

# 基本語法

---

Jia-Yin Wang

# C 程式架構

```
#include <stdio.h>
```

← C I/O 標頭檔

```
int main()
```

← C/C++ 程式進入點

```
{
```

```
    printf("Hello World!");
```

← 標準輸出函數

```
    return 0;
```

← 函數回傳值

```
}
```

# C 的輸入輸出

- 在 C 的環境中, 引入 `<stdio.h>`
- 在 C++ 的環境中, 引入 `<stdio.h>` 或 `<cstdio>`
- 輸出使用 `printf()` 函數
- 輸入使用 `scanf()` 函數

C 程式一般副檔名為 `.c`

C++ 程式一般副檔名為 `.cc` 或 `.cpp`

# 程式註解

```
printf("Hello World!"); // 這是單行的註解文字
```

```
/*  
    這種形式的註解可以跨越很多行，  
    通常會放在程式檔案最開頭的位置。  
*/  
  
#include <stdio.h>  
int main()  
{  
    printf("Hello World!");  
    return 0;  
}
```

# 格式輸出

```
printf("format\nstring");
```



format  
string

跳脫字元	功能
\t	移到下一定位，即Tab鍵功能
\n	移到下一行開頭位置
\"	插入 " 字元
\'	插入 ' 字元
\\	插入 \ 字元

## 格式說明符號 (format specifier)

```
printf("The number is %d", 10*3);
```

會產生如下的輸出結果：

```
The number is 30
```

## 格式說明符號 (format specifier)

```
printf("%d x %d = %d", 7, 9, 7*9);
```

會產生如下的輸出結果：

```
7 x 9 = 63
```

# 格式說明符號 (format specifier)

格式說明符號	功能
%c	插入一個字元
%d	插入一個整數
%f	插入一個浮點數（有小數點的數）
%s	插入一個字串
%%	插入 % 字元



# Example

```
#include <stdio.h>

int main()
{
    printf("Display a character: %c\n", 'A'); // 'A' is a character
    printf("Display an integer: %d\n", 3456);
    printf("Display a float number: %f\n", 345.6);
    printf("Display a string: %s\n", "This is a string");
    printf("Display the %% character\n");
    return 0;
}
```

```
Display a character: A
Display an integer: 3456
Display a float number: 345.600000
Display a string: This is a string
Display the % character
```

## 練習

請在螢幕上輸出以下的字串："The program uses 34% CPU resource."

# 變數宣告

```
int x;
```

```
x = 10;
```



```
int x = 10;
```

```
int x;  
int y;
```



```
int x, y;
```

Q: 這是什麼?

```
int x, y=5, z=3, s, t=8;
```

# 標準輸入

```
int x; // x is an integer variable  
scanf("%d", &x); // input x value
```

```
int x, y; // x and y are integer variables  
scanf("%d", &x);  
scanf("%d", &y);
```

```
int x, y;  
scanf("%d %d", &x, &y);
```

```
int x, y;  
scanf("%d%d", &x, &y);
```



# Example

```
#include <stdio.h>

int main()
{
    int x, y;
    scanf("%d%d", &x, &y);
    printf("%d x %d = %d\n", x, y, x*y);
    return 0;
}
```

## 練習

輸入長方體的長、寬、高，然後利用程式把該長方體的體積印出來。

# 整數格式

名稱	宣告方式	格式說明符號	說明
整數	int	%d	一般使用 4 個位元組，共 32 位元，其中第 1 個位元表示正負號，負數以 2 的補數形式表示，可以表示的整數範圍從 $-2^{31}$ 到 $2^{31}-1$ 。
長長整數	long long 或 long long int	%lld 或 %I64d (Windows)	一般使用 8 個位元組，共 64 位元，其中第 1 個位元表示正負號，負數以 2 的補數形式表示，可以表示的整數範圍從 $-2^{63}$ 到 $2^{63}-1$ 。
字元	char	%c	使用 1 個位元組，共 8 位元。一般字元的編號是依照 ASCII 標準。也可以算是一種整數，不過範圍只從 -128 到 127。
無號整數	unsigned 或 unsigned int	%u	使用 4 個位元組，共 32 位元，沒有正負號位元，可以表示的整數範圍從 0 到 $2^{32}-1$ 。

# 溢位

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int a = 2;
```

```
    a = a * a;    printf("%d\n", a);    // ==> 4
```

```
    a = a * a;    printf("%d\n", a);    // ==> 16
```

```
    a = a * a;    printf("%d\n", a);    // ==> 256
```

```
    a = a * a;    printf("%d\n", a);    // ==> 65536
```

```
    a = a * a;    printf("%d\n", a);    // ==> 0
```

```
    return 0;
```

```
}
```



# 字元型態

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    printf("Char %c => ASCII %d\n", 'a', 'a');
```

```
    printf("Char %c => ASCII %d\n", 97, 97);
```

```
    return 0;
```

```
}
```

Char a => ASCII 97

Char a => ASCII 97

# 小寫換大寫

```
#include <stdio.h>

int main()
{
    char ch;
    scanf("%c", &ch);
    printf("%c", ch + 'A' - 'a');
    return 0;
}
```

## 練習

輸入三個大寫字元，將它們變成小寫字元後輸出到螢幕上。

# 浮點數

名稱	宣告方式	格式說明符號	說明
單精度浮點數	float	%f	使用4個位元組，共32位元，其中包括正負號位元，以及基數和指數部份，可以表達的大小（不考慮正負號）約從 $1.2 \times 10^{-38}$ to $3.4 \times 10^{38}$ ，有效位數約6位。
倍精度浮點數	double	%lf	使用8個位元組，共64位元，其中包括正負號位元，以及基數和指數部份，可以表達的大小（不考慮正負號）約從 $2.3 \times 10^{-308}$ to $1.7 \times 10^{308}$ ，有效位數約15位。

# 計算圓面積

```
#include <stdio.h>

int main()
{
    double r;
    scanf("%lf", &r);
    printf("Area = %lf\n", 3.14159*r*r);
    return 0;
}
```

3.0

Area = 28.274310

# 輸出位數

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    printf("%d\n", 51);
```

```
    printf("%5d\n", 51);
```

```
    printf("%-5d\n", 51);
```

```
    printf("%05d\n", 51);
```

```
    printf("%f\n", 20.0/3);
```

```
    printf("%.2f\n", 20.0/3);
```

```
    printf("%2.2f\n", 20.0/3);
```

```
    printf("%8.2lf\n", 20.0/3);
```

```
    return 0;
```

```
}
```

```
51
```

```
    51
```

```
51
```

```
00051
```

```
6.666667
```

```
6.67
```

```
6.67
```

```
        6.67
```

## 練習

輸入圓柱的半徑和高度，由程式計算其表面積並把它輸出到螢幕上。

# 基本運算

$3 + 4 * 2$



11

$(3 + 4) * 2$



14

$11 \% 3$



2



# 整數除法

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    printf("5/2 = %d", 5/2);
```

```
    return 0;
```

```
}
```

5/2 = 2

# 指派運算

```
x = 3;  
y = 5 + 2 * 8;  
z = x + y;  
t = t + 1;
```

```
x += 3;  
x = x + 3;
```

```
x *= 3 + 2;
```

## 兩數交換

```
int t;  
t = x;   x = y;   y = t;
```

## 遞增、遞減

```
x = x+1;
```

```
x += 1;
```

```
x++;
```

```
++x;
```

```
int x = 3;
```

```
int y = 5 + ++x;
```

```
int x = 3;
```

```
int y = 5 + x++;
```

```
int x=3, y=4, z;
```

```
z = x++ * --y;
```

# 比較

運算符號	意義	範例	範例結果
>	大於	$5 > 3$	1
>=	大於或等於	$5 \geq 3$	1
<	小於	$5 < 3$	0
<=	小於或等於	$5 \leq 3$	0
==	相等	$5 == 3$	0
!=	不等	$5 != 3$	1

# 邏輯運算

運算符號	意義	範例	範例結果
&&	邏輯的且，兩邊的值都是真值的時候成立	(5>3) && (4>6)	0
	邏輯的或，兩邊的值只要存在真值就成立	(5>3)    (4>6)	1
!	邏輯的反值，把右邊的真假值反過來	!(5>3)	0

# 轉型

```
float x = 3 / 5.0;
```

```
x = 1/2 + 0.5;
```

```
x = 1.0/2 + 0.5;
```

```
x = (float)1 / 2;
```

```
int x = (int)2.9999;
```

## 練習

由使用者輸入兩個整數，將其存到整數變數中，再計算它們的實數相除的商，例如輸入的值為 2 和 5，則輸出的結果應為 0.4。



Q & A