



## 2. 輸入輸出 變數

2024資訊研究社語法班  
Made with ❤️ by jheanlee

# 我們的程式怎麼跟我們互動

程式與使用者互動的方式有很多種，最常見的是

- Command-Line Interface 指令列介面  
使用文字輸入輸出 程式最簡單 最不容易壞
- Text-based User Interface 文字使用介面  
使用方向鍵、快捷鍵 具有兩者優點
- Graphical User Interface 圖像化使用介面  
使用圖形 最直覺 最方便使用者

我們這學期教的競程輸入輸出都是使用CLI

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jhean\OneDrive\Desktop\Tools>iperf-3.17.1-win64>iperf3.exe -c 172.16.30.1
Connecting to host 172.16.30.1, port 5201
[ 5] local 172.16.10.3 port 54978 connected to 172.16.30.1 port 5201
[ ID] Interval      Transfer    Bitrate
[ 5] 0.00-1.01 sec  115 MBytes  951 Mbits/sec
[ 5] 1.01-2.01 sec  113 MBytes  949 Mbits/sec
[ 5] 2.01-3.01 sec  114 MBytes  949 Mbits/sec
[ 5] 3.01-4.01 sec  113 MBytes  949 Mbits/sec
[ 5] 4.01-5.00 sec  112 MBytes  950 Mbits/sec
[ 5] 5.00-6.01 sec  114 MBytes  949 Mbits/sec
[ 5] 6.01-7.00 sec  112 MBytes  950 Mbits/sec
[ 5] 7.00-8.01 sec  114 MBytes  949 Mbits/sec
[ 5] 8.01-9.01 sec  113 MBytes  949 Mbits/sec
[ 5] 9.01-10.01 sec 114 MBytes  949 Mbits/sec
-- -- -- -- --
[ ID] Interval      Transfer    Bitrate          sender
[ 5] 0.00-10.01 sec  1.11 GBytes  949 Mbits/sec
[ 5] 0.00-10.02 sec  1.11 GBytes  948 Mbits/sec          receiver

iperf Done.
```

↑ CLI工具 (iperf)

TUI介面 (TrueNAS Setup) ↓





# 怎麼讓程式輸入輸出

CLI的輸入/輸出可以稱為 `stdin` / `stdout`

C++中 最簡單的有下面兩組

C Style `<cstdio>` or `<stdio.h>`:

輸入 `scanf()` 輸出 `printf()`

C++ Style `<iostream>`:

輸入 `std::cin` 輸出 `std::cout`

因為`std::cin` `std::cout`比較直覺 所以我們這學期教這組

# cin cout

cin\_cout.cpp

std::cin 會把輸入的值存入變數中（後面會講）

用 >> 連接 cin和存入的變數 很多個變數可以連續用 >> 串起來

std::cout 和cin相反 會將後方的值輸出到螢幕上

用 << 連接 cout 和資料 很多個資料可以用 << 串起來

'\n' 是換行字元

std::endl 也是換行字元 都可以用

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main() {
6      /*-----程式開始-----*/
7      /*-----變數宣告-----*/
8      int a;
9      string b;
10     float c;
11
12     /*-----輸入-----*/
13     cin >> a >> b >> c; //cin會依序把資料存入 a, b, c 中
14
15     /*-----輸出-----*/
16     cout << b << a << c << endl; //cout會依序輸出 b, a, c 的資料到畫面上
17     cout << "hi" << '\n' << "bye"; // endl = '\n' = 換行
18
19     /*-----程式結束-----*/
20     return 0;
21 }
```



# cin與輸入

cin 怎麼知道在哪裡分段呢

大原則：空白，換行字元(就是 Enter 鍵)  
之後再考慮資料類型 (後面會講)

例如：

```
int a, b, c; // a, b, c 都只能是整數  
cin >> a >> b >> c;
```

輸入為

```
1845611 -17  <- 這裡換行  
-1
```

則 `a = 1845611; b = -17; c = -1`



## 資料型態 (bool, int)

bool - 布林值 (Boolean) 只有是 與 非

是: true, 1

非: false, 0

int - 整數 (integer) 為  $2^{31} - 1$  (2147483647) 到  $-2^{31}$  (-2147483648) 的整數

# 變數

var\_operators.cpp

C++中宣告變數的格式是

`type name;` //這樣name會是空的, 沒有資料

`type name = value;` //這樣name的資料是value

```
a: 0
b: 0
c: 1
d: 1
e: 0
f: 1
g: -1
h: -2147483648
```

宣告一個變數的方式:

```
type    name    = value;
bool some_bool = true;
```

```
bool a = false;
```

```
bool b = 0;
```

```
bool c = true;
```

```
bool d = 1;
```

```
cout << "a: " << a << endl;
```

```
cout << "b: " << b << endl;
```

```
cout << "c: " << c << endl;
```

```
cout << "d: " << d << endl;
```

```
int e = 0;
```

```
int f = 1;
```

```
int g = -1;
```

```
int h = -2147483648;
```

```
cout << "e: " << e << endl;
```

```
cout << "f: " << f << endl;
```

```
cout << "g: " << g << endl;
```

```
cout << "h: " << h << endl;
```

# 基本運算子

+ - \* / 就是 加減乘除

a % b 是 a / b 的餘數 (% 念作 mod)

a++ / a-- 是把a這個變數的值 +1 / -1

前面有用到的 =

用法是 var1 = var2; 意思是 把 var1 變成 var2 的值

i = 100;

j = 101;

i = j; // i == 101, j == 101

```
-----分隔線-----
i: 100
i + 2: 102
i - 2: 98
i * 2: 200
i / 2: 50
i % 2: 0
i after i++: 101
i % 2 after i++: 1
-----分隔線-----
j: 10
k: 11
j after j = k: 11
k after j = k: 11
```

```
28 cout << "-----分隔線-----" << endl;
29
30
31 int i = 100;
32 cout << "i: " << i << endl;
33 cout << "i + 2: " << i + 2 << endl;
34 cout << "i - 2: " << i - 2 << endl;
35 cout << "i * 2: " << i * 2 << endl;
36 cout << "i / 2: " << i / 2 << endl;
37 cout << "i % 2: " << i % 2 << endl;
38 i++;
39 cout << "i after i++: " << i << endl;
40 cout << "i % 2 after i++: " << i % 2 << endl;
41
42 cout << "-----分隔線-----" << endl;
43
44 int j = 10;
45 int k = 11;
46 cout << "j: " << j << endl;
47 cout << "k: " << k << endl;
48 j = k;
49 cout << "j after j = k: " << j << endl;
50 cout << "k after j = k: " << k << endl;
```



# 基本運算子

在 `=` 前面放 `+` `-` `*` `/` `%` 會直接更改他的值

`l = 100;`

`l += 1;` //這行跟 `l = l + 1` 是一樣的效果

則 `l == 101`

```
53  int l = 100;
54  int m = 100;
55  cout << "l: " << l << endl;
56  cout << "m: " << m << endl;
57  l += 1;
58  m = m + 1;
59  cout << "l after l += 1: " << l << endl;
60  cout << "m after m = m + 1: " << m << endl;
61  return 0;
62  }
63
```

```
l: 100
m: 100
l after l += 1: 101
m after m = m + 1: 101
```