SESSION 1

1. **FUNCTIONS DEFINED IN MATHEMATICA**
2. **USER-DEFINED FUNCTIONS**
3. **THE Plot FUNCTION**
4. **THE INSTRUCTION Limit**
5. **INTERPOLATION**

- **PROGRAM ELEMENTS**

1. Functions defined in MATHEMATICA

The main functions of the analysis are implemented in he language of MATHEMATICA. The following list provides a summary of these functions:

1. `Sin[x]` provides the sine of the angle `x`.
2. `Cos [x]` provides he cosine of the angle `x`.
3. `Tan[x]` provides the tangent of the angle `x`.
4. `Sec [x]` provides the secant of the angle `x`.
5. `Csc [x]` provides the cosecant of the angle `x`.
6. `ArcCos [x]` provides the arccosine of `x`.
7. `ArcSin [x]` provides the arcsine of `x`.
8. `ArcTan [x]` provides the arctangent of `x`.
9. `Sqrt [x]` provides the square root of `x`.
10. `Log[b,x]` provides he logarithm in base `b` of `x`.
11. `Log[x]` provides he logarithm natural, that is, with base the number e.

The following points should be taken into account:

- In the trigonometric functions, it is always supposed that the angle used as argument is expressed in radians.
- In order to maintain accuracy, MATHEMATICA will not evaluate a function that uses exact data. For example,

  ```
  In[ 1]:= Sin[1]
  Out [ 1]= Sin[1]
  ```

  In this example since the number `1` is an exact data, MATHEMATICA also tries to obtain an exact result for `Sin[1]` which is not possible since sin(1) is a number with infinite decimal places, so the program ultimately chooses to leave the function unevaluated. This problem can be easily solved using the approximation function `N`:

  ```
  In[ 2]:= N[Sin[1]]
  Out [ 2]= 0.841471
  ```

- The symbols used by MATHEMATICA for he number e are `E` and ⅇ (this last can be entered using the palette or by typing escape ee exhaust).
- The symbols used by MATHEMATICA for he number π are `Pi` or π (this last character can introduce using the pallette either typing escape pi escape).
- The mathematical symbol ∞ is represented in MATHEMATICA through the word `Infinity` or he own symbol ∞ (that we can insert from the pallette or by typing the combination escape inf escape).
- When we evaluate an expression in which divisions by 0 appear or that has an infinite limit, MATHEMATICA will display:
  a) ∞ or - ∞ if the function has at that point limit $+\infty$ or $-\infty$.
  b) `ComplexInfinity` if at that point one of the lateral limits is $+\infty$ and the other is $-\infty$.

2. User-defined functions

MATHEMATICA provides us with numerous functions implemented in its language, but also offers us the necessary tools so that we can build new functions adapted to our needs.

Let us see how to define a function:

Comando: Declaration of a function
Sintaxis:

`namefuncion[x_]:=expression`

Resultado: Declare a function with name `namefunction` with variable `x` defined by the formula `expresion`.

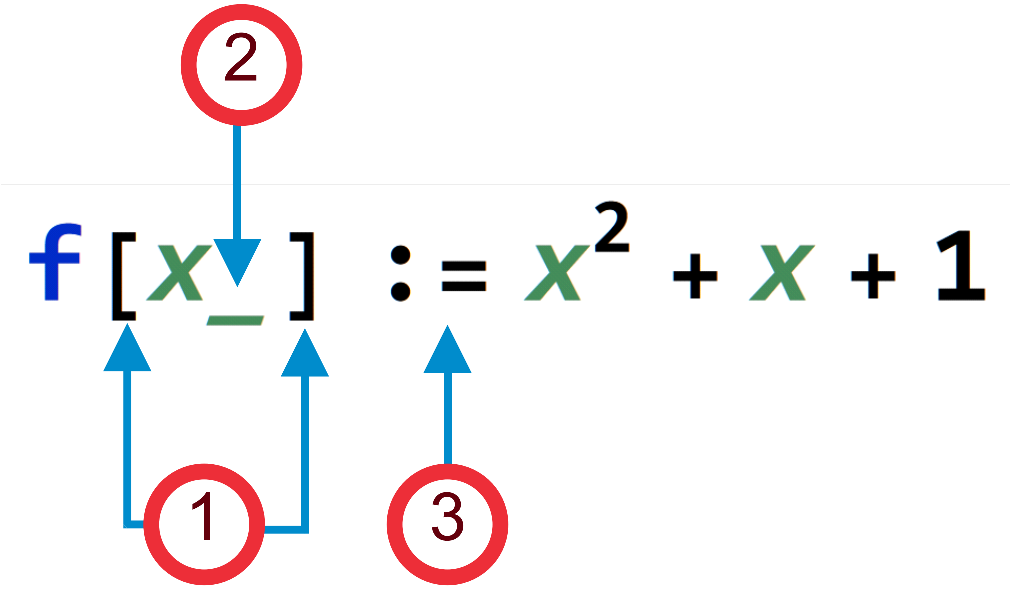
The definition or declaration of a function includes several elements that are not common in classical notation and therefore require special care. These are:

1. **Brackets:** Function variables are always enclosed in brackets, never in parentheses. This is how we write `f[x]` instead of `f(x)` .
2. **The character _ :** In the definition of a function we must indicate which are the variables using the character " _ " (called blank) which is obtained by pressing the key on which the " – " sign appears together with the SHIFT key. It should be noted that we must include the character "_" only in the definition and not when we subsequently use the function.
3. **The symbol `:=` :** The " =" symbol is used to assign values to a variable. For function definitions, we will instead use the " :=" symbol, called the "deferred assignment operator". Its purpose is similar to the " =" assignment operator that we already know and associates the instructions in the definition body with the symbol or name of the function that appears in the header of the definition.

Let's see some examples simple of definition of functions:

Example 1:

To define the function f(x) = $x^2 + x + 1$,



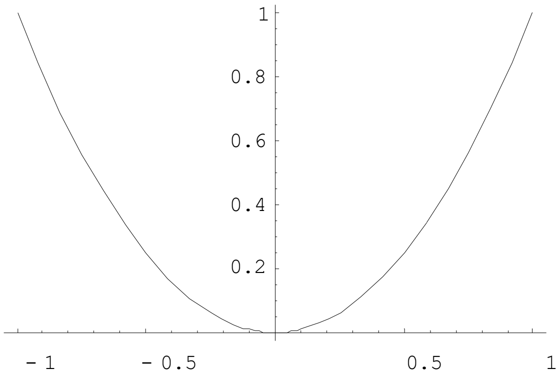
Example 2:

```
In[1 ]:= f[x_]:= Cos [x]-x^2
In[ 2]:= f[0]
Out[2]= 1
```

Let us define the function:

$$f(x) = \cos(x) - x^2$$

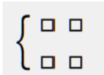
In[4]:= `Plot[f[x],{x,- 1,1}];`



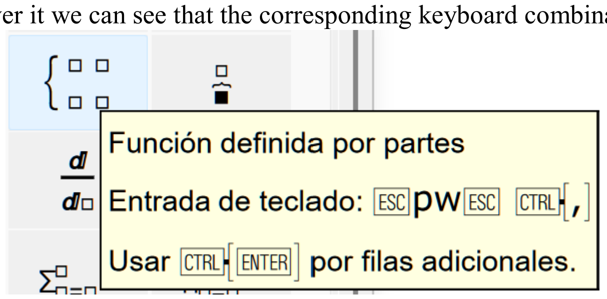
Later on we can use `f[x]` defined before and we no longer need to write its formula again.

2.1. *Piecewise functions*

To define piecewise functions, in the "Class Wizard" palette, in the "Typesetting" drop-down submenu at the bottom of the palette, click the button:



If we place the mouse over it we can see that the corresponding keyboard combination is



That is, we'll press escape pw escape to get the { symbol, and then CTRL + , (CTRL along with the comma) to insert the boxes that will contain the function information. Likewise, as we see in the yellow dialog box, CTRL + ENTER will add new lines inside the curly braces.

Example 3:

To define the function

$$f(x)=\begin{cases} x\cos(x), & x<0 \\ x+e^{x}, & x\geq 0\end{cases}$$

We will start by typing the header of the definition, as we have seen before in the form `f[x_ ]:=` then escape pw escape for the symbol { and CTRL +, for the boxes; so that it will be displayed

$$\texttt{f[x\_] :=} \begin{cases} \square & \square \\ \square & \square \end{cases}$$

Now we will complete the corresponding information in each box (without inserting the comma symbols ",")

$$\texttt{f[x\_] :=} \begin{cases} \texttt{x Cos[x]} & \texttt{x} < 0 \\ \texttt{x} + e^{\texttt{x}} & \texttt{x} \geq 0 \end{cases}$$

If the function we want to define had more lines corresponding to more pieces, we would simply press CTRL + ENTER until we obtained the necessary number of lines.

3. The `Plot` function

By means of the graphic representation of a function we can study in a simple manner their features by detecting at a glance the points at which the function is not defined and its behavior there. One of the instructions we can use in MATHEMATICA to obtain graphical representations is the `Plot instruction` .

Command: `Plot`

Syntax:

1. `Plot[f[x],{x,xi,xf}]`
2. `Plot[{f1[x],f2[x],...,fn[x]},{x,xi,xf}]`
3. `Plot[f[x],{x,xi,xf}, PlotStyle->{style1,…,styleok}`

where `f[x]` , `f1[x]` ,..., `fn[x]` are expressions or functions in the variable `x`.

Result:

1) (first format) Provides the graph of the function `f[x]` for the values of `x` ranging from the initial value `xi` up to the final value `xf`, that is, the graph of the function in the interval of the line [`xi`, `xf`].

2) (second format) Simultaneously provides the graph of the functions `f1[x]` , `f2[x]`,..., `fn[x]` for `x` in [`xi`, `xf`].

3) (third format) `f[x] is rendered` , applying the indicated styles. Some options are

- `RGBColor[red,green,blue]` . Determines the color with which the graph will be painted. of the function, being `red, green, blue` three numbers between 0 and 1 that indicate the mixture of the colors red, green, and blue that will give rise to the resulting color.
- `Dashing[{log1,long2,longk}]` . Allows you to plot the graph of the function using dashed lines, each of length `log1,long2,longk` .

4. The `Limit` instruction

With `Limit` we can compute the limit of a wide range of functions.

Comando: `Limit`
Sintax:

1. `Limit[f[x],x->x0]`
2. `Limit[f[x],x->x0,Direction->1] Limit[f[x],x->x0, Direction->-1]`

where `f[x]` is a function or expression for variable `x`.

Result:

1) (first format) `Limit[f[x],x->x0]` = $\lim_{x\to x_0} f(x)$

2) (second format) `Limit[f[x],x->x0,Direction->1]` = $\lim_{x\to x_0^-} f(x)$

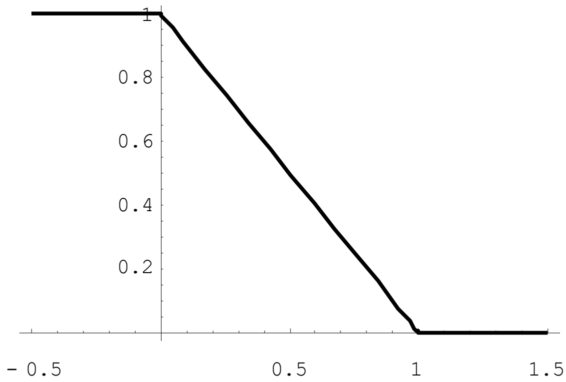
`Limit[f[x],x->x0,Direction->-1]` = $\lim_{x\to x_0^+} f(x)$

Example 4:

```
In[1]:= f[x_]:= x ² e ˣ ;
In[2]:= Limit[f[x],x-> ∞]
Out[1]= 0
In[2]:= Limit[1/x,x->0,Direction->-1]
Out[2]= + ∞
```

4.1. Exercises

1. Build through two different methods a piecewise function defined in two pieces with the following graph:



2. Consider the function $f(x) = \begin{cases} \frac{x^5-1}{x^2+1}, & x < -1, \\ \cos(x), & -1 \le x < 0, \\ e^x, & 0 \le x < 1, \\ x^{\frac{1}{x-1}}, & 1 < x \end{cases}$

and determine by means of the graph of f(x) in different intervals, the points of discontinuity, limits at the points of change of definition and at $\pm\infty$. Check, using `Limit`, that the result obtained form the graph are correct.