

MATLAB

❖ MATLAB هي اختصار للكلمتين **Matrix Laboratory** اي مختبر المصفوفات .

❖ بدا اول اصدار لبرنامج MATLAB في عام 1984 كاول تسويق للمنتج .

❖ الموقع الخاص لمجموعة الشركات المنتجة (Mathworks Incorporated)

<http://www.mathworks.com>

❖ **برمجياً تعرف لغة MATLAB :-**

وهي لغة برمجة عالية الاداء تستخدم لإجراء الحسابات التقنية وتقوم بحساب واخراج البيانات ضمن بيئة سهلة البرمجة، حيث يعبر عن المسألة وحلها بأشكال رياضية مشهورة.

❖ **البرنامج PROGRAM :-** وهو مجموعة الاوامر المتسلسلة (Commands or Statements) ، جملة او امر تكتب باحدى لغات البرمجة تعمل على مدخلات البرنامج (INPUT I/P) لها وظيفة محددة ضمن البرنامج للوصول الى النتائج وهي مخرجات البرنامج (OUTPUT O/P) .

جمل التحكم والشرط والتكرار

PROGRAM CONTROL, CONDITION AND LOOP STATEMENTS

1- الجمل الشرطية :-

تقسم جمل الشرط الى نوعين وهي :-

1- جملة اذا الشرطية (if – statement)

2- جملة التوزيع (switch – case)

1- جملة اذا الشرطية (if – statement) :-

وتستخدم للتحكم بطريقة سير البرنامج اعتمادا على تحقق الشرط (condition) ،

ولها ثلاث صيغ :-

1- تستخدم if لوحدها في حالة استخدام شرط واحد وبالشكل التالي :-

if condition

statement

```

end
ex:-
x=input('enter the number =')
if x > 0
disp(' x is positive number ')
end

```

2- الجملة (if / else) :- يستخدم الامر else كجزء من جملة الشرط if ، فاذا تحقق شرط جملة if يترتب عليه تحقيق النتائج التالية لجملة if (الى ما قبل جملة else مباشرة) والا تحقق النتائج ما بعد else ، ويأخذ الصورة التالية :-

```

if condition
statement(s)
else
statement(s)
end

```

```

ex:-
x=input('enter the number =')
if x > 0
disp(' x is positive number ')
end

```

2- الجملة (if / else) :- يستخدم الامر else كجزء من جملة الشرط if ، فاذا تحقق شرط جملة if يترتب عليه تحقيق النتائج التالية لجملة if (الى ما قبل جملة else مباشرة) والا تحقق النتائج ما بعد else ، ويأخذ الصورة التالية :-

```

if condition
statement(s)
else
statement(s)
end

```

3- الجملة (if / elseif) :- الغرض من هذه الصورة من صور جملة الشرط if

هو اختبار مدى تحقق اكثر من شرط ويأخذ الصيغة التالية :-

```
if condition 1
statement 1
elseif condition 2
statement 2
elseif condition (n-1)
statement (n-1)
else
statement (n)
end
```

مثال / اكتب برنامج بلغة MATLAB لإدخال عدد ، يبين اذا كان العدد موجب ام سالب او يساوي صفر ؟

```
x = input ('enter the number =')
if x > 0
```

```
disp ('this number is negative ')
else
disp('this number is equal 0 ')
end
```

ملاحظة :- يجب ان تحتوي الجمل الشرطية على احد ادوات المقارنة او الادوات المنطقية (Logical or Relational Operators) الموضحة في الجدول التالي :-

الوظيفة Operation	Logic Operators المعاملات المنطقية
اقل من	<
اقل من او يساوي	<=
اكبر من	>
اكبر من او يساوي	>=
يساوي	==
لا يساوي	~=
And	&

Short-circuit And للقِيم العددية فقط (scalars)	&&
Or	
Short-circuit or للقِيم العددية فقط (scalars)	
Not	~

2- جملة التوزيع (switch – case)

تستخدم للتحكم بطريقة سير البرنامج اعتمادا على قيمة المتغير المعطى لها والصيغة العامة لها :-

switch variable

case value 1

Statement 1

case value2

Statement 2

case value n

Statement n

otherwise

Statement

end

يبدأ البرنامج اولا بقراءة قيمة المتغير (variable) ثم يبدأ بمقارنتها مع القيم (value) بعد عبارة case اذا كانت قيمة المتغير variable تنطبق مع اي من القيم value يتم تنفيذ الجملة statement بعدها وهكذا ، واذا لا تنطبق اي من القيم value مع قيمة المتغير variable سوف ينتقل الى العبارة otherwise ويتم تنفيذ الجملة بعدها .

2- جملة التوزيع (switch – case)

تستخدم للتحكم بطريقة سير البرنامج اعتمادا على قيمة المتغير المعطى لها والصيغة العامة لها :-

```
switch variable
case value 1
Statement 1
case value2
Statement 2
case value n
Statement n
otherwise
Statement
end
```

يبدأ البرنامج اولا بقراءة قيمة المتغير (variable) ثم يبدأ بمقارنتها مع القيم (value) بعد عبارة case اذا كانت قيمة المتغير variable تنطبق مع اي من القيم value يتم تنفيذ الجملة statement بعدها وهكذا ، واذا لا تنطبق اي من القيم value مع قيمة المتغير variable سوف ينتقل الى العبارة otherwise ويتم تنفيذ الجملة بعدها .

2- حلقات while (while ... condition)

تستخدم لتنفيذ امر او مجموعة الاوامر مرار متحدة مادام شرط ما متحقق والصيغة العامة لها :-

```
while condition
statement(s)
end
```

مثال / البرنامج التالي يجمع الاعداد من 1 الى 10 ويزيادة مقدارها 0.5 (مجموع الاعداد 1 , 1.5 , 2 , 2.5 , 3 , 3.5 , 4 , ... 10 باستخدام while .

```
s = 0 ;
i = 1 ;
while i <= 10
s = s + i ;
i = i + 0.5 ;
end
```

```
display ( s )
```

sol.

```
>>file_name
```

```
s =
```

```
104.500
```

4) جمل القفز Jumping statements

1) الخروج من جمل التكرار Breaking out of loops

ان تكرر `while` يستمر في العمل حتى يصبح الشرط صحيحا او خاطئا ، وان تكرر `for` يستمر في العمل حتى ينتهي عدد المرات التي تحدها له ، ولكن قد تحتاج في بعض الاحيان الى الخروج المبكر من التكرار قبل انتهائه ، وفي هذه الحالة يمكننا استخدام الامر `break` او `return` ، ويستخدم بالشكل التالي :-

```
for / while
statement(s)
if condition
break
end
```

...

```
end
```

ex:-

```
clc ; clear ; close all ;
```

```
for m = 1 : 10
```

```
if ( m ^ 2 ) >= 36
```

```
break ;
```

```
end
```

```
display ( m ) ;
```

```
end
```

sol.

```
>>file_name
```

```
m =
```

```
1
```

```
m =
```

```
2
```

```
m =
```

```
3
```

```
m =
```

```
4
```

```
m =
```

مثال / اطبع مربع الاعداد من 1 - 10 لغاية القيمة 36

وفي حالة استخدام while loop

```
clc ; clear ; close all ;
```

```
m = 1 ;
```

```
while m < 10
```

```
if ( m ^ 2 ) >= 36
```

```
break ;
```

```
end
```

```
display ( m ) ;
```

```
m = m + 1 ;
```

```
end
```

(2) جملة continue

يستخدم الأمر `continue` في داخل الحلقة التكرارية `for loop` او الحلقة التكرارية المشروطة `while loop` لانتهاء التكرار الحالي ، وتجاوز تنفيذ باقي الاوامر (الجمل البرمجية) الموجودة بداية من الامر `continue` ووصولاً لنهاية الحلقة المستخدمة باستخدام جملة `end` ، ويستخدم بالشكل التالي :-

```
for / while
statement(s)
if condition
continue
end
end
```

مثال/ اطبع القيم الزوجية من 1 - 10

ex:-

```
clc ; clear ; close all ;
for n = 1 : 10
if rem ( n , 2 ) ~= 0
continue ;
end
display ( n ) ;
end
```

sol.

```
>>file_name
n =
2
n =
4
n =
6
n =
8
n =
10
```

5) صناديق try – catch

يستخدم برنامج MATLAB طريقة الاستثناء exception في معالجة الأخطاء التي من الممكن ان يقع فيها المستخدم اثناء سير البرنامج باستخدام صناديق try – catch والتي تستخدم الصورة التالية :-

```
try
commands1
catch
commands2
end
```

في هذه الصورة يتم وضع جميع الاوامر commands1 التي يوجد احتمال حدوث خطأ فيها في جملة try فاذا لم يحدث خطأ ينتقل مباشرة الى جملة end (ينفذ الاوامر commands1 بصورة طبيعية) ، ولكن اذا حدث خطأ في تنفيذ الاوامر commands1 فيتم الانتقال الى جملة catch لتنفيذ الاوامر commands2 والتي تكتب من قبل المستخدم كرسالة توضح وجود خطأ ليتجنب حدوث هذا الخطأ في المرات القادمة .

Ex:-

```
clc ; clear ; close all ;
a = input ('Enter first matrix :');
b = input ('Enter second matrix :');
try
c = a * b ;
catch
c = NaN ;
disp ( ' wrong dimensions for matrices multiply ');
end
display ( c );
sol.
>>file_name
Enter first matrix : [ 1 3 ; 2 4 ]
Enter second matrix : [ 5 7 ; 6 8 ]
c =
23 31
24 46
```

Getting Started with Image Processing using MATLAB

1. Image Processing by using MATLAB Editor Window

Now, we will write the code for performing some basic operations of image processing in the editor window. To get familiar with all the [basic terminology used in MATLAB \(https://circuitdigest.com/tutorial/what-is-matlab-and-how-to-get-started-with-it\)](https://circuitdigest.com/tutorial/what-is-matlab-and-how-to-get-started-with-it) follow the link. Copy and paste the below code in the editor window,

```
a = imread('F:\circuit digest\image processing using matlab\cameraman.jpg');
subplot(2,3,1);
imshow(a);

b = rgb2gray(a);
subplot(2,3,2);
imshow(b);

c = im2bw(a);
subplot(2,3,3);
imshow(c);

d = imadjust(b);
subplot(2,3,4);
imshow(d);

e = a;
e=rgb2gray(e);
subplot(2,3,5);
imhist(e);

imfinfo('F:\circuit digest\image processing using matlab\beard-man.jpg')

[height, width, colour_planes] = size(a)

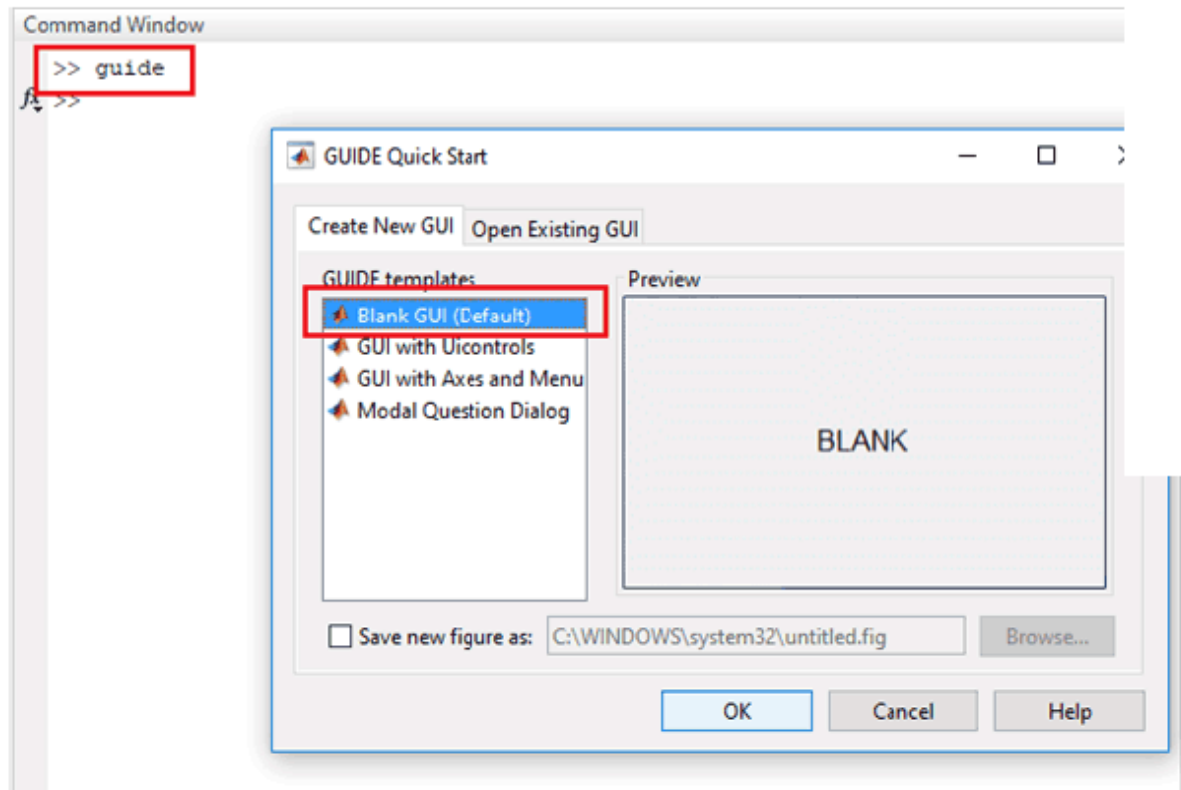
%colormap('spring')
```

Creating MATLAB Graphical User Interface for Image Processing

For building a GUI (Graphical User Interface) for Image Processing launch the GUI by typing the below command *command window*.

```
guide
```

A popup window will open, then select new *blank GUI* as shown in below image,



Now we have to choose number of pushbuttons (every pushbutton will perform different task) and one axis to display image.

```

132 % --- Executes on button press in uploadimage.
133 function uploadimage_Callback(hObject, eventdata, handles)
134 % hObject    handle to uploadimage (see GCBO)
135 % eventdata  reserved - to be defined in a future version of MAI
136 % handles    structure with handles and user data (see GUIDATA)
137 - a=uigetfile('.jpg')
138 - a=imread(a);
139 - axes(handles.axes1);
140 - imshow(a);
141 - setappdata(0,'a',a)

```

Now, in every function you will see command `getappdata()` which is used to retrieve data which is stored the `setappdata()` in the GUI.

Here we will explain eight commonly used functions in image processing

S. No.	Command	Button Name	Task to be Performed
1.	<code>uigetfile()</code>	Upload Image	Click to import image from Disk
2.	<code>rgb2gray()</code>	RGB to Gray	Click to convert RGB image into grayscale
3.	<code>im2bw()</code>	Convert to Binary Image	Click to convert the image into binary
4.	-	RESET	Click to reset the image as original
5.	<code>imhist()</code>	Histogram	Click to see the histogram of the image