

第十七章 檔案的處理

本章學習目標

- 學習檔案的觀念與操作的方式
- 學習文字檔與二進位檔案的使用方式
- 學習檔案指標的操作

Matlab常用的資料檔案格式

- Matlab支援四種常用的資料檔案格式，如下表所列：

表 17.1.1 檔案類型與相關資訊

檔案格式	說 明	讀取函數	寫入函數
MAT	儲存 Matlab 工作區的變數所產生的檔案	load()	save()
CSV	Comma-separated value，即以逗號隔開的數據檔案	csvread()	csvwrite()
DLM	Delimited text，以特定的分隔符號隔開的數據檔案	dlmread()	dlmwrite()
TAB	Tab-separated text，以 Tab 鍵隔開的數據檔案	dlmread()	dlmwrite()

表 17.1.2 存取工作區內的變數之函數

函 數	說 明
<code>save('filename','var₁','var₂',...)</code>	將工作區內的 <code>var₁</code> 、 <code>var₂</code> 、... 等變數儲存成檔名為 <code>filename</code> 的檔案內
<code>load('filename')</code>	從檔案 <code>filename</code> 讀取由 <code>save()</code> 函數所儲存的變數

存取工作區內的變數

```
>> prime_list=primes(100)
Columns 1 through 9
     2     3     5     7    11    13    17    19    23
Columns 10 through 18
    29    31    37    41    43    47    53    59    61
Columns 19 through 25
    67    71    73    79    83    89    97
```

```
>> mag=magic(5)
```

```
mag =
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
```

```
>> save('my_data','prime_list','mag')
```

```
>> clear all
```

```
>> whos
```

```
>> load('my_data')
```

```
>> whos
```

Name	Size	Bytes	Class
mag	5x5	200	double
prime_list	1x25	200	double

```
>> var=load('my_data')
```

```
var =
    prime_list: [1x25 double]
    mag: [5x5 double]
```

```
>> length(var.prime_list)
```

```
ans =
    25
```

```
>> m=load('my_data','mag')
```

```
m =
    mag: [5x5 double]
```

```
>> m.mag
```

```
ans =
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
```



以逗號隔開的數據處理

- 以逗號隔開的資料稱為 CSV(comma separated value)
- `csvread()` 與 `csvwrite()` 可用來存取CSV型態的資料

表 17.1.3 存取由逗號隔開的數據資料

函 數	說 明
<code>m=csvread('filename')</code>	讀取以逗號為分隔符號的數據資料，並以 double 型態儲存到變數 <i>m</i> 裡
<code>csvwrite('filename',m)</code>	將數據資料以 csv 的格式（即以逗號為數據的分隔符號）寫到檔案 <i>filename</i> 裡

```
>> A=magic(3)
```

```
A =
```

```
      8      1      6
      3      5      7
      4      9      2
```

```
>> csvwrite('csvdata.csv',A)
```

```
>> type csvdata.csv
```

```
8,1,6
3,5,7
4,9,2
```

```
>> B=csvread('csvdata.csv')
```

```
B =
```

```
      8      1      6
      3      5      7
      4      9      2
```



```
C=xlsread('ML_Ch17_Excel.xlsx')
sheet=1; xlRange='A2:C3';
D=xlsread('ML_Ch17_Excel.xlsx',sheet,xlRange)
E=xlsread('ML_Ch17_Excel.xlsx','table2','A2:C3')
[data1,text,all_data]=xlsread('ML_Ch17_Excel.xlsx','table2')
```

C =

1	2	3
4	5	6
7	8	9

E =

1.0000	80.0000	0.1000
2.0000	90.0000	2.0000

D =

4	5	6
7	8	9

data1 =

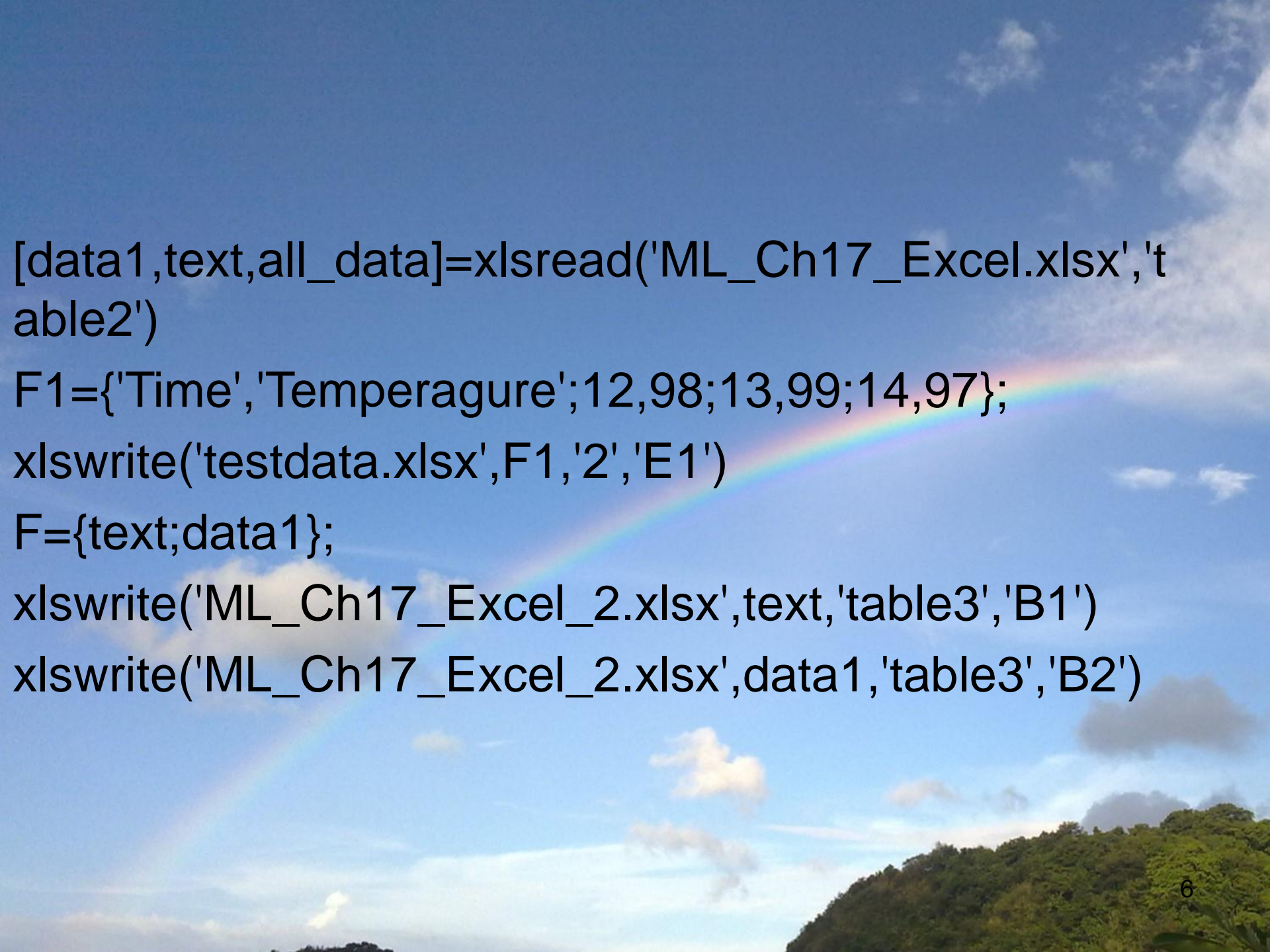
1.0000	80.0000	0.1000
2.0000	90.0000	2.0000

text =

'time'	'RH'	'rain'
--------	------	--------

all_data =

'time'	'RH'	'rain'
[1]	[80]	[0.1000]
[2]	[90]	[2]



```
[data1,text,all_data]=xlsread('ML_Ch17_Excel.xlsx','table2')  
F1={'Time','Temperagure';12,98;13,99;14,97};  
xlswrite('testdata.xlsx',F1,'2','E1')  
F={text;data1};  
xlswrite('ML_Ch17_Excel_2.xlsx',text,'table3','B1')  
xlswrite('ML_Ch17_Excel_2.xlsx',data1,'table3','B2')
```


以特定符號隔開的數據處理

- dlmread() 與dlmwrite()可存取不是以逗號分隔的數據
- dlm是delimiter的縮寫，分隔符號之意。

表 17.1.4 處理以特定符號隔開的數據

函 數	說 明
<code>m=dlmread('filename','dlm')</code>	讀取以 <i>dlm</i> 為分隔符號的數據資料，並以 double 型態儲存到變數 <i>m</i> 裡
<code>dlmwrite('filename',m,'dlm')</code>	以 <i>dlm</i> 為分隔符號來儲存數據資料 <i>m</i>

```
>> mat=[3 4 12;2 4 19]
```

```
mat =
```

```
3      4      12
2      4      19
```

```
>> dlmwrite('dlmfile.dlm',mat,'\t')
```

```
>> type dlmfile.dlm
```

```
3      4      12
2      4      19
```

```
>> dlmread('dlmfile.dlm','\t')
```

```
ans =
```

```
3      4      12
2      4      19
```

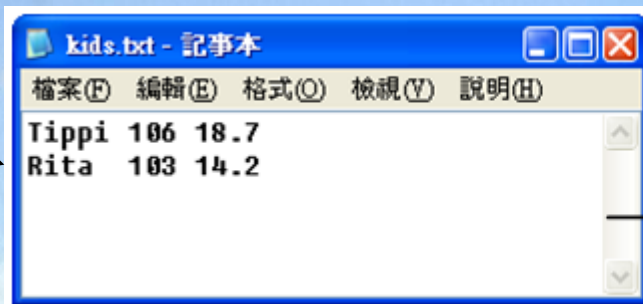
讀取摻雜文字與數據資料的檔案

- `textread()` 可讀取摻雜有文字與數字的文字資料檔

表 17.1.5 讀取摻雜文字與數據資料的檔案

函 數	說 明
<code>[a,b,...]=textread('fname','format')</code>	依 <i>format</i> 所記載的格式從檔案 <i>fname</i> 裡讀取資料。 <i>format</i> 常用的格式如下： %n - 可讀取整數或浮點數 %d - 讀取含正負號的整數 %f - 讀取浮點數的數據 %s - 讀取由空白鍵隔開的字串

先以記事本建立
一個文字檔



純文字檔 kids.txt，各項資料
請以空白鍵隔開

Textread() 的使用範例

```
>> type kids.txt
Tippi 106 18.7
Rita 103 14.2

>> [name height weight]=textread(
    'kids.txt','%s%d%f')
name =
    'Tippi'
    'Rita'
height =
    106
    103
weight =
    18.7000
    14.2000

>> whos name
Name      Size      Bytes   Class
name      2x1          138     cell

>> name{1}
ans =
Tippi

>> height(2)
ans =
103
```

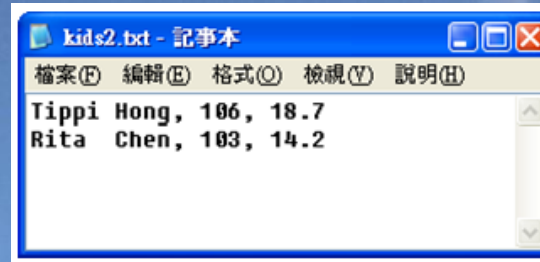


圖 17.13

純文字檔 kids2.txt 的內容

```
>> type kids2.txt
Tippi Hong, 106, 18.7
Rita Chen, 103, 14.2

>> [name height weight]=textread(
    'kids2.txt','%s%d%f','delimiter','(',')')
name =
    'Tippi Hong'
    'Rita Chen'
height =
    106
    103
weight =
    18.7000
    14.2000
```

開啟與關閉文字檔案

表 17.2.1 開檔與關檔的函數

函 數	說 明
<code>fid=fopen('filename','permission')</code>	讀取檔案的內容，其中 <i>filename</i> 為欲開啟的檔案名稱， <i>permission</i> 為檔案的存取模式，並傳回檔案識別碼，由變數 <i>fid</i> 接受
<code>fclose(fid)</code>	關閉檔案識別碼為 <i>fid</i> 的檔案

表 17.2.2 檔案存取模式

存取模式	代碼	說 明
讀取資料	r	開啟檔案以供讀取。在開啟前，此檔案必須先存在於磁碟機內。如果檔案不存在，則開檔失敗
寫入資料	w	開啟檔案以供寫入。如果檔案已經存在，則其內容將被覆蓋掉。如果檔案不存在，則系統會自行建立此檔案
附加於檔案之後	a	開啟一個檔案，可將資料寫入此檔案的末端。如果檔案不存在，則系統會自行建立此檔案
讀取與附加	a+	可讀取檔案，也可附加資料於檔案之後

寫入與讀取文字檔

- 寫入與讀取文字檔，可用 `fprintf()` 與 `fscanf()` 函數

表 17.2.3 檔案寫入與讀取函數

函 數	說 明
<code>fprintf(fid, 'str', e₁, e₂, ...)</code>	依格式字串 <i>str</i> 所記載的格式碼，依序將運算式 <i>e₁</i> , <i>e₂</i> 填入 <i>str</i> 裡，並將它寫入檔案識別碼為 <i>fid</i> 的檔案中。下面列出格式字串裡常用的格式碼： %d：寫入整數 %f：寫入浮點數 %c：寫入字元 %s：寫入字串
<code>fscanf(fid, 'str')</code>	依格式字串 <i>str</i> 所記載的格式碼，讀取檔案識別碼為 <i>fid</i> 之檔案裡的資料
<code>fscanf(fid, 'str', n)</code>	一次讀取 <i>n</i> 筆資料
<code>fscanf(fid, 'str', [m, n])</code>	一次讀取 <i>m</i> × <i>n</i> 筆資料，並以 <i>m</i> × <i>n</i> 的陣列回應讀取的結果

fprintf() 與 fscanf()的範例

```
>> fw=fopen('test.txt','w')
fw =
    3

>> fprintf(fw,'%d ',primes(20));
>> fclose(fw)
ans =
    0

>> type test.txt
2 3 5 7 11 13 17 19

>> fr=fopen('test.txt','r')
fr =
    3

>> fscanf(fr,'%d')
ans =
    2
    3
    5
    7
   11
   13
   17
   19
```

```
>> fscanf(fr,'%d')
ans =
    []

>> fclose(fr)
ans =
    0

>> fr=fopen('test.txt','r')
fr =
    3

>> fscanf(fr,'%d',2)
ans =
    2
    3

>> fscanf(fr,'%d',[2 3])
ans =
    5    11    17
    7    13    19

>> fscanf(fr,'%d')
ans =
    []
```


frewind() 與 feof() 函數

表 17.2.4 檔案讀取函數

函 數	說 明
<code>frewind(fid)</code>	設定檔案從頭讀取
<code>feof(fid)</code>	測試檔案是否已讀取到末端。若是，則回應 1，否則回應 0

請接續前面的範例

```
>> feof(fr)
ans =
    1

>> frewind(fr)

>> fscanf(fr, '%d', 2)
ans =
    2
    3

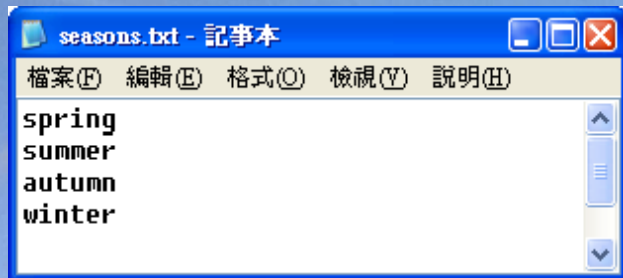
>> fclose(fr)
ans =
    0
```

一行一行讀取檔案

- `fgetl()` 或 `fgets()` 可一行一行的讀取檔案的內容

表 17.2.5 檔案讀取函數

函 數	說 明
<code>fgetl(fid)</code>	從檔案讀取一行字串，但不會讀取換行字元
<code>fgets(fid)</code>	從檔案讀取一行字串，連同換行字元也一併讀入



```
>> fr=fopen('seasons.txt','r');
>> s1=fgetl(fr)
s1 =
spring
```

```
>> s2=fgets(fr)
```

```
s2 =
summer
```

```
>> double(s1)
```

```
ans =
    115    112    114    105    110    103
```

```
>> double(s2)
```

```
ans =
    115    117    109    109    101    114    13    10
```

```
>> fclose(fr)
```

```
ans =
    0
```


KL_Tide_1.txt

基隆逐時潮位觀測資料

位置：25°09'18"N 121°45'08"E；基隆港西33號碼頭

儀器型式：Aquatrak 4100 series 超音波式

潮高基準相對臺灣高程基準(TWVD2001):+0.000cm

資料來源：氣象局

yyyymmddhh：西元年月日時

height：潮高

潮高單位：公厘。

時間不連續，表示缺觀測資料。

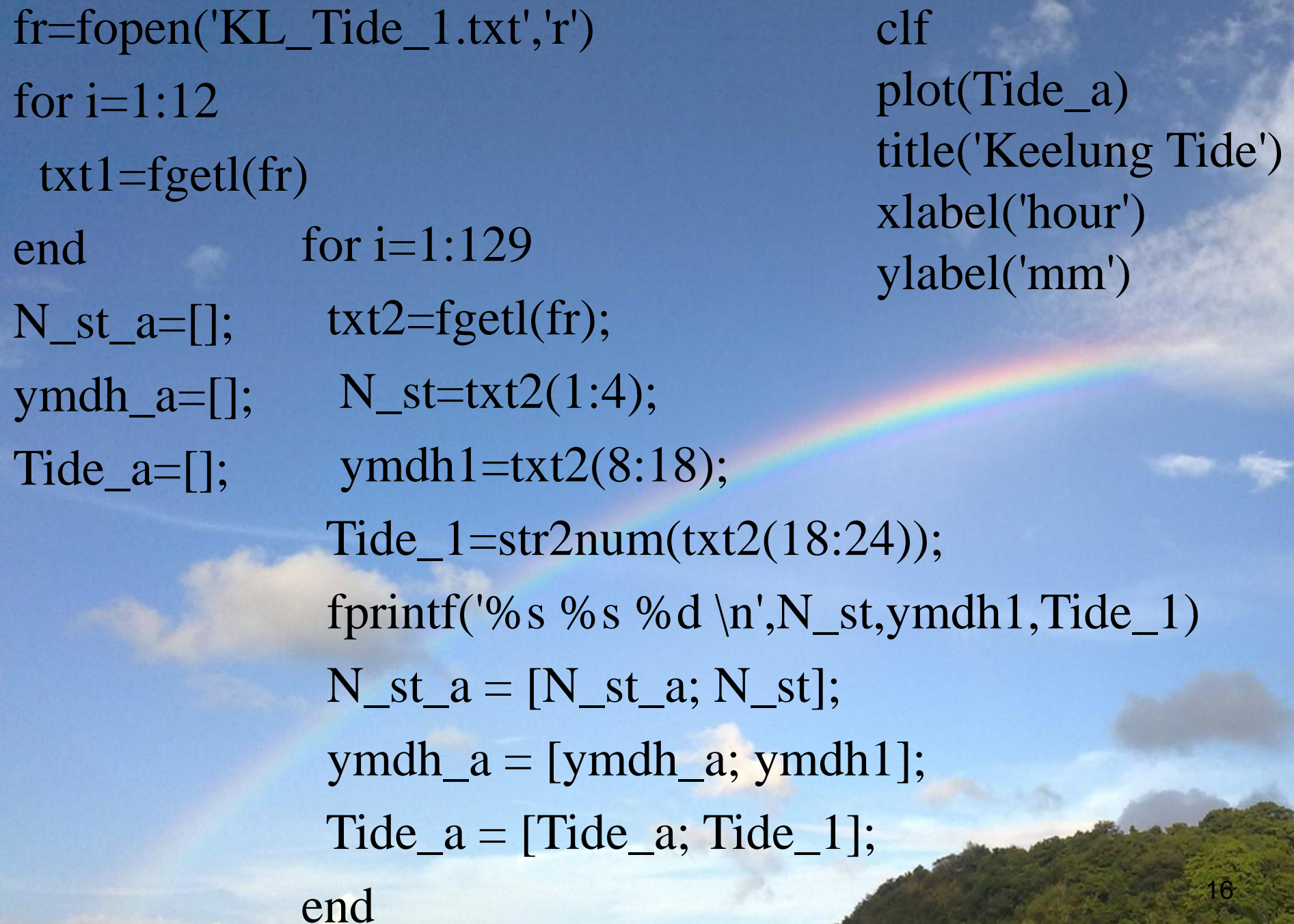
時間：民國103年1月—民國103年12月

*st yyyymmddhh height

1516 2014010100 -494 Q

1516 2014010101 -761 Q

1516 2014010102 -861 Q



```
fr=fopen('KL_Tide_1.txt','r')
for i=1:12
    txt1=fgetl(fr)
end
N_st_a=[];
ymdh_a=[];
Tide_a=[];
for i=1:129
    txt2=fgetl(fr);
    N_st=txt2(1:4);
    ymdh1=txt2(8:18);
    Tide_1=str2num(txt2(18:24));
    fprintf('%s %s %d \n',N_st,ymdh1,Tide_1)
    N_st_a = [N_st_a; N_st];
    ymdh_a = [ymdh_a; ymdh1];
    Tide_a = [Tide_a; Tide_1];
end
```

```
clf
plot(Tide_a)
title('Keelung Tide')
xlabel('hour')
ylabel('mm')
```




KL_Tide_2

1516 2014010100 -494 Q

1516 2014010101 -761 Q

1516 2014010102 -861 Q

1516 2014010103 -852 Q

1516 2014010104 -743 Q

1516 2014010105 -531 Q

1516 2014010106 -308 Q

1516 2014010107 -75 Q

clear all

[N_st ymdh tide qc]=textread('KL_Tide_2.txt','%d%s%d%s')

clf

plot(tide)

xlabel('hour')

ylabel('mm')

二進位檔案的處理

表 17.3.1 二進位檔案寫入與讀取函數

函 數	說 明
<code>A=fread(fid,size,precision)</code>	讀取 <i>size</i> 個 <i>precision</i> 大小的二進位資料，其中 <i>precision</i> 的說明請參閱表 17.3.2
<code>fwrite(fid,A,precision)</code>	以指定的 <i>precision</i> 大小，將陣列 <i>A</i> 的內容寫入檔案 <i>fid</i> 中

表 17.3.2 可供使用的 *precision* 字串

precision	說 明	所佔位元
'schar'	有號的字元	8 bits
'uchar'	無號的字元	8 bits
'int8'	8 bits 整數	8 bits
'int16'	16 bits 整數	16 bits
'int32'	32 bits 整數	32 bits
'int64'	64 bits 整數	64 bits
'uint8'	8 bits 無號整數	8 bits

precision	說 明	所佔位元
'uint16'	16 bits 無號整數	16 bits
'uint32'	32 bits 無號整數	32 bits
'uint64'	64 bits 無號整數	64 bits
'float32'	32 bits 的浮點數	32 bits
'float64'	64 bits 的浮點數	64 bits
'double'	倍精度浮點數	64 bits

讀取與寫入二進位檔

```
>> fid=fopen('data.bin','wb');
```

```
>> magic(5)
```

```
ans =
```

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

```
>>
```

```
cnt=fwrite(fid,magic(5),'int16')
```

```
cnt =
```

```
25
```

```
>> fclose(fid)
```

```
ans =
```

```
0
```



data.bin
BIN 檔案
1 KB

類型: BIN 檔案
修改日期: 2013/7/25 下午 05:22
大小: 50 個位元組

```
>> fr=fopen('data.bin','rb');
```

```
>> dat=fread(fr,3,'int16')
```

```
dat =
```

```
17  
23  
4
```

```
>> fread(fr,[2 4],'int16')
```

```
ans =
```

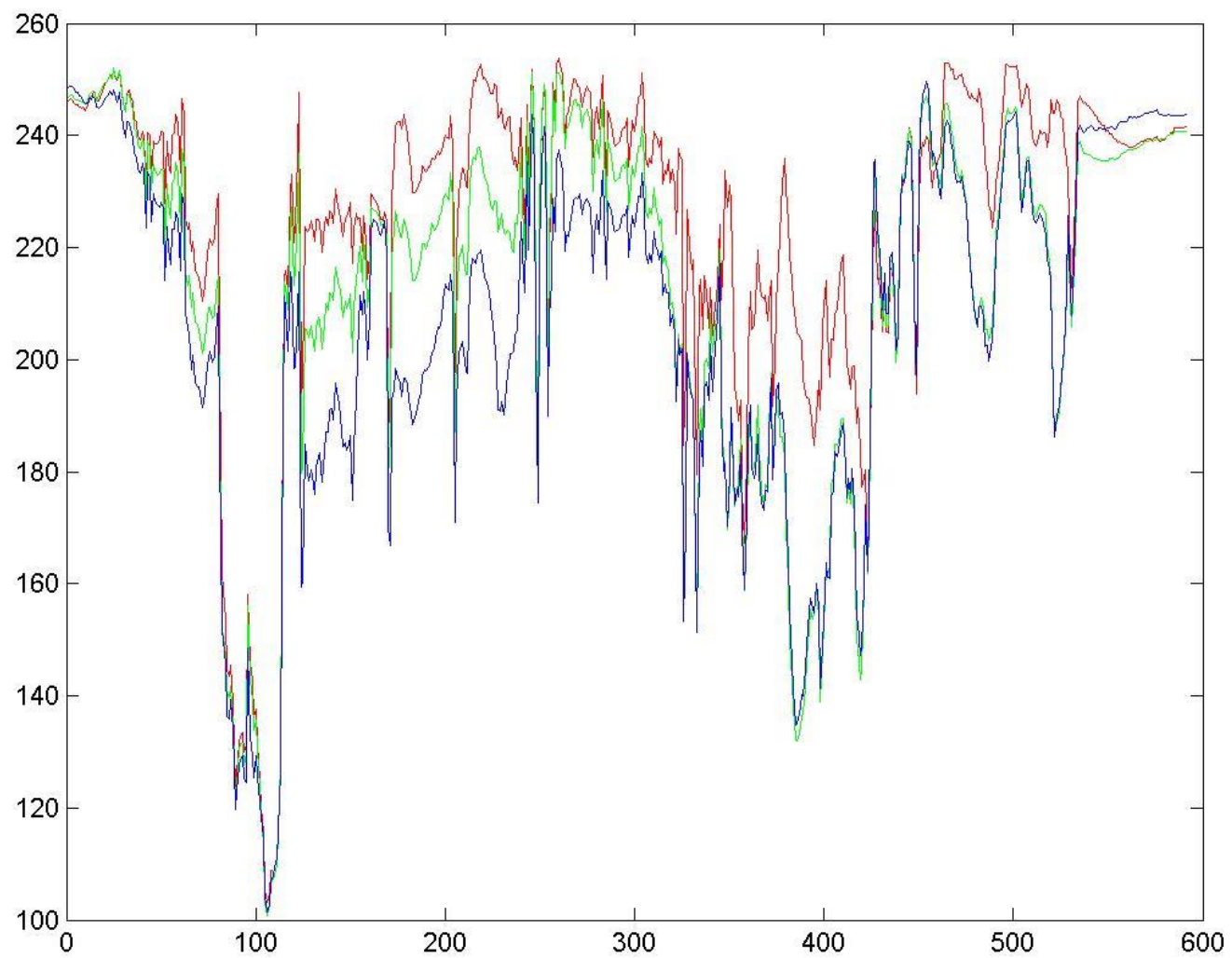
10	24	6	18
11	5	12	1

```
>> fclose(fr);
```



```
clear all
aa=imread('invitation.jpg');
imshow(aa)
size(aa)
clf
imshow(aa(100:150,,:))

avg_aa=mean(aa(100:150,,:));
size(avg_aa)
clf
plot(avg_aa(1, :, 1), 'r')
hold on
plot(avg_aa(1, :, 2), 'g')
hold on
plot(avg_aa(1, :, 3), 'b')
```

控制檔案的指標位置

- 讀取檔案時，Matlab會以一個檔案指標(file indicator)來記錄現在應該是哪一筆資料要被讀取

表 17.3.3 更改指標所指向位置的函數

函 數	說 明
<code>frewind(fid)</code>	將指標移到檔案的最開頭，也就是設定檔案可從頭讀取
<code>fseek(fid, offset, origin)</code>	設定檔案指標所在的位置，其中 <i>fid</i> 是檔案識別碼， <i>offset</i> 是偏移量（以 byte 為單位），而 <i>origin</i> 則代表 <i>offset</i> 的基準點
<code>ftell(fid)</code>	取得檔案指標的值，此值是以從檔案起始到指標目前的位置共有多少個 bytes 來計算

*origin*可以是下列字串之一：

'cof'：指標目前的位置（current position of the opened file）

'bof'：檔案起始的位置（beginning of the file）

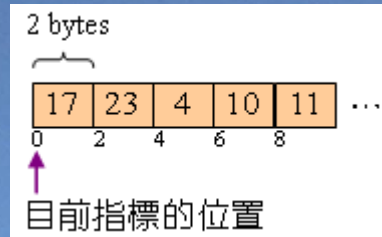
'eof'：檔案的結束位置（end of the file）

檔案讀取流程的範例

```
>> fr=fopen('data.bin','rb');
```

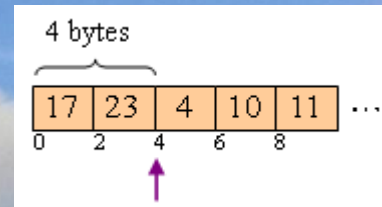
```
>> ftell(fr)
```

```
ans =  
0
```



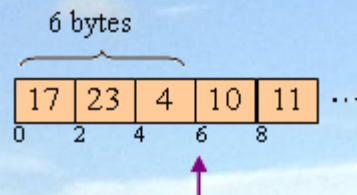
```
>> fseek(fr,4,'bof')
```

```
ans =  
0
```



```
>> fread(fr,1,'int16')
```

```
ans =  
4
```

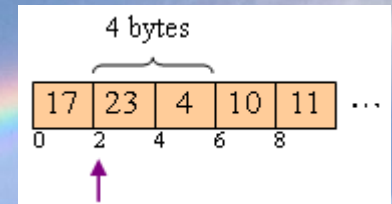


```
>> ftell(fr)
```

```
ans =  
6
```

```
>> fseek(fr,-4,'cof')
```

```
ans =  
0
```

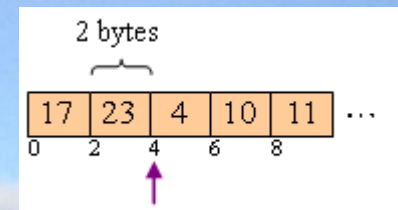


```
>> ftell(fr)
```

```
ans =  
2
```

```
>> fread(fr,1,'int16')
```

```
ans =  
23
```



```
>> fclose(fr);
```