

Práctica 0. 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.

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[1]:= 2^{64}

Out[1]= 18 446 744 073 709 551 616

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

In[2]:= $2^{64} // N$

Out[2]= 1.84467×10^{19}

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

In[3]:= **Expand** [$(x + y)^3$]

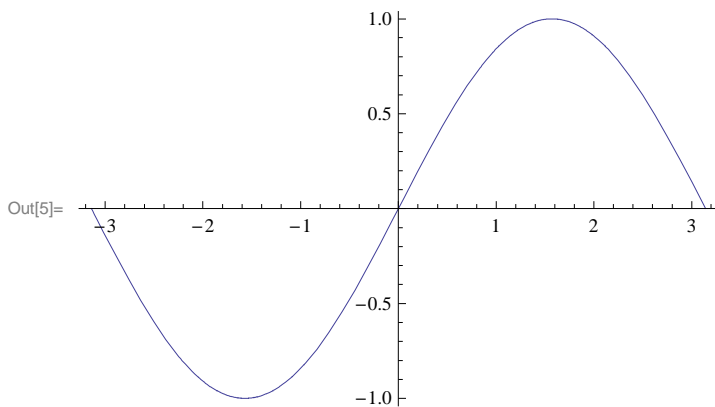
Out[3]= $x^3 + 3 x^2 y + 3 x y^2 + y^3$

In[4]:= **Simplify** [$\frac{a^2 - b^2}{a + b}$]

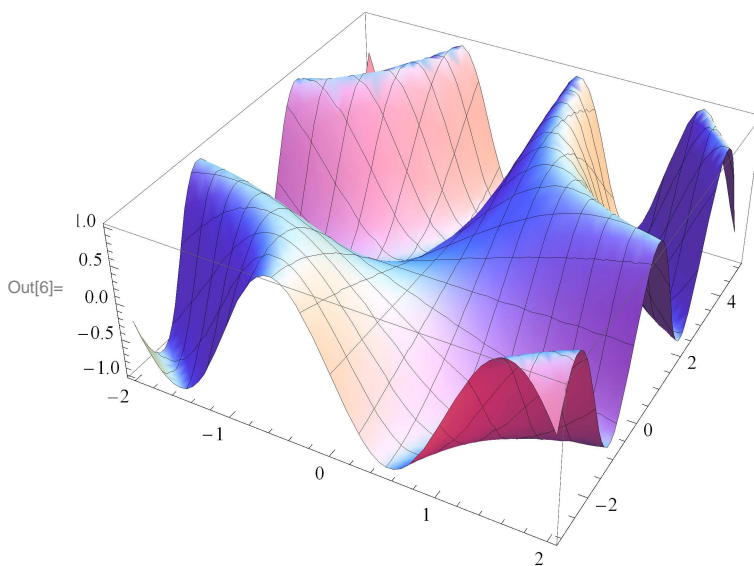
Out[4]= $a - b$

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

```
In[5]:= Plot[Sin[x], {x, -π, π}]
```



```
In[6]:= Plot3D[Sin[x*y], {x, -2, 2}, {y, -3, 5}]
```



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

```
In[7]:= a = 3;
        b = 2;
        a * b
```

Out[9]= 6

```
In[10]:= f[x_] := x^3 - 3 x^2
```

```
In[11]:= f[2]
```

Out[11]= -4

```
In[12]:= f'[x]
```

Out[12]= $-6x + 3x^2$

```
In[13]:= f'[1]
```

Out[13]= -3

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

```
In[14]:= x1 = 1;
x2 = 1;
For [n = 1, n ≤ 6, n = n + 1,
xn+2 = xn+1 + xn;
Print [xn+2 ] ]
```

2

3

5

8

13

21

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 los 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[17]:= 2 + 7
```

```
Out[17]= 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 %*17 y pulsamos Intro, el programa nos devuelve:

```
In[18]:= %17
```

```
Out[18]= 9
```

dado que el % hace referencia al resultado que figura en la celda Output anterior, es decir en la celda Out[17]

Pueden utilizarse también los símbolos %,%%,etc. para referirnos al resultado que figura en la penúltima, antepenúltima, 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 %18-3 y pulsamos Intro, el programa nos devuelve:

```
In[19]:= %18 - 3
```

```
Out[19]= 6
```

Si nos fijamos vemos que la celda Out[19] contiene el valor 9.

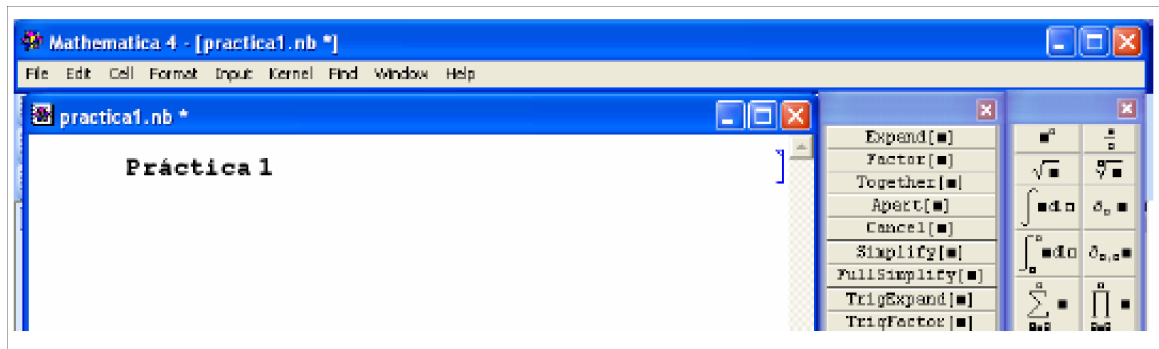
A parte de las celdas Input y Output, Mathematica puede también trabajar con otros tipos de celdas: Title, Section, Subsection, Text, ... Se trata de celdas cuyo contenido no es evaluable y que se utilizan para dar a nuestro Notebook un aspecto más agradable en cuanto a la presentación del texto; permitiendo 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

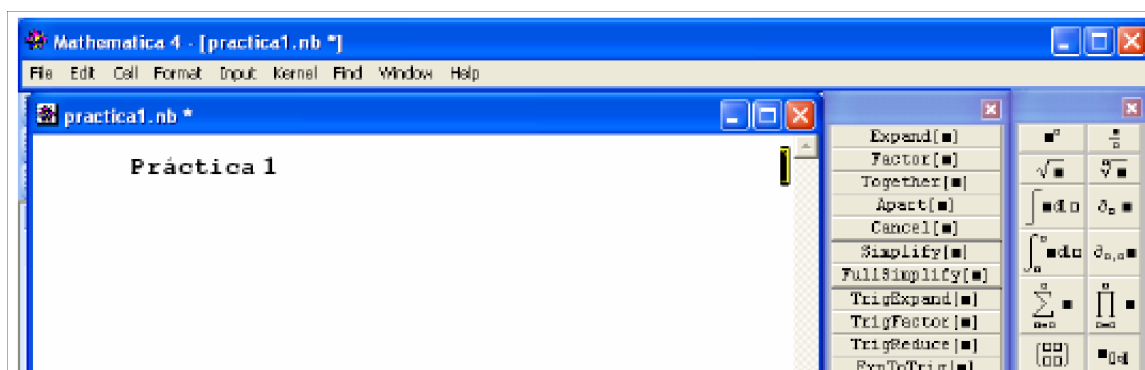
seleccionando con el cursor el estilo deseado.

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

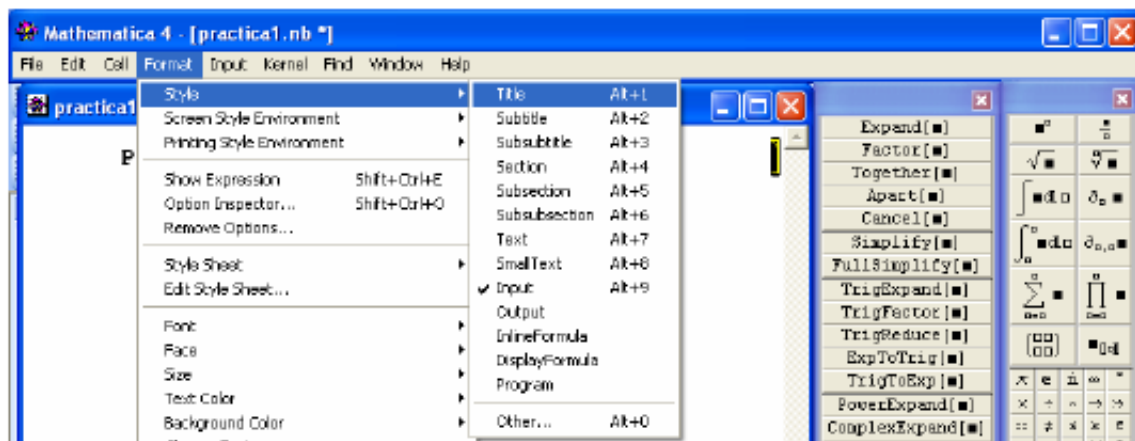
1) Escribimos el título: (por ejemplo: Práctica 1)



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, Classroom, 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[20]:= **Expand** [(x + y)²]

Out[20]= x² + 2 x y + y²

In[21]:= **expand** [(x + y)²]

Out[21]= 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[22]:= **2 × 3**

Out[22]= 6

In[23]:= **base = 3;
altura = 5;
base altura**

2

Out[25]= $\frac{15}{2}$

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[26]:= 2 * (3 + 5) - 3 * 4
```

```
Out[26]= 4
```

```
In[27]:= Sin[π]
```

```
Out[27]= 0
```

```
In[28]:= {{-1, 2}, {3, 1}} // MatrixForm
```

```
Out[28]/MatrixForm=

$$\begin{pmatrix} -1 & 2 \\ 3 & 1 \end{pmatrix}$$

```

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 restas.

```
In[29]:= 3 * 8
```

```
Out[29]= 24
```

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

```
In[30]:= (2 + 6) ^ 2 + 7
```

```
Out[30]= 71
```

Las operaciones de potencia y división también pueden indicarse utilizando los iconos que figuran en la paleta.



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[31]:= Head[3]
```

```
Out[31]= Integer
```

In[32]:= **Head [3 . 5]**

Out[32]= Real

In[33]:= **Head [3 i]**

Out[33]= Complex

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[34]:= **2⁵⁰⁰**

Out[34]= 3 273 390 607 896 141 870 013 189 696 827 599 152 216 642 046 043 064 789 483 291 368 096 133 796 404 3
674 554 883 270 092 325 904 157 150 886 684 127 560 071 009 217 256 545 885 393 053 328 527 589 376

In[35]:=
$$\frac{\frac{3}{5} + \left(\frac{7}{8}\right)^3}{\frac{4}{9}}$$

Out[35]=
$$\frac{29\ 259}{10\ 240}$$

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[36]:= **2.0¹⁰⁰**

Out[36]= 1.26765 × 10³⁰

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

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

In[37]:= $\sqrt{2}$

Out[37]= $\sqrt{2}$

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[38]:= $\sqrt{2.}$

Out[38]= 1.41421

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

In[39]:= **N[$\sqrt{2}$]**

Out[39]= 1.41421

In[40]:= $\sqrt{2} // N$

Out[40]= 1.41421

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[41]:= $N[\sqrt{2}, 20]$

Out[41]= 1.4142135623730950488

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

π	e	i	∞	$^{\circ}$
\times	\div	\times	\rightarrow	$:\rightarrow$
$==$	\neq	\leq	\geq	\in
\neg	\wedge	\vee	\cup	\cap

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	$\text{Sin}[x]$	arcoseno	$\text{ArcSin}[x]$
coseno	$\text{Cos}[x]$	arcocoseno	$\text{ArcCos}[x]$
tangente	$\text{Tan}[x]$	arcotangente	$\text{ArcTan}[x]$

Funciones hiperbólicas		Funciones hiperbólicas inversas	
seno hiperbólico	$\text{Sinh}[x]$	argumento seno hiperbólico	$\text{ArcSinh}[x]$
coseno hiperbólico	$\text{Cosh}[x]$	argumento coseno hiperbólico	$\text{ArcCosh}[x]$
tangente hiperbólica	$\text{Tanh}[x]$	argumento tangente hiperbólica	$\text{ArcTanh}[x]$

Otras funciones		Otras funciones	
Exponencial	$\text{Exp}[x]$	Logaritmo en base a de x	$\text{Log}[a, x]$
Raíz cuadrada	$\text{Sqrt}[x]$	Valor absoluto	$\text{Abs}[x]$
Logaritmo neperiano	$\text{Log}[x]$	Función signo	$\text{Sign}[x]$

Observaciones:

1) Las funciones trigonométricas utilizan el argumento en radianes.

In[42]:= $\text{Sin}\left[\frac{\pi}{2}\right]$

Out[42]= 1

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

In[43]:= $\text{Sin}[90 \text{ Degree}]$

Out[43]= 1


```
In[44]:= Sin[180 °]
```

```
Out[44]= 0
```

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, Mathematica nos devolverá un valor aproximado con una precisión de 6 dígitos (internamente trabajará con 16 dígitos).

```
In[45]:= Log[2]
```

```
Out[45]= Log[2]
```

```
In[46]:= Sin[2.6]
```

```
Out[46]= 0.515501
```

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 realizarse de dos formas: Asignación inmediata: nombredevariable=valor o expresión;

```
In[47]:= a = 4
```

```
Out[47]= 4
```

```
In[48]:= x = a
```

```
Out[48]= 4
```

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[49]:= a = 3
```

```
Out[49]= 3
```

```
In[50]:= x
```

```
Out[50]= 4
```

Asignación diferida: nombredevariable := valor o expresión

```
In[51]:= y := a
```

```
In[52]:= y
```

```
Out[52]= 3
```

```
In[53]:= a = 7
```

```
Out[53]= 7
```

```
In[54]:= y
```

```
Out[54]= 7
```

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[variable1_,variable2_,...,variablen]:=expresión.

```
In[55]:= f[x_] := x^2 - 5 x
```

```
In[56]:= f[3]
```

```
Out[56]= -6
```

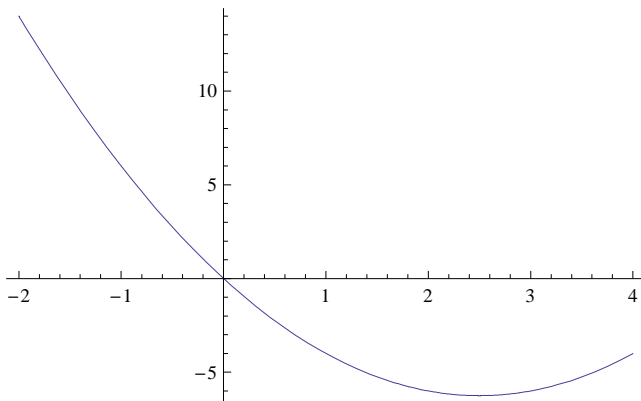
En el ejemplo anterior hemos definido la función $f(x) = x^2 - 5x$. Observemos nuevamente que el argumento tiene que indicarse entre corchetes. A partir de este momento podremos utilizar la nueva función para evaluarla en un punto, calcular su derivada, representarla gráficamente, etc.

```
In[57]:= f'[2]
```

```
Out[57]= -1
```

```
In[58]:= Plot[f[x], {x, -2, 4}]
```

```
Out[58]=
```



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[59]:= g[x_Integer] := 2 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[60]:= g[-3]
```

```
Out[60]= -5
```

```
In[61]:= g[2]
```

```
Out[61]= 5
```

```
In[62]:= g[1.5]
```

```
Out[62]= g[1.5]
```

Observaciones:

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[variable1,variable2,variable3,...]
```

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 con anterioridad.

```
In[63]:= Clear["Global`*"]
```

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

```
In[64]:= Clear[f]
```

```
In[65]:= f[x_, y_, z_] := x y + y z
```

```
In[66]:= f[1, 3, 5]
```

```
Out[66]= 18
```

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

8.- Operaciones usuales en Mathematica

1) Desarrollar una expresión: Expand[expresión]

```
In[67]:= Expand[(a + b)^4]
```

```
Out[67]= a^4 + 4 a^3 b + 6 a^2 b^2 + 4 a b^3 + b^4
```

2) Simplificar una expresión: Simplify[expresión]

```
In[68]:= Simplify[Sin[x]^2 + Cos[x]^2]
```

```
Out[68]= 1
```

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

```
Solve[expresion1==expresion2,variable]
```

Despeja la variable x en la igualdad expresión1=expresion2. 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[69]:= Solve[x + Sqrt[x + 4] == 1]
```

```
Out[69]= {{x -> 1/2 (3 - Sqrt[21])}}
```

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[70]:= `Solve[a x^2 + b x + c == 0, x]`

$$\text{Out[70]} = \left\{ \left\{ x \rightarrow \frac{-b - \sqrt{b^2 - 4 a c}}{2 a} \right\}, \left\{ x \rightarrow \frac{-b + \sqrt{b^2 - 4 a c}}{2 a} \right\} \right\}$$

Se obtiene un resultado distinto si escribimos:

In[71]:= `Solve[a x^2 + b x + c == 0, b]`

$$\text{Out[71]} = \left\{ \left\{ b \rightarrow \frac{-c - a x^2}{x} \right\} \right\}$$

En este último caso se ha despejado la variable b 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 fórmulas 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