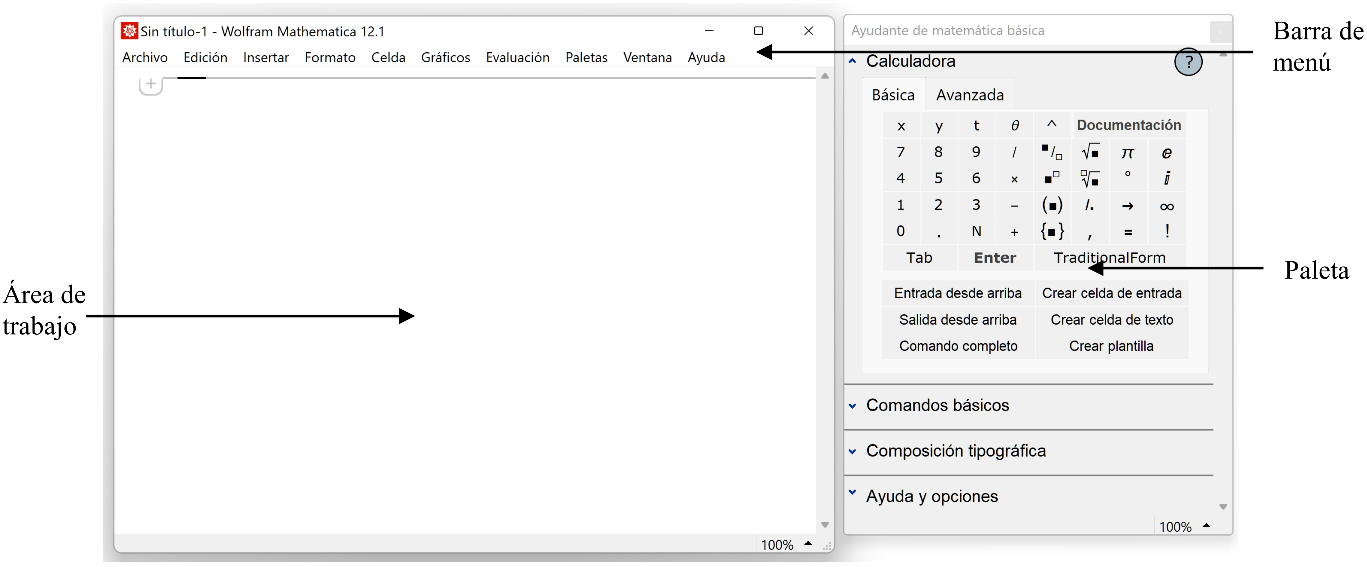
SESIÓN 0

1. METODO DE TRABAJO BASICO
2. OPERACIONES BASICAS Y MANEJO DE PALETAS
3. TIPOS DE NÚMEROS Y PRECISION
4. FUNCIONES
5. VARIABLES, SÍMBOLOS Y ASIGNACIÓN DE VALORES
6. ECUACIONES
- ELEMENTOS DEL PROGRAMA

1. MÉTODO DE TRABAJO BÁSICO

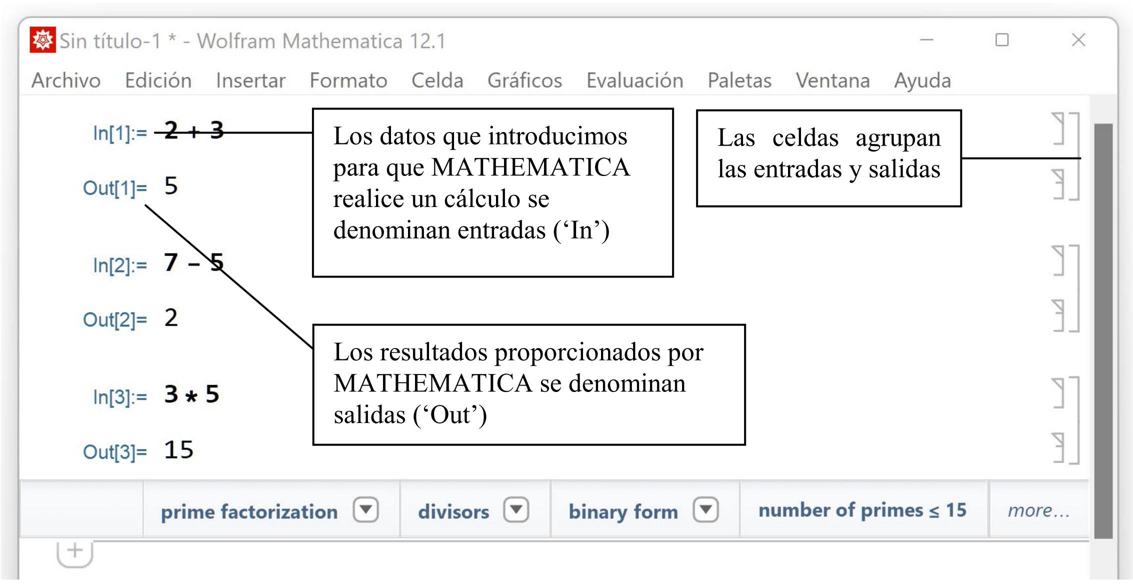
El entorno de Mathematica incorpora los siguientes elementos:



Insertamos expresiones o instrucciones en la zona de trabajo y la ejecutamos mediante las teclar ⎆ + ⇧ (shift+enter). Una vez realizado esto, sucede lo siguiente:

a) El programa etiqueta mediante "In" la expresión a evaluar y mediante "Out" al resultado obtenido.
b) El programa asignará un mismo número de orden a la entrada y a su salida correspondiente.
c) MATHEMATICA agrupa la información en celdas.

Ejemplo 1*:*



A medida que ejecutamos cálculos y recibimos resultados, MATHEMATICA va organizando la información de entrada y de salida en diferentes celdas. Para el manejo de celdas deben tenerse en cuenta los siguientes puntos:

1. Si escribimos una instrucción en el área de trabajo y no la ejecutamos pulsando ⎆ + ⇧, dicha instrucción será ignorada. El ordenador se comportará como si no hubiéramos escrito nada. Sabremos que una instrucción ha sido evaluada cuando aparezca junto a ella el rótulo de entrada

correspondiente (‘In[...]:=’) y la instrucción quede englobada en una celda junto a la respuesta dada por MATHEMATICA.

2. En principio, debe introducirse una sola instrucción en cada celda. Para introducir varias instrucciones véase lo indicado al final de la práctica.
3. La tecla ⟐ por si sola tiene como único efecto introducir espacio vertical que es ignorado por MATHEMATICA al realizar los cálculos. Nunca debe utilizarse para separar varias instrucciones en una misma celda.
4. Si para hacer más claro lo escrito es necesario introducir espacio vertical esto debe hacerse en una celda separada en la que no aparezca ninguna instrucción.

Ejemplo 2:

```
2 + 3
7 - 5

3 * 5
```

MAL

- Varias instrucciones en la misma celda.
- Espacio vertical mezclado con instrucciones

5. Para desplazarse de una celda a otra podemos utilizar el ratón o las teclas con flechas.
6. Podemos situar el cursor en una celda que ya ha sido evaluada, corregirla y volver a evaluarla. Esto es útil cuando necesitamos repetir el mismo cálculo para datos diferentes.

MATHEMATICA permite guardar en disco los cálculos realizados en una sesión de trabajo mediante los procedimientos habituales “Save”/”Guardar” o “Save as”/”Guardar como” de Windows.

2. OPERACIONES BASICAS Y MANEJO DE PALETAS

Las operaciones básicas que podemos emplear en MATHEMATICA son las usuales que encontraríamos en cualquier calculadora, o que aparecen generalmente a la hora de efectuar cálculos. Estas son:

suma: El símbolo correspondiente es “+” .
diferencia: El símbolo es “-”.
producto: El símbolo es un asterisco “*” o bien un espacio entre los factores.
división: El símbolo es “/”.
potenciación: El símbolo es “^”.

Todas ellas se emplean de la manera usual

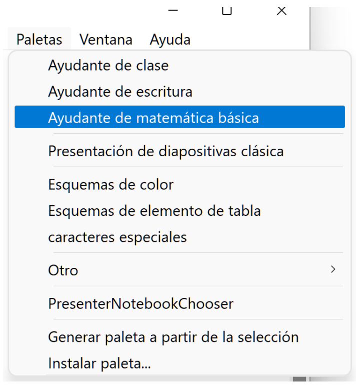
Ejemplo 3:

```
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
```

Podemos escribir el producto de 5 por 3 tanto utilizando el símbolo de asterisco (*) como dejando un espacio entre el 5 y el 3.

Si bien esta notación para las operaciones básicas (sobre todo para la división y potenciación) es habitual en todos los lenguajes de programación, no se asemeja a la que habitualmente empleamos cuando anotamos estas operaciones con lápiz y papel. Así por ejemplo, lo más usual sería escribir 5^2 en lugar de `5^2` o $\frac{9}{3}$ en lugar de `9/3`. Por este motivo MATHEMATICA incorpora métodos de escritura que permiten imitar la notación matemática clásica. Disponemos en realidad de tres formas diferentes de escribir la gran mayoría de expresiones matemáticas:

- **Método 1:** Utilizando el lenguaje interno de MATHEMATICA. Todas las expresiones se puede escribir utilizando los comandos o instrucciones del programa.
- **Método 2:** Utilizando la paleta de símbolos. En la parte derecha de la pantalla aparece la paleta de símbolos y notaciones especiales que permite introducir, con una notación similar a la clásica, potencias, quebrados, raíces, integrales, derivadas, letras del alfabeto griego y otros muchos símbolos especiales. En ocasiones, la paleta de símbolos no se halla presente. Para conseguir que aparezca de nuevo es suficiente con seleccionar en la barra de menú la opción 'Paletas' y dentro de ella seleccionar una de las paletas disponibles.
- **Método 3:** Utilizando métodos abreviados de teclado para introducir las notaciones y símbolos especiales de la paleta. Pasando el cursor sobre los botones de las paletas nos motrará un pequeño cuadro con el método abreviado correspondiente a ese botón.



En muchos casos indicaremos cómo realizar la escritura siguiendo cada uno de estos tres métodos. Comenzamos haciéndolo para las operaciones de división y potenciación (para suma, resta y producto no hay diferencia entre los tres métodos):

Operación	**Lenguaje interno**	**Paleta**	**Paleta-Método de teclado**
División	`/`	$\frac{\blacksquare}{\square}$	$\boxed{\text{ctrl}}$ + ⇧ + 7
Potenciación	`^`	$\blacksquare^{\square}$	▪ escribir la base ▪ $\boxed{\text{ctrl}}$ +6 ▪ escribir el exponente
e	`E`	e	$\boxed{\text{escape}}$ ee $\boxed{\text{escape}}$
π	`Pi`	π	$\boxed{\text{escape}}$ pi $\boxed{\text{escape}}$
i	`I`	i	$\boxed{\text{escape}}$ ii $\boxed{\text{escape}}$

Ejemplo 4:

Por ejemplo, para escribir el quebrado $\frac{10}{3}$ procedemos como sigue:

- Pulsamos el botón correspondiente de la paleta ($\frac{\blacksquare}{\square}$) o bien tecleamos $\boxed{\text{ctrl}}$ + ⇧ + 7. Tras hacerlo aparecerá en pantalla el símbolo de quebrado

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

- Hemos de completar los recuadros del símbolo de quebrado que ha aparecido situando un 10 en el recuadro superior y un 3 en el inferior. El cuadradito negro que aparece dentro del recuadro superior nos indica donde aparecerá lo próximo que escribamos.
- Tecleamos 10 en el recuadro superior para obtener

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

- Para pasar al recuadro inferior podemos pulsar la tecla de tabulación | o bién señalar con el ratón el recuadro inferior y pulsar luego el botón izquierdo. El recuadrito negro aparecerá ahora en el recuadro inferior:

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

Ello indica que ahora podemos teclear 3 para obtener finalmente el quebrado deseado.

2.1. Prioridad de operaciones

Pueden darse casos en los que la notación sea confusa y no sepamos en qué orden se efectuarán las operaciones. Así por ejemplo la expresión:

$$3+1\cdot 5$$

podría entenderse como $3+(1\cdot 5)=8$ o como $(3+1)\cdot 5=20$. Para dar solución a este tipo de conflictos existe lo que se llama prioridad de operaciones. Las operaciones que hemos visto están ordenadas según un orden de prioridades de mayor a menor prioridad de manera que cuando se analiza un cálculo se realizan primero las operaciones que tienen mayor prioridad. El orden de prioridades de mayor a menor es el siguiente:

- mayor prioridad: potenciación ^.
- prioridad intermedia: producto y división (* y /).
- prioridad baja: suma y diferencia (+ y -).

Cuando dos operaciones tienen la misma prioridad (por ejemplo + y - o también * y /) se efectúan de derecha a izquierda.

Así pues en la expresión $3+1\cdot 5$ se efectuará primero la operación de mayor prioridad, el producto, y luego la de menor +, con lo cual el resultado será $3+(1\cdot 5)=8$ y no 20.

El orden de ejecución de las operaciones se puede alterar mediante el uso de los paréntesis "(" y ")" en la manera habitual.

3. TIPOS DE NÚMEROS Y PRECISIÓN

MATHEMATICA permite manejar dos grupos de números:

- **Aproximados:** Cualquier número que incluya el símbolo decimal "." es considerado aproximado.
- **Exactos:** Cualquier número o expresión numérica que no incluya el símbolo decimal es considerado exacto.

A la hora de evaluar una expresión el programa sigue las siguientes reglas:

1) Si todos los datos que intervienen en la expresión son exactos el resultado ofrecido tiene que ser exacto o de lo contrario no se evalúa.
2) Si alguno de los datos que intervienen en la expresión es aproximado el resultado se evalúa y se devuelve como aproximado.

Ejemplo 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

En todos estos cálculos aparecen 2, 3, π. Puesto que no tienen punto decimal, son datos exactos. Si los datos de entrada son exactos el Mathematica intentará que los resultados proporcionados también lo sean. Sin embargo, $\frac{2}{3}$, $\sqrt{2}$ ó π^3 son operaciones que dan como resultado números con infinitas cifras decimales y por tanto es imposible escribir con total exactitud su expresión decimal. Puesto que no es posible dar un resultado exacto el ordenador opta por dejar los cálculos sin evaluar.

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

Out[5]= 0.66666

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

Out[6]= 1.41421

2. es un dato aproximado (el signo de puntuación indica que es un número aproximado independientemente de que tenga decimales o no). Si en los datos de entrada hay números aproximados, MATHEMATICA ya no intentará ofrecer como resultado un número exacto sino aproximado. En este caso los cálculos sí son evaluados.

Se puede forzar a que el programa evalúe una expresión que es exacta proporcionándonos un dato aproximado. Esto se consigue por medio del comando `N`.

Comando: Comando N para aproximación de números exactos

Sintaxis:

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

Resultado:

Formato 1: Da una aproximación de `a` con 16 dígitos de precisión.

Formato 2: Da una aproximación de `a` con `n` dígitos de precisión.

Téngase en cuenta que n dígitos de precisión no es lo mismo que n números decimales, por ejemplo:

3123	es un número con 4 dígitos de precisión.
3123.4	es un número con 5 dígitos de precisión y un decimal.
3123.000000	es un número con 10 dígitos de precisión y 6 dígitos decimales.
3.1415	es un número con 5 dígitos de precisión y 4 dígitos decimales.

Ejemplo 6*:*

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

MATEMÁTICA realiza todos los cálculos con una precisión mínima de 16 dígitos. Sin embargo, para evitar que la pantalla se llene de información, solo mostrará las 5 primeras cifras decimales (véase el resultado que ofrece **`N[π]`**). Para pedir que se muestren más cifras decimales debemos indicarlo al utilizar el comando `N` (como en **`N[π,30]`**).

Posteriormente veremos que podemos aplicar el comando `N` no solamente a número sino también a expresiones de cualquier tipo.

4. FUNCIONES

En matemáticas es común el concepto de función y, así pues, estamos acostumbrados a emplear expresiones del tipo: sen (3), tan (π), arccos (1/2), etc. MATHEMATICA incorpora la práctica totalidad de funciones matemáticas usuales. A continuación enumeramos algunas de ellas:

- `Sin` = seno de un ángulo en radianes.
- `Cos` = coseno de un ángulo en radianes.
- `Tan` = tangente de un ángulo en radianes.
- `Sqrt` = raíz cuadrada de un número real o complejo.
- `ArcTan` = arco cuya tangente.
- `ArcSin` = arco cuyo seno.
- `ArcCos` = arco cuyo coseno.
- `Log` = logaritmo natural.

Cada una de ellas se emplea de la misma manera que lo haríamos en el lenguaje matemático habitual con la salvedad de que en MATHEMATICA los argumentos de una función han de ir encerrados entre corchetes y no entre paréntesis, es decir:

- Para calcular el seno de 3 pondremos `Sin[3]` en lugar de sen (3).

- Para calcular el arco cuya tangente es 1 pondremos `ArcTan[1]` en lugar de arctan(1).
- Para calcular la raíz cuadrada de 2 pondremos `Sqrt[2]`.

Para la escritura de la raíz cuadradas disponemos además de los tres mecanismos alternativos habituales:

Lenguaje interno	Paleta	Paleta-Metodo teclado
`Sqrt`	$\sqrt{\blacksquare}$	ctrl + 2

Ejemplo 7:

In[1]:= **`Sin[π]`**
Out[1]= `0`
In[2]:= **`ArcTan[1]`**
Out[2]= $\frac{\pi}{4}$
In[3]:= **`Cos[1]`**
Out[3]= `Cos[1]`
In[4]:= **`N[Cos[1]]`**
Out[4]= `0.540302`

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.

5. VARIABLES, SÍMBOLOS Y ASIGNACIÓN DE VALORES

En diversas disciplinas científicas es habitual el uso de variables a las que se puede luego asignar un valor. Así por ejemplo, para calcular el área de un triángulo utilizamos la fórmula

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

en la que aparecen las variables ‘area’, ‘base’ y ‘altura’. Cuando estemos ante un triángulo concreto, asignaremos a las variable base y altura el valor que le correspondan y podremos calcular luego el área aplicando la fórmula. Por ejemplo:

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

Al estudiar los tipos de datos vimos que con MATHEMATICA también podemos manejar variables o símbolos. Un símbolo o variable es cualquier agrupación de caracteres y números que comienza con una letra minúscula. Ejemplos de posibles símbolos son: `x`, `z`, `hola`, `z1`, `hola318x`, `altura`, etc.
Ha de tenerse en cuenta que:

- No se puede insertar un espacio en blanco en el nombre de un símbolo (un espacio en blanco es considerado por el programa como una multiplicación).
- MATHEMATICA distingue entre mayúsculas y minúsculas.

Ejemplo 8:

`X1`	Símbolo correcto.
`3a`	Símbolo incorrecto ya que no comienza por una letra. `3a` se interpretará como el producto de `3` por `a`.
`x 1`	Incorrecto puesto que incluye un espacio en blanco. El programa lo interpretará como el producto de `x` por `1`.
`solucionecuacion`	símbolo correcto.
`solucion ecuacion`	Símbolo incorrecto por incluir espacio en blanco. Será interpretado como el producto del símbolo `solucion` por el símbolo `ecuación`.

`primervalor ≠ PrimerValor`	Vemos un ejemplo de dos símbolos en los que intervienen las mismas letras pero que son distintos por no coincidir mayúsculas y minúsculas.

A un símbolo se le puede asignar un valor que puede ser un número, lista, otro símbolo y en general cualquier cosa.
Tenemos pues, que una variable o símbolo puede encontrarse en dos estados:

a) Sin tener un valor asignado: En esta situación la variable es considerada como un dato de tipo exacto y nunca se evalúa.
b) Teniendo asignado un valor: Entonces en cada expresión o comando en el que aparezca tal símbolo, este es inmediatamente sustituido por su valor, antes incluso de que la expresión sea evaluada.

Para asignar un valor a una variable empleamos el símbolo de asignación '=' en la siguiente forma:

Comando: Operador de asignación inmediata `=`

Sintaxis:

`nombresimbolo = expresionvalor`

Resultado: Evalúa la expresión `expresionvalor` y la asigna al símbolo `nombresimbolo`, de manera que cada vez que en adelante aparece el símbolo `nombresimbolo` este es sustituido por `expresionvalor`. Tras ejecutar la asignación se produce como resultado el valor `expresiovalor`.

Ejemplo 9:

In[1]:= **a=5**
Out[1]= 5

A partir de esta orden, siempre que en cualquier expresión aparezca el símbolo "a" este será sustituido por 5

In[2]:= **a²+1**
Out[2]= 26

Una vez que hemos asignado a al variable `a` el valor 5, cada vez que el programa vuelve a encontrar el símbolo `a` lo sustituirá por 5.

In[3]:= $\mathbf{b=\sqrt{2}+1}$
Out[3]= $1+\sqrt{2}$
In[4]:= $\mathbf{\frac{a}{b}}$
Out[4]= $\frac{5}{1+\sqrt{2}}$
In[5]:= **p=x³+6x²+11x+6**
Out[5]= $6+11x+6x^2+x^3$

En una variable podemos almacenar expresiones en las que intervienen otras variables. En este caso en la variable `p` hemos almacenado un polinomio en el que interviene otra variable `x`.

In[6]:= $\mathbf{\frac{p}{x+3}}$
Out[6]= $\frac{6+11x+6x^2+x^3}{3+x}$
In[7]:= **x=3**
Out[7]= 3
In[8]:= **p**
Out[8]= 120

Si modificamos el valor de la variable `x` también variará el valor de `p`.

In[9]:= **x=-4**
Out[9]= -4
In[10]:= **p**
Out[10]= -6

Podemos modificar el valor asignado a una variable realizando una nueva asignación. El nuevo valor quedará reflejado en las futuras operaciones. Véase que ahora, en el polinomio `p`, `x` es sustituida por –4 dando lugar a un resultado diferente.

La supresión de valores asignados a un símbolo se puede realizar por medio del comando "`=.`".

Comando: Comando `=.` para eliminación de valores de variables
Sintaxis:

`simb=.`

Resultado: Elimina el valor asignado a la variable `simb` de mandera que dicha variable queda sin tener ningún valor asignado.

Ejemplo 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

`a3` valía `3` pero tras esta instrucción pierde su valor.

5.1. Manejo de expresiones simbólicas

Tal y como se ve en el ejemplo anterior, podemos manejar expresiones en las que aparecen variables o símbolos. Estas expresiones se denominan simbólicas. Para estos casos MATHEMATICA proporciona instrucciones que permiten simplificar o efectuar cálculos diversos.

Comando: **`Simplify`**

Sintaxis:

`Simplify[expresión]`

Resultado: Simplifica, si ello es posible, la expresión encerrada entre los corchetes.

Comando: **`Expand`**

Sintaxis:

`Expand[expresión]`

Resultado: Desarrolla los cálculos que aparecen pendientes de evaluación en la expresión encerrada entre corchetes.

Ejemplo 11:

In[1]:= **$p=y^3+6y^2+11y+6$**
Out[1]= $6+11y+6y^2+y^3$
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. ECUACIONES

En Mathematica las ecuaciones se escriben utilizando el símbolo de doble “==”. El igual simple “=” está reservado para las asignaciones y nunca se puede utilizar para formular una ecuación.

Ejemplo 12:

In[1]:= **x²-3x+2==0**

> Ecuación correctamente escrita mediante el símbolo de doble igual, **==**.

In[2]:= **x²-3x+2=0**

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

> Ecuación incorrecta con igual simple, **=**. Obtendremos un mensaje de error.

Para resolver ecuaciones utilizamos el commando Solve.

Comando: `Solve`
Sintaxis:
1) Ecuación de una variable: `Solve[ecuación,x]`
2) Sistema de ecuaciones:
`Solve[{ecuacion₁,...,ecuacionₘ},{x₁,...,xₙ}]`

Resultado: **1)** Resuelve la ecuación `ecuacion` despejando en ella la variable `x`.
2) Resuelve el sistema formado por las ecuaciones `ecuacion₁`,..., `ecuacionₘ` despejando en ellas las variables $x_1,\ldots,x_n$.
 Es importante tener en cuenta que en MATHEMATICA el operador de comparación que debe aparecer en una ecuación es == (doble =) nunca = (un = sencillo) que es el operador de asignación.

Por ejemplo:

In[1]:= **Solve[a x²+b x+c==0,x]**

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

> Al resolver la ecuación de segundo grado obtenemos las dos soluciones posibles.

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

Out[2]= $\{\{\alpha\to 2,\beta\to 1\}\}$

> Muy importante emplear en las ecuaciones un doble igual (**==**) en lugar de un igual sencillo (=).

> Si desarrollamos las operaciones, la ecuación
> **α{1,2-1}+β{2,1,1}=={4,5,-1}**
> equivale al sistema
> $$\begin{cases}\alpha+2\beta=4\\ 2\alpha+\beta=5\\ -\alpha+\beta=-1\end{cases}$$
> en el que las variables a despejar son α y β. Véase que lo que estamos calculando aquí son los coeficientes α y β por los que hay que multiplicar a los vectores (1,2,-1) y (2,1,1) para obtener el vector (4,5,1).

ELEMENTOS DEL PROGRAMA

- El símbolo % se utiliza para referirse a una salida determinada. A medida que vamos efectuando operaciones vamos obteniendo una lista de entradas (In) y salidas (Out) que se irán numerando ordenadamente. Es posible que necesitemos emplear un resultado que hemos calculado con anterioridad, empleando el símbolo % es posible evitar el teclear nuevamente tal resultado. La sintaxis de uso para % es:
    - % n = Out nº n.
    - % = Out inmediatamente anterior.
    - %% = Out penúltimo.
    - %%% = Out antepenúltimo.
    - etc.

Ejemplo 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]:= x^3

Out[5]= x^3

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

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

- Podemos eliminar cualquiera de las celdas que aparezcan en el área de trabajo. Para ello pulsaremos con el ratón sobre la celda elegida para marcarla y posteriormente la eliminaremos con la tecla de borrado.
- Todo comando o palabra clave que forma parte o viene incorporado en el programa MATHEMATICA se escribe con la primera letra de cada palabra en mayúsculas y las demás en minúsculas. Por ejemplo: `Pi`, `Sin`, `Sqrt`, `MatrixForm` (palabra compuesta por `Matrix` y `Form`, véase que cada una de las palabras componentes sí empieza con mayúscula).
- Cuando cometemos un error de cualquier tipo el programa nos lo advierte mediante un mensaje de color azul.

Ejemplo 14*:*

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

Power::infty : Infinite expression $\frac{1}{0}$ encountered.

Out[1]:= ComplexInfinity

- La instrucción ? nos permite obtener información acerca del funcionamiento de los distintos comandos que forman parte del lenguaje de MATHEMATICA. Se utiliza en la forma

  `?comando`

  tras lo cual obtendremos la información solicitada.

Ejemplo 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.
```

- En una misma línea y celda podemos incluir más de una instrucción siempre que las separemos mediante ”;”

```
In[1]:= a=3; b=2;
        a+b
Out[1]= 5
```

  Un signo ”;” al final de una instrucción evita que se muestre el resultado producido por tal operación:

```
In[2]:= c=a+b
Out[2]= 3
In[3]:= c=a+b;
```

- Recuérdese que:

Los paréntesis ( )	Se utilizan para agrupar operaciones algebraícas. `(1+3)*2`
Los corchetes simples [ ]	Se utilizan para agrupar los argumentos de una función. `Cos[4]`
El igual simple **=**	Se utiliza para asignar valor a una variable: `x=2`
El igual doble **==**	Se utilizan para escribir ecuaciones. `x+1==3`