Programming in Matlab

Day 1: Basics

Matlab as a calculator

Command Window

```
>> 2+2
ans =

     4

>> 2^3
ans =

     8

>> sqrt(2)
ans =

    1.4142
```

Basic operations

- Sum: +
- Difference: -
- Multiplication: *
- Division: /
- Power: ^
- Square root: sqrt()
- Exponential: exp()
- Logarithm: log()
- ...

Useful: Key ↑ brings back the past.

Variables

Command Window

```
>> a=2
a =

     2

>> b=3;
>> a+b
ans =

     5

>> a*(b+2)
ans =

     10
>> resultado=b+a^2

resultado =

     7
```

Workspace

a	2
ans	10
b	3
resultado	7

Variable: Object that stores information, and which can be used in operations.

Create variable **a** and assign value 2.

Semicolon (**;**) prevents visual output

Parentheses to group operations

Usually we assign the result of an operation to another variable (otherwise, it goes to ans)

Variables

- They can have any name, except:
  - Names beginning by numbers.
  - Names with "strange" characters (ñ,ó,?, etc) or reserved characters (+,-,*,/,^, etc.)
  - Names used for other functions (this is especially dangerous. Matlab will not issue an error, but the function whose name you have used will not work until the variable is deleted).

- Valid names:
  - a
  - chopo
  - Gr17
  - a7jD28r14

- Non-valid names:
  - 1gatito (starts by a number)
  - a+b (includes character “+”, reserved for summation).
- Not recommended:
  - disp (it is the name of a function)

  To check if a name is used by a Matlab function, just type it in Command Window. If matlab says

  ??? Undefined function or variable 'nombre'.

  Then you can use it.

- Matlab distinguishes between upper-case and lower-case characters
  - Variable `gatito` is NOT the same as variable `Gatito`.

Variables

- Basic types
  - Text (also known as Characters). It goes between apostrophes.
    Example: `'Hola'` is text. `Hola` (without apostrophes) is understood by Matlab as the name of a variable or a function
  - Numbers.
- Definition of variables

  Variables are created the first time they are defined. If the variable already exists, it is overwritten.

  Examples:

```
>> a = 17
a =
        17

>> b = 'Hola'
b =
Hola

>> c = a
c =
        17

>> b=14
b =
        14
```

Problem: **Calculotes**

Self-referent variable operations

```
>> b = 1;
>> b = b + 1
b =
    2
```

This is the old value of variable b (in this case, 1)

save, load, clear

- Delete variables

  To delete ALL variables:

  >> clear

  To delete only some variables:

  >> clear pepito juanito

  Only deletes variables `pepito` and `juanito`

- Save and load variables

  To save ALL variables to a file called nombrearchivo.mat:

  >> save nombrearchivo

  To save only some variables:

  >> save nombrearchivo pepito juanito

  To load the variables stored in file nombrearchivo.mat:

  >> load nombrearchivo

The first program of any programmer

```
>> disp(‘¡Hola, mundo!’)
```

Scripts

>> edit name_of_program

- Any instruction in the script will be executed as if it had been typed in the Command Window.
- Comments: Matlab ignores anything that follows the symbol %.

  Example:

```
b=7; % Matlab will assign the value 7 to variable b, and will not read this sentence.
```

Execution of scripts

- Once it has been saved, the script is executed by typing its name in the Command Window.

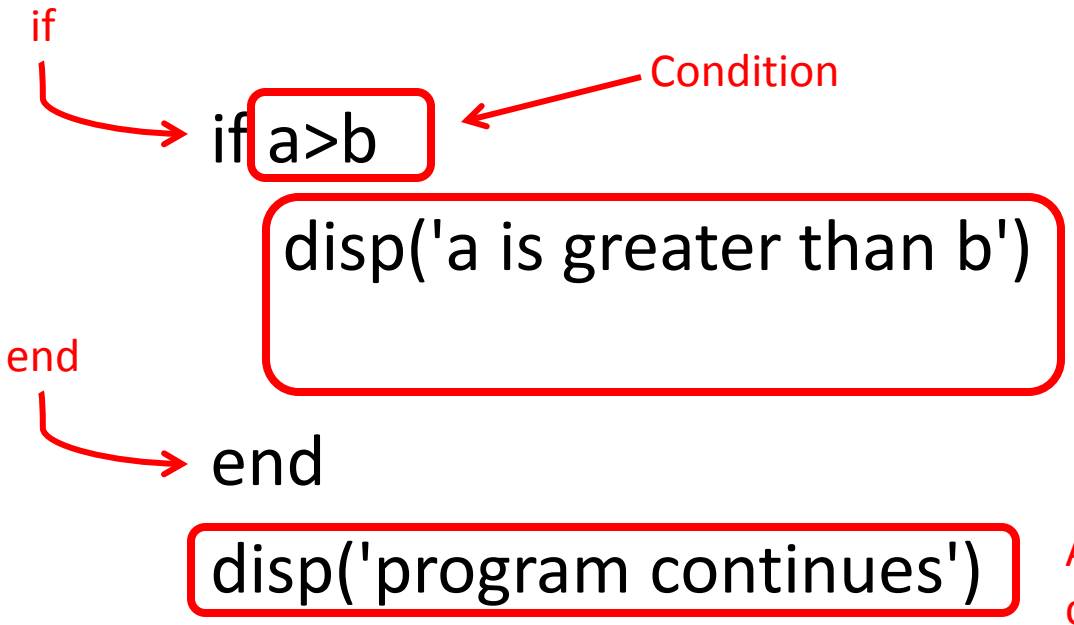
BUT NOTE:

- Matlab will only look for programs in a few directories. If your script is not in one of those directories, Matlab will not be able to find it. These folders are:
  - Current directory: Indicated at the top of Matlab's window.
  - Matlab's path: It is a list of directories where Matlab always looks for programs. You can **add a new directory to the path** in the menu File→Set path…

Matlab's help...

- ...is wonderful.
- As well as error messages (although we all hate them).

if

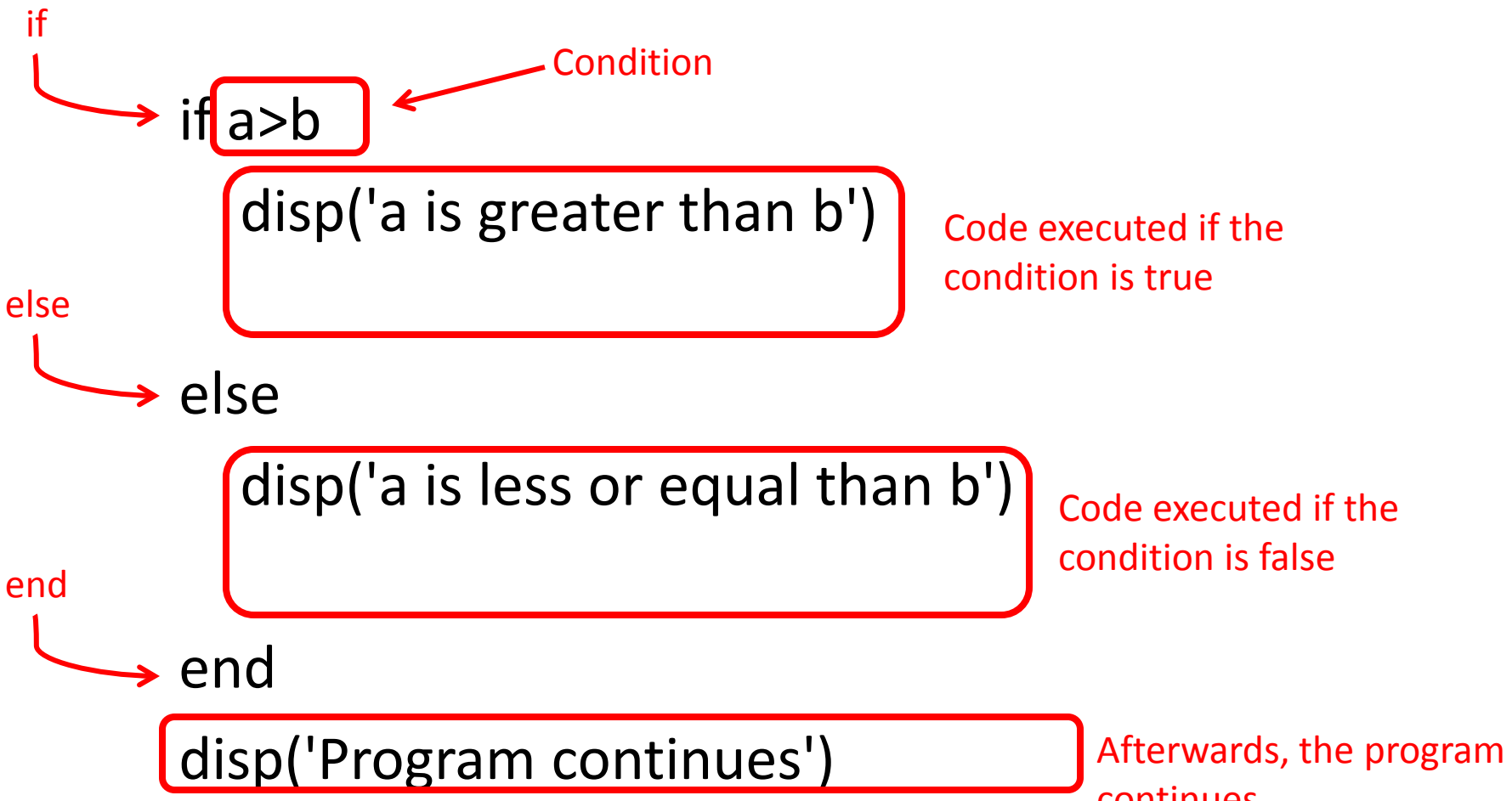


Main types of conditions:

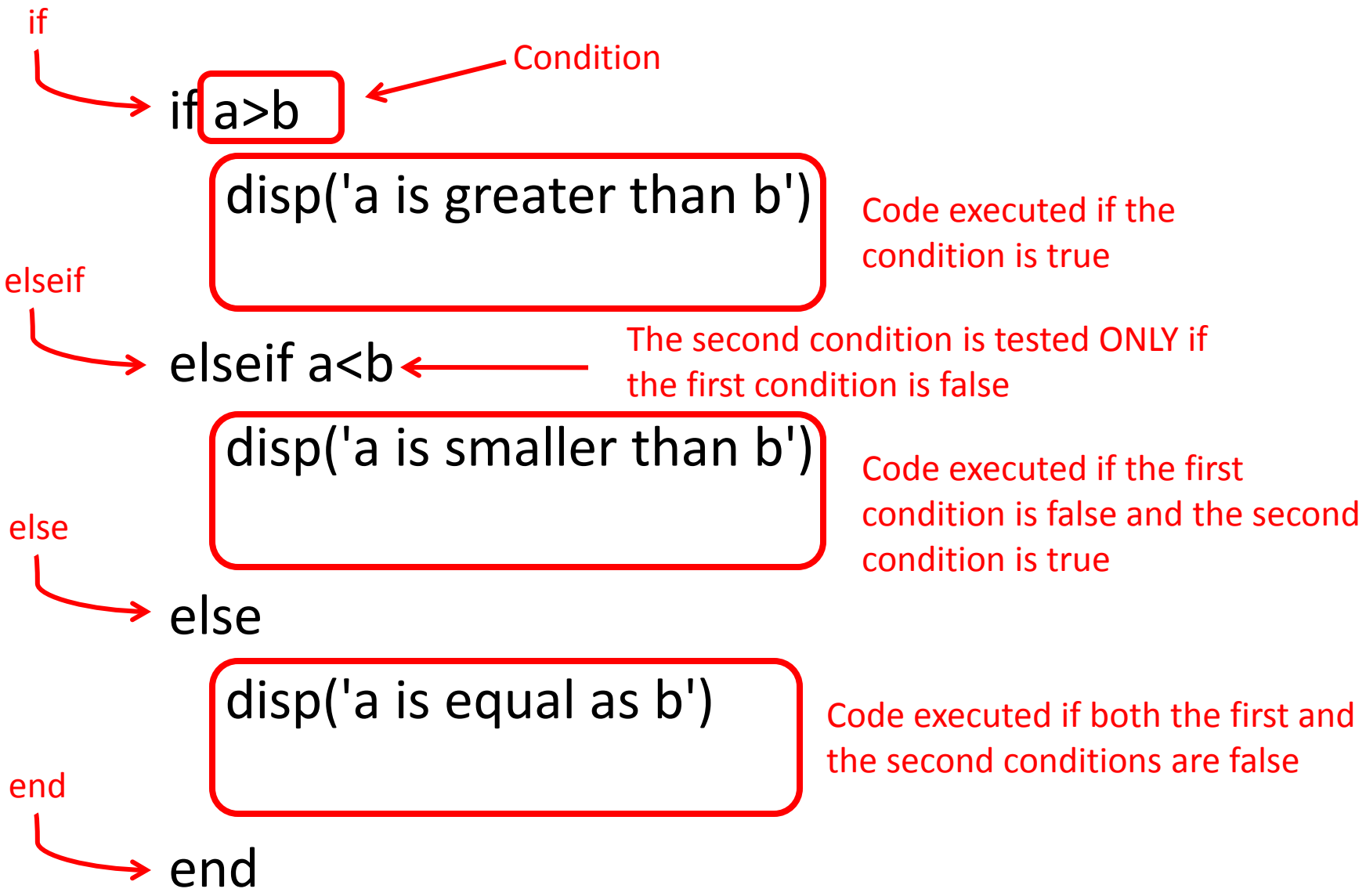
- a==b → a equal to b (**with two equal signs**)
- a~=b → a different than b (**symbol ~ is typed with AltGr+4**)
- a>=b → a greater or equal than b
- a<=b → a less or equal than b
- a>b → a greater than b
- a<b → a less than b

The comparison may involve variables or numbers (for example, `a==1`)

if...else


if
Condition
if a>b
disp('a is greater than b')
Code executed if the condition is true
else
else
disp('a is less or equal than b')
Code executed if the condition is false
end
end
disp('Program continues')
Afterwards, the program continues

if...elseif...else



The `for` loop

Loop: Set of instructions that are repeated several times.