Unit 3. Matlab Syntax (II)

3.1 Variables
3.2 Expressions
3.3 Fundamental data types
3.3 Operators
3.4.Screen output, input and comments

PRINT ON SCREEN

Print on Screen

- When placed at the end of a command, the semicolon ***;*** tells MATLAB not to display any output from that command. For example:

```
>> guest = 20
    guest =20

>> 24 * 5 + 2
    ans =122
```

```
>> guest = 20;


>> 24 * 5 + 2;
```

The operations are performed, but this time MATLAB does not display the result on the screen because the lines end up with ;

Print on Screen

- The disp function displays a **fixed** line of text on screen in the command window

  >> *disp('Show this text')*

  Show this text

  - disp is usefull to show welcome messages, instruct the user to do something or, in general, to show information that does not contain any variable

Print on Screen

- The fprintf function write formated data to the command window

  fprintf ('text', var1, var2,….)

  - var1, var2,… are the variables whose values we want to print. You can also use specific values instead of variables
  - The **text** is a chain of characters in which some ***control characters*** have been included. When Matlab finds one of these characters it will replace it by one of the variables. The **control characters** are:

    %d for variables of integer type

    %f for variables of float point type

    %c for variables of char type

MATLAB replaces the characters by the variables just by following the order of the variables

Print on Screen

- Example

```
varAge = 20;
fprintf( 'My age is %d years', varAge);
```

My age is 20 years

- Example

```
fprintf( 'My favourite number is %f', 10/3);
```

My favourite number is 3.333333

Print on Screen

- It is possible to specify the precision of the number by using the character control ***%n.mf***

  - where **n** is the number of integer digits to display and **m** is the number of decimals. If **n** is not provided MATLAB displays all the integer digits

  Example:

  *fprintf('My favourite number is **%1.2f**', 10/3);*

  *or fprintf('My favourite number is **%.2f**', 10/3);*

  My favourite number is 3.33

Print on Screen

- What MATLAB will print on screen?

```
var1  = 'a';
var2  = 'n';
fprintf('In my name there are %d letters %c and %d letter %c', 2, var1, 1, var2);
```

Print on Screen

- What MATLAB will print on screen?

var1 = 'a';

var2 = 'n';

fprintf('In my name there are %d letters %c and %d letter %c', 2, var1, 1, var2);

In my name there are 2 letters a and 1 letter n

MATLAB matches the numbers and variables with the character-controls just by following their order

%d	2
%c	var1
%d	1
%c	var2

Print on Screen

- More control characters to use with *fprintf*
  - When you want to change the line or to include tabs in your text:

\n	new line
\t	tab

  - If you want to include quotation marks or percent characters in your text you should use:

‘ ‘	single quotation mark
%%	percent character

Print on Screen

- Example:

*fprintf('My favourite **\n** number is %1.2f', 10/3);*

My favourite

number is 3.33

Instead of printing the \n MATLAB changes to a new line

- Example:

*fprintf('My favourite **\t** number is %1.2f', 10/3);*

My favourite number is 3.33

MATLAB replaces the \t by a tab

Print on Screen

- Summary:
  - We used *disp* only when we want to display a fixed text in the screen. Never used *disp* if you want to display the current value of a variable
  - To display text and the current values of one or more variables use **fprintf**

Print on Screen

From now on we should always format the output of our programs.

NO MORE OUTPUTS USING ans = ….

Exercise (I)

- Write a program which asks the user to introduce a number, and next it prints it on screen.

  Execution example:

  Please, introduce a number: 4

  You have introduced the number 4

Exercise (I)

SOLUTION:

```
myNumber = input( ‘Please, introduce a number: ‘);
fprintf( ‘\n You have introduced the number: %d’ , myNumber);
```

Exercise (II)

- Write a program that asks the user to introduce a real number, and next it prints it on screen only with 2 decimals

  Execution example:

  Please, introduce a number: 3.14159

  You have introduce the number 3.14

Exercise (II)

SOLUTION:

```
myNumber = input( ‘Please, introduce a number:  ‘);
fprintf( ‘\n You have introduced the number: %.2f’ , myNumber);
```

Exercise (III)

- Write a program that asks the user to introduce a character, and next it prints it on screen.

  Execution example:

  Please, introduce a character: T

  You have introduced the character T

Exercise (III)

SOLUTION:

myChar = input(‘Please, introduce a character: ’ ,’s’);
fprintf(‘\n You have introduced the character: %c’ , myChar);

Exercise (IV)

- Create a program which asks the user to introduces the price of a product and it prints the correspondent VAT (21%) and the total price.

The output of the execution may look like this:

Introduce the price: *200*

The VAT quantity is 42

The total price is 242

Exercise (IV)

```
v_price = input( ‘Introduce the price:  ‘);
v_VAT = v_price * 21 / 100;
v_total = v_price+ v_VAT;
fprintf( ‘\n The VAT quantity is %f’, v_VAT);
fprintf( ‘\n The total price is %f’, v_total);
```

Exercise (V)

- Write a program that asks the user to input the number of month and number of day of the current date and calculates the number of days since the beginning of the year . To simplify the problem, assume that there are 30 days in each month.

  The output of the execution may look like this:

  Introduce the month: *10*

  Introduce the day: *28*

  298 days since the beginning of the year

Exercise (V)

```
month = input('Introduce the month: ');
day = input('Introduce the day: ');
daysPassed = (month-1)*30+day ;
fprintf( ‘\n %d since the beginning of the year’, daysPassed);
```

DOCUMENTING YOUR PROGRAM

Comments

- It is important to write programs easy to read and understand
  - Use meaningful variable names and indent program lines.
- You can include notes and comments in your program to help the reader understand what you are doing. In order to do this you can use:
  - % Any following text after the % is ignored by Matlab
  - %{ When you want to write larger comments you put them }% between %{ and }%

Comments. Example

```
%{      Author: Telmo Zarraonandia
        Date: 30/09/2020                    } %

data = zeros(1,3);          % Initially we fill the vector with zeros
data(1) = input('Introduce the NIA: ');
data(2) = input('Introduce the age: ');
data(3) = input('Introduce the number of courses: ');

% Now we print the data following the format specified in
% the problem description

fprintf('\n The age of the student %d is %d, and he/she is currently
enrolled in %d courses', data(1), data(2), data(3));
```

Exercise

- Write a program which asks the user to introduce the ages of three students and then prints them on screen. Solve the problem using three different variables to store the ages of the students.

Exercise

```
clear;
studentA = input('Introduce the age of a student: ‘);
studentB = input('Introduce the age of another student: ‘);
studentC = input('Introduce the age of another one: ‘);
fprintf(‘\n The ages of the students are %d, %d and %d’,
studentA, studentB, studentC);
```

The command *clear* cleans the MATLAB memory (the so-called ‘workspace’). It is a good practice to include it at the beginning of our programs, as it can save us for having unexpected errors.

Exercise

- Write a program which asks the user to introduce the ages of three students and then prints them on screen. Solve the problem storing the ages in the positions of a vector of 1 row and 3 columns:
  - the age of the first student in the row 1 column1, the age of the second student in the row 1 column 2, and the age of the third student in the row 1 column 3.

Exercise

```
clear
students = zeros(1, 3);
students(1) = input('Introduce the age of a student: ');
students(2) = input('Introduce the age of another student: ');
students(3) = input('Introduce the age of another one: ');
fprintf('\n The ages of the students are %d, %d and %d',
students(1), students(2), students(3));
```

Note we have created an initial vector of 1 row and 3 columns all zeros. Then we have updated the values in the vector with the numbers the user introduced.

Your program would also work if you don't initialize the vector with zeros… but it is another good practice to do it this way

Exercise

- Modify the previous program so that MATLAB asks the user to introduce a number between 1 and 3 and prints the age of the correspondent student.

  The output of the execution may look like this:

  Introduce the age of a student: 19

  Introduce the age of another student: 20

  Introduce the age of another one: 18

  Introduce a number: 2

  The student number 2 is 20 years old

Exercise

```
Clear;
students = zeros(1, 3);
students(1) = input('Introduce the age of a student: ‘);
students(2) = input('Introduce the age of another student: ‘);
students(3) = input('Introduce the age of another one: ‘);
num = input('Introduce a number: ‘);
fprintf(‘\n The student number %d is %d years old’, num,
students(num));
```

Exercise

- Write a program which asks the user to introduce 4 numbers and puts them in a matrix of 2 rows and 2 columns. Next, it shows in screen the average value of all the values introduced

Exercise

```
clear;
matnum = zeros(2,2);
matnum(1,1) = input('Introduce a number: ');
matnum(1,2) = input('Introduce a number: ');
matnum(2,1) = input('Introduce a number: ');
matnum(2,2) = input('Introduce a number: ');
average = (matnum(1,1)+ matnum(1,2) + matnum(2,1) +
matnum(2,2))/4;
fprintf('The average value is %.2f', average);
```