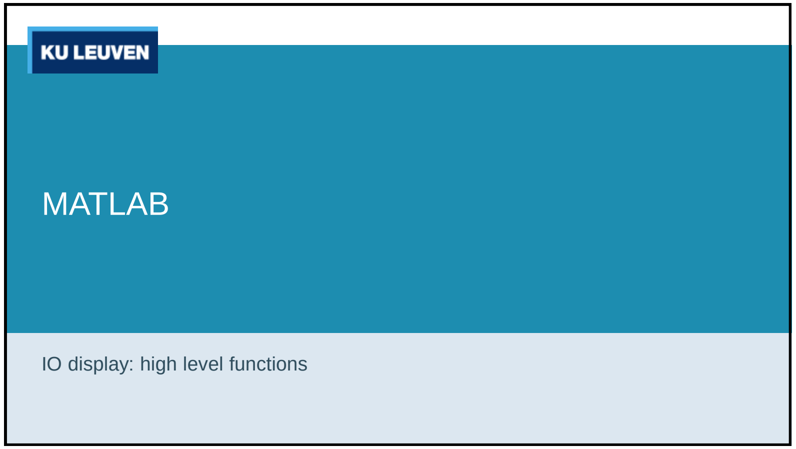
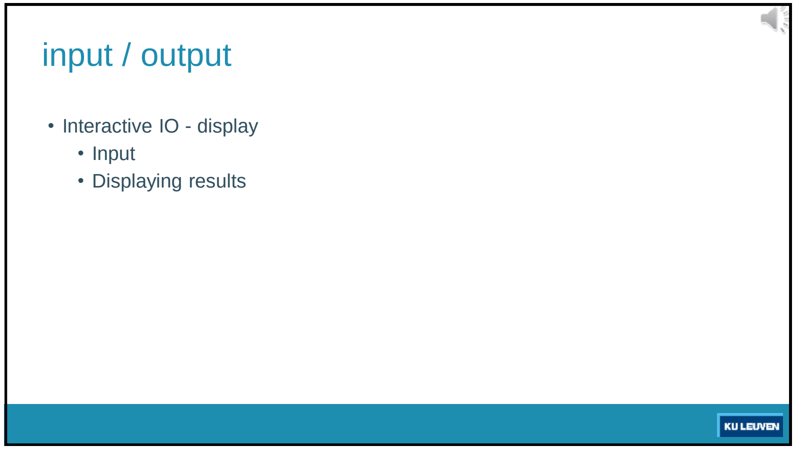


KU LEUVEN

MATLAB

IO display: high level functions

1

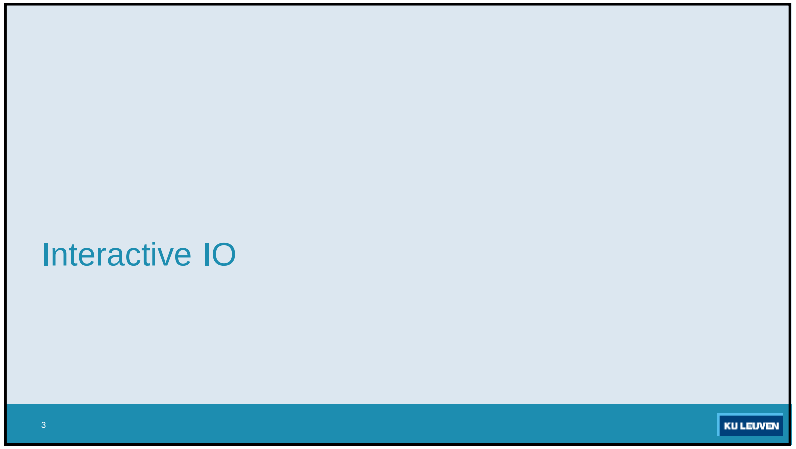


input / output

- Interactive IO - display
  - Input
  - Displaying results

KU LEUVEN

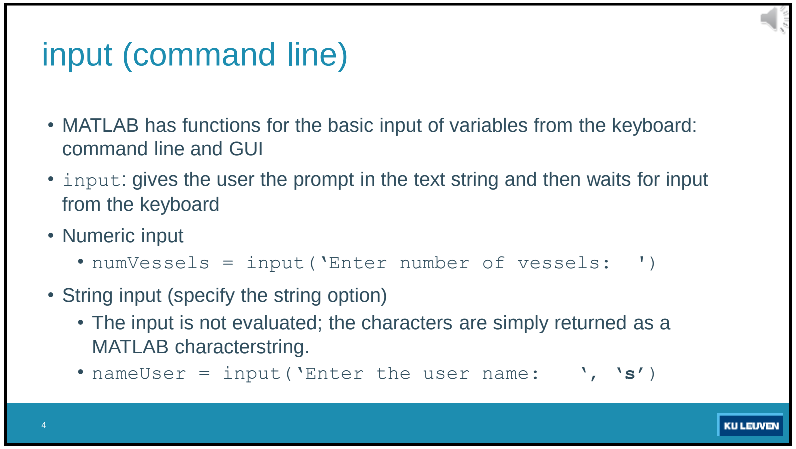
2



Interactive IO

KU LEUVEN

3

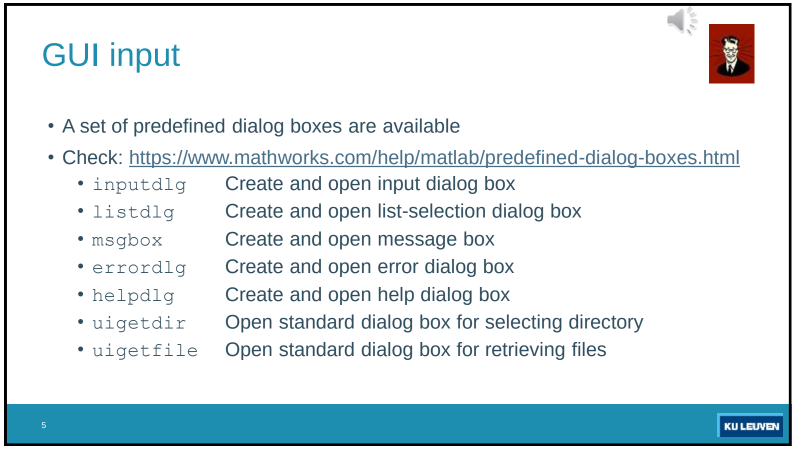


input (command line)

- MATLAB has functions for the basic input of variables from the keyboard: command line and GUI
- `input`: gives the user the prompt in the text string and then waits for input from the keyboard
- Numeric input
  - `numVessels = input(‘Enter number of vessels:  ')`
- String input (specify the string option)
  - The input is not evaluated; the characters are simply returned as a MATLAB characterstring.
  - `nameUser = input(‘Enter the user name:    ‘, ‘s’)`

KU LEUVEN

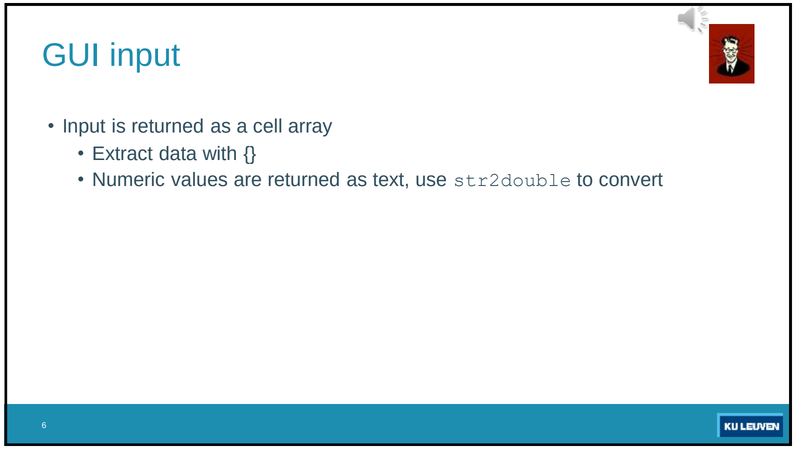
4



GUI input

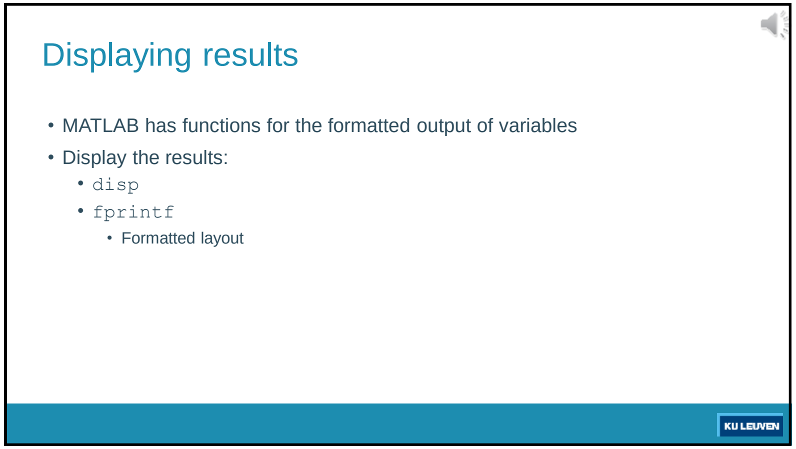
- A set of predefined dialog boxes are available
- Check: https://www.mathworks.com/help/matlab/predefined-dialog-boxes.html

`inputdlg`	Create and open input dialog box
`listdlg`	Create and open list-selection dialog box
`msgbox`	Create and open message box
`errordlg`	Create and open error dialog box
`helpdlg`	Create and open help dialog box
`uigetdir`	Open standard dialog box for selecting directory
`uigetfile`	Open standard dialog box for retrieving files



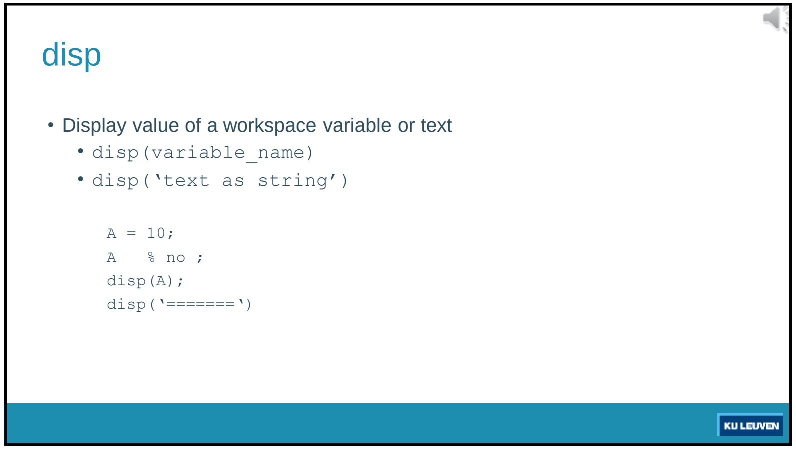
GUI input

- Input is returned as a cell array
  - Extract data with {}
  - Numeric values are returned as text, use `str2double` to convert



Displaying results

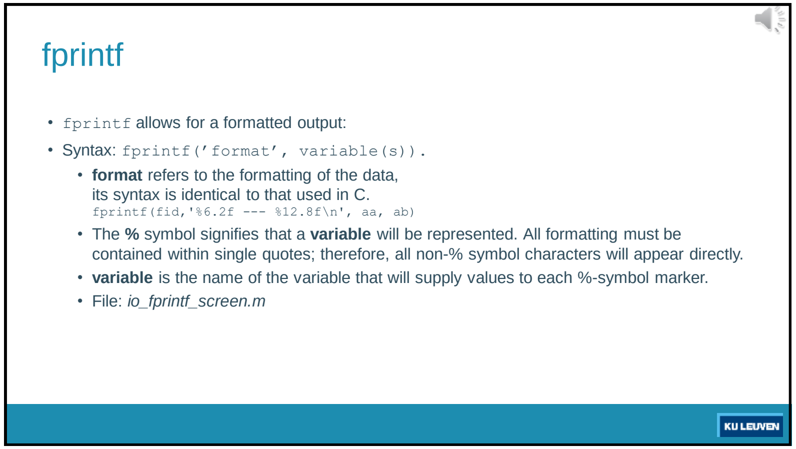
- MATLAB has functions for the formatted output of variables
- Display the results:
  - `disp`
  - `fprintf`
    - Formatted layout



disp

- Display value of a workspace variable or text
  - `disp(variable_name)`
  - `disp(‘text as string’)`

```
A = 10;
A   % no ;
disp(A);
disp(‘=======‘)
```

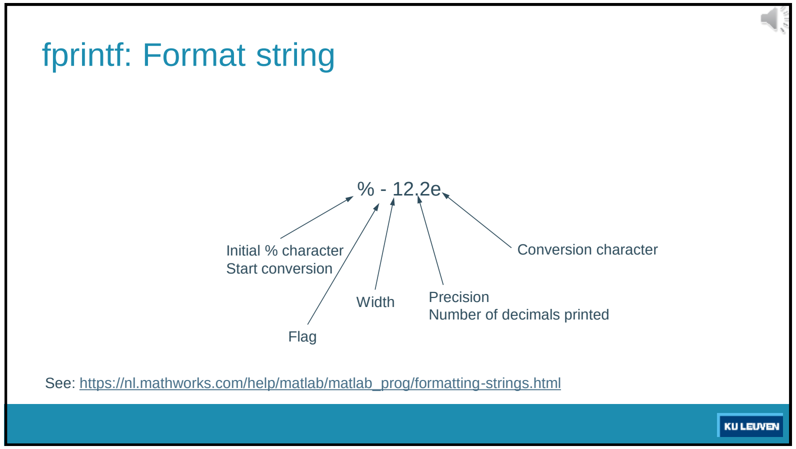


fprintf

- `fprintf` allows for a formatted output:
- Syntax: `fprintf(’format’, variable(s)).`
  - **format** refers to the formatting of the data, its syntax is identical to that used in C.
    `fprintf(fid,'%6.2f --- %12.8f\n', aa, ab)`
  - The **%** symbol signifies that a **variable** will be represented. All formatting must be contained within single quotes; therefore, all non-% symbol characters will appear directly.
  - **variable** is the name of the variable that will supply values to each %-symbol marker.
  - File: *io_fprintf_screen.m*

KU LEUVEN

9



fprintf: Format string

% - 12.2e

Initial % character
Start conversion

Flag

Width

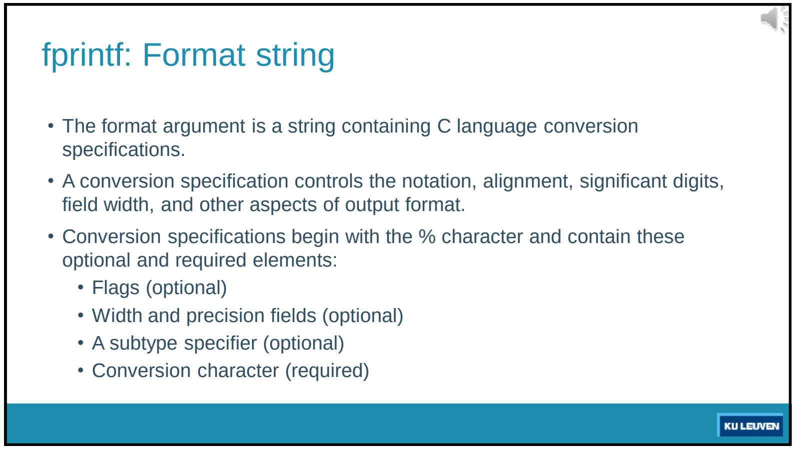
Precision
Number of decimals printed

Conversion character

See: https://nl.mathworks.com/help/matlab/matlab_prog/formatting-strings.html

KU LEUVEN

10

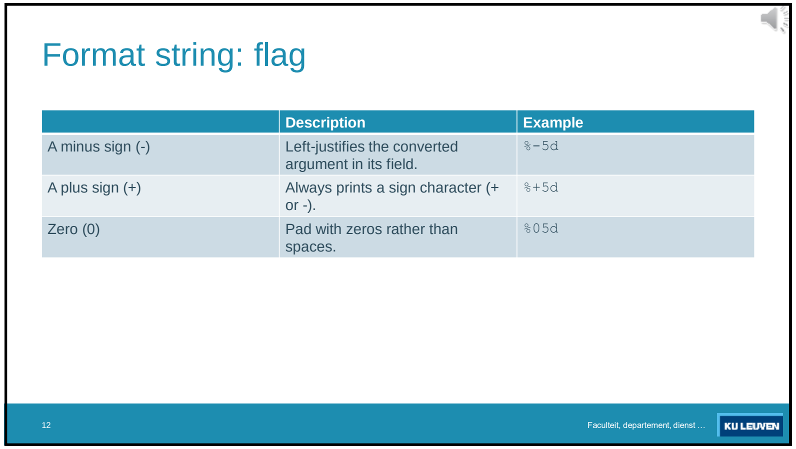


fprintf: Format string

- The format argument is a string containing C language conversion specifications.
- A conversion specification controls the notation, alignment, significant digits, field width, and other aspects of output format.
- Conversion specifications begin with the % character and contain these optional and required elements:
  - Flags (optional)
  - Width and precision fields (optional)
  - A subtype specifier (optional)
  - Conversion character (required)

KU LEUVEN

11

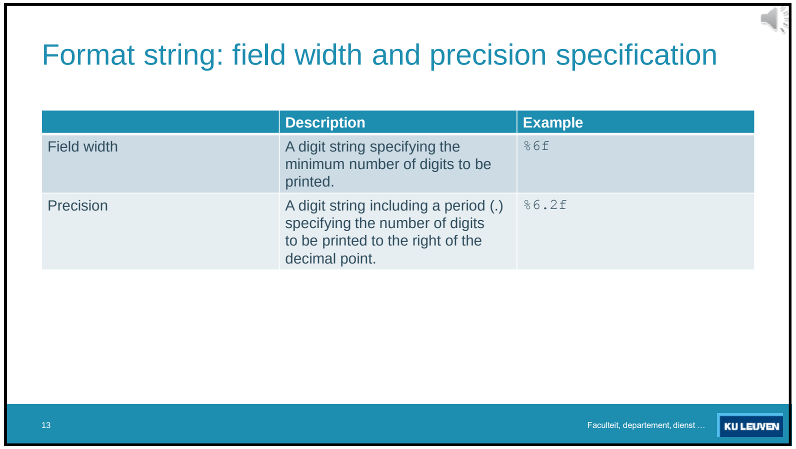


Format string: flag

	Description	Example
A minus sign (-)	Left-justifies the converted argument in its field.	`%-5d`
A plus sign (+)	Always prints a sign character (+ or -).	`%+5d`
Zero (0)	Pad with zeros rather than spaces.	`%05d`

KU LEUVEN

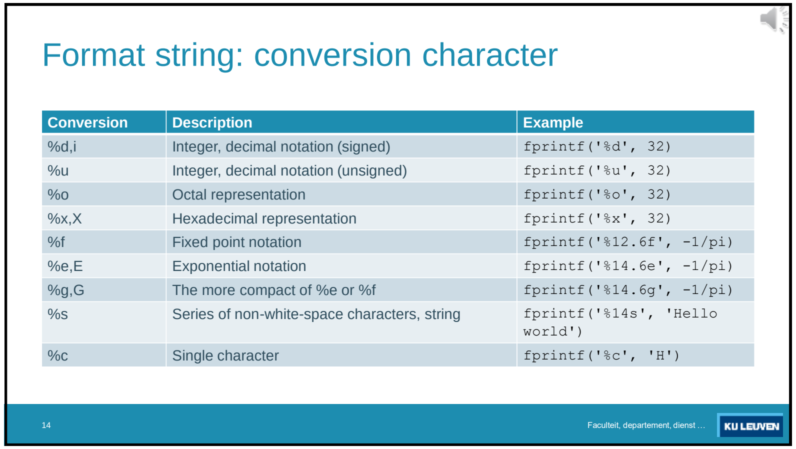
12



Format string: field width and precision specification

	Description	Example
Field width	A digit string specifying the minimum number of digits to be printed.	`%6f`
Precision	A digit string including a period (.) specifying the number of digits to be printed to the right of the decimal point.	`%6.2f`

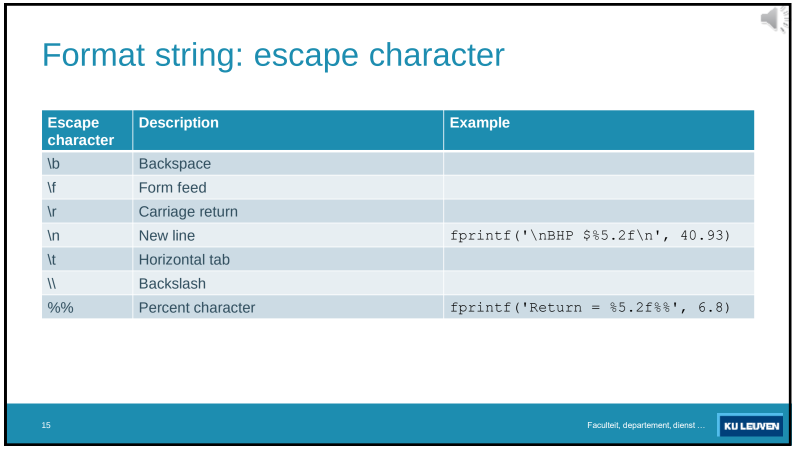
13



Format string: conversion character

Conversion	Description	Example
%d,i	Integer, decimal notation (signed)	`fprintf('%d', 32)`
%u	Integer, decimal notation (unsigned)	`fprintf('%u', 32)`
%o	Octal representation	`fprintf('%o', 32)`
%x,X	Hexadecimal representation	`fprintf('%x', 32)`
%f	Fixed point notation	`fprintf('%12.6f', -1/pi)`
%e,E	Exponential notation	`fprintf('%14.6e', -1/pi)`
%g,G	The more compact of %e or %f	`fprintf('%14.6g', -1/pi)`
%s	Series of non-white-space characters, string	`fprintf('%14s', 'Hello world')`
%c	Single character	`fprintf('%c', 'H')`

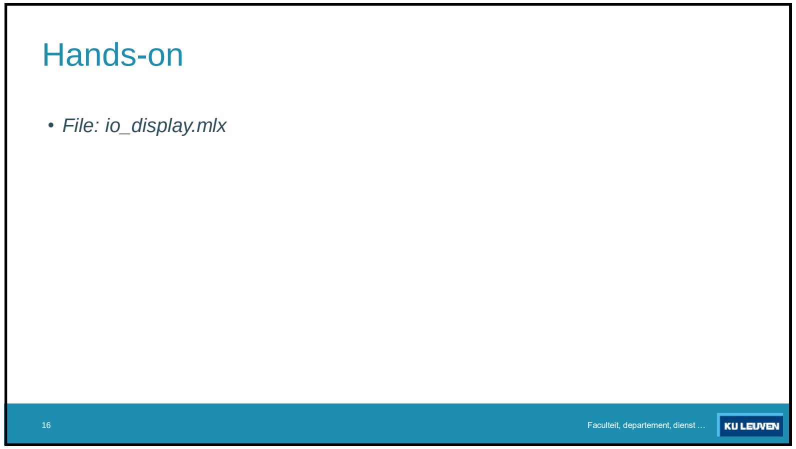
14



Format string: escape character

Escape character	Description	Example
\b	Backspace	
\f	Form feed	
\r	Carriage return	
\n	New line	`fprintf('\nBHP $%5.2f\n', 40.93)`
\t	Horizontal tab	
\\	Backslash	
%%	Percent character	`fprintf('Return = %5.2f%%', 6.8)`

15



Hands-on

- *File: io_display.mlx*

16