Selections in MATLAB

Lecture 4

Dr. Görkem Saygılı

Department of Biomedical Engineering
Ankara University

Introduction to MATLAB, 2017-2018 Spring

Outline:

In this lecture, we will learn:

- if statement and switch statement,
- how use nested statements (if and switch statements inside each other),
- relational operators ($==$, $>=$, $<=$, $!=$, ...),

Sequential Execution:

Generally, programs run in sequential order (most common control construct). That means each line of code is executed after the previous line has been executed.

Can we tell the interpreter to skip the next line if a condition has not been satisfied?

This is called selection or branching (selection construct).

Guessing Game:

Perhaps the most common statement to make selection (branching) is using if statements.

The following is an example of a simple game:

```
1   function guess_0_1(g)
2
3 -     num = round(rand(1));
4
5 -     if g == num
6 -         disp('you guessed correctly');
7 -     end
```

```
New to MATLAB? See resources for Getting Started.

>> guess_0_1(1)
you guessed correctly
>> guess_0_1(1)
you guessed correctly
>> guess_0_1(1)
fx >>
```

If Statement in MATLAB:

In order to use if statement without errors, we need to follow some rules:

```
1   function guess_0_1(g)
2
3 –   num = round(rand(1));
4
5 –   if g == num
6 –       disp('you guessed correctly');
7 –   end
```

```
New to MATLAB? See resources for Getting Started.

>> guess_0_1(1)
you guessed correctly
>> guess_0_1(1)
you guessed correctly
>> guess_0_1(1)
>>
```

If-Else Statement:

```
1    function guess_0_1_if_else(g)
2
3 -  num = round(rand(1));
4
5 -  if g == num
6 -      disp('you guessed correctly');
7 -  else
8 -      disp('your guess is wrong');
9 -  end
```

```
New to MATLAB? See resources for Getting Started.

>> guess_0_1_if_else(1)
you guessed correctly
>> guess_0_1_if_else(1)
you guessed correctly
>> guess_0_1_if_else(1)
your guess is wrong
>>
```

Remember: End Keyword for Functions:

Remember that use of end is not obligatory for functions:

\rand_2_8.m

```
1   function res = rand_2_8(n)
2       % the following generates a scalar
3       % between 1 and 10
4       disp(' ');
5       disp('scalar (1-10): ');
6       res_1_10 = rand_1_10(1);
7       disp(res_1_10);
8       % the following creates random
9       % integer from 2 to 8
10      res = 2+round(6*rand(n));
11      disp('matrix of size n (2-8): ');
12      disp(res)
13  end
14
15  function res = rand_1_10(n)
16      res = 1+round(9*rand(n));
17  end
18
```

Command Window

New to MATLAB? See resources for Getting Started.

```
>> rand_2_8(5);

scalar (1-10):
     5

matrix of size n (2-8):
     4     4     8     6     3
     8     5     4     6     2
     8     5     6     3     5
     2     8     6     3     7
     6     5     5     8     6

fx >>
```

if-elseif-else:

```
1   function guess_my_age(guess)
2
3   age = 32;
4   if guess < age
5       disp('I am not that young!');
6   elseif guess > age
7       disp('Well, there is still some time for me to
8   else
9       disp('Well done! Your guess is correct!');
10  end
11
```

```
New to MATLAB? See resources for Getting Started.

>> guess_my_age(22)
I am not that young!
>> guess_my_age(42)
Well, there is still some time for me to reach that age!
>> guess_my_age(32)
Well done! Your guess is correct!
fx >>
```

The Return Statement:

```
1   function guess_my_age_return(guess)
2
3   age = 32;
4   if guess < age
5       disp('I am not that young!');
6   elseif guess > age
7       disp('Well, there is still some time for me to
8   else
9       disp('Well done! Your guess is correct!');
10      return;
11  end
12  disp('Lets try again!');
13
```

```
New to MATLAB? See resources for Getting Started.

>> guess_my_age_return(22)
I am not that young!
Lets try again!
>> guess_my_age_return(42)
Well, there is still some time for me to reach that age!
Lets try again!
>> guess_my_age_return(32)
Well done! Your guess is correct!
fx >>
```

Switch Statement:

```
1   function check_matlab_course(day_of_week)
2
3   switch day_of_week
4       case 1
5           disp('Yes, we have two hours of lecture!');
6       case 2
7           disp('No course today, so sad!');
8       case 3
9           disp('Well, today is Lab day for Group A');
10      case 4
11          disp('Well, today is Lab day for Group B');
12      case 5
13          disp('No course today, so sad!');
14      case 6
15          disp('It is weekend!');
16      case 7
17          disp('It is weekend!');
18      otherwise
19          disp('It is not a valid day!');
20          return
21  end
22
```

New to MATLAB? See resources for Getting Started.

```
>> check_matlab_course_num(1)
Yes, we have two hours of lecture!
>> check_matlab_course_num(3)
Well, today is Lab day for Group A
>> check_matlab_course_num(4)
Well, today is Lab day for Group B
>> check_matlab_course_num(5)
No course today, so sad!
>> check_matlab_course_num(7)
It is weekend!
>> check_matlab_course_num('Monday')
It is not a valid day!
fx >>
```

Switch Statement with Strings:

```
1   function check_matlab_course(day_of_week)
2
3   switch day_of_week
4       case 'Monday'
5           disp('Yes, we have two hours of lecture!');
6       case 'Tuesday'
7           disp('No course today, so sad!');
8       case 'Wednesday'
9           disp('Well, today is Lab day for Group A');
10      case 'Thursday'
11          disp('Well, today is Lab day for Group B');
12      case 'Friday'
13          disp('No course today, so sad!');
14      case 'Saturday'
15          disp('It is weekend!');
16      case 'Sunday'
17          disp('It is weekend!');
18      otherwise
19          disp('It is not a valid day!');
20          return
21  end
```

New to MATLAB? See resources for Getting Started.

```
>> check_matlab_course('Monday')
Yes, we have two hours of lecture!
>> check_matlab_course('Wednesday')
Well, today is Lab day for Group A
>> check_matlab_course('Thursday')
Well, today is Lab day for Group B
>> check_matlab_course('Friday')
No course today, so sad!
>> check_matlab_course('Saturday')
It is weekend!
>> check_matlab_course('matlab')
It is not a valid day!
fx >>
```

Using a Set in Case:

```
1   function check_matlab_course_set(day_of_week)
2
3   switch day_of_week
4       case {'Monday', 1}
5           disp('Yes, we have two hours of lecture!');
6       case {'Tuesday', 2}
7           disp('No course today, so sad!');
8       case {'Wednesday', 3}
9           disp('Well, today is Lab day for Group A');
10      case {'Thursday', 4}
11          disp('Well, today is Lab day for Group B');
12      case {'Friday', 5}
13          disp('No course today, so sad!');
14      case {'Saturday', 6}
15          disp('It is weekend!');
16      case {'Sunday', 7}
17          disp('It is weekend!');
18      otherwise
19          disp('It is not a valid day!');
20          return
21  end
```

```
New to MATLAB? See resources for Getting Started.

>> check_matlab_course_set(1)
Yes, we have two hours of lecture!
>> check_matlab_course_set('Wednesday')
Well, today is Lab day for Group A
>> check_matlab_course_num(4)
Well, today is Lab day for Group B
>> check_matlab_course_set('Friday')
No course today, so sad!
>> check_matlab_course_set(7)
It is weekend!
>> check_matlab_course_set('Matlab')
It is not a valid day!
fx >>
```

If vs. Switch:

If and Switch statements are two powerful control construct. In many cases, they can be used both, the choice is upto the programmer.

Yet, Switch statement is more suitable when the control variable is discrete.

In contrast, if statement is preferred over switch when control statement is consists of inequalities.

Relational Operators:

$==$: is equal to.

$=$: is not equal to.

$>$: greater than.

$<$: smaller than.

$>=$: greater or equal to.

$<=$: smaller or equal to.

Relational Operators in If Statements:

```
1   function guess_my_age_return(guess)
2
3   age = 32;
4   if guess < age
5       disp('I am not that young!');
6   elseif guess > age
7       disp('Well, there is still some time for me to
8   else
9       disp('Well done! Your guess is correct!');
10      return;
11  end
12  disp('Lets try again!');
13
```

```
New to MATLAB? See resources for Getting Started.

>> guess_my_age_return(22)
I am not that young!
Lets try again!
>> guess_my_age_return(42)
Well, there is still some time for me to reach that age!
Lets try again!
>> guess_my_age_return(32)
Well done! Your guess is correct!
fx >>
```

Relational Operator Output:

```
>> 3 == 2

ans =

     0

>> abs(-3) == 3

ans =

     1

>> 2+3 <= 2*3

ans =

     1

>>
```

Relational Operator with Arrays:

```
>> [1 2 3 4] < [5 6 7 8]

ans =

     1     1     1     1

>> [1*8 2+7 3 4] < [5 6 7 8]

ans =

     0     0     1     1

>> [1*8 2+7 3 4] < [5 6 7 8 9]
Error using  <
Matrix dimensions must agree.

>>
```

Logical Operators:

&&: and

||: or

 : not

xor: xor

Example:

```
1   function check_matlab_course_if(d)
2
3   if strcmp(d, 'Monday') || strcmp(num2str(d), '1')
4       disp('Yes, we have two hours of lecture!');
5   elseif strcmp(d, 'Tuesday') || strcmp(num2str(d), '2')
6       disp('No course today, so sad!');
7   elseif strcmp(d, 'Wednesday') || strcmp(num2str(d), '3')
8       disp('Well, today is Lab day for Group A');
9   elseif strcmp(d, 'Thursday') || strcmp(num2str(d), '4')
10      disp('Well, today is Lab day for Group B');
11  elseif strcmp(d, 'Friday') || strcmp(num2str(d), '5')
12      disp('No course today, so sad!');
13  elseif strcmp(d, 'Saturday') ||strcmp(num2str(d), '6')
14          disp('It is weekend!');
15  elseif strcmp(d, 'Sunday') || strcmp(num2str(d), '7')
16          disp('It is weekend!');
17  else
18          disp('It is not a valid day!');
19  end
```

```
New to MATLAB? See resources for Getting Started.

>> check_matlab_course_if(1)
Yes, we have two hours of lecture!
>> check_matlab_course_if('Wednesday')
Well, today is Lab day for Group A
>> check_matlab_course_if(4)
Well, today is Lab day for Group B
>> check_matlab_course_if('Friday')
No course today, so sad!
>> check_matlab_course_if(7)
It is weekend!
>> check_matlab_course_if('Matlab')
It is not a valid day!
fx >>
```

Precedence of Operators in MATLAB:

Precedence	Operator
0	Paranthesis (...)
1	exp (^) and transpose (')
2	Unary +,- logical negation (~)
3	Multip. (*), division (/)
4	Addition (+), Subtraction (-)
5	Colon Operator (:)
6	Relational Op. (<, >, <=,...)
7	Elementwise logical and (&)
8	Elementwise logical or (|)
9	Logical Op (&&)
10	Logical Op (||)

Nested Statements:

```
1   function guess_my_age_mod(guess)
2
3   age = 32;
4   if guess == age
5       disp('Well done! Your guess is correct!');
6   else
7       if guess < age
8           disp('I am not that young!');
9       else
10          disp('Well, there is still some time for me to re
11      end
12  end
```

```
New to MATLAB? See resources for Getting Started.

>> guess_my_age_mod(22)
I am not that young!
>> guess_my_age_mod(42)
Well, there is still some time for me to reach that age!
>> guess_my_age_mod(32)
Well done! Your guess is correct!
fx >>
```