الكلية	العلوم
القسم	الرياضيات
المادة باللغة الانجليزية	Programming (MATLAB)
المادة باللغة العربية	البرمجة بلغة (ماتلاب)
المرحلة الدراسية	الثانية
اسم التدريسي	صفوت عبدالقادر حمد
عنوان المحاضرة باللغة الانجليزية	The if Statement (2)
عنوان المحاضرة باللغة العربية	اذا الشرطية (الجزء الثاني)
رقم المحاضرة	الثالثة عشر
المصادر والمراجع	MATLAB A Practical Introduction to Programming and Problem Solving
	MATLAB The Language of Technical Computing
	MATLAB numerical computing

University Of Anbar

College Of Science

Math Department

Programing Language (MATLAB)

Lecture 13

The if Statement (2)

By:

Safwat A. Hamad

For the 2nd stage Math Department

1. Representing Logical True and False

It has been stated that conceptually true expressions have the ***logical*** value of 1, and expressions that are conceptually false have the ***logical*** value of 0. Representing the concepts of ***logical true*** and ***false*** in MATLAB is slightly different: the concept of false is represented by the value of 0, but the concept of true can be represented by any nonzero value (not just 1). This can lead to some strange ***logical*** expressions. For example:

```
>> all(1:3)
ans =
        1
```

Also, consider the following ***"if statement"***:

```
>> if 5
    disp('Yes, this is true!')
end
>>Yes, this is true!
```

As 5 is a nonzero value, the condition is ***true***. Therefore, when this l***ogical*** expression is evaluated, it will be ***true***, so the ***disp*** function will be executed and "Yes, this is true" is displayed. Of course, this is a pretty bizarre if statement, one that hopefully would never be encountered!

However, a simple mistake in an expression can lead to a similar result. For example, let's say that the user is prompted for a choice of 'Y' or 'N' for a yes/no question.

```
letter = input('Choice (Y/N): ','s');
```

In a script we might want to execute a particular action if the user responded with 'Y'. Most scripts would allow the user to enter either lowercase or uppercase; for example, either 'y' or 'Y' to indicate "yes". The proper expression that would return ***true*** if the value of letter was 'y' or 'Y' would be

```
letter == 'y' || letter == 'Y'
```

2. The If-Else Statement

The ***if*** statement chooses whether or not an action is executed. Choosing between two actions, or choosing from among several actions, is accomplished using if-else, nested ***if-else***, and ***switch*** statements. The ***if-else*** statement is used to choose between two statements, or sets of statements. The general form is:

```
if condition
    action1
else
    action2
end
```

First, the condition is evaluated. If it is true, then the set of statements designated as "**action1**" is executed, and that is the end of the if-else statement. If, instead, the condition is false, the second set of statements designated as "**action2**" is executed, and that is the end of the ***if-else*** statement. The first set of statements ("action1") is called the action of the if clause; it is what will be executed if the expression is true. The second set of statements ("action2") is called the action of the else clause; it is what will be executed if the expression is false. One of these actions, and only one, will be executed—which one depends on the value of the condition.

For example, to determine and print whether or not a random number in the range from 0 to 1 is less than 0.5, an if-else statement could be used:

```
if rand < 0.5
   disp('It was less than .5!')
else
    disp('It was not less than .5!')
end
```

One application of an ***if-else*** statement is to check for errors in the inputs to a script (this is called ***error-checking***). For example, an earlier script prompted the user for a radius, and then used that to calculate the area of a circle. However, it did not

check to make sure that the radius was valid (e.g., a positive number). Here is a modified script that checks the radius:

checkradius.m

```
% This script calculates the area of a circle
% It error-checks the user's radius
radius = input('Please enter the radius: ');
if radius <= 0
    fprintf('Sorry; %.2f is not a valid radius\n',radius)
else
    area = calcarea(radius);
    fprintf('For a circle with a radius of %.2f,',radius)
    fprintf(' the area is %.2f\n',area)
end
```

Examples of running this script when the user enters invalid and then valid radius is shown as follows:

```
>> checkradius
Please enter the radius: -4
Sorry; -4.00 is not a valid radius
>> checkradius
Please enter the radius: 5.5
For a circle with a radius of 5.50, the area is 95.03
```

The ***if-else*** statement in this example chooses between two actions: printing an error message, or using the radius to calculate the area and then printing out the result. Note that the action of the ***if*** clause is a single statement, whereas the action of the ***else*** clause is a group of three statements.

MATLAB also has an ***error*** function that can be used to display an error message; the terminology is that this is ***throwing an error***. In the previous script, the **if** clause could be modified to use the ***error*** function rather than ***fprintf***; the result will be displayed in red as with the error messages generated by MATLAB. Also, very

importantly, when an error message is thrown, the script stops executing. This is illustrated by the following modified script:

checkraderror.m

```
radius = input('Please enter the radius: ');
if radius <= 0
    error('Sorry; %.2f is not a valid radius\n',radius)
else
    area = pi * radius .^2;
    fprintf('For a circle with a radius of %.2f,',radius)
    fprintf(' the area is %.2f\n',area)
end
disp("And that is it!")
```

```
>> checkraderror
Please enter the radius: -5
Error using checkraderror (line 3)
Sorry; -5.00 is not a valid radius
>> checkraderror
Please enter the radius: 4.5
For a circle with a radius of 4.50, the area is 63.62
And that is it!
```

If the entered radius is not valid, an error message is thrown and nothing else is executed. However, if the radius is valid, it is used to calculate and print the error; also, the ***disp*** statement is executed after the ***if-else*** statement ends.