المحاضرة الرابعة: لغة البرمجة MATLAB

(The MATLAB programming language)

~~~~~~

1. السلاسل الزمنية.
2. جمل الادخال والاخراج.

أعداد: - م.م. محمد وليد عبد الرضا

جامعة البصرة

كلية التربية للعلوم الصرفة

قسم الرياضيات
~~~~~~

السلاسل الرمزية

تكمن قوة برنامج MATLAB الحقيقية في القدرة على التعامل مع الأرقام، ولكنه يحتاج أحياناً إلى التعامل مع النصوص كما في حالة وضع العناوين وأسماء المحاور على الرسومات.

تركيب السلسة الرمزية

تتألف السلاسل الرمزية في لغة MATLAB من مصفوفات عددية خاصة من قيم ASCII والتي تعيد أظهار السلسة الرمزية، فمثلاً:

```
>> t = 'How about this character string?'
t =
    How about this character string?
>> size (t)
ans =
     1    32
>> whos                    إيعاز عرض أسماء المتغيرات وحجومها وعدد بياناتها وصنفها
```

Name	Size	Bytes	Class
ans	1×2	16	double array
t	1×32	64	character array

```
  Grand total is 34 elements using 80 bytes
```

إن السلاسل الرمزية ببساطة هي نص محاطة بفاصلة علوية مفردة. ويعتبر كل حرف في السلسة عنصراً من مصفوفة، والتي نحتاج إلى بايتين لتخزين كل حرف، ونختلف بذلك عن 8 بايت المخصصة لكل عنصر من عناصر المصفوفة العددية أو مضاعفة الدقة. ولرؤية التمثيل ASCII لسلسلة رمزية نحتاج فقط لإجراء بعض العمليات الرياضية على السلسلة أو استخدام الايعاز double، وكما في المثال التالي:

```
>> u = double (t)
u =
    Columns 1 through 12
     72   111   119    32    97    98   111   117   116    32   116   104
```

```
    Columns 13 through 24
      105   115    32     99   104  97   114    97     99     116   101    114
    Columns 25 through 32
      32    115    116    114  105  110  103  63
>> char (u)
ans =
      How about this character string?
>> char (u (1))
ans =
      H
```

وبما إن السلاسل هي مصفوفات، لذلك يمكن التعامل معها بكل أدوات التعامل مع المصفوفات المتوفرة في لغة MATLAB، فمثلاً:

```
>>  u = t (16: 24)
u =
   character
```

وتعنون السلاسل تماماً كما تعنون المصفوفات، وتحوي العناصر من 16 إلى 24 في المثال أعلاه الكلمة character

```
>> u = t (24: -1: 16)
u =
   retcarahc
```

وهنا تمت تهجئة الكلمة character بشكل عكسي.

```
>> u = t (16: 24)'
u =
    c
    h


      a
```

r

a

c

t

e

r

وتم هنا تحويل كلمة character إلى مصفوفة عمود عبر عملية مدور(منقول).

يمكن دمج المصفوفات الرمزية وكالاتي:

```
>> u = 'Hameed ';
>> v = 'Aiad';
>> w  = [u  v]
w =
    Hameed Aiad
```

ويسمح لنا الايعاز disp إظهار السلسلة بدون طباعة اسم المتغير كما في المثال التالي:

```
>> disp (u)
   Hameed
```

ويمكن أن تتملك السلاسل الرمزية (كما في باقي المصفوفات) عدة اسطر، ولكن يجب أن يحوي كل سطر عدداً متساوياً من الأعمدة، لذلك يجب استخدام الفراغات لجعل طول كل الأسطر متساوية كما في المثال التالي:

```
>> v = ['character strings having more than   '
        'one row must have the same number'
        'of columns just like arrays!          ']
v =
    character string having more than
    one row must have the same number
    of columns just like array!
```

وينشئ الايعاز char مصفوفة نصية متعددة الأسطر انطلاقاً من سلاسل مستقلة مختلفة الطول، كما في المثال التالي:

```
>> legends = char ('Wilt', 'Russel', 'Kareem', 'Bird', 'Magic', 'Jordan')
legends =
        Wilt
        Russel
        Kareem
        Bird
        Magic
        Jordan
>> size (legends)
ans =
     6     6
```

تحويل الأعداد إلى سلاسل رمزية وبالعكس

قد نرغب في العديد من الحالات بتحويل النتائج العددية إلى سلاسل رمزية واستخراج البيانات العددية من السلاسل الرمزية. يزودك برنامج MATLAB بالإيعاز num2str و int2str و fprintf وغيرها لتحويل النتائج العددية إلى سلاسل رمزية، واليك الأمثلة التالية على التحويل:

```
>> int2str (35)
ans =
     35
>> class (ans)
 ans =
      char
>> num2str (3.5)
ans =
     3.5
>> class (ans)
```

```
ans =
      char
>> fprintf ('% 4.3f\n', sqrt (2))
 1.414
>> size (fprintf ('% 4.3f\n' , sqrt (2))
ans =
     1     1
```

مثال:

```
>> radius = sqrt (2);
>> area = pi * radius ^ 2;
>> fprintf ('A circle of radius% 6.4f has an area of % 6.4f', radius, area)
A circle of radius  1.4142 has an area of  6.2832
```

تحدد هنا الصيغة 6.4f % ست خانات لإظهار المتغير radius والمتغير area.

مثال (طريقة أخرى):

```
>> S = ['A circle of radius ', (num2str (radius)), 'has an of ',  (num2str (area))  '.']
S =
A circle of radius 1.4121 has an area of 6.2832.
```

تحويل السلاسل الرمزية إلى عددية

```
>> S = str2num ('3.5')
S =
     3.5
>> t = ['3.5▼'      'sqrt(2)'    ;'▼1.5'  '▼▼▼▼9.5']
```

فراغات ▼

يجب إن تكون أطوال الأسطر متساوية

```
t =
   3.5 sqrt(2)
   1.5    9.5
>> str2num (t)
```

مصفوفة رمزية ←

```
ans =
    3.5000    1.4142
    1.5000    9.5000
```
← مصفوفة عددية

يمكن أن تكون أطوال الأسطر غير متساوي

```
>> t = '[3.5▼sqrt(2);▼1.5▼9.5]'
t =
  [3.5▼sqrt(2);▼1.5▼9.5]
```
← مصفوفة رمزية

```
>> str2num (t)
ans =
    3.5000    1.4142
    1.5000    9.5000
```

يعيد الايعاز findstr أدلة البداية لسلسلة رمزية موجودة ضمن سلسلة أخرى.

```
>> b = 'Peter Piper picked a peck of pickled peppers';
>> findstr (b, '▼')
ans =
     6    12    1 9    21    26    29    37
>> findstr (b, 'p')
```
← حرف صغير

```
ans =
     9    13    22    30    38    40    41
>> findstr (b, 'cow')
ans =
     [ ]
```

مصفوفة الخلايا للسلاسل الرمزية

يبدو شرط تساوي عدد الأعمدة في جميع اسطر المصفوفات النصية متعباً، خاصة إذا اختلف عدد الفراغات المضافة من سطر لآخر، ويمكن حل هذه المشكلة عبر استخدام مصفوفات الخلايا، حيث يمكننا وضع كل أنواع البيانات ضمن مصفوفة الخلايا، ويتجلى الاستخدام الأكبر لمصفوفة الخلايا مع السلاسل الرمزية.

تعتبر مصفوفة الخلية ببساطة نوعاً من البيانات التي تسمح للمستخدم بتسمية مجموعة من البيانات ذات الأنواع والحجوم المختلفة، وذلك كما يبينه المثال التالي:

```
>> C = {'How'; 'about'; 'this for a'; 'cell array of strings?'}
C =
    'How'
    'about'
    'this for a'
    'cell array of strings?'
>> size (C)
ans =
     4     1
```

تستخدم أقواس المجموعة { } لإنشاء مصفوفة الخلايا، وذلك استخدمناها في حصر السلسلة الرمزية بأكملها، وتملك المصفوفة C في هذا المثال أربعة اسطر وعموداً واحداً، ويحوي كل عنصر من مصفوفة الخلية سلسلة رمزية مختلفة الطول.

وتعنون مصفوفات الخلايا كما تعنون بقية المصفوفات، وذلك كما يلي:

```
>> C (2: 3)
ans =
    'about'
    'this for a'
>> C ([4   3   2   1])
ans =
    'cell array of strings?'
    'this for a'
    'about'
    'How'
>> C (1)
ans =
```

```
    'How'
>> size (C (1))
ans =
     1     1
```

ما زالت النتائج حتى ألان عبارة عن مصفوفات خلايا، وذلك لان التعبير C (indices) يعنون الخلايا المعطاة ولكنه لا يحدد محتوى هذه الخلايا. ولاسترجاع محتويات خلية جزئية محدّدة عليك استخدام أقواس مجموعة كما في المثال التالي:

```
>> S = C {4}
S =
    cell array of strings?
>> size (s)
ans =
     1   22
```

كما ويمكن عنونة جزء من محتويات مصفوفة الخلية الجزئية كما يلي:

```
>> C {4} (1: 10)
ans =
     cell array
```

يحول الايعاز char محتويات مصفوفة الخلية إلى مصفوفة نصية مناسبة، كما يبينه المثال التالي:

```
>> S = char (C)
S =
    How
     about
     this for a
     cell array of strings?
>> size (S)
ans =
     4     22
```

ويقوم الايعاز cellstr بإجراء التحويل العكسي ويعيد صياغة السلاسل الرمزية بشكل جيد كما يلي:

```
>> cellstr (S)
ans =
        'How'
        'about'
        'this for a'
        'cell array of strings?'
```

جمل الإدخال والإخراج

جمل الإدخال

هناك عدة صيغ للإدخال بالإضافة إلى عملية التنسيب منها:

1- تعليمة input:

مثال (1):

```
>> x = input ('enter x: ')
enter x:
```

مثال (2): إدخال الأعداد.

```
n = input ('enter n:');
m = input ('enter m:');
for i = 1: n
  for j = 1: m
    result (i, j) = i ^  j;
  end;
end;
```

مثال (3): إدخال أسماء رمزية.

```
clc;
clear;
z = input ('enter name', 's');
```

للدلالة على إدخال string

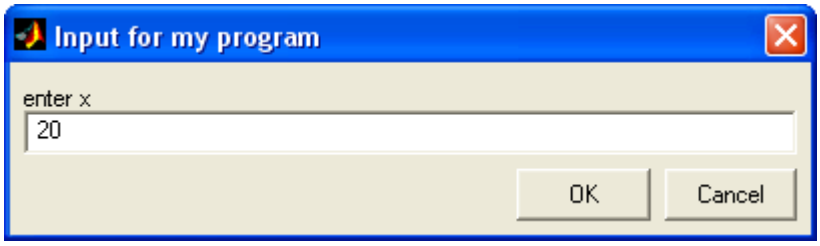
2- صيغة ثابتة للإدخال (على شكل مربع حوار):

مثال:

```
prompt = {'enter x'};
def = {'20'};
dlgTitle = 'Input for my program';
```

```
lineNo = 1;                    % عدد السطور المدخلة
answer = inputdlg (prompt, dlgTitle, lineNo, def);
x = str2num (answer {1});      % تحويل string إلى num في حالة التعامل مع رقم
```

القيمة الأولى من مصفوفة الخلايا



جمل الإخراج

هناك عدة صيغ للإخراج منها:

1- تعليمة disp:

مثال (1):

```
>> d = 15;
>> disp (d);
15
```

مثال (2):

```
>> a = 'ali';
>> disp (a);
ali
```

مثال (3):

```
>> sum = 9.8;
>> disp (['sum = ', num2str (sum)]);
sum = 9.8
```

مثال (4):

```
>> disp ('computer');
computer
```

ملاحظة (1):

يجب أن يكون محتويات disp قيمة ذات نوع بياني واحد ضمن الجملة الواحدة (كل جملة نوع بياني واحد).

ملاحظة (2):

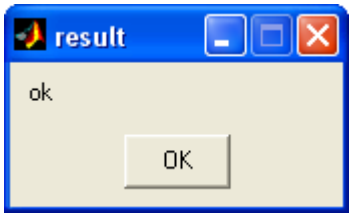
في حالة كون محتويات disp أكثر من قيمة ذات نوع بيانية مختلفة ضمن الجملة الواحدة (يجب ان تجمع القيم في قوسين كبيرين [ ] (مثال (3))).

2- تعليمة msgbox:

```
>> msgbox ('ok', 'result')
```

عنوان الصندوق

الشيء المطلوب طباعته (نوع بياني رمزي)



(3) تعليمة fprintf:

مثال (1):

```
>> y = 1.2;
>> x = 100.5;
>> fprintf ('variable x is % 6.3f\n', x);
>> fprintf ('variable y is % 6.3f\n', y);
variable x is  1.200
variable y is 100.500
```

وهذا يعني بأنه تم حجز 6 مراتب منها 3 مراتب بعد الفارزة العشرية.

مثال (2):

```
>> fprintf ('% 8.3f\n', round (3.8));
   4.000
```

ملاحظة (1):

يمكن استخدام صيغ مختلفة للطباعة وكما يلي:

رمز واحد	%c
تدوين عشري	%d
تدوين يائي	%e
تدوين النقطة الثابتة	%f
تدوين عشري	%i
تدوين ثماني	%o
تدوين رمزي	%s
تدوين سداسي عشر	%x

ملاحظة (2):

يمكن طباعة الأعداد والأسماء والنتائج من خلال كتابة الايعازات بدون فارزة منقوطة وستظهر النتائج في نافذة الأمر Command Window.