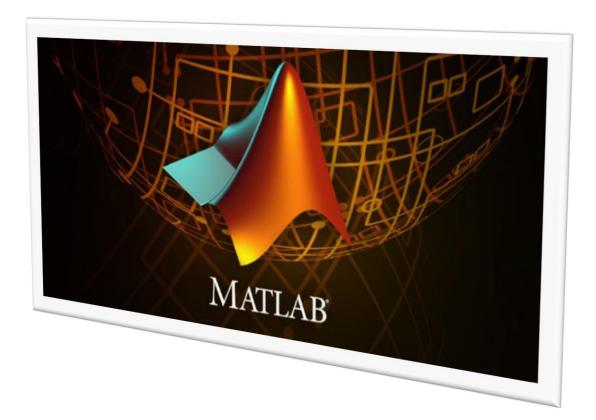


Course N°06 The if-else-elseif-end Condition in MATLAB



Dr. Salah Djerouni







1.Introduction

In this chapter, we cover the next building block in basic programming, one that exists in most programming languages and that is the conditional operators. The first conditional operator discussed is the *if-end* command. Next, we cover the *if-else-end*. Finally, we cover *if-elseif-end*.

2. Conditional operators and alternate paths

2.1. The *if* command (conditional execution)

The if statement is the simplest and most widely used of the flow control structures. It allows program execution to be directed according to the logical value of a condition. Use an if statement if you need to ask a question in order to determine what to do next.

Note.

- > Conditional statements enable MATLAB to make decisions.
- > The process is similar to the way we humans makes decisions.
- A condition stated. If the condition is met, one set of actions is taken. If the condition is not met either nothing is done, or a second set of actions is taken.
- > For every if command a computer program must have an end command.
- A program can have many *if....end* statements following each other.
- A computer program can perform the same task using different combinations of *if-end*, *if-else-end*, and *if-elseif-else-end* statements.
- > MATLAB starts at the beginning of the *if* sequence.
- ➢ It proceeds one condition to the next.
- > When it finds a true statement, the appropriate section of the code is executed.
- ➤ The sequence is then terminated!!!
- > The last section of code is closed using the keyword end.
- ▶ If two mutually exclusive actions can occur as a result of a decision, use the else statement.



The form of a conditional statement

Relational operator	Meaning
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
—	Equal to
~=	Not equal to
&&	And
	Or
~	not

3. The three forms of the if statement

There are three forms for an if statement :

✤ If-end by itself

Use for problems where you want to do something or nothing (see fig.1).

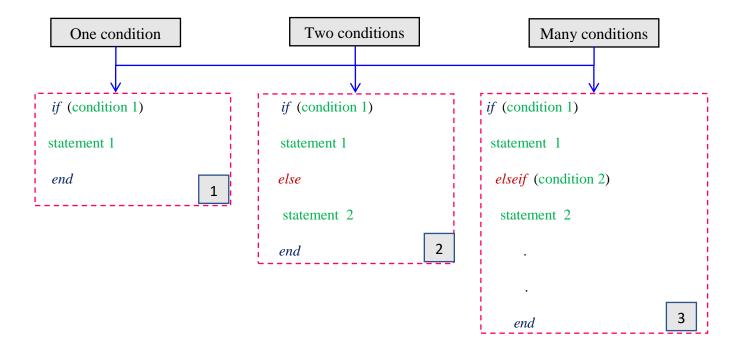
✤ If-else-end

Use for problems where you want to do one thing or another thing (see fig.2).

✤ If-elseif-end

Use for problems where do you want to do one thing out of three or more choices (see fig.3).





Example

HOME PLOTS	APPS EDITOR PUBLISH VIEW	🛃 🔚 🔏 🏦 🖄 😒 📿 🔁 🕐 Search Documentation 🛛 🔎
👍 🛅 🔚 📮 Find Files	s Insert 🔍 fx 🔃 👻 🤤 🔄 💽 🔛 🔛 Run Section 🕑	
New Open Save		
FILE	EDIT NAVIGATE BREAKPOINTS RUN	
🔃 🔿 🔁 🔁 📙 🕨 C: 🕨 WIN		ار 🔻
Current Folder	Editor - Untitled*	X Workspace
Name 0409	Untitled* × +	Name A Value
AdvancedInstallers	1 clc	y 2
∎ am-et	2 clear	
AppLocker appraiser	3 y = 2 ;	
ar-SA	4 if (y>0)	
bg-BG Boot	<pre>5 disp('> y is positive number')</pre>	
Bthprops CatRoot	6 end	
E catroot2	o ena	
CodeIntegrity Com		
E compatrel		
config Configuration	•	
Details	A	
Command History	\odot	
clc	^	
clear		
edit		
>9 gr6	Command Window	\odot
1.5	> y is positive number	
_gr6	fx >>	
2× gr6	JA, 222	
% 11/28/20.	· · · ·	
edit		
<pre>% 11/29/20.</pre>		
edit	v	٢
· · ·		



Example

A MATLAB R2014a	– 0 ×
HOME PLOTS APPS EDITOR PUBLISH VIEW	🖪 🔚 🔬 🟦 🖆 😂 🔗 🗮 🕐 Search Documentation 🛛 🔎 🗖
🕂 🐚 🔚 🖫 Find Files Insert 🗟 🎘 🛣 🔹 😪 🔄 💽 📐 🖳 🍡 Run Section 😥	
New Open Save Comment % % % % Go To + Breakpoints Run Run and Advance Run and	
V V C Print V Indent S V C Find V V Advance Time	
FILE EDIT NAVIGATE BREAKPOINTS RUN	م •
Current Folder O Fiscal Current Folder	(e) ★ Workspace
Name - Untitled' × +	Name A Value
0409 1 clc	□ u y -2
Image: Barrier and State and	
AppLocker	
B bg-BG 4 LL (Y>O)	
Be Boot 5 disp('> y is positive number')	
CatRoot Gelse catroot2	
disp('> y is negative number')	
B Com 8 end	
config	
Details	
Command History 💿	
clear	2
edit	
o gro	
-gr6 y is negative number	
2x gr6 fx >>	
<u>\$−− 11/28/20</u>	
edit	
% 11/29/20	
eait v	< >>
IIII-1	
2. An example of the if-else statement (<i>two conditions</i>)	
2. The example of the n-cise statement (<i>two contations</i>)	

Example

New Open Save Print + In FILE	Breakpots Run Run and G Advance Run and Advance Run and C Advance Run and C Advance Run and C Advance Run Run Advance Run Run Advance Run			
⊨ → 💽 💭 📙 → C: → WINDOWS → e	ystem32 >	0		▼ <u>}</u>
Name A	Vitiled* X +	(e) X	Workspace	Value
OD9 OD O	<pre>1 clc 2 clear 3 y = 0; 4 if (y>0) 5 disp('> y is positive number') 6 elseif (y<0) 7 disp('> y is negative number') 8 elseif (y==0) 9 disp('> y is null') 10 end</pre>		₩y	0
Command History clc clear edit	20 Child			
9.gr6	> y is null	U		
gro	$f_{\mathbf{x}} >> $			
% 11/28/20 edit				
% 11/29/20 edit			۲	



Important remarks

1-Always there is a « space » between *if* and the condition so don't forget is please.

2-In case two conditions after *else* there is no word in the same line.

3-In case we want to display a text in command window, this can be done using the command *disp(`')*

4.List of References

Kattan, Peter Issa. Matlab for Beginners: A gentle approach. Petra books, 2008.

Etter, Delores M., David C. Kuncicky, and Douglas W. Hull. Introduction to MATLAB. Vol.4. Hoboken, NJ, USA: Prentice Hall, 2002.

Attaway, Stormy. Matlab: a practical introduction to programming and problem solving. Butterworth-Heinemann, 2013.

Driscoll, Tobin A. Learning Matlab. Society for Industrial and Applied Mathematics, 2009.

Butt, Rizwan. Introduction to numerical analysis using MATLAB. Laxmi Publications, Ltd., 2008.

Sigmon, Kermit. Matlab: aide-mémoire. Springer Science & Business Media, 1999.

Chapman, Stephen J. Essentials of MATLAB programming. Cengage Learning, 2016.