and "log in". The page lists various functions under the heading "Trigonometric functions" and "Hyperbolic functions".

Search:  Go

Not logged in  
[register](#) [log in](#)

**C++**

Information  
Tutorials  
Reference  
Articles  
Forum

**Reference**

**C library:**

- <cassert> (assert.h)
- <cctype> (ctype.h)
- <cerrno> (errno.h)
- <cfenv> (fenv.h)
- <cfloat> (float.h)
- <stdint.h> (stdint.h)
- <iso646> (iso646.h)
- <limits> (limits.h)
- <locale> (locale.h)
- <cmath> (math.h)**
- <csignal> (signal.h)
- <stdarg> (stdarg.h)
- <stdbool> (stdbool.h)
- <stddef> (stddef.h)
- <stdint> (stdint.h)
- <stdio> (stdio.h)
- <stdlib> (stdlib.h)
- <string> (string.h)
- <tgmath> (tgmath.h)
- <time> (time.h)
- <uchar> (uchar.h)
- <wchar> (wchar.h)

header  
**<cmath> (math.h)**

**C numerics library**

Header <cmath> declares a set of functions to compute common mathematical operations and transformations:

**Trigonometric functions**

<b>cos</b>	Compute cosine (function )
<b>sin</b>	Compute sine (function )
<b>tan</b>	Compute tangent (function )
<b>acos</b>	Compute arc cosine (function )
<b>asin</b>	Compute arc sine (function )
<b>atan</b>	Compute arc tangent (function )
<b>atan2</b>	Compute arc tangent with two parameters (function )

**Hyperbolic functions**

<b>cosh</b>	Compute hyperbolic cosine (function )
<b>sinh</b>	Compute hyperbolic sine (function )
<b>tanh</b>	Compute hyperbolic tangent (function )
<b>acosh</b> <small>C++11</small>	Compute arc hyperbolic cosine (function )
<b>asinh</b> <small>C++11</small>	Compute arc hyperbolic sine (function )
<b>atanh</b> <small>C++11</small>	Compute arc hyperbolic tangent (function )

**Exponential and logarithmic functions**

# #include <algorithm>

- 也是一個函式庫
- 用來做一些正常的事情
- **max( x , y ) , min( x , y )** , lower\_bound().....
- 跟cmath不一樣，**不限於**double data type



# max(), min()...

- 用法就長這樣：
- 找最大值：`max_number = std::max( x, y );`
- 找最小值：`min_number = std::min( x , y );`

# 練習：max(), min()

```
1 #include <iostream>
2 #include <algorithm>
3
4 int main(){
5
6     int x = 10, y = 20, max_num, min_num;
7     max_num = std::max(x, y);
8     min_num = std::min(x, y);
9
10    std::cout << max_num << ' ' << min_num << std::endl;
11
12    double a = 3.14, b = 9.838, double_max, double_min;
13    double_max = std::max(a, b);
14    double_min = std::min(a, b);
15
16    std::cout << double_max << ' ' << double_min << std::endl;
17
18    return 0;
19 }
```

亂數

亂數

隨機產生的數字

# rand()

- `int x = rand();` // 取一個數字
- `int x = rand() % 6;` // 對隨機取出來的數字作mod

# rand()

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4
5 int main(){
6     std::cout << rand() << std::endl;
7     std::cout << rand() % 6 << std::endl;
8
9     for(int i = 0; i < 5; i++){
10         std::cout << i << " : " << rand() % 6 << std::endl;
11     }
12
13
14     return 0;
15 }
```

# rand()

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4
5 int main(){
6     std::cout << rand() << std::endl;
7     std::cout << rand() % 6 << std::endl;
8
9     for(int i = 0; i < 5; i++){
10         std::cout << i << " : " << rand() % 6 << std::endl;
11     }
12
13
14     return 0;
15 }
```

```
→ ~ ./a.out
1804289383
4
0 : 3
1 : 1
2 : 5
3 : 1
4 : 4
→ ~ ./a.out
1804289383
4
0 : 3
1 : 1
2 : 5
3 : 1
4 : 4
→ ~ ./a.out
1804289383
4
0 : 3
1 : 1
2 : 5
3 : 1
4 : 4
```

rand()

多執行幾次你會發現...



rand()

多執行幾次你會發現...

每次結果都一樣（挖哈哈

rand()

這樣只是假亂數：D

rand()

如何做出真亂數呢？

rand()

如何做出真亂數呢？

impossible

rand()

目前機制不是真亂數

rand()

目前機制不是真亂數


但我們可以做出...

看起來很像真的亂數....

rand()

加入srand(time(0))

rand()

加入  srand(time(0))

 seed 種子



rand()

加入srand(time(0))



把時間當作種子

# rand()

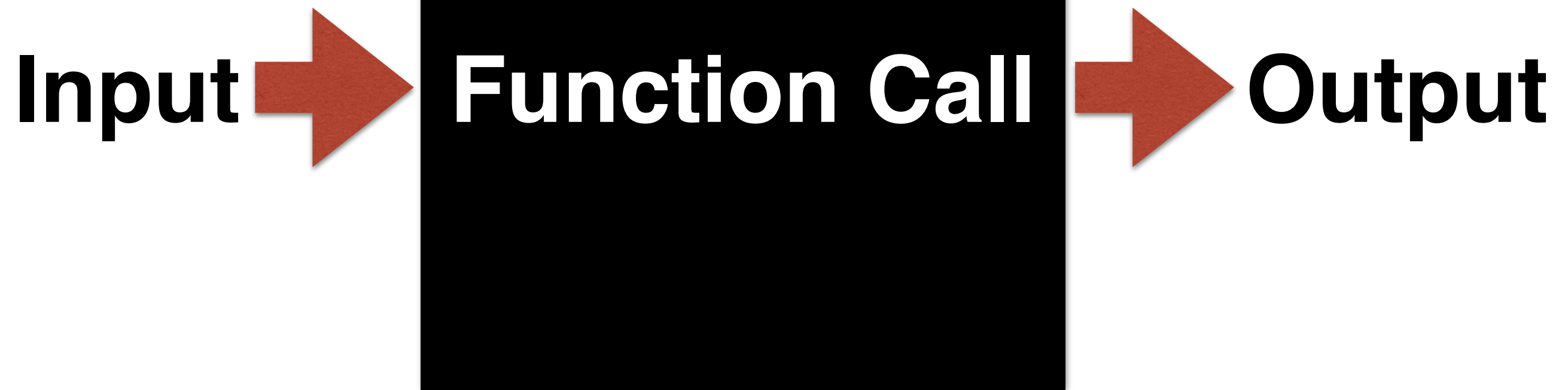
```
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4
5 int main() {
6     srand(time(0));
7
8     std::cout << rand() << std::endl;
9     std::cout << rand() % 6 << std::endl;
10
11     for(int i = 0; i < 5; i++){
12         std::cout << i << " : " << rand() % 6 << std::endl;
13     }
14
15     return 0;
16 }
17 }
```

# rand()

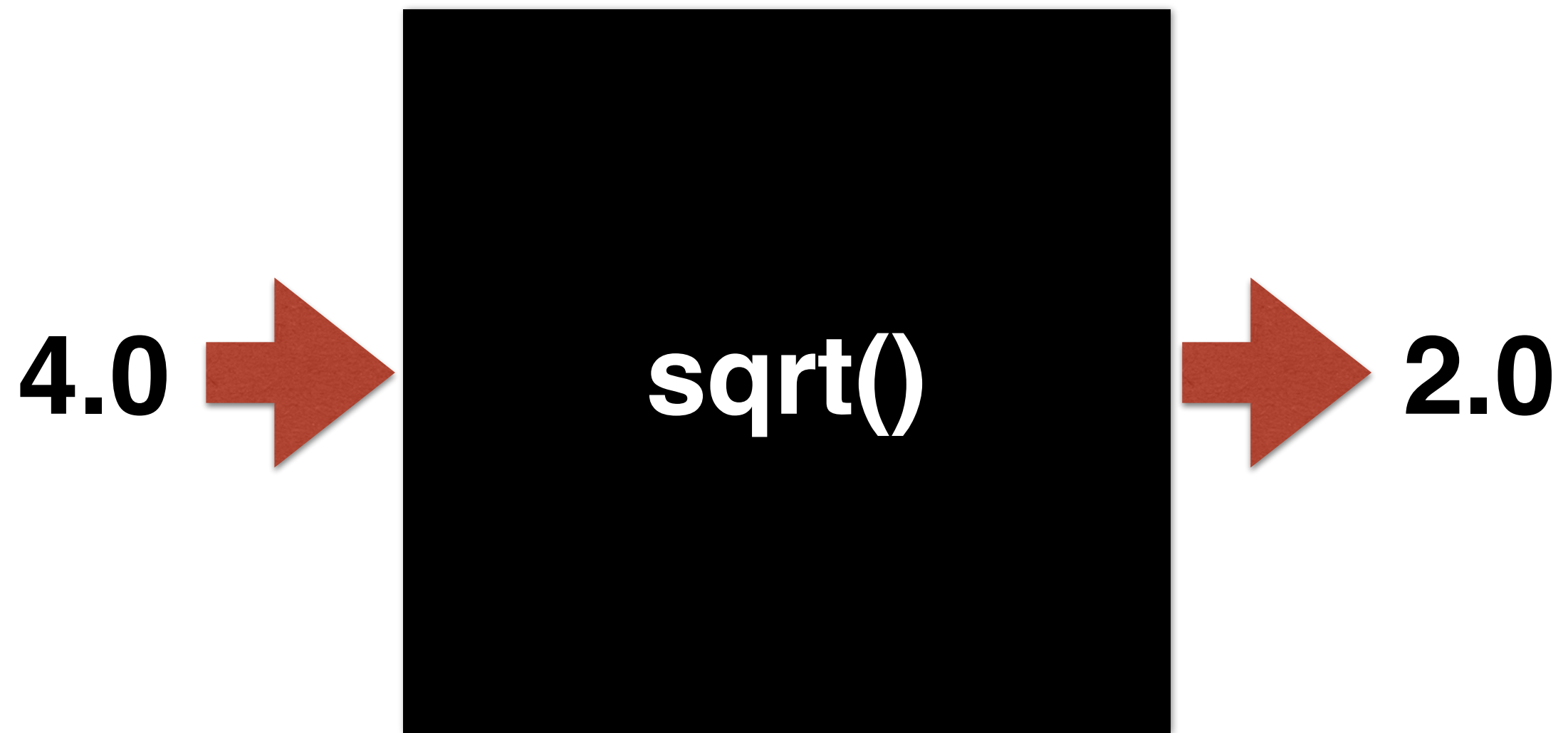
```
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4
5 int main(){
6     srand(time(0));
7
8     std::cout << rand() << std::endl;
9     std::cout << rand() % 6 << std::endl;
10
11     for(int i = 0; i < 5; i++){
12         std::cout << i << " : " << rand() % 6 << std::endl;
13     }
14
15
16     return 0;
17 }
```

```
g++ 1.cpp
~ ./a.out
702950961
3
0 : 3
1 : 0
2 : 0
3 : 2
4 : 3
~ ./a.out
1946655437
2
0 : 3
1 : 4
2 : 5
3 : 2
4 : 0
~ ./a.out
1632637223
3
0 : 3
1 : 5
2 : 0
3 : 1
4 : 4
~ ./a.out
2100544631
4
0 : 2
1 : 1
2 : 1
3 : 1
4 : 0
~
```

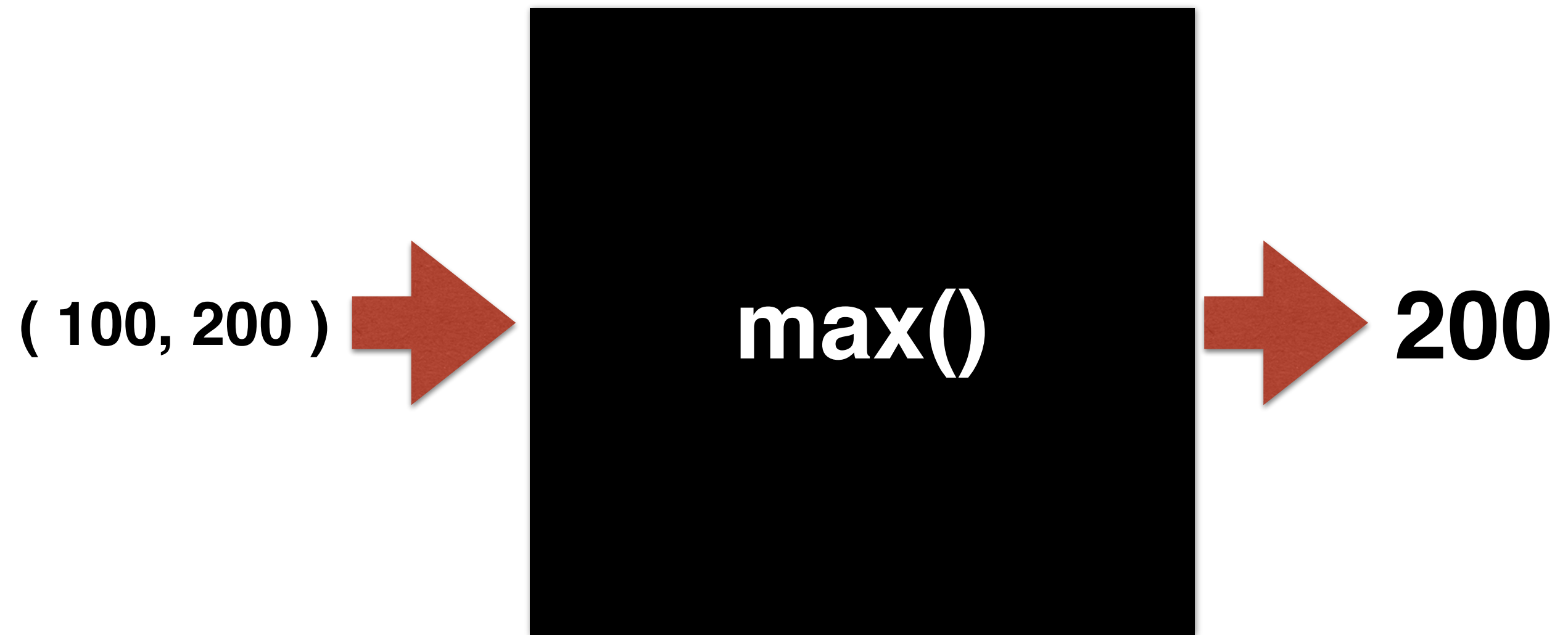
回來看看黑盒子....



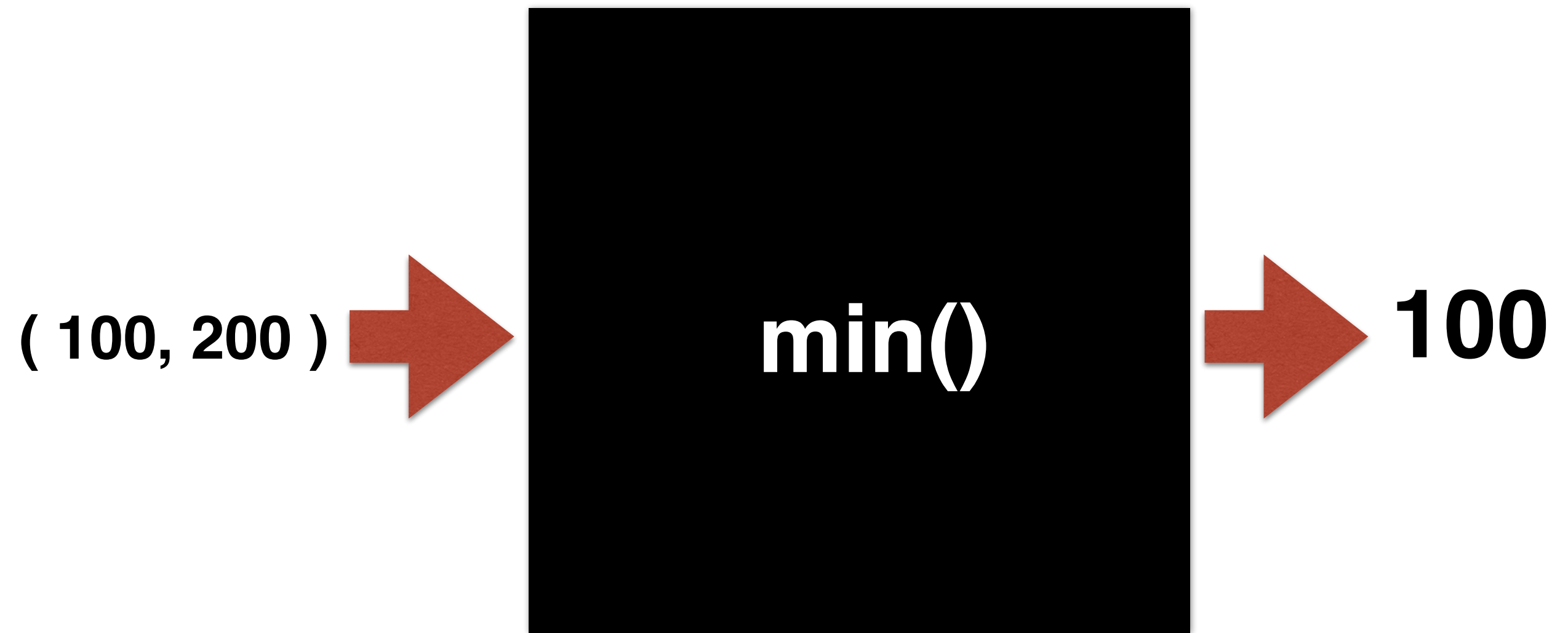
# 回來看看黑盒子....



# 回來看看黑盒子.....



# 回來看看黑盒子.....



下次會教你們.....

如何製作黑人的箱子