



05 字元與字串

Fundamental Computer Programming- C++ Lab(I)

元智大學 | C++ 程式設計實習

張維元

課程投影片：[請從元智個人 Portal 下載](#)

C++ 程式設計實習

- 01 程式設計與開發環境
- 02 變數型態與運算
- 03 流程控制
- 04 陣列與向量
- 05 字元與字串
- 06 函數與遞迴函數
- 07 指標與引用
- 08 循序檔案和隨機檔案

2015



維元

(@v123582)

Web Development

Data Science

#遠端 #斜槓 #教學
#資料科學 #網站開發

擅長網站開發與資料科學的雙棲工程師，熟悉的語言是 Python 跟 JavaScript。同時也是 資料科學家的工作日常 粉專及 資料科學家的 12 堂心法課 發起人，擁有多次國內大型技術會議講者經驗，持續在不同的平台發表對 #資料科學、#網頁開發 或 #軟體職涯 相關的分享。

- 元智大學 C++/CPE 程式設計課程 講師
- ALPHACamp 全端 Web 開發 / Leetcode Camp 課程講師
- CUPOY Python 網路爬蟲實戰研習馬拉松 發起人
- 中華電信學院 資料驅動系列課程 講師
- 工研院 Python AI 人工智慧資料分析師養成班 講師
- 華岡興業基金會 AI/Big Data 技能養成班系列課程 講師

site: v123582.tw / line: weiwei63

mail: weiyuan@saturn.yzu.edu.tw

2015



維元

(@v123582)

Web Development

Data Science

#遠端 #斜槓 #教學
#資料科學 #網站開發

擅長網站開發與資料科學的雙棲工程師，熟悉的語言是 Python 跟 JavaScript。同時也是 資料科學家的工作日常 粉專及 資料科學家的 12 堂心法課 發起人，擁有多次國內大型技術會議講者經驗，持續在不同的平台發表對 #資料科學、#網頁開發 或 #軟體職涯 相關的分享。

- 2018 總統盃黑客松 冠軍隊伍
- 2017 資料科學愛好者年會(DSCONF) 講師
- 2017 行動科技年會(MOPCON) 講師
- 2016 微軟 Imagine Cup 台灣區冠軍

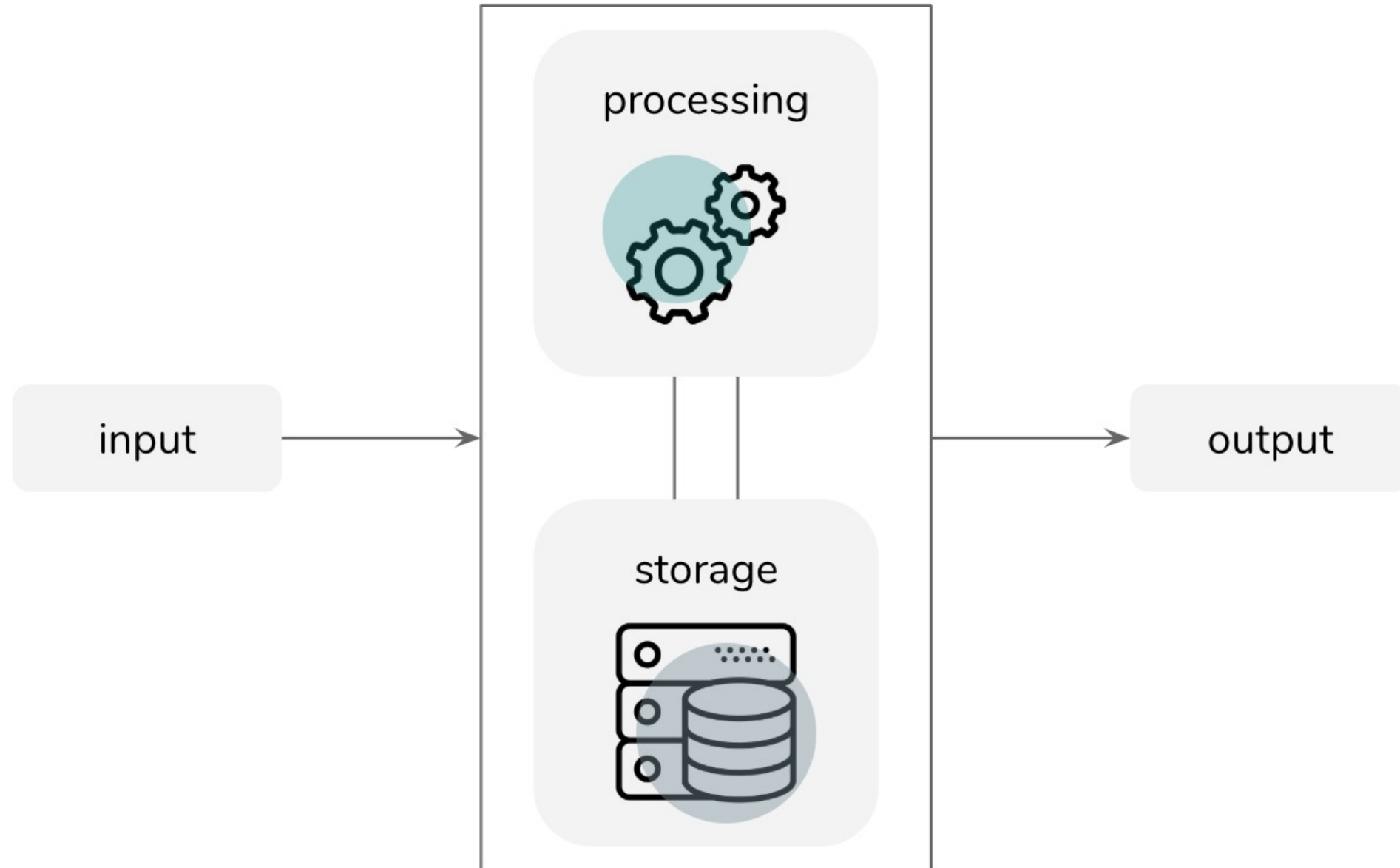
site: v123582.tw / line: weiwei63
mail: weiyuan@saturn.yzu.edu.tw

什麼是程式？

程式語言是用來命令電腦執行各種作業的工具，是人與電腦溝通的橋樑。當電腦藉由輸入設備把程式讀入後，會儲存在主記憶體內，然後指令會依序被控制單元提取並解碼或翻譯成電腦可以執行的信號，並把信號送到各個裝置上，以執行指令所指派的動作。也就是說，人類與電腦溝通的語言稱為程式語言。

程式 = 利用一系列的指令告訴電腦如何執行工作

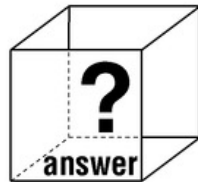




變數 (Variables) 與常數

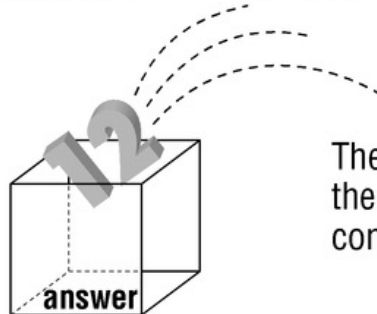
C++在使用變數之前，必須先告訴電腦「我要使用變數」，電腦會幫我們準備記憶空間儲存該變數。這件事稱之為變數宣告。

A `int answer;`



The variable `answer` has not been assigned a value. So we put a “?” in it to indicate that it’s in an unknown state.

B `answer = (1+2) * 4;`



The variable `answer` is assigned the value of the expression $(1+2) * 4$. The box is shown containing the value 12.

變數的型態

型態	中文意思	英文字義	可儲存的資料
int	整數	Integer	100 、 -5 、 1246 ...
float	浮點數(小數)	floating point	3.14159 、 4.3 、 -1.1 ...
char	字元(半形字)	Character	'a' 、 'R' 、 '1' 、 '@' 、 '*' ...
string	字串(文句)	String	"Hello" 、 "^_^" 、 "Rock!" ...
bool	布林(是非)	boolean	true 、 false

字元

型態	中文意思	英文字義	可儲存的資料
char	字元(半形字)	Character	'a' 、 'R' 、 '1' 、 '@' 、 '*' ...

```
1 #include<iostream>
2 using namespace std;
3
4 int main(){
5     char a = 'a';
6
7     cout << a << endl;
8     cout << sizeof(a) << endl;
9     cout << int(a) << endl;
10    cout << (int)a << endl;
11 }
```

字元

型態	中文意思	英文字義	可儲存的資料
char	字元(半形字)	Character	'a' 、 'R' 、 '1' 、 '@' 、 '*' ...

```
1 #include<iostream>
2 using namespace std;
3
4 int main(){
5     char a = 'a';
6
7     cout << a << endl;
8     cout << sizeof(a) << endl;
9     cout << int(a) << endl;
10    cout << (int)a << endl;
11 }
```

→ 在 C++ 當中，每一個字元是 1 Byte

#動手實作

■#練習：想一下，為什麼 C++ 的字元只需要 1 Byte？

型態	中文意思	英文字義	可儲存的資料
int	整數	Integer	100 、 -5 、 1246 ...
float	浮點數(小數)	floating point	3.14159 、 4.3 、 -1.1 ...
char	字元(半形字)	Character	'a' 、 'R' 、 '1' 、 '@' 、 '*' ...
string	字串(文句)	String	"Hello" 、 "^_^" 、 "Rock!" ...
bool	布林(是非)	boolean	true 、 false

→ 4 Bytes

→ 4 Bytes

→ 1 Bytes

→ 32 Bytes

→ 1 Bytes

ASCII

Ctrl	Dec	Hex	Char	Code	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
^@	0	00		NUL	32	20	!	64	40	@	96	60	'
^A	1	01		SOH	33	21	!	65	41	A	97	61	a
^B	2	02		STX	34	22	"	66	42	B	98	62	b
^C	3	03		ETX	35	23	#	67	43	C	99	63	c
^D	4	04		EOT	36	24	\$	68	44	D	100	64	d
^E	5	05		ENQ	37	25	%	69	45	E	101	65	e
^F	6	06		ACK	38	26	&	70	46	F	102	66	f
^G	7	07		BEL	39	27	'	71	47	G	103	67	g
^H	8	08		BS	40	28	(72	48	H	104	68	h
^I	9	09		HT	41	29)	73	49	I	105	69	i
^J	10	0A		LF	42	2A	*	74	4A	J	106	6A	j
^K	11	0B		VT	43	2B	+	75	4B	K	107	6B	k
^L	12	0C		FF	44	2C	,	76	4C	L	108	6C	l
^M	13	0D		CR	45	2D	-	77	4D	M	109	6D	m
^N	14	0E		SO	46	2E	.	78	4E	N	110	6E	n
^O	15	0F		SI	47	2F	/	79	4F	O	111	6F	o
^P	16	10		DLE	48	30	0	80	50	P	112	70	p
^Q	17	11		DC1	49	31	1	81	51	Q	113	71	q
^R	18	12		DC2	50	32	2	82	52	R	114	72	r
^S	19	13		DC3	51	33	3	83	53	S	115	73	s
^T	20	14		DC4	52	34	4	84	54	T	116	74	t
^U	21	15		NAK	53	35	5	85	55	U	117	75	u
^V	22	16		SYN	54	36	6	86	56	V	118	76	v
^W	23	17		ETB	55	37	7	87	57	W	119	77	w
^X	24	18		CAN	56	38	8	88	58	X	120	78	x
^Y	25	19		EM	57	39	9	89	59	Y	121	79	y
^Z	26	1A		SUB	58	3A	:	90	5A	Z	122	7A	z
^[27	1B		ESC	59	3B	;	91	5B	[123	7B	{
^\	28	1C		FS	60	3C	<	92	5C	\	124	7C	
^]	29	1D		GS	61	3D	=	93	5D]	125	7D	}
^^	30	1E	▲	RS	62	3E	>	94	5E	~	126	7E	~
^-	31	1F	▼	US	63	3F	?	95	5F	_	127	7F	Δ*

■ 半形字

■ 控制字元/跳脫字元

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5
6      for(int i = 0; i <= 126 ;i++){
7          cout << i << " " << (char)i << endl;
8      }
9
10     return 0;
11 }
```

跳脫字元

(Escape Character)

字元	用法
\'	單引號
\"	雙引號
\\	反斜線
\a	嗶聲字元
\b	倒退鍵
\t	跳格/定位字元
\r	回歸鍵
\n	換行符號
\0	空字元

字串 (String)

字串程式語言中表示文字的資料型別。通常以串的整體作為運算物件，如：在串中尋找某個子串、求取一個子串、在串的某個位置上插入一個子串以及刪除一個子串等。兩個字串相等的充要條件是：長度相等，並且各個對應位置上的字元都相等。

index:	“ H e l l o W o r l d ”											長度 = 11
	0	1	2	3	4	5	6	7	8	9	10	= 長度 - 1

字元陣列

i = 0	i = 1	i = 2	i = 3	i = 4	
H	e	l	l	o	\0

```
1 #include<iostream>
2 #include<string.h>
3 using namespace std;
4
5 int main(){
6
7     char c[] = {'H', 'e', 'l', 'l', 'o', '\0'};
8     // char c[] = "Hello";
9
10    cout << c << endl;
11    cout << strlen(c) << endl;
12    cout << sizeof(c) << endl;
13
14 }
```


cin.get()

```
1 #include<iostream>
2 using namespace std;
3 int main(){
4     char c;
5     while(cin.get(c)){
6         cout << "得到字元:" << c;
7         cout << " , ASCII = " << (int)c << endl;
8     }
9 }
10
11
```

cin.get()

```
1 #include<iostream>
2 using namespace std;
3
4 int main(){
5     char c;
6     while(cin.get(c)){
7         cout << "得到字元:" << c;
8         cout << ", ASCII = " << (int)c << endl;
9     }
10 }
11
```

→ cin.get() 的用法是一個一個字元取，直到換行 (/n) 結束

cin.getline()

```
1 #include<iostream>
2 using namespace std;
3 int main(){
4     char s[100];
5     while(cin.getline(s, 100)){
6         cout << "得到字符串:" << s << endl;
7     }
8 }
9
10
11
```

cin.getline()

```
1 #include<iostream>
2 using namespace std;
3
4 int main(){
5     char s[100];
6     while(cin.getline(s, 100)){
7         cout << "得到字串:" << s << endl;
8     }
9 }
10
11
```

→ cin.getline() 的用法是一次取一個字串，用換行（/n）做結束

字串

型態	中文意思	英文字義	可儲存的資料
string	字串(文句)	String	"Hello" 、 "^_^" 、 "Rock!" ...

```
1 #include<iostream>
2 using namespace std;
3
4 int main(){
5
6     string s = "Hello World";
7     cout << s << endl;
8     cout << s.length() << endl;
9 }
10
11
```

字元陣列 vs. 字串

```
1 #include<iostream>
2 using namespace std;
3
4 int main(){
5
6     char c[] = {'H', 'e', 'l', 'l', 'o', '\0'};
7     cout << c << endl;
8     cout << strlen(c) << endl;
9
10 }
11
12
```

```
1 #include<iostream>
2 using namespace std;
3
4 int main(){
5
6     string s = "Hello World";
7     cout << s << endl;
8     cout << s.length() << endl;
9
10 }
11
```

字元陣列 vs. 字串

```
1 #include<iostream>
2 #include<string.h>
3 using namespace std;
4
5 int main(){
6
7     char c[] = {'H', 'e', 'l', 'l', 'o', '\0'};
8     for(int i = 0; i < strlen(c); i++)
9         cout << c[i] << "\t" << int(c[i]) << endl;
10
11 }
12
```

```
1
2 #include<iostream>
3 using namespace std;
4
5 int main(){
6
7     string s = "Hello World";
8     for(int i = 0; i < s.length(); i++)
9         cout << s[i] << "\t" << int(c[i]) << endl;
10
11 }
12
```


Thanks for listening.

元智大學 | C++ 程式設計實習

Wei-Yuan Chang