



2017

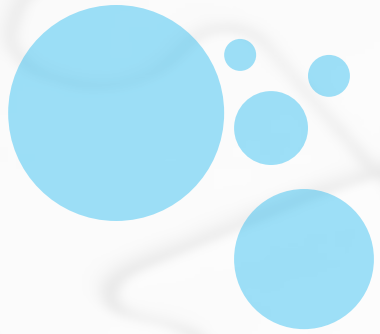
程式設計  
加強班

# 程式設計與實習(一)

這學期最後一次加強班講義

BY 孫茂勛

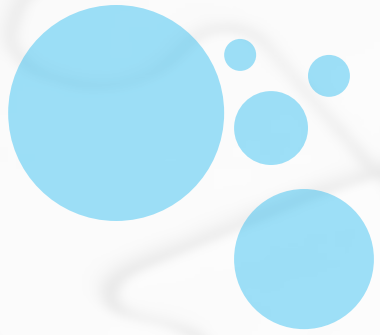
Email:JOHN85051232@GMAIL.COM



# File I/O

Q：為什麼需要檔案讀寫？

💡 程式執行完資料就消失了，總該把它存到電腦裡面吧，下次就可以把上次執行的結果拿來用。



# File I/O

檔案讀寫：將資料存到電腦中，以及從電腦中讀取資料  
檔案讀寫的兩種方式：

- Sequential-Access(fprintf、fscanf...)
- Random-Access(fwrite、fread)

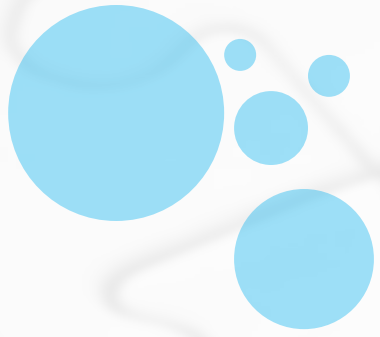
# 先來介紹這兩個Access的差別

## Sequential-Access(循序存取)：

每次讀取 / 寫入時都必須從第一筆資料開始逐筆往後做。

每次讀／寫都必須從起始位置  
開始依序往後





# 先來介紹這兩個Access的差別

## Random-Access(隨機存取)：

每次讀取 / 寫入時可以從指定的位置開始做。

Ex: 可以直接讀/寫第三筆的資料，而不用從第一筆開始往後找



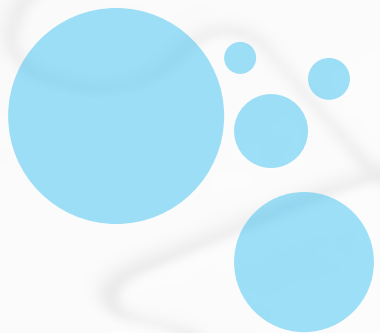
Data





# 先來介紹這兩個Access的差別

- 循序存取可以想成古代畫卷，需要把攤開之前的東西才看的到後面的內容
- 隨機存取則像是CD片，想要找哪一片可以直接取出

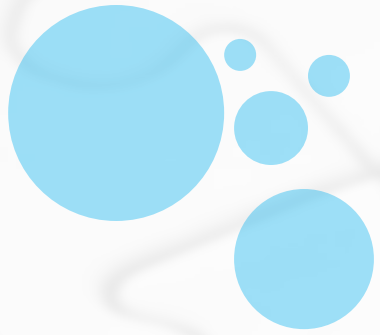


## 想一想

Q：陣列為什麼可以直接取得某個位置的資料？

E x：想取得陣列第三個位置的內容？Arr[3]

- ❖ 透過[...]內告訴程式要取得第幾個位置，就可以把它取出來，這樣取得資料的方式也是隨機存取。
- ❖ (之後會學到有些資料結構是無法進行隨機存取的。)



# File I/O

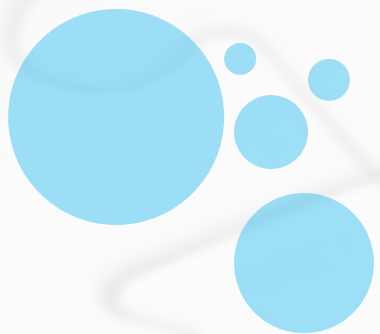
讀寫檔三步驟：

1. 開檔(fopen)

2. 讀/寫檔

3. 關檔(fclose) -> 記得關檔，不然無法成功寫入!!

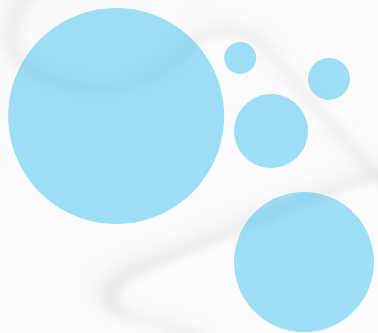




# File I/O

使用fopen打開檔案，fopen會回傳檔案的指標，所以要用 FILE 的指標變數去接。

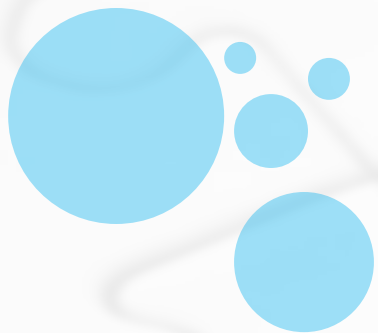
```
FILE *fptr = fopen("test.txt", "w");  
if( fptr == NULL)//如果開檔失敗會回傳NULL  
{  
    printf("開檔失敗\n");  
}  
else  
{  
    fprintf(fptr, "HELLO WORLD");  
    printf("開檔成功\n");  
}  
fclose(fptr);//記得關檔  
return 0;
```



# File I/O

FILE \*fptr = Fopen( "檔名" , "開啟模式" );

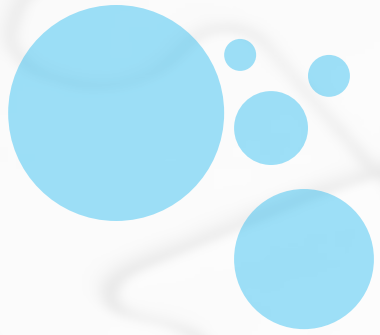
模式	功能說明
"w"	開啟一個可以寫入資料的檔案。若檔案存在，則該檔內的資料都會被刪除變成一個空檔；如果檔案不存在，則會自動建立一個新檔。
"a"	開啟一個可以寫入資料的檔案。若檔案存在，則新增的資料會附加在該檔案的最後面；如果檔案不存在，則會自動建立一個新檔。
"r"	開啟一個已經存在的檔案，若檔案不存在，則 fopen()函式會傳回 NULL。
"w+"	開啟一個可以讀寫資料的檔案。若檔案存在，則該檔內的資料都會被刪除變成一個空檔；如果檔案不存在，則會自動建立一個新檔。
"a+"	開啟一個可以讀寫資料的檔案。若檔案存在，則新增的資料會附加在該檔案的最後面；如果檔案不存在，則會自動建立一個新檔。



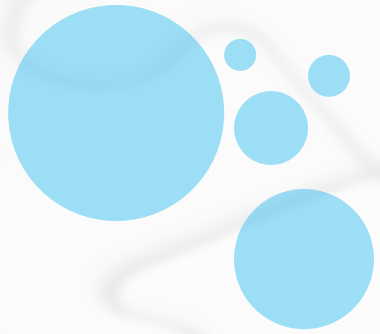
# File I/O

FILE \*fptr = Fopen( "檔名" , " 開啟模式" );

模式	功能說明
"r+"	開啟一個可以讀寫資料的已存在檔案，若檔案不存在，則 fopen() 函式會傳回 NULL。
"wb+"	開啟一個可以讀寫資料的二進位檔案。若檔案存在，則該檔內的資料都會被刪除變成一個空檔；如果檔案不存在，則會自動建立一個新檔。
"ab+"	開啟一個可以讀寫資料的二進位檔案。若檔案存在，則新增的資料會附加在該檔案的最後面；如果檔案不存在，則會自動建立一個新檔。
"rb+"	開啟一個可以讀寫資料的已存在二進位檔案，若檔案不存在，則 fopen() 函式會傳回 NULL。

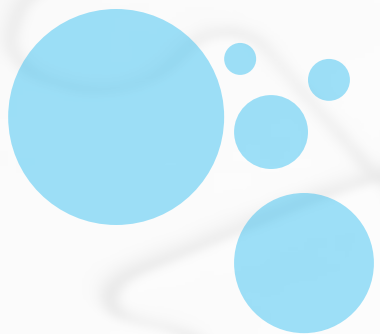


# 循序存取的讀寫



## 讀檔

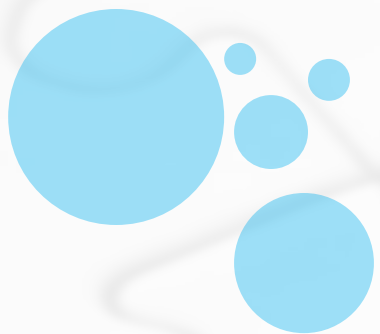
❖ `fscanf()`：把檔案內容依照需要的格式讀出來，多了一個參數(檔案指標)，其他用法與`scanf`一樣。



# 讀檔

 **fgetc**：把檔案中的內容讀取一個字元。

```
FILE *fptr = NULL;
char c = 0;
if( (fptr = fopen("test.txt", "r")) == NULL) // 讀檔
{
    printf("開檔失敗\n");
}
else
{
    while( (c = fgetc(fptr)) != EOF )
    {
        printf("%c", c);
    }
}
fclose(fptr); // 關閉檔案
```

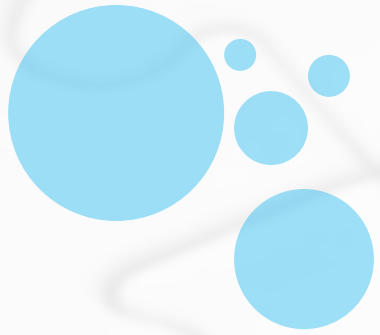


# 讀檔

 fgetc : 把檔案中的內容讀取一行。

```
while( fgetc(str,20,fptr) != NULL )  
{  
    printf( "%s",str);  
}
```





# 寫檔

 `fprintf` :

```
fprintf( fptr, "hello world\n");
```

檔案指標

依照需要的格式寫入檔案，多了一個參數(檔案指標)，其他用法與printf一樣。



# 練習

Q：使用者輸入 3 位學生的姓名、學號、電話，並依序存到txt中

```
請輸入第0位學生的姓名:  
john  
請輸入第0位學生的學號:  
a1035501  
請輸入第0位學生的電話:  
123  
請輸入第1位學生的姓名:  
張小名  
請輸入第1位學生的學號:  
a1035502  
請輸入第1位學生的電話:  
456  
請輸入第2位學生的姓名:  
王大明  
請輸入第2位學生的學號:  
a1035503  
請輸入第2位學生的電話:  
789  
請按任意鍵繼續 . . .
```

student.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

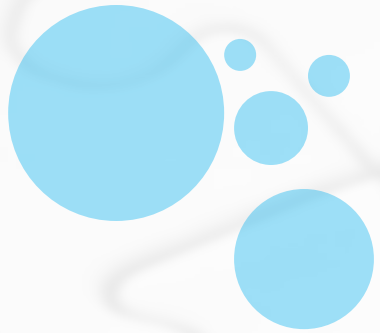
```
john a1035501 123  
張小名 a1035502 456  
王大明 a1035503 789
```

# 練習

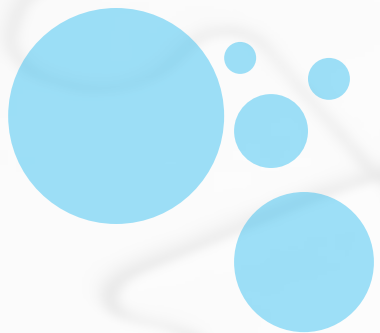
📁 讀取剛剛寫入的資料看看是否正確?(使用Struct)

```
struct student
{
    char name[10];
    char ssn[10];
    char phone[10];
};
```

```
FILE *fptr = NULL;
student s[3] = {0};
if( (fptr = fopen("student.txt", "a+")) == NULL) // 讀檔
{
    printf("開檔失敗\n");
}
else
{
    for(int i = 0 ; i < 3 ; ++i)
    {
        fscanf(fptr, "%s %s %s", s[i].name, s[i].ssn, s[i].phone); // 讀取
        printf("%s %s %s\n", s[i].name, s[i].ssn, s[i].phone);
    }
}
fclose(fptr); // 關閉檔案
```



# 隨機存取的讀寫



# 移動檔案指標

```
int fseek(FILE *fp, long offset, int mode);
```

▣ fp 開啟的檔案指標

▣ offset 相對於mode 的 位移量

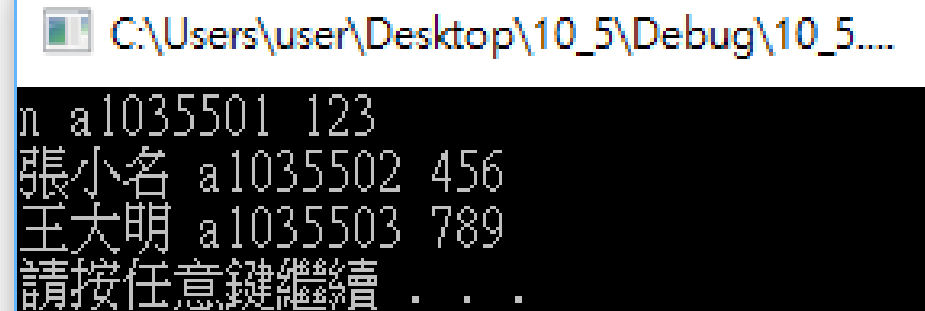
▣ mode

- SEEK\_SET 檔案開頭
- SEEK\_CUR 目前游標所在位置
- SEEK\_END 檔案結尾

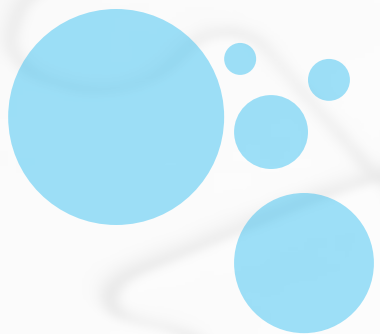
# 移動檔案指標

```
{  
    printf("開檔失敗\n");  
}  
else  
{  
    fseek(fp, 3, SEEK_SET);  
    for(int i = 0 ; i < 3 ; ++i)  
    {  
        fscanf(fp, "%s %s %s", s[i].name, s[i].ssn, s[i].age);  
        printf("%s %s %s\n", s[i].name, s[i].ssn, s[i].age);  
    }  
}
```

從檔案開頭往後移了3個byte



```
C:\Users\user\Desktop\10_5\Debug\10_5....  
n a1035501 123  
張小名 a1035502 456  
王大明 a1035503 789  
請按任意鍵繼續 . . .
```



# 隨機存取的讀寫

- ❖ `fopen`時模式需用二進位模式(`wb/rb/ab...`)
- ❖ 透過`fseek`將檔案指標移動到要讀取/寫入的位置
- ❖ 再透過`fread/fwrite`進行讀取/寫入
- ❖ 什麼時候用隨機存取？常用在讀寫陣列 / `struct`的時候



# 隨機存取的讀寫

二進位模式讀寫檔案，不會發生字元轉換，按照資料需要的大小來讀出或寫入。

```
int fread(char *buffer, int size, int count, FILE *fp);
```

- buffer 參數為寫入的資料所要存放的記憶體位置

- size 每一筆資料的大小

- count 要寫入多少筆資料

- fp 開啟的檔案指標

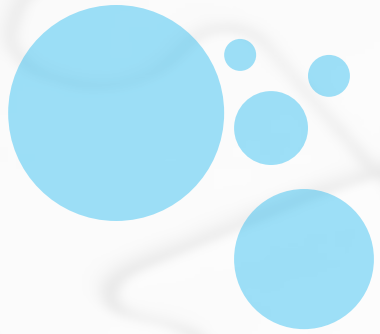
```
int fwrite(char *buffer, int size, int count, FILE *fp);
```

- buffer 參數為讀出的資料所要存放的記憶體位置

- size 每一筆資料的大小

- count 要寫入多少筆資料

- fp 開啟的檔案指標



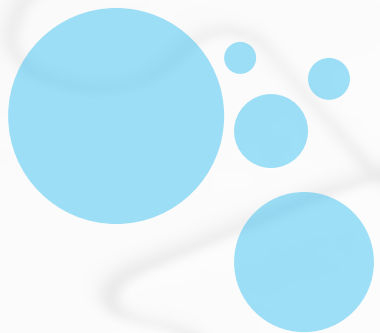
# 練習練習練習

一步一步做：

1. 建立一個ball的structure，包含球的顏色、編號、半徑。

```
struct ball
{
    char color[10];
    int num;
    float radius;
};
```





# 練習練習練習

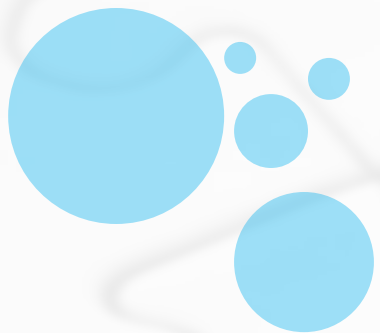
一步一步做：

2.先做初始化，讓使用者輸入要將幾筆初始化資料寫到檔案中，把它寫進去檔案裏面，寫入的方法大概像這樣：

```
fwrite(&b, sizeof(ball), 1, fptr);
```

sizeof()可以回傳一個資料型態/結構的大小

寫入成功後打開你的txt，看看它長什麼樣子呢？



## 練習練習練習

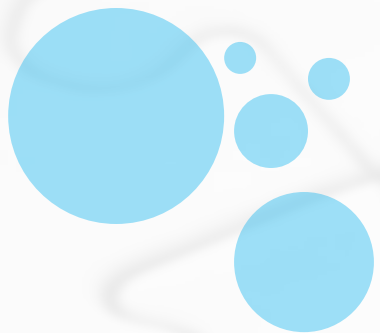
一步一步做：

3.先看看我們寫了什麼在txt裡面?用fread把我們寫的東西讀出來看看吧，讀取的寫法似乎長這個樣子：

```
fread(&b, sizeof(ball), 1, fptr);
```

搭配feof()服用：

`int feof(FILE* fp);`讀到檔案結尾時回傳非0值



# 練習練習練習

一步一步做：

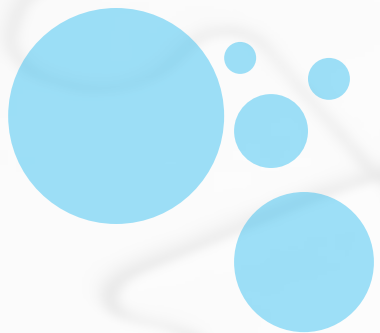
4. 試著隨機寫入吧！如果我想寫入球的號碼為3(第四筆)的資料呢？

檔案指標



第0筆資料(color、num、radius)	第1筆資料(color、num、radius)	第2筆資料(color、num、radius)	第3筆資料(color、num、radius)	...			
-------------------------	-------------------------	-------------------------	-------------------------	-----	--	--	--

必須先把檔案指標移到第四筆資料的開頭才能寫入，怎麼移呢？



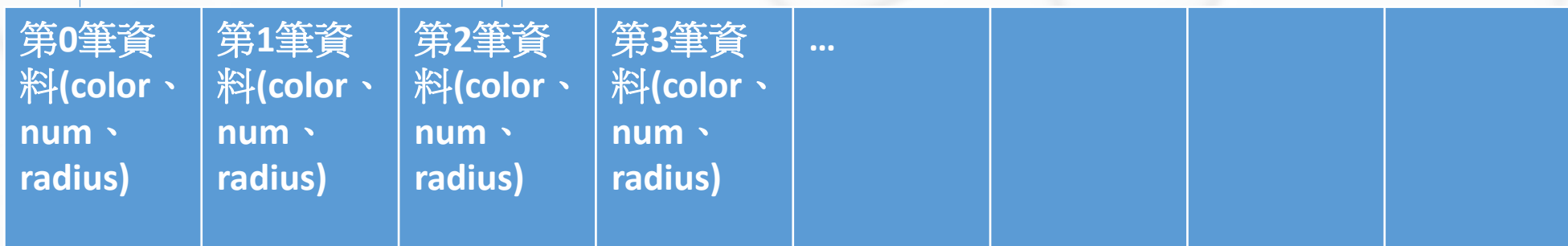
# 練習練習練習

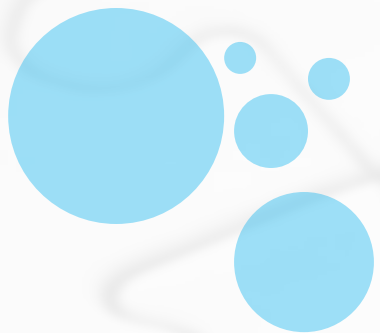
透過fseek()移動到正確的位置在寫入。

```
fseek( fptr, (4-1)*sizeof(ball), SEEK_SET); //移到第四筆檔案的開頭
```

前面有(4-1)筆資料  
，每筆資料的大小  
都是sizeof(ball)

檔案指標





## 練習練習練習

回家自己做：

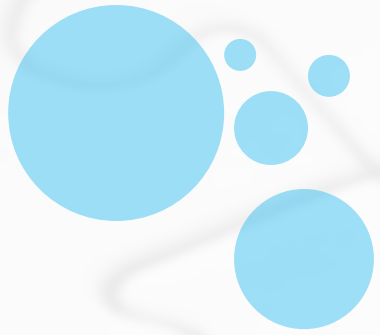
5. 把它寫活一點，讓使用者輸入我想要更新第幾筆資料就更新第幾筆，該怎麼寫呢？




## 練習練習練習

回家自己做：

6.都寫到這裡了再多寫一點，新增個查詢功能玩玩吧？  
讓使用者輸入想查詢第幾筆資料，如果資料存在  
(b.num != 0)就把它顯示出來。



## 參考資料

 <http://homepage.ntu.edu.tw/~d02922022/AdvC/ppt/Part%202/09%20Extra%20%E6%AA%94%E6%A1%88%E8%99%95%E7%90%86.pdf>



**THANK YOU**