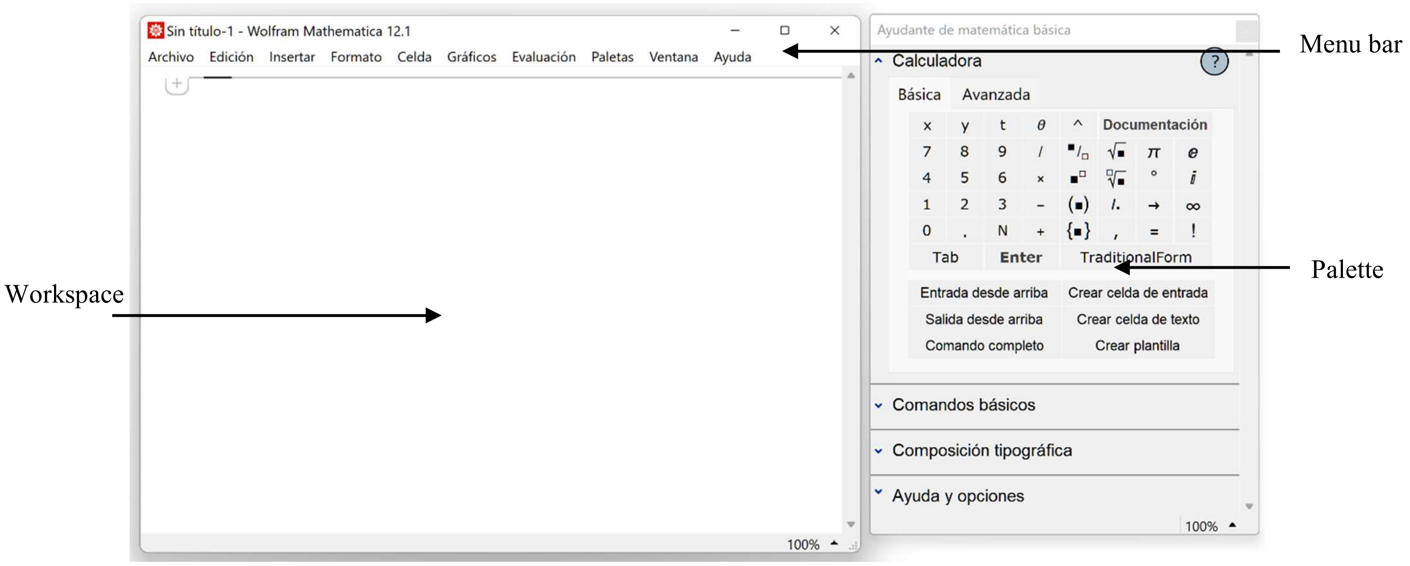
SESSION 0

1. BASIC WORK METHOD
2. BASIC OPERATIONS AND PALLET HANDLING
3. TYPES OF NUMBERS AND PRECISION
4. FUNCTIONS
5. VARIABLES, SYMBOLS AND VALUE ASSIGNMENT
6. EQUATIONS

- PROGRAM ELEMENTS

1. BASIC WORKING METHOD

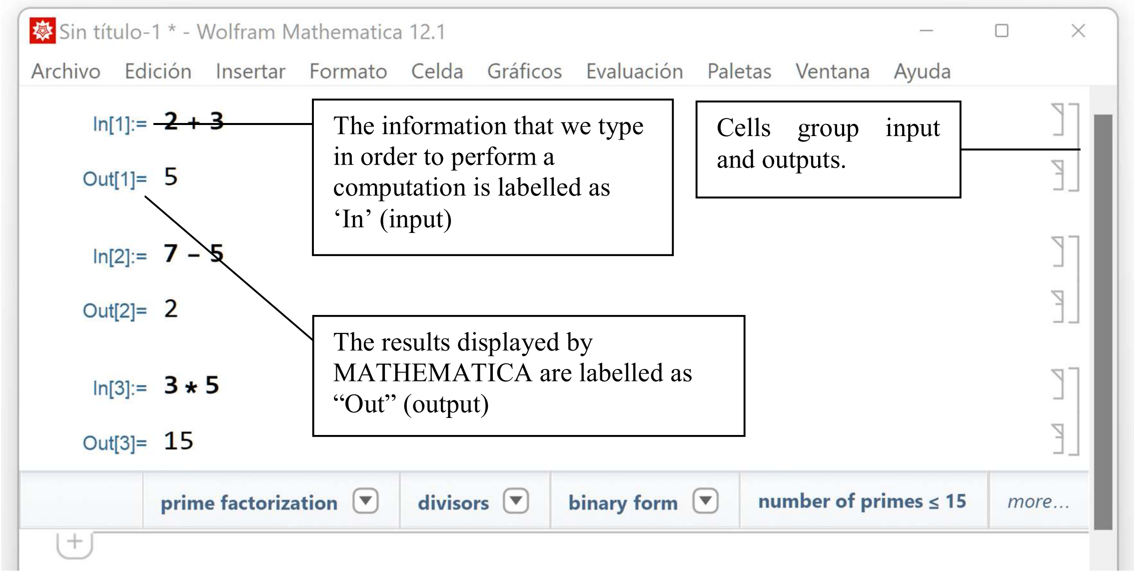
The Mathematica environment incorporates the following elements:



We insert expressions or instructions in the work area and execute them using the keys ⇧ + ⏎ (shift+enter). Once this is done, the following happens:

a) The program labels the expression to be evaluated with "In" and the result obtained with "Out".
b) The program will assign the same order number to the input and its corresponding output.
c) MATHEMATICA groups information into cells.

***Example* 1:**

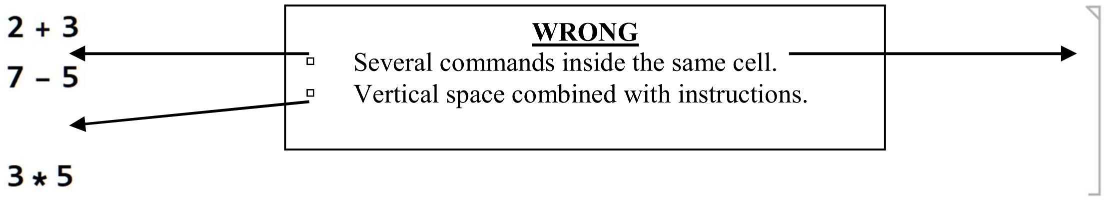


As we run calculations and receive results, MATHEMATICA organizes the input and output information into different cells. The following points should be kept in mind when managing cells:

1. If we write an instruction in the workspace and don't execute it by pressing ⇧ + ⏎ , the instruction will be ignored. The computer will behave as if we hadn't written anything. We'll know that an instruction has been evaluated when the corresponding input label ('In[...]:=') appears next to it and the instruction is enclosed in a cell next to the answer given by MATHEMATICA.

2. In principle, only one instruction should be entered in each cell. To enter multiple instructions, see the instructions at the end of this exercise.
3. ⏎ key alone only has the effect of introducing vertical space, which is ignored by MATHEMATICA when performing calculations. It should never be used to separate multiple statements in the same cell.
4. If vertical space is required to make writing clearer, this should be done in a separate cell where no instructions appear.

Example 2:



5. To move from one cell to another we can use the mouse or the arrow keys.
6. We can place the cursor in a cell that has already been evaluated, correct it, and then re-evaluate it. This is useful when we need to repeat the same calculation for different data.

MATHEMATICA allows you to save calculations performed during a work session to disk using the usual Windows "Save" or "Save as" procedures.

2. BASIC OPERATIONS AND PALLET HANDLING

The basic operations we can use in MATHEMATICA are the usual ones we would find on any calculator, or those that generally appear when performing calculations. These are:

Operation	Symbol
Addition	+
Subtraction	-
Product	* or space
Division	/
Power	^

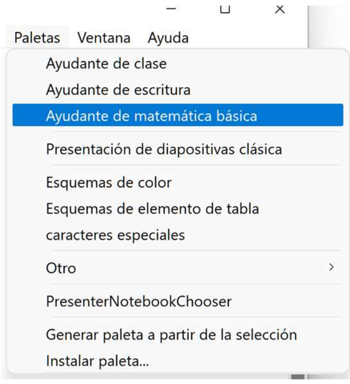
Example 3:

```
In[1]:= 3+2
Out[1]= 5
In[2]:= 5-3
Out[2]= 2
In[3]:= 5*3
Out[3]= 15
In[4]:= 5 3
Out[4]= 15
In[5]:= 9/3
Out[5]= 3
In[6]:= 5^2
Out[6]= 25
```

We can type the product 5 times 3 both using the asterisk symbol (*) and also leaving an space between 5 and 3. Also with the symbol × that we find in the palette.

To achieve a notation closer to the usual one, Mathematica offers, apart from these symbols of its own internal language, three writing methods:

- **Method 1: Mathematica's internal language.** The method we just saw uses the indicated symbols, even though this notation is not standard in mathematics.
- **Method 2: Using palettes.** MATHEMATICA offers several palettes that allow you to enter powers, fractions, roots, integrals, derivatives, letters of the Greek alphabet, and many other special symbols using a notation similar to classical notation. You can access the palettes by selecting "Palettes" from the menu bar and selecting one of the available palettes.
- **Method 3: Keyboard shortcut.** Use keyboard shortcuts to enter special palette notations and symbols. Hovering over the palette buttons will display a small box with the corresponding shortcut.



For basic operations these three methods are applied as follows (for addition, subtraction and product there is no difference between the three methods):

Operation	Internal language	Pallette	Palette-Keyboard Method
Division	/	$\frac{\blacksquare}{\square}$	ctrl + ⇧ + 7
Power	^	$\blacksquare^{\square}$	▪ write the base ▪ ctrl +6 ▪ write the exponent
Number e	E	ⅇ	escape ee escape
π	Pi	π	escape pi escape
Number i	I	ⅈ	escape ii escape

Example 4:

For example, to write the fraction $\frac{10}{3}$ we proceed as follows:

- We press the corresponding button on the palette ($\frac{\blacksquare}{\square}$) or type ctrl + ⇧ + 7. After doing so, the fraction symbol will appear on the screen

$$\frac{\blacksquare}{\square}$$

- We have to complete the place holders above and below the fraction symbol that has appeared by placing a 10 in the top box and a 3 in the bottom. The small black square that appears inside the top box tells us where the next thing we write will appear.
- We type 10 in the upper box to obtain

$$\frac{\mathbf{10}}{\square}$$

- To move to the lower box, we can press the Tab key or point to the lower box with the mouse and then press the left mouse button. The small black box will now appear in the lower box:

$$\frac{\mathbf{10}}{\blacksquare}$$

This means that we can now type 3 to finally obtain the desired fraction.

2.1. Parentheses and priority of operations

There may be cases where the notation is confusing and we don't know the order in which the operations will be performed. For example, the expression:

$$3+1\cdot 5$$

could be understood as $3+(1\cdot 5)=8$ or as $(3+1)\cdot 5=20$. To resolve this type of conflict, there is something called operation priority. The operations we have seen are arranged in order of priority from highest to lowest, such that when a calculation is analyzed, the operations with the highest priority are performed first. The order of priority from highest to lowest is as follows:

- highest priority: power ^.

- intermediate priority: product and division (* and /).
- low priority: sum and difference (+ and -).

When two operations have the same priority (for example + and -) they are performed from right to left.

Thus, in the expression, $3+1\cdot 5$ the operation with the highest priority, the product, will be performed first, and then the one with the lowest priority, +, so the result will be $3+(1\cdot 5)=8$ and not 20.

The order of execution of operations can be altered by using the parentheses "(" and ")" in the usual manner.

3. TYPES OF NUMBERS AND PRECISION

MATHEMATICA allows you to handle two groups of numbers:

- **Approximate:** Any number that includes the decimal symbol "." is considered approximate.
- **Exact:** Any number or numerical expression that does not include the decimal symbol is considered exact.

When evaluating an expression, the program follows the following rules:

1) If all the data involved in the expression are exact, the result offered must be exact or else it is not evaluated.
2) If any of the data involved in the expression is approximate, the result is evaluated and returned as approximate.

***Example** 5:*

In[1]:= $\frac{2}{3}$

Out[1]= $\frac{2}{3}$

In[2]:= $\sqrt{2}$

Out[2]= $\sqrt{2}$

In[4]:= π^3

Out[4]= π^3

In these computations we find 2, 3, π. Since no decimal symbol "." Appears, they are considered exact numbers. Whenever the imput data is exact, Mathematica will try to offer also exact output. However, $\frac{2}{3}$, $\sqrt{2}$ ó π^3 are computations that yields as a result irrational numbers with infinite decimal places and, therefore, it is impossible to write their complete decimal expression in an exact way. As it is not possible to offer an exact result, MATHEMATICA chooses to leave the calculations unperformed.

In[5]:= $\frac{2.}{3}$

Out[5]= 0.66666

In[6]:= $\sqrt{2.}$

Out[6]= 1.41421

`2.` is an approximate number (symbol "." Indicates that it is approximate even though it has no decimal places). As the input contains approximate numbers, MATHEMATICA is free to provide an approximate result. In this case the computations are evaluated as approximate decimal expressions.

You can force the program to evaluate an exact expression by providing an approximate value. This is achieved using the `N command` .

Command: Command N for approximation of exact numbers

Syntax:

Format 1: `N[a]`
Format 2: `N[a,n]`

Result:

Format 1: Gives an approximation of `a` with 16 digits of precision.

Format 2: Gives an approximation of `a` with `n` digits of precision.

Note that n digits of precision is not the same as n decimal numbers, for example:

3123 is a number with 4 digits of precision.
3123.4 is a number with 5 digits of precision and one decimal place.
3123.000000 is a number with 10 digits of precision and 6 decimal digits.
3.1415 is a number with 5 digits of precision and 4 decimal digits.

Example 6:

In[1]:= **N[π]**
Out[1]= 3.14159
In[2]:= **N[π,30]**
Out[2]= 3.14159265358979323846264338328

MATHEMATICA performs all calculations with a minimum precision of 16 digits. However, to prevent the screen from becoming cluttered, it only displays the first 5 decimal places (see the result given by **N[π]**). To request more decimal places, we must indicate this when using the `N` command (as in **N[π,30]**).

Later we will see that we can apply command `N` not only to numbers but also to expressions of any type.

4. FUNCTIONS

The concept of a function is common in mathematics, and we are therefore accustomed to using expressions such as sin(3), tan(π), arccos(1/2), etc. MATHEMATICA incorporates almost all common mathematical functions. Below we list some of them:

- `Sin` = sine of an angle in radians.
- `Cos` = cosine of an angle in radians.
- `Tan` = tangent of an angle in radians.
- `Sqrt` = square root of a real or complex number.
- `ArcTan` = arc whose tangent.
- `ArcSin` = arc whose sine.
- `ArcCos` = arc whose cosine.
- `Log` = natural logarithm.

Each of them is used in the same way as we would in normal mathematical language, with the exception that in MATHEMATICA the arguments of a function must be enclosed in brackets and not in parentheses, that is:

- To calculate the sine of 3 we will put `Sin[3]` instead of sen (3).
- To calculate the arc whose tangent is 1 we will put `ArcTan[1]` instead of arctan(1).
- To calculate the square root of 2 we will put `Sqrt[2]` .

For writing the square root we also have the three usual alternative mechanisms:

Internal language	Pallette	Palette-Keyboard Method
`Sqrt`	$\sqrt{\blacksquare}$	ctrl + 2

Example 7:

In[1]:= **Sin[π]**
Out[1]= 0
In[2]:= **ArcTan[1]**
Out[2]= $\frac{\pi}{4}$
In[3]:= **Cos[1]**

En la expresión Cos[1] solamente intervienen datos exactos (en este caso el único dato es el número 1 que es un dato exacto) y el ordenador intentará dar un resultado exacto. Sin embargo cos(1) es un número irracional con infinitas cifras decimales así que es imposible escribir de forma exacta su valor. Por tanto el ordenador deja la expresión sin evaluar.

Out[3]= Cos[1]
In[4]:= **N[Cos[1]]**
Out[4]= 0.540302

5. VARIABLES, SYMBOLS AND VALUE ASSIGNMENT

In various scientific disciplines, it is common to use variables to which a value can then be assigned. For example, to calculate the area of a triangle, we use the formula

$$\text{area} = \frac{\text{base} \cdot \text{altura}}{2}$$

in which the variables 'area', 'base', and 'height' appear. When we are faced with a specific triangle, we will assign the corresponding values to the variables 'base' and 'height', and we can then calculate the area by applying the formula. For example:

$$\begin{matrix} \text{base} = 4 \\ \text{altura} = 5 \end{matrix} \Rightarrow \text{area} = \frac{4 \cdot 5}{2} = 10\,.$$

When studying data types, we saw that with MATHEMATICA we can also handle variables or symbols. A symbol or variable is any grouping of characters or numbers that begins with a lowercase letter. Examples of possible symbols are: `x` , `z` , `hello` , `z1` , `hello318x` , `height` , etc.
It should be noted that:

- A blank space cannot be inserted into a symbol name (a blank space is considered by the program as a multiplication).
- MATHEMATICA is case sensitive.

Example 8:

`X1`	Correct symbol.
`3a`	Incorrect symbol because it does not begin with a letter. `3a` will be interpreted as the product of `3` and `a` .
`x 1`	Incorrect because it includes a space. The program will interpret it as the product of `x` and `1` .
`solution equation`	correct symbol.
`equation solution`	Incorrect symbol due to a space. It will be interpreted as the product of the `solution symbol and the equation` symbol .
`firstvalue` ≠ `FirstValue`	We see an example of two symbols that use the same letters but are different because the upper and lower case letters do not match.

A symbol can be assigned a value that can be a number, a list, another symbol, or generally anything.

We have then, that a variable or symbol can be found in two states:

a) Without having an assigned value : In this situation the variable is considered as an exact data type and is never evaluated.
b) Having a value assigned : Then in every expression or command in which such a symbol appears, it is immediately substituted by its value, even before the expression is evaluated.

To assign a value to a variable we use the assignment symbol '=' in the following way:

Command: Immediate assignment operator =

Syntax:

```
symbolname = valueexpression
```

Result: Evaluates the expression `valueExpression` and assigns it to the symbol `symbolName`, such that each subsequent occurrence of the symbol `symbolName` is replaced by `valueExpression`. The assignment results in the value `valueExpression`.

Example 9:

In[1]:= **a=5**
Out[1]= 5
In[2]:= **a^2+1**
Out[2]= 26
In[3]:= **b=$\sqrt{2}+1$**
Out[3]= $1+\sqrt{2}$
In[4]:= $\frac{a}{b}$
Out[4]= $\frac{5}{1+\sqrt{2}}$
In[5]:= **p=x^3+6x^2+11x+6**
Out[5]= $6+11x+6x^2+x^3$
In[6]:= $\frac{p}{x+3}$
Out[6]= $\frac{6+11x+6x^2+x^3}{3+x}$
In[7]:= **x=3**
Out[7]= 3
In[8]:= **p**
Out[8]= 120
In[9]:= **x=-4**
Out[9]= -4
In[10]:= **p**
Out[10]= -6

Once we execute this instrucion, symbol "a" will be substitud by 5 whenever it appears in any expression.

It can be checked that symbol "a" is replaced with 5 at every point after we did the assignment.

In a variable we can store expressions that contain other variables. In this case, in variable the `p` we have stored a polynomial in the variable `x`.

If we modify the variable `x` the value of `p` will also vary.

We can modify the value assigned to a variable by making a new assignment. The new value will be reflected in future operations. Note that now, in the polynomial `p`, `x` is replaced by –4, giving a different result.

Deleting values assigned to a symbol can be done using the "`=.`" command.

Command: Command `=.` to delete the values of a variable.
Syntax:

```
symbol=.
```

Result: Removes the value assigned to the variable `symbol` so that such a variable is left without any assigned value.

Example 10:

In[1]:= **a1=1;a2=2;a3=3;b=2;c=b;**
In[2]:= **a3**
Out[2]= 3
In[3]:= **a3=.**
In[4]:= **a3**
Out[4]= a3
In[5]:= **a1**
Out[5]= 1
In[6]:= **a2**
Out[6]= 2

The value of the variable `a3` was `3` but after this instruction it will lose its value.

5.1. *Handling symbolic expressions*

As seen in the previous example, we can handle expressions that include variables or symbols. These expressions are called symbolic expressions. For these cases, MATHEMATICA provides instructions that allow you to simplify or perform various calculations.

Command: `Simplify`

Syntax:

```
Simplify[expression]
```

Result: Simplify, if possible, the expression enclosed in brackets.

Command: `Expand`

Syntax:

```
Expand[expression]
```

Result: Develop the pending calculations that appear in the expression enclosed in brackets.

Example 11:

In[1]:= **p=y³ +6y² +11y+6**
Out[1]= 6+11y+6y² +y³

In[2]:= $\frac{p}{y+3}$

Out[2]= $\frac{6+11y+6y^2+y^3}{3+y}$

In[3]:= **Simplify[** $\frac{p}{y+3}$ **]**

Out[3]= $2+3y+y^2$
In[4]:= **Expand[(y+z)²]**
Out[4]= $y^2+2yz+z^2$
In[5]:= **Expand[(y+z)³]**
Out[5]= $y^3+3y^2z+3yz^2+z^3$
In[6]:= **Simplify[%]**
Out[6]= $(y+z)^3$

6. EQUATIONS

In Mathematica, equations are written using the double "==" symbol. The single equal sign "=" is reserved for assignments and can never be used to formulate an equation.

Example 12:

In[1]:= **x²-3x+2==0**

This is the correct notation for an equation with the doublé symbol **==**.

In[2]:= **x²-3x+2=0**

… Set: Tag Plus in 2-3 x+x^2 is Protected.

Here the equation is not correct since we use the simple symbol, **=**. We get an error message.

To solve equations we use the Solve command.

Command: `Solve`
Syntax:

1) One-variable equation: `Solve[equation,x]`

2) System of equations:
`Solve[{equation 1 ,...,equation m },{x 1 ,...,x n }]`

Result: **1)** The equation `equation` is solved for the variable `x`.
2) The system formed by the equations $equation_1$,..., $equation_m$ is solved for the variables x_1 ,..., x_n.

It is important to note that in MATHEMATICA the comparison operator that must appear in an equation is == (double =) never = (single =) which is the assignment operator.

For example:

In[1]:= **Solve[ax² +bx+c==0,x]**

Out[1]= $\{\{x \to \frac{-b-\sqrt{b^2-4ac}}{2a}\},\{x \to \frac{-b+\sqrt{b^2-4ac}}{2a}\}\}$

In[2]:= **Solve[{α+2β==4,2α+β==5,-α+β==-1},{α,β}]**

Out[2]= {{α→2, β→1}}

We solve the equation of degree two to obtain the well known formulas for the solution.

It is very important to use in the equations the double symbol (**==**) instead of the simple one (=).

With the last instruction we solve the linear system

$$\begin{cases}\alpha+2\beta=4\\2\alpha+\beta=5\\-\alpha+\beta=-1\end{cases}$$

for the variables α y β .

PROGRAM ELEMENTS

- The % symbol is used to refer to a specific output. As we perform operations, we obtain a list of inputs (In) and outputs (Out), which are numbered in order. We may need to use a previously calculated result; using the % symbol allows us to avoid retyping that result. The syntax for % is:
  - % n = Out nº n.
  - % = Out immediately preceding.
  - %% = Out second to last.
  - %%% = Out second to last.
  - Etc.

Example 13*:*

In[1]:= **1+$\frac{3}{2}$**

Out[1]= $\frac{5}{2}$

In[2]:= **%+1**

Out[2]= $\frac{7}{2}$

In[3]:= **%%-1**

Out[3]= $\frac{3}{2}$

In[4]:= **(x+y)** ^{2}

Out[4]= $(x+y)^2$

In[5]:= $\mathbf{x}^3$

Out[5]= x^3

In[6]:= **Expand[%4+%5]**

In[6]:= $2xy+x^2+y^2+x^3$

- We can delete any of the cells that appear in the workspace. To do this, click on the selected cell to select it and then delete it with the Delete key.
- Every command or keyword that is part of or incorporated into the MATHEMATICA program is written with the first letter of each word in uppercase and the rest in lowercase. For example: `Pi`, `Sin`, `Sqrt`, `MatrixForm` (a compound word composed of `Matrix` and `Form`; note that each of the component words begins with a capital letter).
- When we make a mistake of any kind, the program warns us with a blue message.

Example 14*:*

In[1]:= $\frac{1}{0}$

… Power: Infinite expression 1/0 encountered.

Out[1]:= `ComplexInfinity`

- `?` instruction allows us to obtain information about the operation of the different commands that are part of the MATHEMATICA language. It is used in the form

  `?command`

  after which we will obtain the requested information.

Example 15:

In[1]:= **?Inverse**

```
Inverse[m] gives the inverse of a square matrix m.
```

In[2]:= **?Transpose**

```
Transpose[list] transposes the first two levels in list.
Transpose[list, {n1,n2, ...}] transposes list so that the nk-th
level in list is the k-th level in the result.
```

- In the same line and cell we can include more than one instruction as long as we separate them with "`;`"

  In[1]:= **a=3; b=2;**
  a+b

  Out[1]= 5

  A "`;`" character at the end of an instruction prevents the result produced by that operation from being displayed:

  In[2]:= **c=a+b**

  Out[2]= 3

  In[3]:= **c=a+b;**

- Remember that:

Parentheses `( )`	They are used to group algebraic operations. `(1+3)*2`
Single brackets `[ ]`	They are used to group the arguments of a function. `Cos[4]`
The simple equals **=**	It is used to assign a value to a variable: `x=2`
The double equals **==**	They are used to write equations. `x+1==3`