```
        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
```

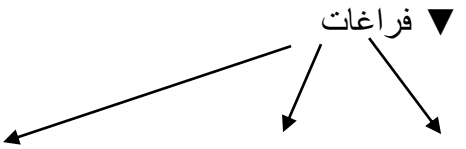
تحدد هنا الصيغة 6.4f % ست خانات لإظهار المتغير radius والمتغير area.

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']
```



يجب إن تكون أطوال الأسطر متساوية

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

مصفوفة رمزية ←

```
>> str2num (t)
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:

١- input

مثال (١):

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

مثال (٢): 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 (٣): إدخال أسماء رمزية.

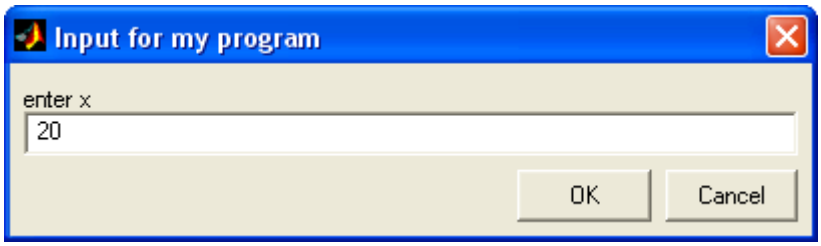
```
clc;
clear;
z = input ('enter name', 's');
```

للدلالة على إدخال string

٢- 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:

١- disp

مثال (١):

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

مثال (٢):

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

مثال (٣):

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

مثال (٤):

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

ملاحظة (١):

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

ملاحظة (٢):

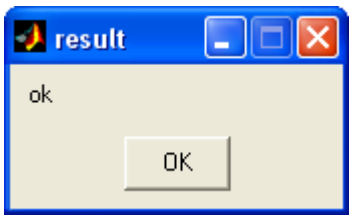
في حالة كون محتويات disp أكثر من قيمة ذات نوع بيانية مختلفة ضمن الجملة الواحدة (يجب ان تجمع القيم في قوسين كبيرين [ ] (مثال (٣))).

٢- msgbox

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

عنوان الصندوق

الشيء المطلوب طباعته (نوع بياني رمزي)



(٣) fprintf

مثال (١):

```
>> 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 (٢):

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

ملاحظة (١):

يمكن استخدام صيغ مختلفة للطباعة وكما يلي:

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

ملاحظة (٢):

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

الجمل الشرطية

يدعم برنامج MATLAB العمليات المنطقية والمقارنة مثلما يدعم العمليات الرياضية، وتهدف العمليات والمعاملات المنطقية الحصول على أجوبة للأسئلة التي يجاب عنها بصح أو خطأ (True/False).

تعتبر لغة MATLAB في تعاملها مع جميع التعابير المنطقية وعمليات المقارنة إن أي عدد غير صفري هو True ويعتبر الصفر False، كما ويكون إخراج جميع التعابير المنطقية وعمليات المقارنة عبارة عن مصفوفات منطقية تحوي العدد واحد من اجل True والعدد صفر من اجل False.

وتعتبر المصفوفات المنطقية نوعاً خاصاً من المصفوفات العددية، كما يمكن عنونة المصفوفة المنطقية بنفس طريقة عنونة باقي المصفوفات التي استخدمها سابقاً ضمن التعابير العددية.

معاملات المقارنة (العوامل العلائقية) : Relational Operators

تتضمن معاملات المقارنة كل الإشارات المقارنة الشائعة والمدرجة في الجدول التالي:

معامل المقارنة	الوصف
<	أصغر من
<=	أصغر أو يساوي
>	أكبر من
>=	أكبر أو يساوي
==	إشارة المساواة (لكي نميزها عن =)
~=	اشارة عدم المساواة