

第二章 補充

變數的使用

- 宣告方式：

- `int num;` `/* 宣告 num 為整數變數 */`
- `int a,b,c;` `/* 同時宣告 a,b 與 c 為整數變數 */`
- `float sum=0.0;` `/* 宣告浮點數變數sum，並設值為0.0 */`

- 變數的資料型態：

- `char` 字元，如 `'A'`、`'2'` 與 `'&'` 等
- `int` 整數 `如12、-27 等`
- `long` 長整數
- `short` 短整數
- `float` 單精度浮點數 `如12.762、-37.483 等`
- `double` 倍精度浮點數

變數的命名規則

- 變數名稱可以是英文字母、數字或底線
 - 名稱中不能有空白字元
 - 第一個字元不能是數字
 - 不能使用到關鍵字

intel_4x	/* 正確 */
_AMD	/* 正確，變數的第一個字母可以是底線 */
2dos	/* 錯誤，變數的第一個字母不能是數字 */
my dogs	/* 錯誤，變數不能有空格 */
goto	/* 錯誤，變數不能是C語言的關鍵字 */

識別字 (identifier)

- 識別字是用來命名變數或函數的文字

```
01  /* prog2_3, printf() 函數的練習 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      int num=2;    /* 定義變數 num, 並設值為 2 */
07      printf("I have %d cats.\n", num); /* 呼叫 printf() 函數 */
08      system("pause");
09      return 0;
10  }
```

識別字

關鍵字 (keyword) (1/2)

- 關鍵字是編譯程式本身所使用的識別字

```
01  /* prog2_3, printf() 函數的練習 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      int num=2;    /* 定義變數 num, 並設值為 2 */
07      printf("I have %d cats.\n",num);  /* 呼叫 printf() 函數 */
08      system("pause");
09      return 0;
10  }
```

關鍵字

關鍵字 (keyword) (2/2)

- 下表為 C 語言的關鍵字

auto	break	case	char	const
continue	default	defined	do	double
else	enum	extern	float	for
goto	if	int	long	register
return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned
void	while	volatile		

程式錯誤的分類

- 語法錯誤（**syntax error**）
 - 程式含有不合語法的敘述，它無法被編譯程式翻譯
- 語意錯誤（**semantic error**）
 - 語意錯誤(又稱邏輯錯誤)，就是程式的執行結果非我們所願

語法錯誤

- 下面是有語法錯誤的程式：

```
01  /* prog2_4, 有錯誤的程式 */
02  #include <stdio.h>
03  #include <stdlib.h>
04
05  int main(void)
06  {
07      int num;          /* 宣告整數 num */
08      num=2;            /* 將 num 設值為 2 */
09      printf("I have %d dogs. \n",num);
10      printf("You have %d dogs, too. \n,num);
11      system("pause")
12      return 0;
13  }
```

語意錯誤

```
/* prog2_5 OUTPUT---
```

```
I have -2 dogs.
```

```
-----*/
```

- 下面是語意錯誤的程式：

```
01  /* prog2_5, 語意錯誤的程式 */
02  #include <stdio.h>
03  #include <stdlib.h>
04
05  int main(void)
06  {
07      int num=-2;          /* 宣告整數變數 num，並設值為-2 */
08
09      printf("I have %d dogs.\n", num);
10      system("pause");
11      return 0;
12  }
```

課本與本學期C的異同

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void main(void)
5  {
6      printf("hi\n");
7      system("pause");
8  }
```

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

void main(void)
{
    cout << "hi\n";
    system("pause");
}
```

1. C++課本的include與本學期不同
2. 本學期輸出學習printf（與C++課本cout 同）
3. 本學期輸入學習scanf（與C++課本cin 同）

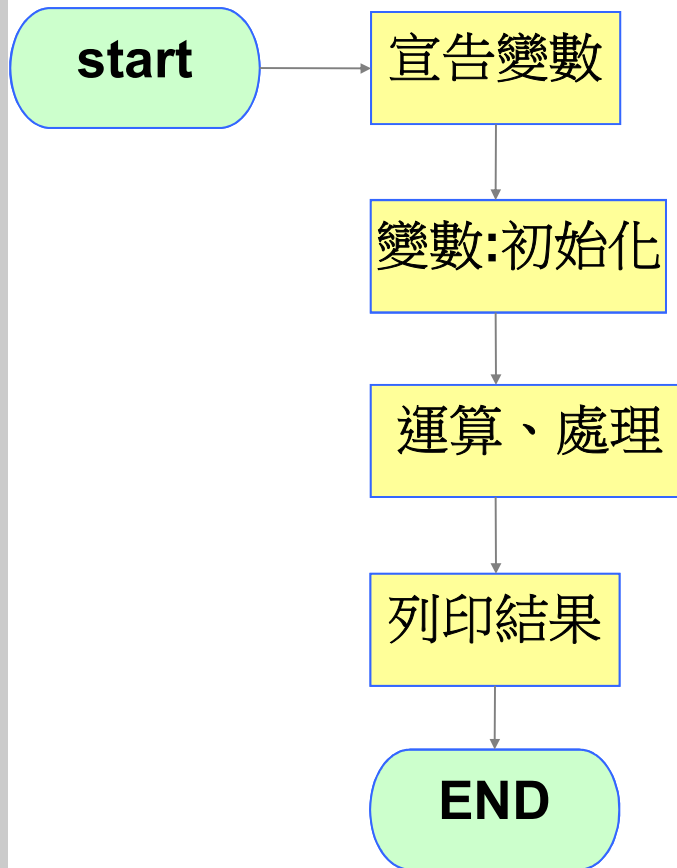
第三章

基本資料型態

I-Fen Chao

Variable

- **Naming:** 命名
- **Declaring:** 宣告
- **Assigning:** 指定
- **Printing Values**



```
main.cpp x
(全域範圍)
#include <stdio.h>
#include <stdlib.h>

void main(void)
{
    1 //declare variable
      int a;

    2 //assignment
      a=2;

    3 //output
      printf("a = %d\n", a);

      //assignment
      a=10;
      //output
      printf("a = %d\n", a);

      system("pause");
}
```

How to declare variables?

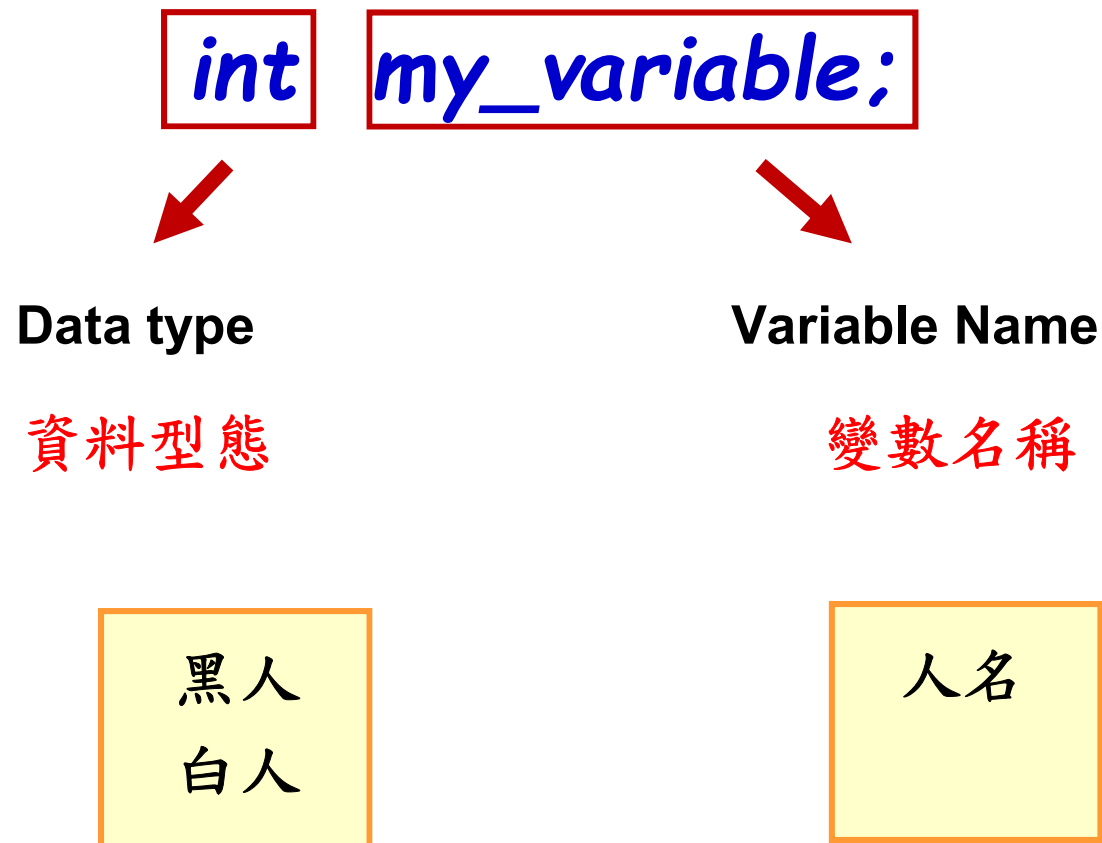
- 語法(syntax)

[(變數) 型態] [變數名稱] ;

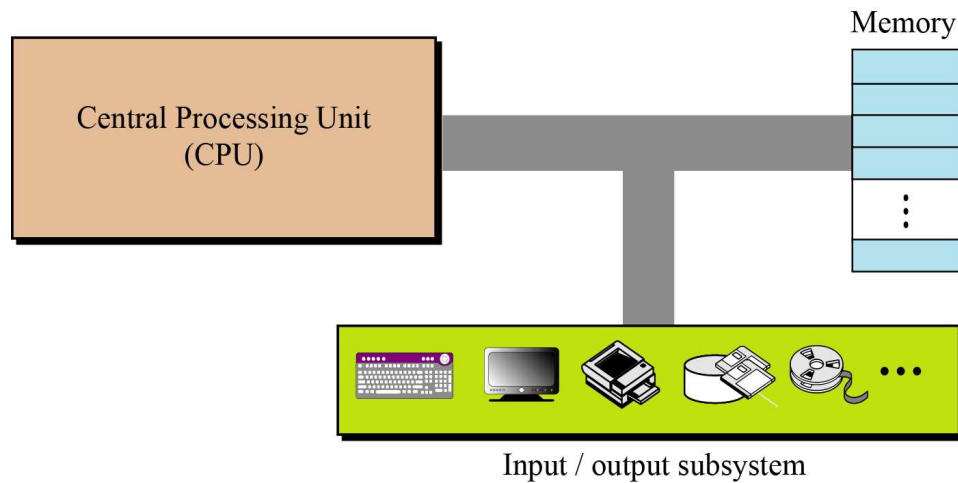
```
#include <stdio.h>
void main(void)
{
    int    month;
    float  expense, income;

    month = 12;
    expense = 111.1;
    ...
}
```

What is "Variable"?



Why "Data type"?



Memory Address

0x0100	32 bits of data
0x0104	32 bits of data
0x0108	32 bits of data
0x010c	32 bits of data
...	

`int my_variable;`

Data type

資料型態

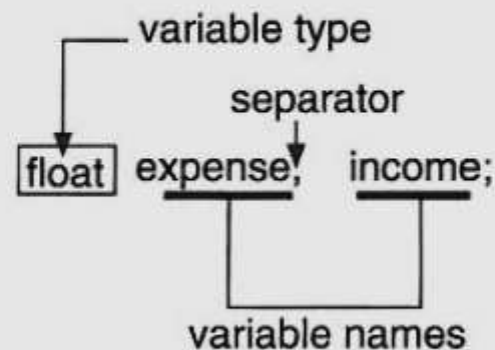
Variable Name

變數名稱

Why "Declaring variable"? (宣告變數)

- Variable Name
 - Let C compiler identifies the name
- Variable data type
 - Let C compiler knows the memory space to be reserved

FIG. 3.1
Declaring variables



Basic data type in C

—**char** character 字元

—**int** integer 整數

—**float** floating point

浮點數(小數)

—**double** double precision

floating point 倍精數

—**void** void (函式專用)

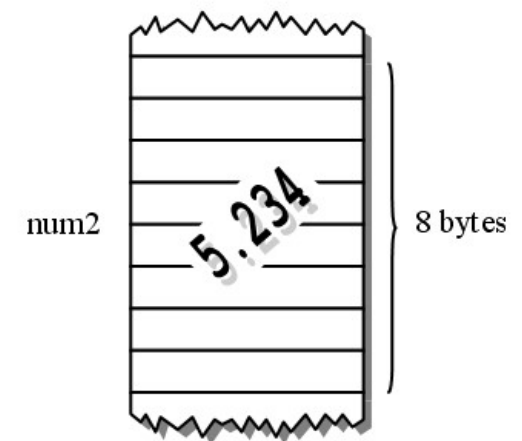
```
int num1=12400;
```

宣告整數變數 num1，
並設值為 12400



```
double num2=5.234;
```

宣告 double 型態的變數
num2，並設值為 5.234



基本資料型態

- 各種基本資料型態所佔的記憶體空間及範圍

表 3.2.1 C 語言所提供的基本資料型態

資料型態		型態說明	位元組	表示範圍
整數 類型	long int	長整數	4	-2147483648 到 2147483647
	int	整數	4	-2147483648 到 2147483647
	short int	短整數	2	-32768 到 32767
	char	字元	1	0 到 255 (256 個字元)
浮點數 類型	float	浮點數	4	1.2e-38 到 3.4e38
	double	倍精度浮點數	8	2.2e-308 到 1.8e308

整數型態 *int*

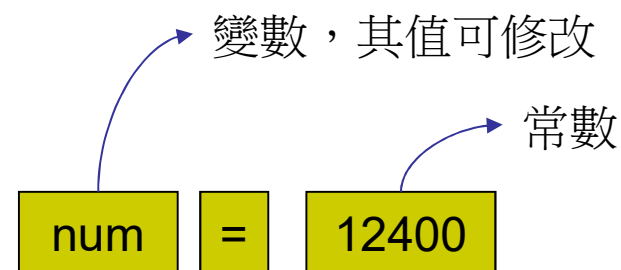
- 整數型態可分為
 - 長整數 (**long int**)
 - 整數 (**int**)
 - 短整數 (**short int**)
- 下面為整數型態宣告的範例：

```
int num=15;           /* 宣告num為整數，並設值為15 */
```

```
long int num=124L;    /* 宣告num為長整數，並設值為1240L */
```

```
short int sum;        /* 宣告sum為短整數 */
```

變數



```
01  /* prog3 1, 變數的使用 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      int num1=12400;    /* 宣告 num1 為整數變數，並設值為 12400 */
07      double num2=5.234; /* 宣告 num2 為倍精度浮點數變數，並設值為 5.234 */
08
09      printf("%d is an integer\n",num1); /* 呼叫 printf() 函數 */
10      printf("%f is a double\n",num2);  /* 呼叫 printf() 函數 */
11
12      system("pause");
13      return 0;
14  }
```

Assignment Statement

- An assignment statement
 - assigns a value to a variable (= store the value into the memory location)
- 語法(syntax)

[變數名稱] = [數值];

```
#include <stdio.h>
void main(void)
{
    int    month;
    float  expense, income;

    month = 12;
    expense = 111.1;
    ...
}
```

溢位 (overflow)

```
/* prog3_2 OUTPUT---
```

```
s+1= -32768
```

```
s+2= -32767
```

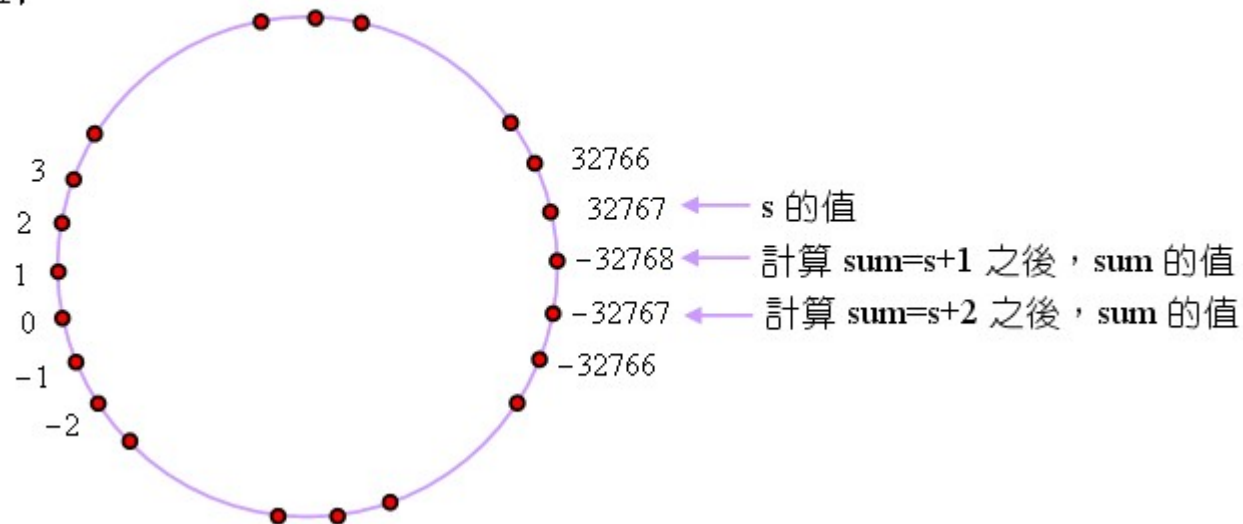
```
-----*/
```

- 溢位：當儲存的數值超出容許範圍時

```
01  /* prog3_2, 短整數資料型態的溢位*/
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      short sum,s=32767;          /* 宣告短整數變數 sum 與 s */
07
08      sum=s+1;
09      printf("s+1= %d\n",sum);    /* 列印出 sum 的值 */
10
11      sum=s+2;
12      printf("s+2= %d\n",sum);    /* 列印出 sum 的值 */
13
14      system("pause");
15      return 0;
16  }
```

溢位 (overflow) (2/2)

```
short sum, s=32767;  
sum=s+1;
```



Data Type - char

- One byte for a character
- Declare a char, and then assign a value

```
char ch;      /* 宣告字元變數ch */
```

```
ch='A';      /* 將字元常數'A'設值給字元變數ch */
```

- Declare and initialize (at the same time)

```
char ch='A';  /* 宣告字元變數ch，並將字元常數'A'設值給它 */
```

```
char ch=97;   /* 將ch設值為ASCII碼為97的字元 */
```

```
char ch='7';  /* 將ch設值為字元常數'7' */
```

```
char ch=7;    /* 將ch設值為設值為ASCII碼為7的字元 */
```

列印字元變數

/* prog3_3 OUTPUT---

ch= a

ASCII of ch= 97

-----*/

```
01  /* prog3_3, 字元的列印*/
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      char ch='a';          /* 宣告字元變數 ch，並設值為 'a' */
07      printf("ch= %c\n",ch); /* 印出 ch 的值 */
08      printf("ASCII of ch= %d\n",ch); /* 印出 ch 的十進位值 */
09
10      system("pause");
11      return 0;
12  }
```

字元型態溢位的問題：

```

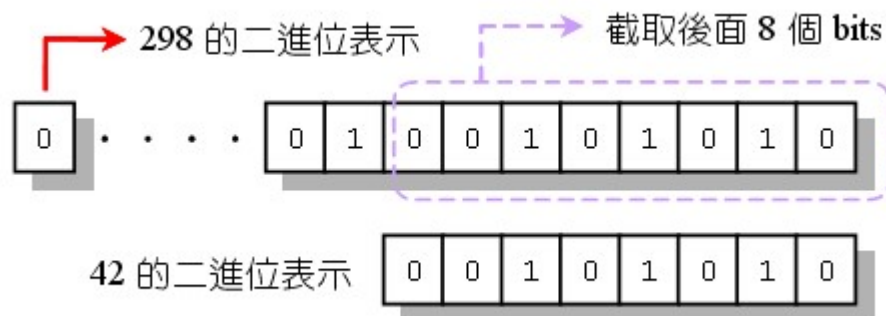
01  /* prog3_6, 字元型態的列印問題 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      int i=298;
07      printf("ASCII(%d)=%c\n",i,i);    /* 印出 ASCII 碼為 i 的字元 */
08
09      system("pause");
10      return 0;
11  }

```

```

/* prog3_6 OUTPUT---
ASCII (298) =*
-----*/

```



跳脫字元

- 反斜線「\」稱為跳脫字元
- 反斜線「\」加上控制碼，稱為跳脫序列


表 3.2.3 常用的跳脫序列

跳脫序列	所代表的意義	十進位 ASCII
\a	警告音 (alert)	7
\b	倒退一格 (backspace)	8
\n	換行 (new line)	10
\r	歸位 (carriage return)	13
\0	字串結束字元 (null character)	0
\t	跳格 (tab)	9
\\	反斜線 (backslash)	92
\'	單引號 (single quote)	39
\"	雙引號 (double quote)	34

跳脫字元

```
01  /* prog3_7, 跳脫序列的列印 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      char beep='\a';          /* 宣告字元變數 beep，並設定其值為 '\a' */
07      printf("%c", beep);      /* 響一聲警告音 */
08      printf("ASCII of beep=%d", beep); /* 印出 beep 的 ASCII 值 */
09
10      system("pause");
11      return 0;
12  }
```

```
/* prog3_7 OUTPUT---
ASCII of beep=7
-----*/
```



還會有一聲
警告音哦

浮點數型態 *float*

- 浮點數佔 **4** 個位元組，有效範圍

```
float num;           /* 宣告浮點數變數num */
```

```
float num=5.46F;      /* 宣告浮點數變數num，並設值為5.46F */
```

- 要印出浮點數，可用「%f」格式碼
- 要以指數的型式列印浮點數，可用「%e」格式碼

列印浮點數字

/* prog3_9 OUTPUT---

num1=1.234500e+002

num2=0.004560

-----*/

```

01  /* prog3_9, 浮點數的列印 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      float num1=123.45F;      /* 宣告 num1 為浮點數，並設值為 123.45F */
07      float num2=4.56E-3F;    /* 宣告 num2 為浮點數，並設值為 4.56E-3F */
08
09      printf("num1=%e\n", num1); /* 以指數的型態印出 num1 的值 */
10      printf("num2=%f\n", num2); /* 以浮點數的型態印出 num2 的值 */
11
12      system("pause");
13      return 0;
14  }

```

倍精度浮點數型態 **double**

/* prog3_10 OUTPUT-----

num1=123.456787109375

num2=123.456789012345

-----*/

- 下面的範例是 **float** 與 **double** 精度的比較：

```

01  /* prog3_10, float 與 double 精度的比較 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      float  num1=123.456789012345F; /* 宣告 num1 為 float, 並設定初值 */
07      double num2=123.456789012345; /* 宣告 num2 為 double, 並設定初值 */
08
09      printf("num1=%16.12f\n", num1); /* 列印出浮點數 num1 的值 */
10      printf("num2=%16.12f\n", num2); /* 列印出倍精度浮點數 num2 的值 */
11
12      system("pause");
13      return 0;
14  }

```


資料型態的轉換

- 將資料型態轉換成另一種型態的語法：

資料型態的強制轉換

(欲轉換的資料型態) 變數名稱;

```
int num=12;
```

```
float total;
```

```
total=(float) num;    /* 將int型態轉換成float型態 */
```

資料型態的轉換

```
/* prog3_12 OUTPUT-----  
num1=3.002000, num2=3.988000  
n1=3, n2=3  
-----*/
```

```
01  /* prog3_12, 資料型態的轉換*/  
02  #include <stdio.h>  
03  #include <stdlib.h>  
04  int main(void)  
05  {  
06      int n1,n2;  
07      float num1=3.002F,num2=3.988F;  
08  
09      n1=(int) num1;      /* 將浮點數 num1 轉換成整數 */  
10      n2=(int) num2;      /* 將浮點數 num2 轉換成整數 */  
11  
12      printf("num1=%f, num2=%f\n",num1,num2); /* 印出浮點數的值 */  
13      printf("n1=%d, n2=%d\n",n1,n2); /* 印出浮點數轉成整數後的值 */  
14  
15      system("pause");  
16      return 0;  
17  }
```

資料型態的轉換

```
/* prog3_13 OUTPUT-----
```

```
num/2=2
```

```
(float) num/2=2.500000
```

```
-----*/
```

- 把整數轉換成浮點數，再進行除法運算：

```
01  /* prog3_13, 資料型態的轉換 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      int num=5;
07
08      printf("num/2=%d\n", num/2);          /* 整數相除 */
09      printf("(float) num/2=%f\n", (float) num/2); /* 將整數轉成浮點數，再做除法 */
10
11      system("pause");
12      return 0;
13  }
```

*What happens if we try to display an **int** with **%f** or a **float** with **%d**?*

- A common error that beginners make
 - Get completely **nonsensical values or zeros**
 - It is an extremely frustrating error, **it appear like you have major errors in your program**. Then you may spend a lot of time at debugging
 - If you get total nonsense or zeros for your output, **first** check your conversion specifications

控制碼必須符合輸出的型態

- 錯誤的範例：整數資料以其它型態輸出

```
01  /* prog4_7, 整數資料以其它型態輸出, 錯誤的範例 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      int a=15;          /* 宣告整數變數 a, 並設值為 15 */
07
08      printf("a=%d\n", a);          /* 印出 a 的值 */
09      printf("以浮點數型態印出: %f\n", a);          /* 以%f 格式碼印出 a 的值 */
10      printf("以指數型態印出 : %e\n", a);          /* 以%e 格式碼印出 a 的值 */
11
12      system("pause");
13      return 0;
14  }
```

Variable Initialization

- Use an **assignment** to initialize a variable

Ex: `e=3;`

- Initialize a variable in a **declaration** statement

Ex: `float a=7, b=6;`

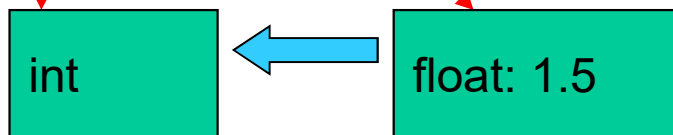
What are the values of the following?

- $6/4 = ?$
 - Same type arithmetic
- $6.0/4.0 = ?$
 - Same type arithmetic
- $6/4.0 = ?$
 - Mixed type arithmetic

What happens? (1)

- To assign a *real (float)* type value to a variable that has been declared be **int**?

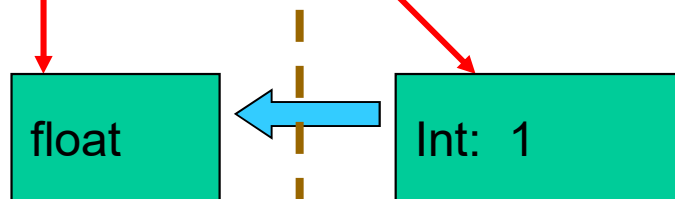
```
int n;  
n = 6 / 4.0;
```



The fractional part is cut off : $n = 1.0$

What happens? (2)

```
float p;  
p = 6 / 4;
```



**The fractional
part is cut off here,
rvalue is 1**

輸出函數 `printf()`

- `printf` 是由 **print**（列印）與 **format**（格式）二字組成
- `printf()` 函數的使用格式：

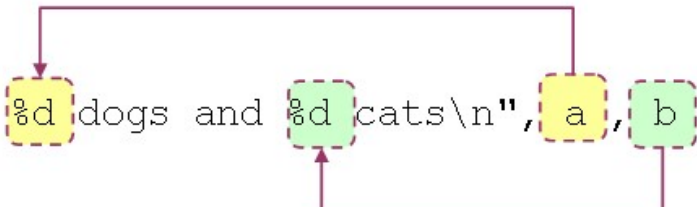
`printf()` 函數的格式

```
printf("格式字串", 項目1, 項目2, ...);
```

把 **a** 的值以 `%d` 的格式填到這兒

```
printf("I have %d dogs and %d cats\n", a, b);
```

把 **b** 的值以 `%d` 的格式填到這兒



printf() 的格式碼

表 4.1.1 *printf()* 函數常用的格式碼

格式碼	說 明	格式碼	說 明
%c	字元	%%	印出百分比號
%d	十進位整數	%o	無號八進位整數
%ld	長整數	%s	字串
%e	浮點數，指數 e 型式	%u	無號十進位整數
%f	浮點數，小數點型式	%x	無號十六進位整數

- printf() 格式：

f 旗標 (+, -, 0, ...)

w 資料總長度

p 小數點後位數

x 列印控制字元 (d, f, e, ...)

資料內容

格式

執行結果

12345

%d

	1	2	3	4	5
--	---	---	---	---	---

123.456

%7.2f

	1	2	3	.	4	6
--	---	---	---	---	---	---

sizeof 指令的應用

```

01  /* prog3_11, 列印出各種資料型態的長度 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      char ch;          /* 宣告字元變數 ch */
07      float num;        /* 宣告浮點數變數 num */
08
09      printf("sizeof(2L)=%d\n", sizeof(2L)); /* 查詢常數 2L 所佔位元組 */
10
11      printf("sizeof(ch)=%d\n", sizeof(ch)); /* 查詢字元變數 ch 所佔位元組 */
12      printf("sizeof(num)=%d\n", sizeof(num)); /* 查詢變數 num 所佔位元組 */
13
14      printf("sizeof(int)=%d\n", sizeof(int)); /* 查詢 int 型態所佔位元組 */
15      printf("sizeof(long)=%d\n", sizeof(long)); /* 查詢 long 型態所佔位元組 */
16      printf("sizeof(short)=%d\n", sizeof(short)); /* 查詢 short 所佔位元組 */
17
18      system("pause");
19      return 0;
20  }

```

/* prog3_11 OUTPUT---

```

sizeof(2L)=4
sizeof(ch)=1
sizeof(num)=4
sizeof(int)=4
sizeof(long)=4
sizeof(short)=2
-----*/

```