

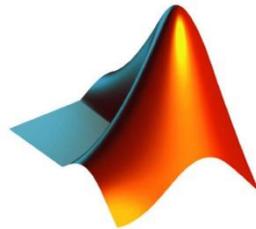
العلوم	الكلية
الرياضيات	القسم
Programming (MATLAB)	المادة باللغة الانجليزية
البرمجة بلغة (ماتلاب)	المادة باللغة العربية
الثانية	المرحلة الدراسية
صفوت عبدالقادر حمد	اسم التدريسي
The if Statement (1)	عنوان المحاضرة باللغة الانجليزية
اذا الشرطية (الجزء الاول)	عنوان المحاضرة باللغة العربية
الثانية عشر	رقم المحاضرة
MATLAB A Practical Introduction to Programming and Problem Solving	المصادر والمراجع
MATLAB The Language of Technical Computing	
MATLAB numerical computing	



University Of Anbar
College Of Science
Math Department



Programing Language (MATLAB)



MATLAB

Lecture 12

The if Statement (1)

By:

Safwat A. Hamad

For the 2nd stage Math Department

1. Introduction

In the scripts and functions we've seen thus far, every statement was executed in sequence. This is not always desirable, and in this lecture, we'll see how to make choices as to whether statements are executed or not, and how to choose between or among statements. The statements that accomplish this are called *selection* or *branching* statements.

The MATLAB software has two basic statements that allow us to make choices: the if statement and the switch statement. The if statement has optional else and elseif clauses for branching. The if statement uses expressions that are logically true or false. These expressions use relational and logical operators. MATLAB also has "is" functions that test whether an attribute is true or not; these can be used with the selection statements.

2. If Statement

The if statement chooses whether another statement, or group of statements, is executed or not. The general form of the if statement is:

```
if condition
    action
end
```

A condition is a relational expression that is conceptually, or logically, true or false. The action is a statement, or a group of statements, that will be executed if the condition is true. When the if statement is executed, first the condition is evaluated. If the value of the condition is true, the action will be executed; if not, the action will not be executed. The action can be any number of statements until the reserved word end; the action is naturally bracketed by the reserved words *if* and *end*. (Note that this is different from the end that is used as an index into a vector or matrix.) The action is usually indented to make it easier to see.

For example, the following if statement checks to see whether the value of a variable is negative. If it is, the value is changed to a zero; otherwise, nothing is changed.

```
if num < 0
    num = 0
end
```

"If statements" can be entered in the Command Window, although they generally make more sense in scripts or functions. In the Command Window, the "if" line would be entered, followed by the Enter key, the action, the Enter key, and finally end and Enter. The results will follow immediately. For example, the preceding "if statement" is shown twice here.

```
>> num = -4;
>> if num < 0
    num = 0
end
num =
    0
>> num = 5;
>> if num < 0
    num = 0
end
```

Note, that the output from the assignment is not suppressed, so the result of the action will be shown if the action is executed. The first time the value of the variable is negative, so the action is executed and the variable is modified,

But, in the second case, the variable is positive so the action is skipped. This may be used, for example, to make sure that the square root function is not used on a negative number. The following script prompts the user for a number and prints the square root. If the user enters a negative number, the "if statement" changes it to zero before taking the square root.

sqrtifexamp.m

```
% Prompt the user for a number and print its sqrt
num = input('Please enter a number: ');
% If the user entered a negative number, change it
if num < 0
    num = 0;
end
fprintf('The sqrt of %.1f is %.1f\n', num, sqrt(num))
```

Here are two examples of running this script:

```
>> sqrtifexamp
Please enter a number: -4.2
The sqrt of 0.0 is 0.0
>> sqrtifexamp
Please enter a number: 1.44
The sqrt of 1.4 is 1.2
```

Note that in the script the output from the assignment statement is suppressed. In this case, the action of the *"if statement"* was a single assignment statement. The action can be any number of valid statements. For example, we may wish to print a note to the user to say that the number entered was being changed. Also, instead of changing it to zero, we will use the absolute value of the negative number entered by the user.

sqrtifexampii.m

```
% Prompt the user for a number and print its sqrt
num = input('Please enter a number: ');
% If the user entered a negative number, tell
% the user and change it
if num < 0
    disp('OK, we'll use the absolute value')
    num = abs(num);
end
```

```
fprintf('The sqrt of %.1f is %.1f\n', num, sqrt(num))
```

```
>> sqrtifexampii
```

```
Please enter a number: -25
```

```
OK, we'll use the absolute value
```

```
The sqrt of 25.0 is 5.0
```

Note that as seen in this example, two single quotes in the *disp* statement are used to print one single quote.

SAFWAT HAMAD