



Conditional Statements: Conditions and if, if-else, and if-elseif-else

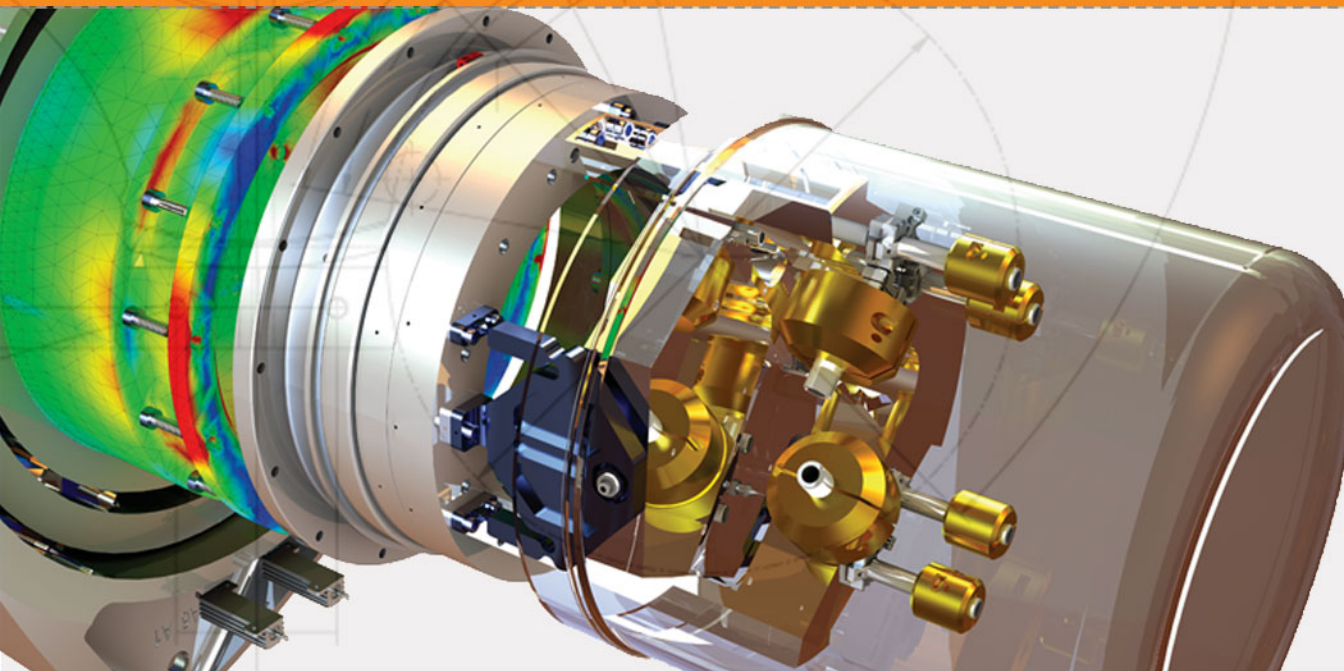


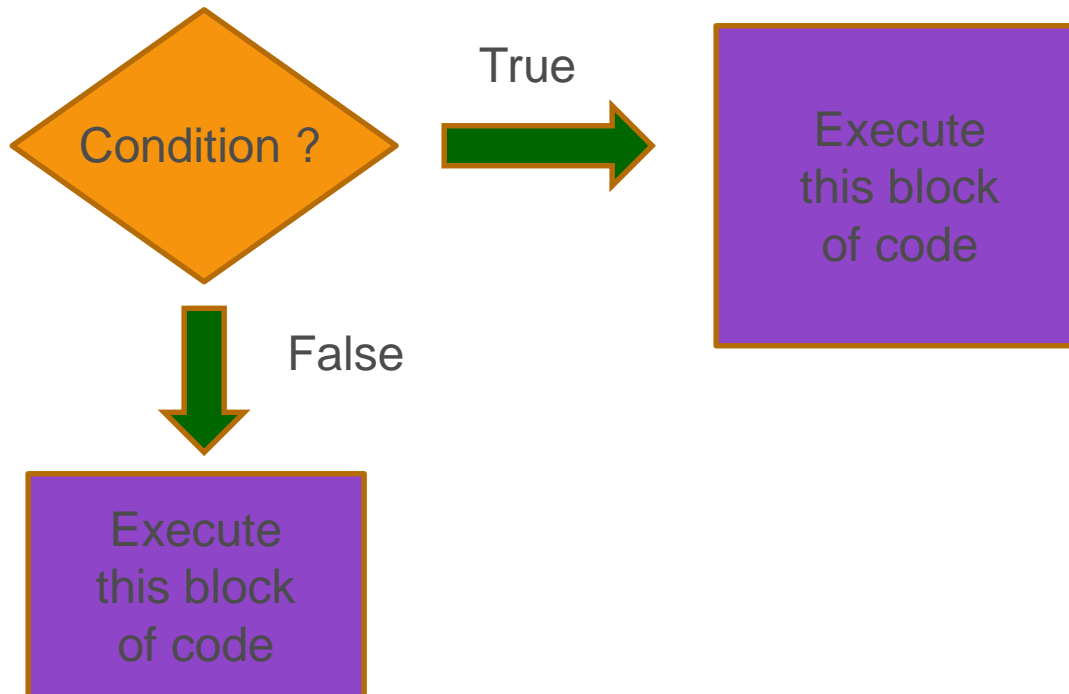
Image courtesy of National Optical Astronomy
Observatory, operated by the Association of Universities
for Research in Astronomy, under cooperative
agreement with the National Science Foundation.

Outline

- Need a way for program to make decisions
- Conditional Operators
 - Boolean Logic
 - Relational Logic
- Conditions
- Conditional Structures
 - if structures
 - if-else structures
 - if-elseif-else structures

Conditional Statements

- Conditional statements control when a section of program code executes and when it does not execute.
- All programming languages use conditional statements but syntax will vary among languages.



Relational and Logic Operators

Relational Operator	Description	Logical Operator	Description
==	Equal	&&	Scalar Logical AND
~=	Not equal		Scalar Logical OR
<	Less than	&	Element by Element AND
>	Greater than		Element by Element OR
<=	Less than or equal to	~	Logical NOT
>=	Greater than or equal to	xor	Logical Exclusive OR

Compare numbers

Compare true/false values

Logical Operators

Truth Tables for AND, OR, and NOT (~)

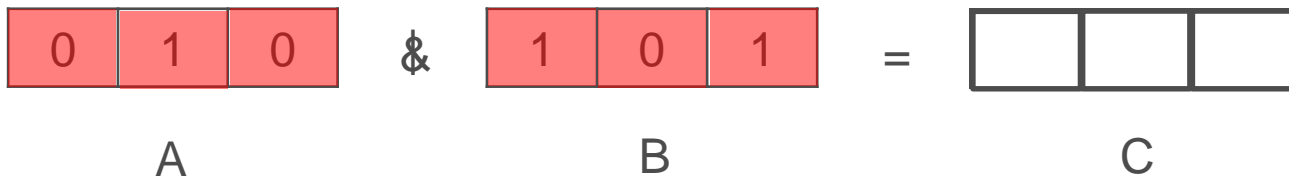
A	B	A && B	A B	~A
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

A = expression1 B = expression2

1 = TRUE 0 = FALSE

Element-wise Operators

- The single & and | are also called element-wise operators because they act like the . in front of an operator: ANDing or ORing the elements of a vector or array:



Relational and Logical Operators

- It is possible to string several relational and logical operators together. It is a good idea to use parenthesis and to understand the order of operation (just like with arithmetic operators).

Operator(s)	Operation	Priority
>, <, >=, <=, ==, ~=	Relational operators	Highest
~	NOT	
&	Elementwise AND	
	Elementwise OR	
&&	Short-circuit AND	
	Short-circuit OR	Lowest

*** All of these operators have lower priority than mathematical operators**

Logical Operator Examples

- If `a = true`, `b = true`, `c = false`, what do the following expressions result in:

— <code>a && b</code>	1 (True)
— <code>a && c</code>	0 (False)
— <code>a b</code>	1 (True)
— <code>c b</code>	1 (True)
— <code>~b</code>	0 (False)
— <code>a && ~c</code>	1 (True)
— <code>(a b) && ~c</code>	1 (True)
— <code>~(a && b) (a && c)</code>	0 (False)

Relational Operator Examples

- If $a = 2$, $b = 3$, $c = 5$, what do the following conditional expressions result in:

— $a > b$	0 (False)
— $a \neq c$	1 (True)
— $a \leq b$	1 (True)
— $c == a + b$	1 (True)
— $(c > a) \ \&\& \ (c > b)$	1 (True)
— $(c > a) \ \ (a > b)$	1 (True)
— $(a \leq b) \ \&\& \ \sim(c == 5)$	0 (False)
— $\sim((a < b) \ \ (b < c))$	0 (False)

Cautions: the == operator

It is very important to understand the difference between = (an assignment operator) and == (a relational operator)

```
>> x = 10
```

% Creates a variable, x, and assigns 10 to that variable.

```
>> x == 10
```

% Checks to see if the variable x is equivalent to 10

The statement `x == 10` will produce an error if x is not defined in the workspace or program; a 1 (TRUE) if x is indeed defined and equal to 10; and a 0 (FALSE) if x is defined but not equal to 10.

Cautions: complex conditions and relational operators

$$15 < A < 25$$

We interpret this expression to be true only if A is between 15 and 25.

In MATLAB, the expression: $15 < A < 25$ is always true for any A! This can create serious problems in your code!

In MATLAB, we must split the expression apart with a logical operator as follows:

$$15 < A \ \&\& \ A < 25$$

Then the expression will be true only if A lies between 15 and 25.

Building Conditions

- What is the condition to test if a point (x,y) is in:

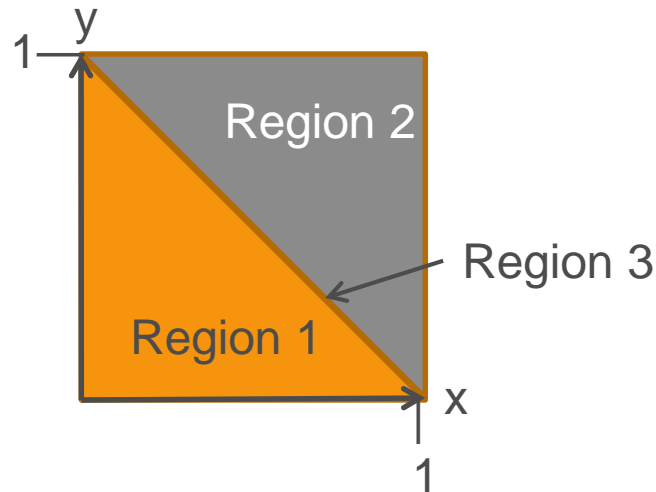
- Region 1 $y < 1 - x$

- Region 2 $y > 1 - x$

- Region 3 $y == 1 - x$

- Not in Region 2 $\sim (y > 1 - x)$

OR $(y < 1 - x) \ || \ (y == 1 - x)$



if ... Structures

if *expression == true*

MATLAB commands

end

MATLAB commands

The expression is the condition being checked. Examples:

if $a == 5$

if $b \sim= 0$

if $a == 5 \parallel b \sim= 0$

if $4 < a \ \&\& \ a \leq 12$

If the expression is true, then the MATLAB commands following the if statement will run.

Specifies where the end of the MATLAB commands in the if statement are

If the expression is false, then the program will skip to the MATLAB commands following the end statement

if ... else ... Structures

```
if      expression == true
```

```
    MATLAB commands
```

← If the expression is true, then the MATLAB commands following the if statement will run.

```
else
```

```
    MATLAB commands
```

← If the expression is false, the MATLAB commands following the else statement will run

```
end
```

```
MATLAB commands
```

if ... elseif ... else ... Structures

```
if expression1  
    MATLAB commands
```

← If expression1 is true, then the MATLAB commands following the if statement will run. Program then jumps to end.

```
elseif expression2  
    MATLAB commands
```

← If the expression1 is false but expression2 is true, the MATLAB commands following the elseif statement will run. Program then jumps to end.

```
else  
    MATLAB commands
```

← If neither of the expressions are true, then the MATLAB commands following the else statement will run. Note: you don't have to include an else statement here if it isn't needed for your code.

```
end  
MATLAB commands
```

Comments on if ... statements

- Multiple **elseif** statements can be included within an **if** statement

```
answer = menu('What day is today?', 'a) Mon', 'b) Tues', 'c) Wed',  
              'd) Thur', 'e) Fri');  
  
if answer == 1  
    disp('Today is not Monday! ');  
elseif answer == 2  
    disp('Today is not Tuesday! ');  
elseif answer == 3  
    disp('That is correct! ');  
elseif answer == 4  
    disp('Today is not Thursday! ');  
elseif answer == 5  
    disp('Today is not Friday! ');  
else  
    disp('Are you sure you entered a day? ');  
end
```


Comments on if ... statements

- **if** statements can be nested within other **if** statements
- Don't include a condition after an **else** statement..

```
temp = input('What is the outside temperature? ');
precip = menu('Is it precipitating outside? ', 'Yes', 'No');
if temp >= 50
    if precip == 2
        disp('It is a nice day, go outside! ');
    else
        disp('It is an ok day, but you're going to get wet! ');
    end
else
    disp('Just stay inside! ');
end
```