C語言語法整理

P.S. 大括弧裡的三個小點代表程式碼，不代表程式碼多寡
其他地方的三個小點表示可以一直寫下去。

標頭檔

下列只講解最常用的標頭檔以及課程範圍內使用到的函數，

如欲知其他可用標頭檔請自行搜尋。

stdio.h

使用scanf(scanf_s)函數、printf函數必須要引入的標頭檔

語法:#include <stdio.h>

想了解還有什麼函數請到可以到 輸出與輸入 stdio.h 以及 檔案處理 stdio.h 網站觀看。

stdlib.h

使用system函數、rand函數、srand函數時必須要引入的標頭檔

語法:#include <stdlib.h>

想了解還有什麼函數請到可以到 通用工具 stdlib.h 網站觀看。

math.h

使用sqrt函數、pow函數時必須要引入的標頭檔

語法:#include <math.h>

想了解還有什麼函數請到可以到 數學計算 math.h 網站觀看。

常用函數

以下講解的函數以課程用為主，其他(如讀檔、時間等)的請自行查詢。

scanf(scanf_s)函數

使用時機:需要使用者從鍵盤輸入數字以及文字到程式時

語法:scanf(“輸入格式”,&變數);

或 scanf_s(“輸入格式”,&變數);

輸入格式常用有“%c”(字元)、”%d”(數字)以及”%s”(字串)

P.S. &為取得該變數的記憶體位址

printf函數

使用時機:需要印出數字以及文字到螢幕上時

語法:printf(“列印格式”,變數);

列印格式因礙於版面問題，無法一一講解，請至printf( ) 的列印格式、控制字元、修飾子 觀看。

rand函數

使用時機:需要產生亂數時使用

語法:整數變數 = (rand() % (最大值-最小值+1)) + 最小值;

P.S. 產生亂數結果不會因為程式碼內執行多少rand()或關閉程式重開而改變

srand函數

使用時機:當每次執行程式時亂數都跟上次的結果不一樣

(僅限每次不同的亂數種子情況下)

語法:srand(亂數種子);

P.S. 產生亂數結果不會因為程式碼內執行多少rand()或關閉程式重開而改變

變數

定義: 記錄某些資料，可能是文字，也可能是數字，
把這些資料記錄在記憶體的某個位址中，並給它一個名稱。

下一張利用表格介紹常用的變數的種類以及宣告方法。

變數種類	語法
int	int 變數名稱;
unsigned int	unsigned int 變數名稱;
float	float 變數名稱;
double	double 變數名稱;
char	char 變數名稱;

P.S. unsigned為無負號、char儲存單位為字元(非字串)

選擇條件 if

```
if (條件式)
{
   …
}
else if (條件式)
{
 …
}
…
…
else
{
 ...
}
```

範例

```
if (a == 'a')
{
    printf("Yes!\n");
}
else if (a == 'b')
{
    printf("No!\n");
}
else
{
    printf("What?\n");
}
```

選擇條件 switch

```
switch (變數)
{
  case 符合字元或數字:
      …
      break;
  case 符合字元或數字:
      …
      break;
  default:
      …
      break;
}
```

範例

```
switch (a)
{
  case 0:
    zeroCount++;
    break;
  case 1:
     oneCount++;
     break;
  default:
     otherCount++;
     break;
}
```

更多寫法請至 switch 多重選擇控制 網站觀看

重複結構 for

```
for (變數初始值;執行條件式;增或減值)
{
   …
   …
   …
}
```

```
for (i = 1; i <= 100; i++)
{
        sum += i;
}
```

```
for (i = 4; i <= 100; i += 4)
{
        sum += i;
}
```

```
for (i = 100; i <= 1; i--)
{
        sum += i;
}
```

```
for (i = 100; i <= 4; i -= 4)
{
        sum += i;
}
```

重複結構 while

```
while (進入迴圈的條件式)
{
   …
   …
   …
}
```

範例

```
int i = 1;

while (i <= 100)
{
    sum += i;
    i = i +1; //或寫成i++
 }
```

重複結構 do…while

```
do
{
   …
   …
   …
} while (結束迴圈的條件式);
```

範例

```
int i = 1;

 do
{
   sum += i;
   i++;
} while (i <= 100);
```

自訂函數

```
回傳型態 名稱(參數1,參數2,…)
{
    …
    …
    …
    return 回傳型態;
}
```

貼心小叮嚀:

使用自訂函數前請養成良好習慣先寫原型宣告在定義
副程式內容，回傳型態跟參數都可為空(void)
回傳型態為void時不用寫return。

範例

```
#include <stdio.h>

int square(int y); //原型宣告

int main(void) //主程式
{
  for (int I = 1;I <= 10;++I)
  {
    printf("%d",square(I));
  }
}

int square(int y) //副程式
{
  return y*y;
}
```

參數傳入自訂函數的方式

請參閱下面網站

[筆記] C++學習筆記（傳值、傳址、傳參考）

P.S.傳址(Call by Value of Pointer or Call by Address)課程中還沒教到

可以先行略過

遞迴

定義:一個函數呼叫自己的行為
右邊為範例

範例

```
int fact(int n)
{
    if (n==0)
    {
        return(1);
    }
    else
    {
        return(n*fact(n-1)); //呼叫自己
    }
}
```

巢狀結構(範例)

```
for (int I = 1;I <= 100;I++)
{
   …
   for (int j = 1;j <= 200;j++)
   {
     …
   }
}
```

```
do
{
    ….
     do
    {
         …
    } while (j <= 50);
    ….
}while (i <= 100);
```

結構	是否可使用巢狀結構
for	可
while	可
do…while	可
if	可
switch	可
Function (自訂函數)	不可

一維陣列

語法:變數種類 變數名稱[元素個數或長度]

int a[10] = {0}; //把陣列所有元素設為0
下面為圖例

索引值	0	1	2	3	4	5	6	7	8	9
內容	0	0	0	0	0	0	0	0	0	0

二維陣列

語法:變數種類 變數名稱[列數] [行數]

int a[7] [6] 圖例

(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,0)	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,0)	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,0)	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)
(6,0)	(6,1)	(6,2)	(6,3)	(6,4)	(6,5)

三維陣列

語法:變數種類 變數名稱[二維陣列數量] [列數] [行數]

int a[3] [7] [6]

三維陣列圖例

指標

指標在C當中，可以算是有難度的一個章節，
有人說:「不會用指標，就別說你/妳有學過C」
以下用圖片來說明

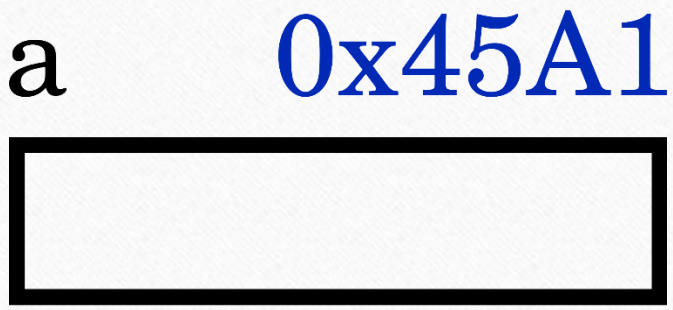
定義:指向一個記憶體位址

語法:變數種類 *變數名稱

範例: int *ptr 或 int *ptr = NULL

宣告　　記憶體

int a



int *p

程式碼

```
//設定a的值
a = 31;
//將p指標指向a
p = &a;
```

記憶體