Práctica 1

Cálculo con Mathematica

1.- Introducción al *Mathematica*

El programa *Mathematica* constituye una herramienta muy potente para la realización de todo tipo de cálculos matemáticos: operaciones aritméticas, cálculo simbólico, gráficos,… Las últimas versiones del programa incorporan un editor de texto (bastante aceptable) y permiten el manejo de "paletas" de símbolos que facilitan el uso de los comandos e instrucciones del programa y la utilización de la simbología habitual en las fórmulas y operaciones matemáticas.

El programa *Mathematica* está estructurado en 2 partes:

1.- El **Front End**: es la **interface** que permite la comunicación con el usuario. Presenta las entradas y salidas y permite crear y editar ficheros que contienen texto, cálculos, gráficas, etc. Estos ficheros tienen la extensión **.nb** y se denominan **Notebooks (Cuadernos de notas).** La información que contienen estos ficheros se estructura mediante un sistema de **celdas** de diferentes tipos (título, subtítulo, sección, texto, input, output, ….) .

2.- El **Kernel** (**núcleo**): constituye la estructura interna del programa y es el que se encarga de la realización de los cálculos y operaciones matemáticas que solicita el usuario.

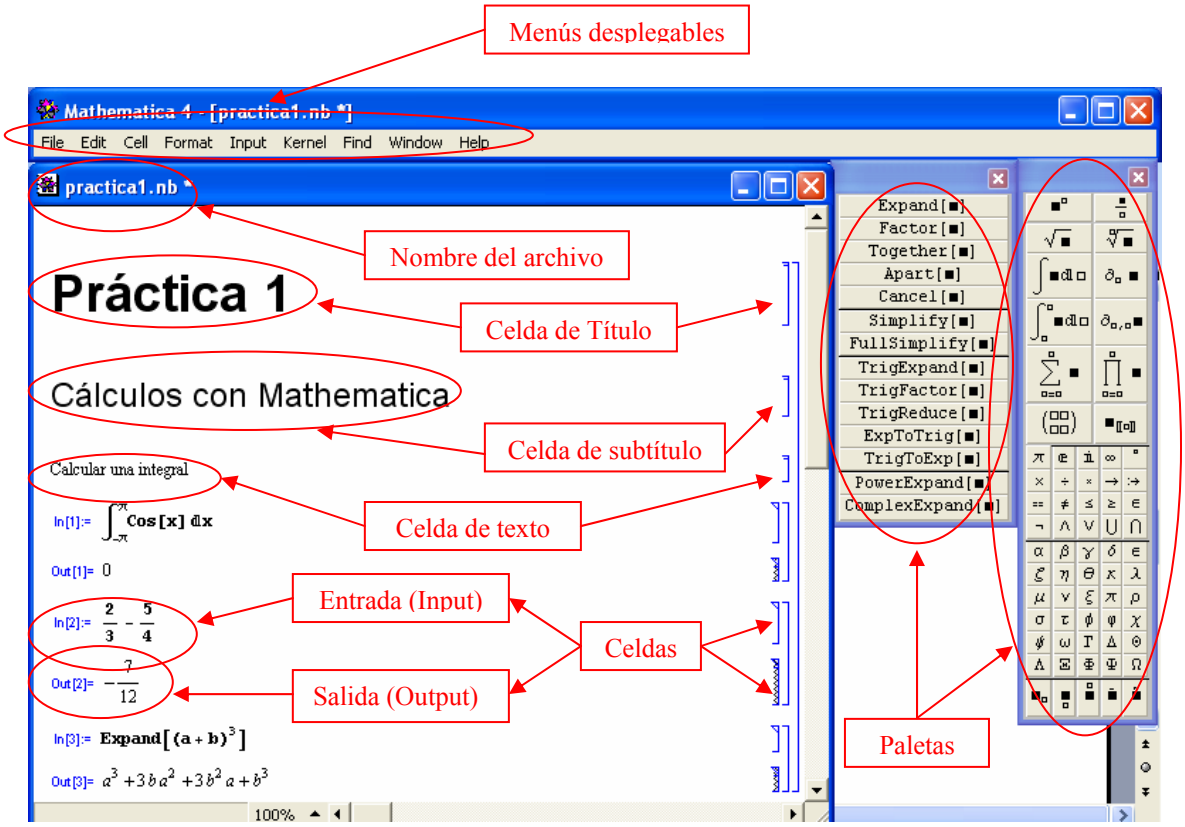


Figura 1: Aspecto que presenta el programa *Mathematica* 4.0

Las posibilidades que ofrece el programa *Mathematica* son muy variadas:

a) **Cálculo numérico:** Realización de operaciones de cálculo simples o muy complicadas. El programa nos devuelve el resultado como si se tratara de una calculadora científica. La diferencia es que permite trabajar con la precisión que queramos.

In[4]:= 2^{100}

Out[4]= 1267650600228229401496703205376

Mathematica nos devuelve el valor exacto y no una aproximación como haría una calculadora. O bien un valor aproximado:

In[5]:= 2^{100} // N

Out[5]= 1.26765×10^{30}

b) **Cálculo simbólico:** Permite trabajar y operar con expresiones simbólicas.

In[6]:= Expand[$(a+b)^3$]

Out[6]= $a^3 + 3\,b\,a^2 + 3\,b^2\,a + b^3$

In[7]:= D[x^n, x]

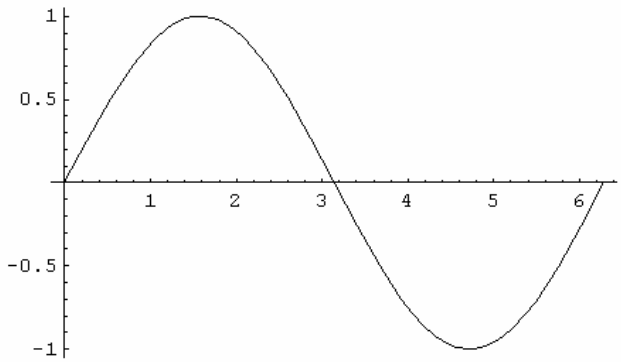
Out[7]= $n\,x^{n-1}$

In[8]:= $\int_0^1 x\,\mathrm{Log}[x]\,dx$

Out[8]= $-\frac{1}{4}$

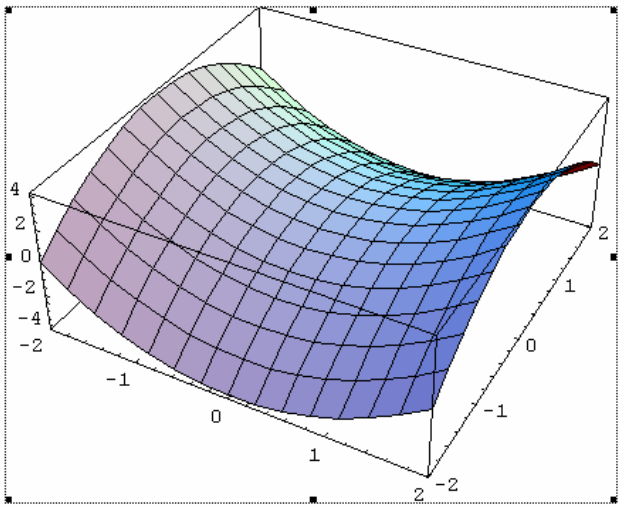
c) **Gráficos**: Permite visualizar gráficos en dos o tres dimensiones

In[10]:= Plot[Sin[x], {x, 0, 2 π}]



Out[10]= - Graphics -

```
In[11]:= Plot3D[x^2 - y^2, {x, -2, 2}, {y, -2, 2}]
```




```
Out[11]= - SurfaceGraphics -
```

d) **Definición de variables y funciones:** Permite definir y operar con variables y funciones definidas por el usuario.

```
In[12]:= a = 7; b = 3;
         a + b

Out[13]= 10

In[14]:= f[x_] := x^2 - 5 x + 6

In[15]:= f[3]

Out[15]= 0

In[16]:= f'[x]

Out[16]= 2 x - 5
```

e) **Confección de programas:** *Mathematica* incorpora su propio lenguaje de programación que puede utilizarse para implantar algoritmos de cálculo, bucles, procesos iterativos, …

```
In[17]:= x1 = 2;
         For[n = 1, n ≤ 5, n = n + 1,
          x_{n+1} = 3/(4 - x_n);
          Print[N[x_n]]]

         2.

         1.5

         1.2

         1.07143

         1.02439
```

2.- Primeros pasos con *Mathematica*

Como hemos indicado con anterioridad la información contenida en los Notebooks se estructura mediante un sistema de celdas de distintos tipos. Por defecto *Mathematica* asigna a las celdas el tipo **Input** (**entrada**). Estas celdas contienen las **órdenes** (comandos, instrucciones, operaciones, etc.) que el usuario quiere realizar. Para que el programa ejecute la instrucción contenida en una celda **Input** hay que pulsar la tecla **Intro** (o la combinación: **Mayúscula+Return)**.

Si escribimos **2+7** y pulsamos **Intro**, el programa nos devuelve:

```
In[19]:= 2 + 7

Out[19]= 9
```

Asociada a cada celda **Input** el programa genera una celda **Output** (**salida**) donde muestra el resultado de la operación requerida. Ambas celdas aparecen con los indicadores **In[n]:= y Out[n]**. El número **n** se genera automáticamente y de manera secuencial a lo largo de una sesión de trabajo, desde que se inició el programa hasta que salimos de él. Este número puede sernos de utilidad si queremos utilizar un cálculo previamente obtenido, dado que el programa *Mathematica* almacena en el **Kernel** (**núcleo**) todas las evaluaciones y cálculos que hemos realizado durante nuestra sesión de trabajo.

El signo **%** se utiliza para referirnos al resultado que figura en la celda **Output** inmediatamente anterior.

Si escribimos **%*5** y pulsamos **Intro**, el programa nos devuelve:

```
In[20]:= % * 5

Out[20]= 45
```

dado que el % hace referencia al resultado que figura en la celda **Output** anterior, es decir en la celda **Out[19]**

Pueden utilizarse también los símbolos **%%**, **%%%**, etc. para referirnos al resultado que figura en la penúltima, antepenúltimo, etc. celda **Output**. Sin embargo, la forma más cómoda es utilizar **%n** si queremos referirnos al resultado contenido en la celda **Out[n].**

Si escribimos **%13 – 3** y pulsamos **Intro**, el programa nos devuelve:

```
In[14]:= %13 - 3

Out[14]= 7
```

Si nos fijamos en la página anterior podemos comprobar que la celda **Out[13]** contiene el valor **10**.

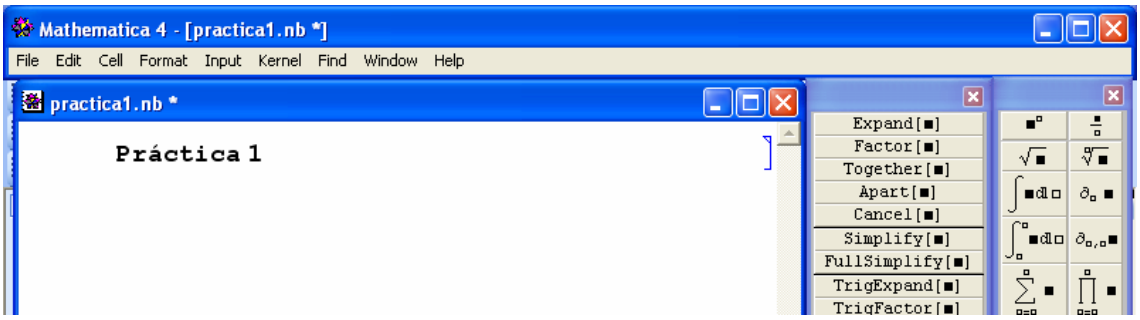
A parte de las celdas **Input** y **Output**, *Mathematica* puede también trabajar con otros tipos de celdas: **Títle**, **Section**, **Subsection**, **Text**, … Se trata de celdas cuyo contenido **no es evaluable** y que se utilizan para dar a nuestro **Notebook** un aspecto más agradable en cuanto a la presentación del texto: permitiendo incluir un título, organizar el contenido en secciones, mostrar comentarios, etc. Como hemos indicado Mathematica supone por defecto que todas las celdas son de tipo **Input.** Para cambiar el tipo para una determinada celda utilizamos el menú desplegable:

Format → Style

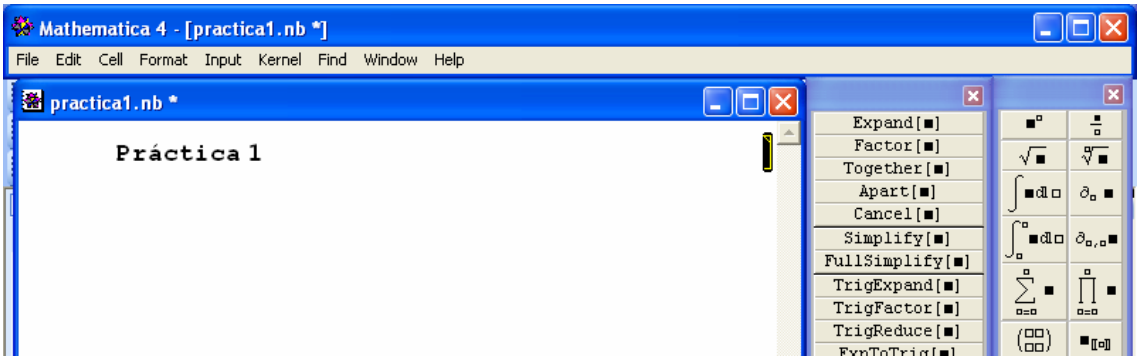
seleccionando con el cursor el estilo deseado.

Por ejemplo: si queremos incluir una celda de **Título** seguiremos los siguiente pasos:

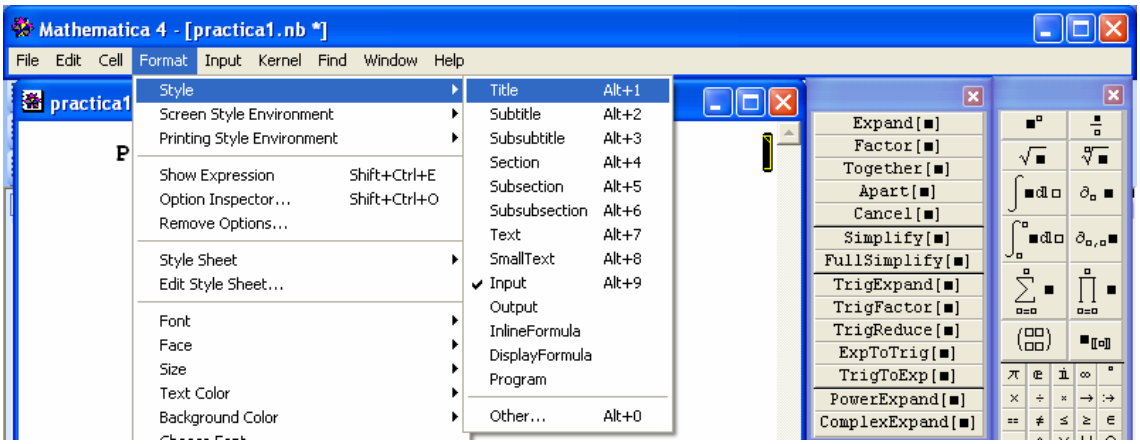
1) **Escribimos el título**: (por ejemplo: **Práctica 1**)



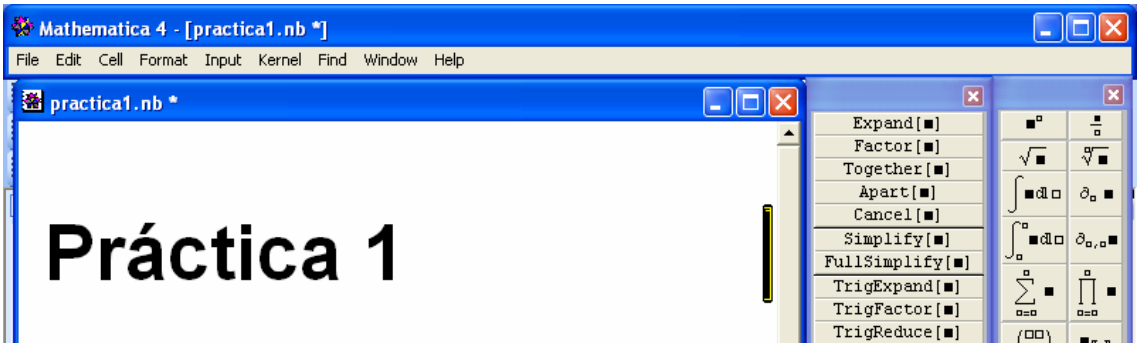
2) **Seleccionamos la celda**: botón izquierdo del ratón sobre el corchete que figura en la parte derecha de la celda.



3) Desplegamos el menú: **Format** → **Style** y seleccionamos la opción **Title**



4) El programa nos mostrará entonces



El aspecto final de nuestro **Notebook** dependerá del estilo de documento que estemos utilizando. *Mathematica* incorpora una serie de estilos de documento predefinidos: **Article-Clasic**, **Demo**, **Report**, **Clasroom**, etc. Cada uno de estos estilos asigna distintos tipos de letra, de fondo, de combinaciones de color, etc, a nuestro documento y a cada uno de los tipos de celda. Para seleccionar el estilo del documento deseado utilizamos el menú desplegable:

Format → StyleSheet.

3.- Sintaxis general de las instrucciones en *Mathematica*

A la hora de escribir las instrucciones que queremos que ejecute el programa *Mathematica,* contenidas en una celda **Input,** hemos de prestar mucha atención a las siguientes reglas:

a) **Las mayúsculas y las minúsculas**: Mathematica distingue unos caracteres de otros. No es lo mismo escribir **Expand** (instrucción que utilizaremos para desarrollar una determinada expresión matemática) que **expand** (instrucción no reconocida por el programa y cuya utilización generará un mensaje de error y no el resultado esperado).

```
In[15]:= Expand[(x + y)^2]

Out[15]= x^2 + 2 y x + y^2

In[16]:= expand[(x + y)^2]

         General::spell1 :
           Possible spelling error: new symbol name "expand" is
               similar to existing symbol "Expand".

Out[16]= expand((x + y)^2)
```

Importante: Todas las funciones, instrucciones, opciones, constantes, etc. incorporadas en el programa *Mathematica* empiezan con mayúsculas

b) **Los espacios:** Un espacio colocado entre dos variables o números en una celda **Input** se interpreta como el **signo de multiplicar**.

```
In[17]:= 7 5

Out[17]= 35

In[18]:= x x

Out[18]= x^2
```

c) **Paréntesis, corchetes y llaves**: Los paréntesis ( ) se utilizan para agrupar términos e indicar prioridad. Los corchetes [ ] son para uso exclusivo de las funciones para delimitar sus argumentos. Las llaves { } se utilizan para definir listas de elementos: vectores, matrices, etc.

```
In[19]:= 2 + 3 (7 - 4)

Out[19]= 11

In[20]:= Cos [π]

Out[20]= -1

In[21]:= {{1, -2, 3}, {4, 0, -1}, {1, 1, 1}}
```

Out[21]= $\begin{pmatrix} 1 & -2 & 3 \\ 4 & 0 & -1 \\ 1 & 1 & 1 \end{pmatrix}$

4.- Operaciones aritméticas elementales

Mathematica puede utilizarse como una calculadora convencional para la realización aunque con una importante diferencia: la precisión en el cálculo. Las operaciones se realizan en forma exacta o bien aproximada (con el grado de precisión que queramos).

Mathematica reconoce los operadores aritméticos habituales: **suma** (+), **diferencia** (-), **producto** (* o espacio en blanco), **división** (/) y **potencia** (^). Si intervienen varios de estos operadores en una misma instrucción *Mathematica* aplica los criterios de prioridad habituales entre ellos: potencias → productos y divisiones → sumas y rectas.

```
In[22]:= 2 + 5 * 3 ^ 2
```

Primero se efectúa la potencia, después el producto y por último la suma

```
Out[22]= 47
```

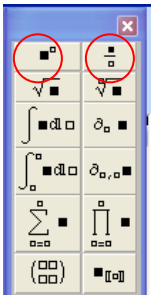
Sin embargo los criterios de prioridad se pueden alterar mediante el uso de los paréntesis:

```
In[23]:= (2 + 5) * 3 ^ 2
```

Primero se efectúa la suma, después la potencia y por último el producto

```
Out[23]= 63
```

Las operaciones de potencia y división también pueden indicarse utilizando los iconos $\blacksquare^{\square}$ y $\frac{\blacksquare}{\square}$ que figuran en la paleta.



La siguiente operación: `(2 / 3 - 1 / 5 ^ 2) / (3 - 1 / 2) ^ 2` puede escribirse en la forma:

$$\frac{\frac{2}{3} - \frac{1}{5^2}}{\left(3 - \frac{1}{2}\right)^2}$$

5.- Tipos de números y precisión en los cálculos

Mathematica puede trabajar con todos los tipos de números: enteros (**Integer**), racionales (**Rational**), reales (**Real**) y complejos (**Complex**). Para saber de qué tipo es un número concreto podemos utilizar la instrucción: **Head**[número]

```
In[24]:= Head[0.35]

Out[24]= Real
```

Mathematica no tiene ninguna limitación en cuanto al tamaño de los números enteros (la única limitación viene impuesta por la memoria del ordenador). Todas las operaciones realizadas con números enteros o racionales se realizan con precisión infinita por lo que el programa siempre nos devuelve un número exacto.

In[25]:= 2^{1000}

Out[25]= 107150860718626732094842504906000181056140481`.
170553360744375038837035105112493612249319 83`.
788156958581275946729175531468251871452856 92`.
314043598457757469857480393456777482423098 54`.
210746050623711418779541821530464749835819 41`.
267398767559165543946077062914571196477686 54`.
216766042983165262438683720566806937 6

In[26]:= $\dfrac{\frac{2}{3} - \frac{1}{5^2}}{\left(3 - \frac{1}{2}\right)^2}$

Out[26]= $\dfrac{188}{1875}$

Mathematica asigna el tipo **Real** a los números decimales. La forma de operar con estos números es completamente distinta. Si en una operación interviene un número decimal *Mathematica* nos devolverá siempre un valor aproximado (con una precisión de 6 dígitos decimales, aunque internamente el resultado lo realiza con una precisión de 16 cifras decimales).

In[27]:= $2.^{1000}$

Out[27]= 1.07151×10^{301}

Mathematica distingue entre el número **2** (entero) y el número **2.** (**2.0**) que es tratado como decimal de ahí que el resultado mostrado sea un número aproximado y no un número exacto como nos muestra en **Out[25]**.

In[28]:= $\frac{7}{5} - \frac{1}{9} + \frac{1}{2}$

Out[28]= $\dfrac{161}{90}$ *Mathematica* nos devuelve un valor exacto

In[29]:= $\frac{7}{5} - \frac{1}{9} + 0.5$

Out[29]= 1.78889 *Mathematica* nos devuelve un valor aproximado

Este diferente comportamiento puede resultar chocante las primeras veces que se realizan ciertas operaciones con *Mathematica*.

In[30]:= $1 + \sqrt{2}$

Out[30]= $1 + \sqrt{2}$ ¡¡¡ *Mathematica* nos devuelve el valor exacto !!!

En la operación anterior podemos forzar a Mathematica que nos devuelva un valor aproximado de las siguientes maneras:

1) Escribiendo uno de los números en formato decimal.

   In[31]:= $1. + \sqrt{2}$

   Out[31]= 2.41421

2) Pidiendo que nos de una aproximación (utilizamos el comando **N[***expresión***]** que es equivalente a escribir *expresión***//N**)

   In[32]:= $1 + \sqrt{2}$ // N

   Out[32]= 2.41421

3) Pidiendo que nos de el resultado con una determinada precisión mediante la instrucción

   N[*expresión*,**n]**,

   donde **n** indica el número de cifras decimales.

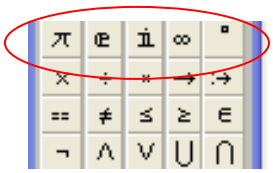
   In[33]:= N[$1 + \sqrt{2}$, 30]

   Out[33]= 2.41421356237309504880168872421

6.- Constantes y funciones incorporadas en *Mathematica*

El programa *Mathematica* tiene predefinidas una serie de constantes habituales en Matemáticas:

- El número *pi*: **`Pi`**
- El número *e*: **`E`**
- La unidad imaginaria: **`I`**
- Infinito: **`Infinity`**
- Factor de conversión a grados: **`Degree`**



Asimismo, el programa *Mathematica* incorpora todas las funciones matemáticas elementales. Sus nombres responden a su abreviatura proveniente del inglés por lo que algunas de ellas tienen una denominación distinta a la usual en español.

Funciones trigonométricas		Funciones trigonométricas inversas	
seno	**`Sin[x]`**	arcoseno	**`Arcsin[x]`**
coseno	**`Cos[x]`**	arcocoseno	**`ArcCos[x]`**
tangente	**`Tan[x]`**	arcotangente	**`ArcTan[x]`**

Funciones hiperbólicas		Funciones hiperbólicas inversas	
seno hiperbólico	**`Sinh[x]`**	argumento seno hiperbólico	**`ArcSinh[x]`**
coseno hiperbólico	**`Cosh[x]`**	argumento coseno hiperbólico	**`ArcCosh[x]`**
tangente hiperbólica	**`Tanh[x]`**	argumento tangente hiperbólica	**`ArcTanh[x]`**

Otras funciones		Otras funciones	
Exponencial	**`Exp[x]`**	Logaritmo en base *a* de *x*	**`Log[a,x]`**
Raíz cuadrada	**`Sqrt[x]`**	Valor absoluto	**`Abs[x]`**
Logaritmo neperiano	**`Log[x]`**	Función signo	**`Sign[x]`**

Observaciones:

1) Las funciones trigonométricas utilizan el argumento en radianes.

In[34]:= $\mathbf{Sin}\left[\frac{\pi}{4}\right]$

Out[34]= $\frac{1}{\sqrt{2}}$

Si queremos utilizar el argumento en grados utilizamos el factor de conversión **Degree** (º).

```
In[35]:= Sin[90 Degree]

Out[35]= 1
```

2) Si el argumento utilizado en cualquiera de las funciones es un número exacto *Mathematica* nos devolverá un valor exacto (como sabemos siempre podemos pedir que nos dé un valor aproximado utilizando el comando N). En cambio, si el argumento es un número decimal, *Mathematic*a nos devolverá un valor aproximado con una precisión de 6 dígitos (internamente trabajará con 16 dígitos).

```
In[36]:= Log[2]

Out[36]= Log[2]

In[37]:= ArcSin[0.5]

Out[37]= 0.523599
```

7.- Variables y funciones definidas por el usuario

Las variables se utilizan para el almacenamiento temporal de información que vaya a ser utilizada en cálculos posteriores.

a) El nombre de la variable tiene que empezar con una letra.

Nombres válidos: **`x, y, radio, longitud, xm, xy, x1, z4,`**…

b) No pueden utilizarse como nombres de variable aquellos reservados para designar a las instrucciones, funciones y constantes de *Mathematica.* Una forma de evitar esto es escribir todos los nombres de las variables utilizadas en minúsculas.

c) No es necesario declarar previamente las variables que vamos manejar es el sistema el que se encarga de identificarlas en el momento en que se definen.

d) La asignación de valores a una variable se puede realizare de dos formas:

Asignación inmediata: *nombredevariable* = *valor o expresión*;

```
In[38]:= a = 2

Out[38]= 2

In[39]:= x = a

Out[39]= 2
```

En el ejemplo anterior hemos realizado dos asignaciones inmediatas. En la primera de ellas asignamos a la variable *a* el valor 2. En la segunda a la variable *x* le asignamos el valor que **en ese momento** tenga la variable *a*, es decir, 2. Si posteriormente cambiamos el valor de la variable *a*, la variable *x* no se verá afectada por este cambio.

```
In[40]:= a = 4

Out[40]= 4

In[41]:= x

Out[41]= 2
```

Asignación diferida: *nombredevariable* **:=** *valor o expresión*

```
In[42]:= y := a

In[43]:= y

Out[43]= 4

In[44]:= a = 6

Out[44]= 6

In[45]:= y

Out[45]= 6
```

A la variable *y* se le asocia el valor de la variable *a*. Sin embargo esta asignación no se hace de manera inmediata sino que, cuando se vaya a utilizar la variable *y,* el valor que se le asignará será el que tenga en ese momento la variable *a*, es decir, los cambios posteriores efectuados en la variable *a* SI repercuten en la variable *y*.

La asignación diferida se utiliza principalmente para definir funciones. La forma de definir una función es la siguiente:

nombredefuncion[*variable*1_,*variable*2_,…,*variable*n]:=*expresión*.

```
In[38]:= f[x_] := x^2

In[39]:= f[7]

Out[39]= 49
```

En el ejemplo anterior hemos definido la función f(x)=x^2. Observemos nuevamente que el argumento tiene que indicarse entre corchetes. A partir de este momento podremos utilizar la nueva función para evaluarla en un punto, calcular su derivada, representarla gráficamente, etc.

```
In[40]:= f[2]

Out[40]= 4
```

```
In[41]:= f'[3]

Out[41]= 6
```

Observemos que no es necesario indicar el tipo de variable que interviene en la función (Entera, real, alfanumérica,…) es el propio programa el que se encarga de atribuir el tipo según el dato que contiene la variable. Sin embargo, podemos indicar el tipo para restringir el tipo de datos válidos para ser almacenados en una determinada variable. Esta asignación de tipo se realiza de la forma:

nombredevariable_tipo

In[42]:= `g[x_Integer] :=` $\frac{2x}{x+1}$

En la instrucción anterior hemos definido una función g de variable entera. Dicha función sólo reconocerá como posible argumento a un número entero. Si intentamos evaluarla en un número no entero, *Mathematica* no nos devuelve nada.

```
In[43]:= g[3]
```

Out[43]= $\frac{3}{2}$

```
In[44]:= g[0.5]

Out[44]= g[0.5]
```

<u>Observaciones</u>:

1) Para evitar posibles conflictos y resultados inesperados es aconsejable que las variables que figuran como argumento de una función estén limpias, es decir, que no se les haya asignado previamente ningún valor. Para borrar una variable utilizaremos la instrucción:

Clear[*variable*1,*variable*2,*variable*3,….**]**

Clear[a,x,y]	Borra las variables *a*, *x* e *y* de la memoria del ordenador.
Clear["a*"]	Borra las variables y funciones cuyo nombre empiece por *a*.
Clear["Global,*"]	Borra todas las variables y funciones definidas cn anterioridad.

2) No existe ninguna limitación sobre el número de variables que pueden utilizarse como argumentos de una función.

```
In[45]:= Clear[f]
```

In[46]:= `f[x_, y_] :=` $x^2 - 2xy + y^3$

La instrucción anterior define borra la definición de la función f y vuelve a definirla como una función de dos variables.

8.- Operaciones usuales en *Mathematica*

1) **Desarrollar una expresión**: **Expand[***expresión***]**

In[47]:= Expand[$(a-b)^4$]

Out[47]= $a^4 - 4a^3b + 6a^2b^2 - 4ab^3 + b^4$

2) **Simplificar una expresión**: **Simplify[***expresión***]**

FullSimplify[*expresión***]**

In[48]:= $\text{Simplify}\left[\frac{x^2 - 2x + 1}{x^2 - 1}\right]$

Out[48]= $\frac{-1 + x}{1 + x}$

3) **Resolver una ecuación o despejar una variable en una igualdad**:

Solve[*expresion*1**==***expresion*2,*variable*]

Despeja la variable *x* en la igualdad *expresión*1=*expresion*2. Obsérvese que entre ambas expresiones figura el signo **==** (doble signo igual) para que el programa no lo confunda con el signo **=** que como sabemos es utilizado para asignar valores a una variable.

In[49]:= $\text{Solve}\left[x + \sqrt{x} == 2\right]$

Out[49]= $\{\{x \to 1\}\}$

Esta expresión es equivalente a

In[50]:= $\text{Solve}\left[x + \sqrt{x} == 2,\ x\right]$

Out[50]= $\{\{x \to 1\}\}$

En este caso no es necesario indicar la variable que queremos despejar porque en la igualdad sólo figura la variable *x*. No ocurre lo mismo con la siguiente ecuación en la que intervienen cuatro variables: *a*, *b*, *c* y *x*

In[51]:= $\text{Solve}[a x^2 + b x + c == 0,\ x]$

Out[51]= $\left\{\left\{x \to \frac{-b - \sqrt{b^2 - 4 a c}}{2 a}\right\}, \left\{x \to \frac{-b + \sqrt{b^2 - 4 a c}}{2 a}\right\}\right\}$

Se obtiene un resultado distinto si escribimos:

In[52]:= $\text{Solve}[a x^2 + b x + c == 0,\ a]$

Out[52]= $\left\{\left\{a \to \frac{-c - b x}{x^2}\right\}\right\}$

En este último caso se ha despejado la variable *a* en la expresión anterior.

9.- Otras operaciones con *Mathematica*

Factor[*expresión***]**

Realiza la descomposición factorial de la *expresión* dada.

Collect[*expresión,variable***]**

Agrupa la *expresión* algebraica en potencias ordenadas de la *variable* indicada

Expand[*expresión***,Trig->True]**

Desarrolla la *expresión* trigonométrica utilizando las fórmlas de trigonometría

Factor[*expresión***,Trig->True]**

Factoriza expresiones trigonométricas

Binomial[m,n]

Calcula el número combinatorio **m** sobre **n**

n!

Calcula el factorial del número **n**

10.- Ejercicios propuestos

1.- Calcular usando *Mathematica:*

a) 3^{30}
b) Un valor aproximado de la raiz cuadrada del número *pi* con 20 cifras decimales.
c) logaritmo neperiano de 10.
d) Logaritmo decimal de 10.
e) Seno de 45º.
f) 7 sobre 2
g) Valor absoluto de -3.

2.- Hallar los siguientes resultados:

a) Sen(π)
b) e^3
c) cos(45º)
d) Convierte π radianes en grados
e) Ln 2
f) artg(-1)
g) 7!

3.- Realiza las siguientes operaciones, entendiendo lo que realiza cada una:

a) N[10/3]
b) x+x^3+7y^2x+8y x
c) Collect[%,x]
d) Expand[(x+y)^2]
e) Simplify[(x-1)^2(x+2)/((x+1)(x-3)^2))]
f) **Binomial[5,3]**