MATLAB programming
Cell array(異質陣列)

Contain

- Main purpose
  - Store (varying data types) in a single matrix
  - Access by index
- Main issues
  - (build-up) cell array
  - (display) cell array
  - (Access) cell array

(build-up) cell array - Cell Indexing

- Method：Define the contains of the cell array by {} & using index
- Ex: cell01.M

```
A(1,1) = {'This is the first cell.'};
A(1,2) = {[5+j*6 , 4+j*5]};
A(2,1) = {[1 2 3; 4 5 6; 7 8 9]};
A(2,2) = {{'Tim'; 'Chris'}}
```

Cell-Indexing (cont)

- Build up 2-Dim cell array A:2*2 with the contains:

A(1,1)： 'This is the first cell 字串	A(1,2)： [5+j*6 4+j*5] 1*2複數陣列
A(2,1)： 1 2 3 4 5 6 7 8 9 3*3整數陣列	A(2,2)： {'Tim', 'Chris'} 2*2異質陣列

(build-up) cell array - Content Indexing

- 做法：matrix use () for addressing (index) → cell array use {} for addressing Ex: cell02.m

```
A{1,1} = 'this is the first cell.';
A{1,2} = [5+j*6, 4+j*5];
A{2,1} = [1 2 3; 4 5 6; 7 8 9];
A{2,2} = {'Tim'; 'Chris'}
```

Same results as previous definition

(build-up) cell array – use big parantheses {}

- Directly use { } to include all elements
- Ex:

```
>> B = {'James Bond', [1 2;3 4;5 6]; pi, magic(5)}
>> C = {rand(3), ones(2); zeros(5), randperm(4)}
```

```
B =     'James Bond'    [3x2 double]
        [    3.1416]    [5x5 double]


C =     [3x3 double]    [2x2 double]

        [5x5 double]    [1x4 double]
```

Merge of cell arrays

- Merge of cell arrays by using [ ]
- examples (B,C as before) ：

```
>> M = [B C]    % merge [ B C ] by left-right arrangement
M =
'James Bond'    [3x2 double]    [3x3 double]    [2x2 double]
[    3.1416]    [5x5 double]    [5x5 double]    [1x4 double]
```

Display cell array

- Matlab usually display the data types of the cell array instead of its values
- 範例：

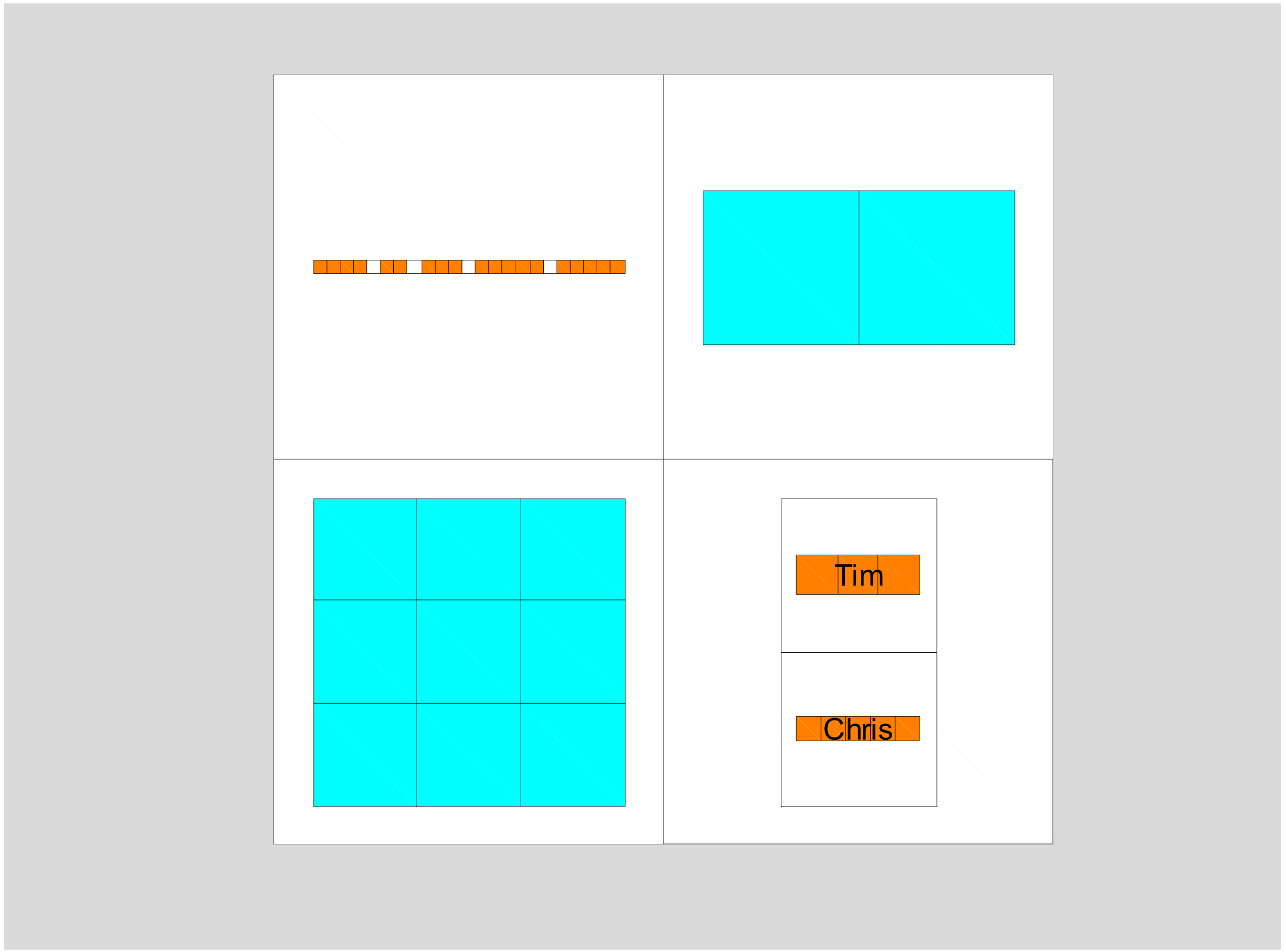
```
>> A
A =
'this is the first cell.'     [1x2 double]
              [3x3 double]    {2x1 cell  }
```

Cellplot graph-type

- cellplot -- display the datatype of the cell array
- Ex:cell03.m

```
A{1,1} = 'this is the first cell.';
A{1,2} = [5+j*6, 4+j*5];
A{2,1} = [1 2 3; 4 5 6; 7 8 9];
A{2,2} = {'Tim'; 'Chris'};
cellplot(A)        % 以圖形的方式顯示異質陣列 A 的內部資料型態
```

Result



celldisp

```
Ex:
>> celldisp(A)              % display the values of the cell elements
```

```
A{1,1} =
this is the first cell.
A{2,1} =
    1     2     3
    4     5     6
    7     8     9
A{1,2} =
   5.0000 + 6.0000i   4.0000 + 5.0000i
A{2,2}{1} =
Tim
A{2,2}{2} =
Chris
```

Content Indexing

- Content Indexing，
- methods:
  - Show the contents of the cell array A, or display the contents of the element by using indexing & {}

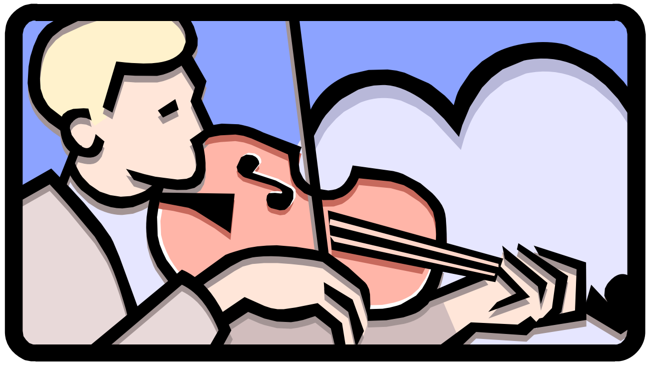
```
>> A{:}
>> A{1,2}
```

Result

```
ans =
this is the first cell.
ans =
     1     2     3
     4     5     6
     7     8     9
ans =
   5.0000 + 6.0000i    4.0000 + 5.0000
ans =
  'Tim'
  'Chris'
```

access the contains of the cells

- Direct access to cells
- Takes the internal contains of the cell from the outer cell
- Access or delete multiple elements at once
- Cell can replace comma-separated (;) variable columns



Directly access of cell (1/2)

- The example first creates a cell array B, and then takes the elements in the first row and the second row:

```
>> B = {'James Bond', [1 2;3 4;5 6]; pi, magic(5)}

B =

    'James Bond'    [3x2 double]
    [    3.1416]    [5x5 double]
```

Hint: magic(n)

```
>> F = B{1,2}
% Take the elements of the 1st row and the 2nd row of the
   cell array B
```

Result:

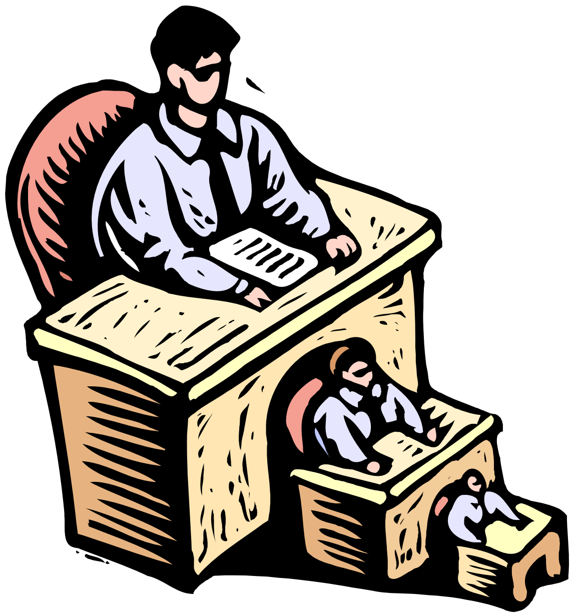
```
F =
        1    2
        3    4
        5    6
```

- If the element is a matrix then first Content Indexing then matrix indexing
- Ex:

```
>> G = B{1,2}(3,1)
```

```
G =

     5
```



Access or delete multiple elements

- B cell array as before

```
>> H = B(2,:)        % access 2-nd row of B

H =

    [3.1416]    [5x5 double]


>> B(1,:) = []        % delete first row of B cell array

B =

    [3.1416]    [5x5 double]
```

separate by comma ` , `

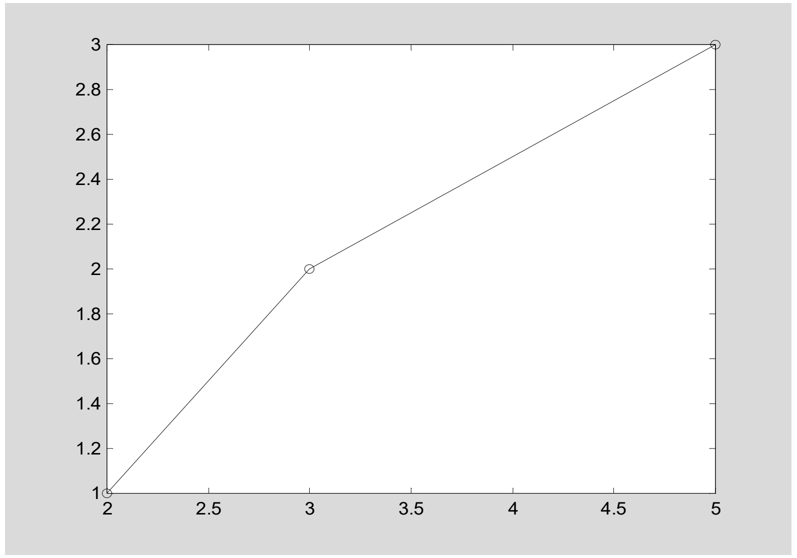
Ex: % build up a 1×4 cell array (separate by comma ` , `

```
>> F = {[2 3 5], [1 2 3], 'Timmy', 'Annie'};
>> F{1:2}
ans =
     2     3     5
ans =
     1     2     3
```

- As previous, F{1:2} with contents「[2 3 5], [1 2 3]」
- Use cell as the input variables ，Example：

```
>> plot(F{1:2}, '-o')
```

Result:



- Also, cell array can be used as the output variables
- Ex:

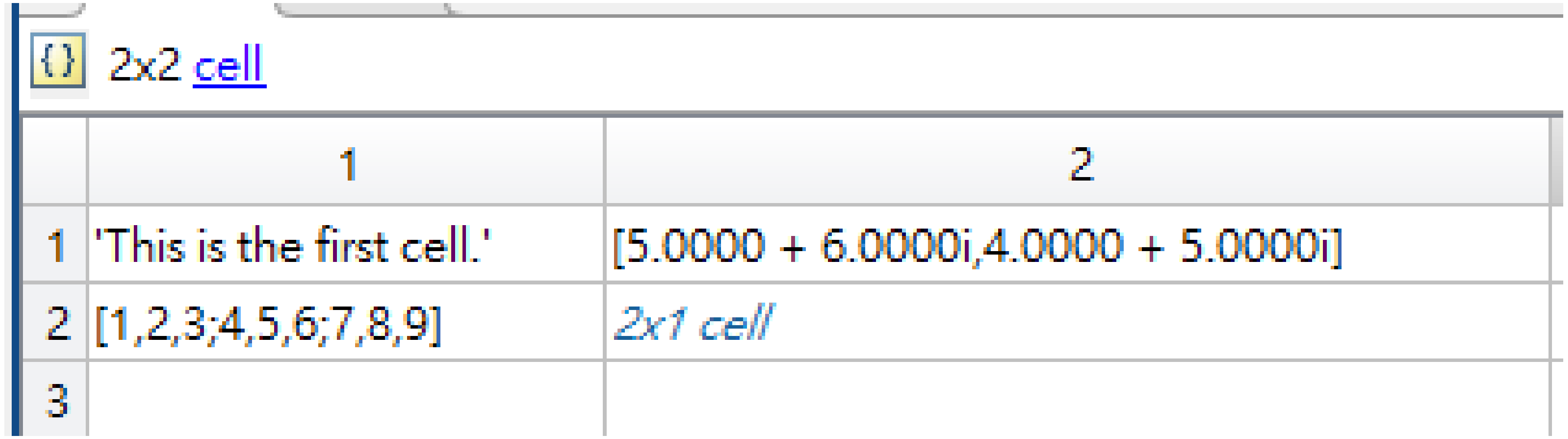
```
>> [F{1:2}] = max(rand(5))
```

```
F =

[1x5 double]    [1x5 double]

          %F為一1x2的異質陣列
```

Exercise 1



2x2 cell

	1	2
1	'This is the first cell.'	[5.0000 + 6.0000i,4.0000 + 5.0000i]
2	[1,2,3;4,5,6;7,8,9]	*2x1 cell*
3		

(1) Get the array element A(2,2)
(2) Get the value of the array element to another cell array : C
(3) Display the cell array
(4) Change the values of the cell array
(5) Build up an empty 3*2 cell array : D
(6) Put the cell array A to part of D.

Other related instructions

- Change the shape of the cell array
- （Pre-Allocate) zero cell array
- Check whether it is a cell array or not
- Transfer number array to cell array
- Transfer (Structure) in all field to cell array
- Transfer cell array to (Structure)
- Transfer (Structure) in certain field to cell array
- 將一結構陣列的某一欄位值轉換成異質陣列

Change the shape of the cell array

- 「reshape」
- Ex: B為2x2異質陣列

```
B =
        'James Bond‘     [3x2 double]
        [    3.1416]     [5x5 double]
```

- Result:

```
>> N = reshape(B,1,4)
N =
    'James Bond'    [3.1416]    [3x2 double]    [5x5 double]
```

```
>> M = B(:)
M =
    'James Bond'
    [    3.1416]
    [3x2 double]
    [5x5 double]
```

（Pre-Allocate) zero cell array

- Use cell
- Ex:

```
>> E = cell(4, 3)
        %  （Pre-Allocate) zero 4*3 cell array
```

- Result:

```
E =

    []    []    []
    []    []    []
    []    []    []
    []    []    []
```

Check whether it is a cell array or not

- iscell Ex:iscell01.m

```
C = {[1,2,3]; 'This is a test.'};


iscell(C)
```

```
ans =

    1
```

Hint: here，iscell is ture return 1，if not reture 0。

transfer numeric array to cell array(1/5)

- num2cell : transfer numeric array to cell array ：
  - C = num2cell(A, dim)
- dim : means didn’t change the dimension，other it will assign to the contents to a 1×1 cell array

- Ex:num2cell01.m

```
A = [1 2 3;4 5 6];    % 建立一個數值陣列 A
C = num2cell(A)       % 將數值陣列 A 轉成異質陣列 C
```

```
C =

    [1]    [2]    [3]
    [4]    [5]    [6]
```

- transfer each column of numeric array A to cell array D
- Ex:：num2cell02.m

```
A = [1 2 3;4 5 6];      % 建立的一個數值陣列 A
D = num2cell(A, 1)      % 1 代表「橫列被切割」
```

```
D =
[2x1 double]    [2x1 double]    [2x1 double]
```

- transfer each row of numeric array A to cell array D
- Ex:num2cell03.m

```
A = [1 2 3;4 5 6];        % 建立的一個數值陣列 A
E = num2cell(A, 2)        % 2 代表「直行被切割」
```

```
E =
        [1x3 double]
        [1x3 double]
```

將數值陣列轉換成異質陣列 (5/5)

- 若需要更複雜的轉換，可以使用mat2cell指令
- Ex:

```
X = [1 2 3 4; 5 6 7 8; 9 10 11 12]
C = mat2cell(X,[1 2],[1 3])
```

```
X =                              C =
      1    2    3    4               [          1]    [1x3 double]
      5    6    7    8               [2x1 double]    [2x3 double]
      9   10   11   12
```

Convert Cell Array to Numeric Array

- Convert numeric arrays in four cells of a cell array into one numeric array.
- C = {[1], [2 3 4];
      [5; 9], [6 7 8; 10 11 12]};

```
C=2x2 cell array
          {[          1]}    {[   2 3 4]}
          {2x1 double}       {2x3 double}
```

- A = cell2mat(C)

```
A = 3x4

       1      2      3      4
       5      6      7      8
       9     10     11     12
```

- struct2cell transfer structure to cell 。
- Ex:struct2cell01.m

```
student.name = 'Tim';
student.age = 8;          %student為一struct
sCell = struct2cell(student)
```

```
sCell =
      'Tim'
      [  8]
```

cell2struct in the next chapter

Transfer one field of the struct to a cell

- deal 指令
- Ex:deal01.m

```
S = struct('name',{'Tim','Annie'},'age', {8,5});
[sCell{1:length(S)}] = deal(S.name)
```

```
sCell =
        'Tim '
        'Annie'
```

deal 指令

- deal 指令的輸入及輸出引數需有相同個數
- 如上例中，sCell{1:length(S)} 的作用即是產生以逗點分開的輸出變數列。
- 善用 deal 指令，可達成事半功倍之效，
- 以下欲列出 MATLAB 根目錄下所有目錄（不含檔案）

本章指令彙整

指令	功能
cellplot(A)	以圖形的方式顯示異質陣列 A 的內部資料型態
celldisp(A)	顯示異質陣列 A 各個構成元素的實際內容
reshape(A,m,n)	改變異質陣列 A 的維度成 m×n
cell(m,n)	預先配置一個維度為 m×n 的空矩陣
iscell(A)	測試變數 A 是否為異質陣列： 傳回值 0，代表不是異質陣列 傳回值 1，代表是異質陣列
num2cell(A, d)	將數值陣列 A 轉成異質陣列，d 表示被切割的維度
struct2cell(A)	將結構陣列 A 的所有欄位名稱值，轉換成異質陣列

MATLAB :Structure

13-1 build up a structure array

- a（Structure Array）contains multiple elements (data) 。
- Each element（or each data）contains several fields (Fields） ，each field can have different data types。For examples: element contain the student data including : name、id、scores。
- Access by Fields

Example

- Define the values of each field for the element (structure element) 。
- 範例13-1 : struct01.m

```
clear student                         % clear student 變數
student.name = '洪鵬翔';              % add name field
student.id = 'mr871912';              % add id field
student.scores = [58, 75, 62];        % add scores field
student                               % show result
student =
      name: '洪鵬翔'
        id: 'mr871912'
    scores: '[58,75,62]‘
```

- Here : student is the first element of a structure array

結構陣列之範例二

- 範例13-2：struct02.m

```
clear student                                    % 清除 student 變數
student.name = ‘洪鵬翔’ ;                        % 加入 name 欄位
student.id = ‘mr871912’;                         % 加入 id 欄位
student.scores = [58, 75, 62];                   % 加入 scores 欄位
% 以下是新加入的第二筆資料
student(2).name = '邱中人';
student(2).id = 'mr872510';
student(2).scores = [25, 36, 92];
student                                          % 秀出結果
```

```
student =
1x2 struct array with fields:
    Name
    Id
    scores
dent =
1x2 struct array with fields:
    Name
    Id
    scores
```

結構陣列之範例二

- At this point student represents a 1x2 structure array 。
  For display the certain element ， 例如 student(2).scores 等。
- Other way to buil up the structure struct with the format：
  structureArray = struct(field1, value1, field2, value2,….)
- Where field1, field2, ... are the field names, and value1, value2, ... are the data contained in the field.
- If value1, value2, . 。If value1、value2、…are cell array, see the previous chapter for details), then MATLAB sequentially sets each element of the cell array to the corresponding column value in each structure, as shown in the following example.

結構陣列之範例三

- 範例13-3 : struct03.m

```
student = struct('name', {'張庭碩', '張庭安'}, 'scores', {[50 60], [60 70]});
student(1)                % 顯示 student(1)
student(2)                % 顯示 student(2)
ans =
        name: '張庭碩 ‘
        scores: [50 60]
ans =
        name: '張庭安'
        scores: [60 70]
```

- In this example，{ ‘張庭碩’ ， ‘張庭安’ } 和 {[50 60], [60 70]} are cell array ， So each of their elements will be set to each structure in turn. But if there is an outlier array whose length is 1 ，那麼 MATLAB "Scalar Expansion" will be performed to automatically complement, as shown in the following example 。

結構陣列之範例四

- 範例13-4：struct04.m

```
student = struct('name', '張庭安 ‘,'scores', {[50 60], [90 100]});
student(1)                    % 顯示 student(1)
student(2)                    % 顯示 student(2)
    ans =
        name: '張庭安'
        scores: [50 60]
    ans =
        name: '張庭安'
        scores: [90 100]
```

- n the above example, 「張庭安」 can be regarded as an element of the cell array, so when setting it to the student structure array, MATLAB will perform scalar expansion and set 「張庭安」 to the value of the name field of the two elements of student.

Structure example 五

- Structure array can be Nested）的，也就是說，結構陣列的欄位可是另一個結構陣列，我們可以藉此產生複雜的資料結構
- 範例13-5 : struct05.m

```
student = struct('name', {'張庭碩', '張庭安'}, 'scores', {[50 60], [60
        70]});
student(2).course(1).title = 'Web Programming';
student(2).course(1).credits = 2;
student(2).course(2).title = 'Numerical Method';
student(2).course(2).credits = 3;
student(2).course
```

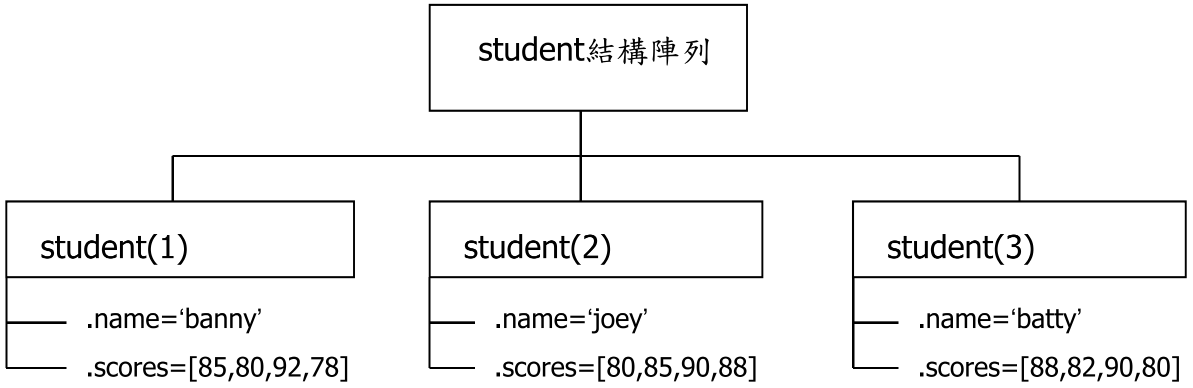
```
ans =
1x2 struct array with fields:
        title
credits
```

13-2 access and change the data in the dtructure

- 範例13-6 : buildStruct01.m

```
clear student                    % 清除 student 變數
student(1) = struct('name', 'Banny', 'scores', [85,80,92,78]);
student(2) = struct('name', 'Joey', 'scores', [80,85,90,88]);
student(3) = struct('name', 'Betty', 'scores', [88,82,90,80]);
```

- 上述的 student 結構陣列，可圖示如下：



struct2cell指令

- To get data for all fields in all elements in the structure array 可用 struct2cell 指令，例如：

```
>> values = struct2cell(student)
values(:,:,1) =
    'Banny'
    [1x4 double]
values(:,:,2) =
    'Joey'
    [1x4 double]
values(:,:,3) =
    'Betty'
    [1x4 double]
```

- Note that the returned values is a cell array. Generally speaking, if the dimension of the structure variable input to the struct2cell command is mxn and contains p fields, the dimension of the returned cell array is pxmxn 。（在上例中，p = 2，m = 1， n = 3。）

Access by Fields name

- In the structure array, we can use "period" (".") to find out the value of a specific field in a certain structure element, for example, we only want to see who the second student is, at this time we can enter：

```
>> studentName = student(2).name
studentName =
Joey
```

- In the above example, add a period after the second array element student(2) of this student structure array, and then connect the name field name to obtain the actual name data of this student, Joey. In this example, further Save the extracted student name Joey in the user-defined variable student Name and display it on the screen
- Similar to accessing the data content of individual fields in the structure array, we can change the data content of individual fields in the structure array：

```
>> student(2).name = 'Alex';
```

- student(2) change from Joey to Alex。

cat指令

- MATLAB MATLAB provides the ‘cat’ command to achieve the purpose of "side-by-side field values". Its syntax is: ：

  A = cat(dim, structureField)

  其中，dim Represents the dimension after side by side. For example, if you want to place the quiz scores side by side (horizontal), you can enter：

```
>> cat(2, student.scores)        % 2 代表左右並排以改變直行的維度
ans =
   85   80   92   78   80   85   90   88   88   82   90  80
```

- To place the quiz scores up and down (vertically) side by side, you can enter ：

```
>> cat(1, student.scores)        % 1 for side-by-side to change the
dimension of the row
ans =
     85    80    92    78
     80    85    90    88
     88    82    90    80
```

To calculate the average score for each test (out of four), enter

計算平均

- When performing "side-by-side", it must be confirmed that the values of the side-by-side fields have the same number of rows (up and down) or columns (side-by-side), otherwise an error message will be generated due to inconsistent dimensions 。
- To calculate the average score for each test (out of four), enter ：

```
>> average1 = mean(cat(1, student.scores))
average1 =
   84.3333   82.3333   90.6667   82.0000
```

- To calculate the average score of each student (three in total), you can enter the following expression, but pay attention to the use of single quotation marks after the expression to represent the transpose of the matrix, because the mean command defaults to each constant of the matrix. row average ：

```
>> average2 = mean(cat(1,student.scores)')
average2 =
   83.7500   85.7500   85.0000
```

side-by-side operation

- Since "side-by-side operation" is often used, MATLAB provides the following two methods
- Square bracket operation [ ]: You can combine the numerical matrix of the same column in the structure array left and right to generate a new numerical matrix.
- Braces operation { } : You can combine the data of the same column in the structure array left and right to generate an array matrix.

：
```
>> allScores = [student.scores]
allScores =
   85   80   92   78   80   85   90   88   88   82   90  80
```

- To extract the value of the name field to form an cell array consisting of strings, enter the following：

```
>> allNames = {student.name}
allNames =
 'Banny'    'Alex'    'Betty'
```

範例13-7：printStruct01.m

```
clear student                      % clear  student 變數
student(1) = struct('name', '張庭碩', 'scores', [85, 80]);
student(2) = struct('name', '鍾書蓉', 'scores', [80, 85]);
student(3) = struct('name', '黃念中', 'scores', [88, 82]);
for i = 1:length(student)          % 列印出每個學生的名字
    fprintf ('student %g: %s\n', i, student(i).name);
end
```

```
student 1: 張庭碩
student 2: 鍾書蓉
student 3: 黃念中
```

Access and change the content of the field

- One can also use getfield及setfield to access or change the data of a field, as the format下：

  fieldValues = getfield (structureArray, {arrayIndex}, field, {fieldIndex})

  newStructure = setfield (structureArray, {arrayIndex}, field,{fieldIndex})
- Enter the following formula to get the first quiz score of the second student：

  >> **score3 = getfield(student, {2}, 'scores', {1})**

  or simply

  >> **score3 = student(2).scores(1);**
- f you want to change the first quiz score of the second student, you can enter the following：

  >> **student = setfield(student, {2}, 'scores', {1}, 75);**

  or simply

  >> **student(2).scores(1)=75;**

Example 8

範例13-8:deal01.m

```
myStruct = struct('name', {'Tim', 'Annie'}, 'age', {10, 13});
[myStruct.name] = deal('Roger', ‘Tom');
fprintf('myStruct(1).name = %s\n', myStruct(1).name);
fprintf('myStruct(2).name = %s\n', myStruct(2).name);
```

```
myStruct(1).name = Roger
myStruct(2).name = Tom
```

13-3 access and change the field name

- Use the fieldnames command to return all the field names of of structures, e.g. ：

```
student = struct('name', 'Roland', 'scores', [80, 90]);
allFields = fieldnams(estudent)
allfields =
 'name'
 'scores'
```

- The returned result is a Cell Array of Strings, containing all the fields of student
- To add a new field, directly add this field to any array element

Structure array

- 範例13-10:addField01.m

```
clear student                                          % 清除 student 變數
student = struct('name', 'Roland', 'scores', [80, 90]);
student(2).age = 20;                 % 加入新欄位
student(1)                                             % 顯示 student(1)
student(2)                                             % 顯示 student(2)
```

```
ans =
      name: 'Roland'
    scores: [80 90]
       age: []
ans =
      name: []
    scores: []
       age: 20
```

- As can be seen from the above results, MATLAB will add this new field to other elements and set its default value to [] (empty matrix)

- 範例13-11:rmField01.m

```
student = struct('name', 'Roland', 'scores', [80, 90])
student2 = rmfield(student, 'scores')  % 刪除 scores 欄位
```

```
student =
      name: 'Roland'
      scores: [80 90]
student2 =
    name: 'Roland'
```

13-4 other related command

- We can use isstruct to identify the structure array：
- 範例13-12 : isstruct01.m

```
s = struct('name', {'Tim', 'Ann'}, 'scores', {[1 3 5 ],[2 4 6]});
isstruct(s)
ans =
     1
```

- isfield identify whether it is a field：
- 範例13-13 : isstruct02.m

```
s = struct('name', {'Tim', 'Ann'}, 'scores', {[1 3 5 ],[2 4 6]});
fprintf('isfield(s, "name") = %d\n', isfield(s, 'name'));
fprintf('isfield(s, "height") = %d\n', isfield(s, 'height'));
isfield(s, 'name') = 1
isfield(s, 'height') = 0
```

- In the above example, since s does not contain a "height" field, the value 0 is returned 。

Example of Structure Array

- We can use cell2struct command to transfer the cell to structure, e.g.：
- 範例13-14：cell2struct01.m

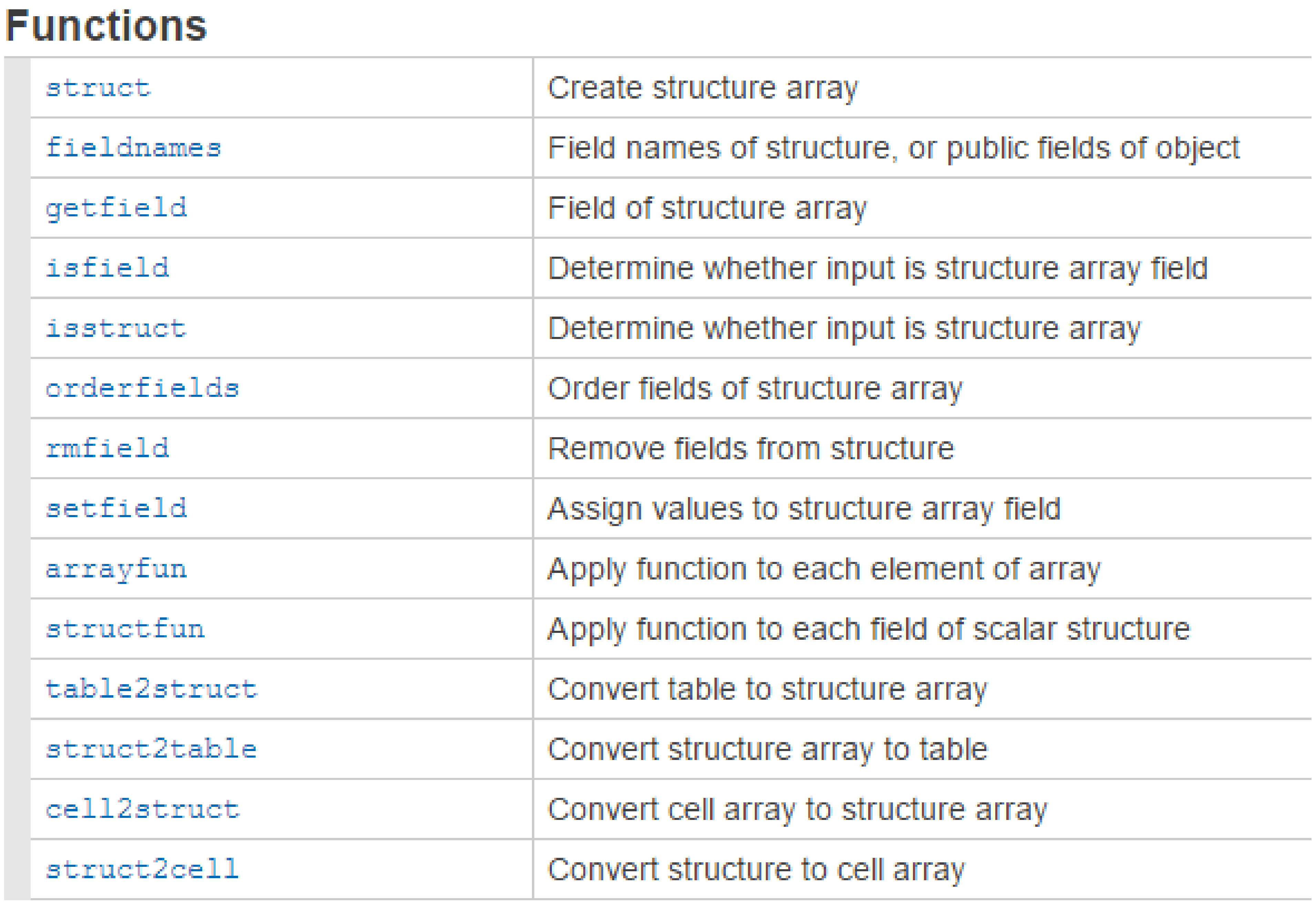
```
fields = {'name', 'age'};
values = {'Tim', 9; 'Annie', 6};
s = cell2struct(values, fields, 2);
s(1)                                   % 印出第一筆資料
s(2)                                   % 印出第二筆資料
ans =
    name: 'Tim'
     age: 9
ans =
    name: 'Annie'
     age: 6
```

- In the above example, the expression "s = cell2struct(values, fields, 2)" means that the command cell2struct will use the data of the array variable fields as the field name, and use the second dimension of values to correspond to the field name fields, to produces an structures s

Example of Structure Array十五

- If the first dimension of values corresponds to the field name fields, the result is as follows ：
- 範例13-15：cell2struct02.m

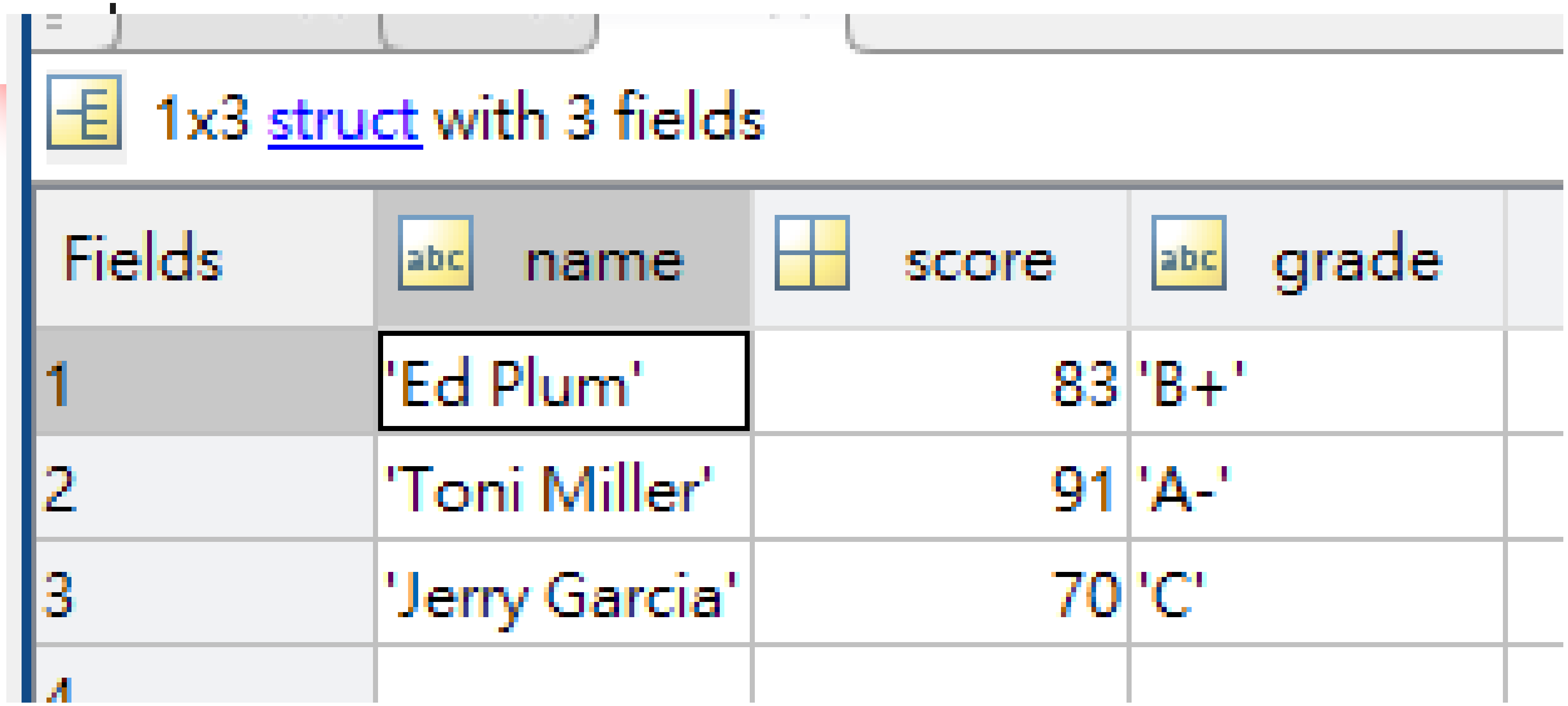
```
fields = {'name', 'age'};
values = {'Tim', 9; 'Annie', 6};
s = cell2struct(values, fields, 1);
s(1)                          % 印出第一筆資料
s(2)                          % 印出第二筆資料
ans =
    name: 'Tim'
     age: 'Annie'
ans =
    name: 9
     age: 6
```



Functions

`struct`	Create structure array
`fieldnames`	Field names of structure, or public fields of object
`getfield`	Field of structure array
`isfield`	Determine whether input is structure array field
`isstruct`	Determine whether input is structure array
`orderfields`	Order fields of structure array
`rmfield`	Remove fields from structure
`setfield`	Assign values to structure array field
`arrayfun`	Apply function to each element of array
`structfun`	Apply function to each field of scalar structure
`table2struct`	Convert table to structure array
`struct2table`	Convert structure array to table
`cell2struct`	Convert cell array to structure array
`struct2cell`	Convert structure to cell array

Exercise



1x3 struct with 3 fields

Fields	name	score	grade
1	'Ed Plum'	83	'B+'
2	'Toni Miller'	91	'A-'
3	'Jerry Garcia'	70	'C'
4			

(1) Get the field element S(2).name
(2) Use fprintf to print out values of structure array
(3) Change the values of the structure array
(4) Remove the field ‘grade’.
(5) Transfer the structure array to the cell array