C++程式設計

資料型態

運算式

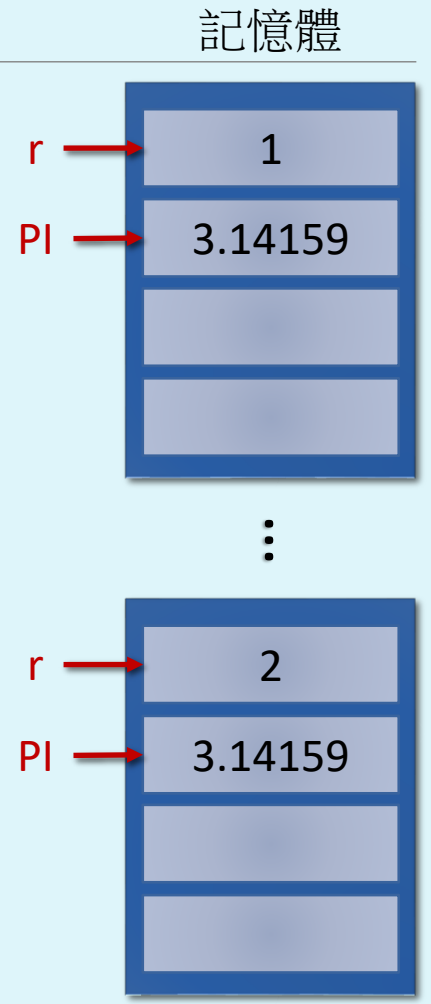
資料型態

變數與常數

- 變數: 內容會隨程式流程而改變
- 常數: 內容固定不變

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int r;
7      const float PI = 3.14159;
8
9      r = 1;
10     cout << "半徑 = " << r << ", 圓面積 = " << PI * r * r << endl;
11     r = 2;
12     cout << "半徑 = " << r << ", 圓面積 = " << PI * r * r << endl;
13     return 0;
14 }
15
```

```
半徑 = 1, 圓面積 = 3.14159
半徑 = 2, 圓面積 = 12.5664
```



資料型態

字面常數

種類	範例	說明
數值	3.14159 6.02e23	float x = 3.14159; $6.02*10^{23}$
字元	‘z’	char ch = ‘z’;
字串	“Hello world”	string s = “Hello world”;

常數以關鍵字const宣告

語法	範例
const 資料型別 常數名稱 = 常數值	const float PI = 3.14159; const char tab = ‘\t’;

資料型態

型別名稱	中譯名稱	記憶體空間（位元組）	表示範圍及涵意
bool	布林	1	0（false）或者 1（true）
char	字元	1	0 ~ 255（共256個字元）
int	整數	4	-2147483648 ~ 2147483647
short int	短整數	2	-32768 ~ 32767
long int	長整數	4	-2147483648 ~ 2147483647[註1]
unsigned int	無號整數	4	0 ~ 4294967295
unsigned short	無號短整數	2	0 ~ 65535
unsigned long	無號長整數	4	0 ~ 4294967295
float	單精度浮點數	4	1.175e-38 ~ 3.402e38[註2]
double	倍精度浮點數	8	2.225e-308 ~ 1.7976e308

1位元組 = 8位元

4位元組 = 32位元
2^{32} = 4294967296
$-2^{31} \sim 2^{31}-1$
-2147483648 ~ 2147483647

在支援64位元CPU的系統(LP64)中, long int長整數為8位元組
Windows系統中, long long int為8位元組

二進制系統

1個位元

2^1
0
1

2個位元

2^2	
0	0
0	1
1	0
1	1

3個位元

2^3		
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

n個位元→2^n個狀態

資料型態

```cpp
#include <iostream>
using namespace std;

int main()
{
    unsigned int i;
    unsigned short int j;
    float a;
    double b;

    cout << "整數 " << sizeof(int) << " 位元組" << endl;
    cout << "長整數 " << sizeof(long int) << " 位元組" << endl;
    cout << "無號整數 " << sizeof(i) << " 位元組" << endl;
    cout << "無號短整數 " << sizeof(j) << " 位元組" << endl;
    cout << "浮點數 " << sizeof(a) << " 位元組" << endl;
    cout << "倍度浮點數 " << sizeof(b) << " 位元組" << endl;
    return 0;
}
```

```
整數 4 位元組
長整數 4 位元組
無號整數 4 位元組
無號短整數 2 位元組
浮點數 4 位元組
倍度浮點數 8 位元組
```

整數型態

```cpp
#include <iostream>
using namespace std;

int main()
{
    int x = 2147483647;

    cout << "x = " << x << endl;
    cout << "x + 1 = " << x+1 << endl;
    cout << "x + 2 = " << x+2 << endl;
    return 0;
}
```

```
x = 2147483647
x + 1 = -2147483648
x + 2 = -2147483647
```

int 表示範圍

-2147483648	-2147483647	...	0	...	2147483646	2147483647

字元型態

```cpp
#include <iostream>
using namespace std;

int main()
{
    char ch = 'a';
    int x = ch;
    cout << "ch = " << ch << endl;
    cout << "x = " << x << endl;
    return 0;
}
```

```
ch = a
x = 97
```

```cpp
#include <iostream>
using namespace std;

int main()
{
    char ch = 97;

    cout << "ch = " << ch << endl;
    return 0;
}
```

```
ch = a
```

二進位	十進位	十六進位	圖形
0110 0000	96	60	`
0110 0001	97	61	a
0110 0010	98	62	b
0110 0011	99	63	c
0110 0100	100	64	d
0110 0101	101	65	e
0110 0110	102	66	f
0110 0111	103	67	g
0110 1000	104	68	h

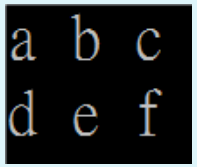
ASCII內碼表

字元型態輸出→字元（圖形）
整數型態輸出→整數（內碼）

字元型態

簡易字元加密

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      char p1,p2,p3;
6      char c1,c2,c3;
7      cin >> p1 >> p2 >> p3;
8      c1 = p1 + 3;
9      c2 = p2 + 3;
10     c3 = p3 + 3;
11     cout << c1 << ' ' << c2 << ' ' << c3 << endl;
12 }
```



小寫轉成大寫?

字元型態

跳脫序列

跳脫序列	說明
\a	警告音
\b	倒退一格
\n	換行
\r	歸位
\t	跳格
\0	字串結束字元
\\	反斜線
\’	單引號
\”	雙引號

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5    char beep = '\a';
6    string s = "kshs";
7    char bs = '\b';
8    char ch = '\"';
9
10   cout << beep << endl;
11   cout << s << bs << '*' << endl;
12   cout << "Hello World!" << endl;
13   cout << ch << "Hello World!" << ch << endl;
14 }
15
```

```
ksh*
Hello World!
"Hello World!"
```

浮點數型態

資料型態	中文名稱	記憶體空間	表示範圍
float	單精度浮點數	4位元組	1.2e-38 ~ 3.4e38
double	倍精度浮點數	8位元組	2.2e-308 ~ 1.8e308

```
float num1 = 2.586e11;
float num2 = -3.45e55;
double num3 = 3.6e33;
double num4 = -5.14E88;
```

// 錯誤,超出範圍

// 正確,也可使用大寫E表示

布林型態

資料型態	中文名稱	記憶體空間	表示範圍
bool	布林	1位元組	0 (false) 或 1 (true)

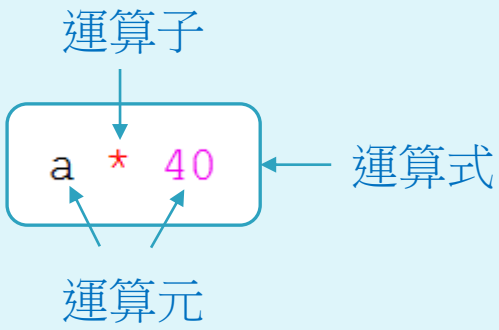
```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5    bool flag = false;
6    cout << "flag = " << flag << endl;
7    flag = 1;
8    cout << "flag = " << flag << endl;
9  }
10
```

```
flag = 0
flag = 1
```

運算式

運算式由運算元與運算子組成

- 運算子: + 、 - 、 * 、 /
- 運算元: 變數、常數、函數



指定運算子

指定運算子	說明
=	指定

```
1   #include<iostream>
2   using namespace std;
3   int main()
4   {
5       int x = 10;
6       int y = 5;
7       int z;
8       z = x + y;
9       x = x + 1;
10      cout << "x = " << x << endl;
11      cout << "y = " << y << endl;
12      cout << "z = " << z << endl;
13  }
14
```

固定為一個變數 = 變數 常數 運算式

```
x = 11
y = 5
z = 15
```

一元運算子

一元運算子	說明
+	正號
-	負號
!	NOT

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5     int a = 10;
6     bool b = false;
7     cout << "-a = " << -a << endl;
8     cout << "!b = " << !b << endl;
9  }
10
```

```
-a = -10
!b = 1
```

算術運算子

算術運算子	說明
+	加法
-	減法
*	乘法
/	除法
%	取餘數

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5     int a = 25;
6     int b = 4;
7     cout << "a / b = " << a / b << endl;
8     cout << "a % b = " << a % b << endl;
9     cout << "a / b = " << (float) a / b << endl;
10    return 0;
11 }
12
```

型別轉換

```
a / b = 6
a % b = 1
a / b = 6.25
```

關係運算子

if 敘述

- 格式:

```
if (條件判斷)
   敘述;
```

- 說明: 若條件判斷成立,則敘述被執行; 若不成立則不執行

關係運算子	說明
>	大於
<	小於
>=	大於等於
<=	小於等於
==	等於
!=	不等於

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5     int a;
6     cin >> a;
7     if ( a > 5)
8        cout << a << " > 5 成立" << endl;
9     if ( a % 2 == 0)
10       cout << a << "是偶數" << endl;
11    if (true)
12       cout << "此行永遠被執行" << endl;
13
14    return 0;
15 }
```

```
10
10 > 5 成立
10是偶數
此行永遠被執行
```

遞增/遞減運算子

運算子	範例	說明
++	i++; ++i; i = i +1;	遞增,變數值加1
--	i--; --i; i = i – 1;	遞減,變數值減1

```cpp
#include<iostream>
using namespace std;

int main()
{
    int i = 2;
    cout << "i = " << i << endl;
    cout << "i++*5 = " << i++*5 << endl;
    cout << "i = " << i << endl;
    return 0;
}
```

```
i = 2
i++*5 = 10
i = 3
```

```cpp
#include<iostream>
using namespace std;

int main()
{
    int i = 2;
    cout << "i = " << i << endl;
    cout << "++i*5 = " << ++i*5 << endl;
    cout << "i = " << i << endl;
    return 0;
}
```

```
i = 2
++i*5 = 15
i = 3
```

複合運算子

運算子	範例	說明
+=	a += b	a = a + b
-=	a -= b	a = a - b
*=	a *= b	a = a * b
/=	a /= b	a = a / b
%=	a %= b	a = a % b

```
1  #include<iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a = 10;
7      int b = 3;
8      cout << "a = " << a << ", b = " << b << endl;
9      a+=b;
10     cout << "a+=b運算後, a = " << a << endl;
11     a-=b;
12     cout << "a-=b運算後, a = " << a << endl;
13     a*=b;
14     cout << "a*=b運算後, a = " << a << endl;
15     a/=b;
16     cout << "a/=b運算後, a = " << a << endl;
17     a%=b;
18     cout << "a%=b運算後, a = " << a << endl;
19
20     return 0;
21 }
```

```
a = 10, b = 3
a+=b運算後, a = 13
a-=b運算後, a = 10
a*=b運算後, a = 30
a/=b運算後, a = 10
a%=b運算後, a = 1
```

複合運算子

執行前 a	執行前 b	運算式	執行後 a	執行後 b	說明
12	4	a*=b++	48	5	將a*b放入a中, b加1 (a = a * b; b++;)
12	4	a*=++b	60	5	b加1, 將a*b放入a中 (b++; a = a * b;)
12	4	a/=b--	3	3	將a/b放入a中, b減1 (a = a / b; b--;)
12	4	a/=--b	4	3	b減1, 將a/b放入a中, (b--; a = a / b;)

邏輯運算子

邏輯運算子	說明
&&	AND, 且
||	OR, 或
!	NOT, 反

AND	True (1)	False (0)
True (1)	True (1)	False (0)
False (0)	False (0)	False (0)

OR	True (1)	False (0)
True (1)	True (1)	True (1)
False (0)	True (1)	False (0)

NOT	True (1)	False (0)
	False (0)	True (1)

邏輯運算子-運用實例

```
1  #include<iostream>
2  using namespace std;
3
4  int main()
5  {
6      int score;
7      cin >> score;
8      if( score < 0 || score > 100)
9          cout << "輸入錯誤!" << endl;
10     if( score < 60 && score > 49)
11         cout << "補考!" << endl;
12     return 0;
13 }
```

~~49 < score < 60~~

```
50
補考!
```

```
101
輸入錯誤!
```

括號運算子

括號運算子	說明
( )	提高括號中運算式的優先權

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

```
1+2-3*4+5 = -4
1+(2-3)*4+5 = 2
```

{1+[(2-3)*4]}+5

(1+((2-3)*4))+5

運算子的優先順序

優先順序	運算子	類別	結合性
1	( )	括號運算子	由左至右
2	!、 + (正號)、 - (負號)	一元運算子	由右至左
2	++、 --	遞增/減運算子	由右至左
3	*、 /、 %	算術運算子	由左至右
4	+、 -	算術運算子	由左至右
5	>、 >=、 <、 <=	關係運算子	由左至右
6	==、 !=	關係運算子	由左至右
7	&&	邏輯運算子	由左至右
8	||	邏輯運算子	由左至右
9	=	指定運算子	由右至左

運算子的優先順序

```cpp
#include<iostream>
using namespace std;

int main()
{
    int a=1,b=2,c=3;
    cout << "a = " << a << ", b = " << b << ", c = " << c << endl;
    cout << "a-b+c*5/2 = " << a-b+c*5/2 << endl;
    cout << "a-b+c*(5/2) = " << a-b+c*(5/2) << endl;
    return 0;
}
```

```
a = 1, b = 2, c = 3
a-b+c*5/2 = 6
a-b+c*(5/2) = 5
```