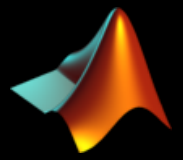

Programming with MATLAB

If-expressions

Dimitris Voudouris, PhD
d.s.voudouris@gmail.com

If I am tired, then I (should) rest

If-expressions **are called** ***conditionals***
If a condition/expression is satisfied, ***then*** something will happen

```
year = 2025;
if year == 2025 4  % if the variable year is equal to 2025, then
    disp('condition is met')          % display this text
end                  % each if-expression must close with an end

if year == 2000 + 25          % year is still equal to 2025, so
    disp('condition is met')          % display this text
end                           % end the expression
```

Note that the expression works on logical operations (true or false)!
Therefore, we use the two equal sign (==)

If I am hungry, then I eat

You can use other logical operators as well

```
if year > 2000 - 25 + 32                % is year greater than 2007?
        disp('condition is met')        % if yes, display this text
end                                     % end expression

if year ~= 2021                         % is year different from 2021?
        disp('condition is met')        % if yes, display this text
end                                     % end expression

% NOTE:
if year                                 % does year exist in the workspace?
        disp('condition is met')        % if yes, display this text
end                                     % end expression
```

… else, I do not eat

Alternatives can also be included

```
if year == 2022                    % is year equal to 2022?
        disp('condition is met')   % if yes, display this text
else                               % if not
        disp('condition is not met') % display this
end                                % end expression
```

… else, I do not eat

Alternatives can also be included

```
if year == 2022                    % is year equal to 2022?
        disp('it is 2022')         % if yes, display this text
elseif year > 2022                 % else if year is greater than 2022
        disp('it is future')       % display this
elseif year < 2022                 % else if year is smaller than 2022
        disp('it is still past')   % display this
end                                % end expression
```

The last else-if expression is redundant:

If the year is neither equal to 2022, nor greater than 2022, then it must be smaller than 2022. So we do not need to explicitly state this.

… else, I do not eat

So the previous statement can be written as:

```
if year == 2022                    % is year equal to 2022?
      disp('it is 2022')           % if yes, display this text
elseif year > 2022                 % else if year is greater than 2022
       disp('it is future')        % display this
else                               % otherwise
      disp('it is still past')     % display this
end                                % end expression
```

Nested if-expressions

Combine conditions:

```
if year == 2022                                  % is year equal to 2022?
        disp('it is 2021')                       % if yes, display this text
elseif year > 2022                               % else if year is greater than 2022
        disp('it is future')
         if year < 2026                          % if year > 2022 but < 2026
             disp('this')
        elseif year >= 2026 & year < 2030        % if year >= 2026 but < 2030
             disp('that')
        elseif year >= 2030                      % if year > 2030
             disp('the other')
        end                                      % end expression of the current if
end                                              % end expression of the earlier if
```

Nested if-expressions

In the previous slide, the *end* aligns with the *if* that it belongs to.

Select your code and press ctrl + I to format it accordingly
For Macintosh users, press cmd + I

This way Matlab will let you better see the architecture of your code.

Ask for input

Ask the user to give an input, then check this input:

```
age = inputdlg('Please type in your age:')
% age will be a character vector in a cell
% if you expect numerical input, you need to convert it to a number
if str2double(age) > 18                % str2double or str2num ?
        disp('you are an adult')       % display this
else
        disp('you are not an adult')   % display this
end
```

You can also use the function *input*, which does not open a dialog box

See the differences in the format of the input between the two functions (*help inputdlg* and *help input*)

str2double vs str2num

In many cases, the two commands will return the same answer.
However, there are some subtle, yet important differences.

str2double(str) converts the text in str to double precision values, and str should represent certain numerical values
str2num(str) converts the input to a numerical matrix, and str should not necessarily be a number, but can also be a character.

You can type the value of one thousand either like ‘1000’ or ‘1.000’. But
str2double(‘1.000’); % will return 1
str2double(‘1000’); % will return 1000

And:
str2double(‘1,000’)
str2num(‘1,000’)

You can also evaluate strings

Ask the user to type in their name:

```
name = input('What is your name?', 's');
% input expects a numeric input (contrary to inputdlg that creates cells!)
% If you want a string, you need to specify this by adding the 's' in the end
if strcmp(name, 'Dimitris')                          % if the name is Dimitris
        disp('your name is Dimitris')                % display this
elseif strcmp(name, 'Alex')                          % elseif the name is Alex
        disp('your name is Alex')                    % display this
else
        disp('your name is neither Dimitris nor Alex')   % display this
end                                                  % end expression
```

What will happen in this case?

Ask the user to type in their name (type in 'Daniel'):

```
name = input('What is your name?', 's');

if strcmp(name, 'Dimitris')                % if the name is Dimitris
        disp('your name is Dimitris')
elseif strcmp(name, 'Alex')                % elseif the name is Alex
        disp('your name is Alex')
elseif strfind(name, 'D')                  % elseif you find the letter D…
        disp('your name contains a D' )
end                                        % end statement
```

Think for 1 minute before you execute!

Then proceed to the next slide and compare
this and the next expression as well as their outputs

What will happen in this case?

Ask the user to type in their name (type in Daniel):

```
name = input('What is your name?', 's');

if strfind(name, 'D')                    % If you find the letter D…
        disp('your name contains a D' )
elseif strcmp(name, 'Dimitris')          % if the name is Dimitris
        disp('your name is Dimitris')
elseif strcmp(name, 'Alex')              % elseif the name is Alex
        disp('your name is Alex')
end                                      % end expression
```

Once an if-expression is true, subsequent expressions are skipped

Enjoy!