SESIÓN 1

1. **FUNCIONES DEFINIDAS EN MATHEMATICA**
2. **FUNCIONES DEFINIDAS POR EL USUARIO**
3. **LA FUNCIÓN Plot**
4. **LA INSTRUCCIÓN Limit**
5. **INTERPOLACIÓN**

- **ELEMENTOS DEL PROGRAMA**

1. Funciones definidas en MATHEMATICA

La mayor parte de las funciones habituales del análisis se encuentran implementadas en el lenguaje de MATHEMATICA. En la siguiente lista ofrecemos una relación de estas funciones:

1. `Sin[x]` proporciona el seno del ángulo `x`.
2. `Cos[x]` proporciona el coseno del ángulo `x`.
3. `Tan[x]` proporciona la tangente del ángulo `x`.
4. `Sec[x]` proporciona la secante del ángulo `x`.
5. `Csc[x]` proporciona la cosecante del ángulo `x`.
6. `ArcCos[x]` proporciona el arcocoseno de `x`.
7. `ArcSin[x]` proporciona el arcoseno de `x`.
8. `ArcTan[x]` proporciona el arcotangente de `x`.
9. `Cosh[x]` proporciona el coseno hiperbólico de `x`.
10. `Sinh[x]` proporciona el seno hiperbólico de `x`.
11. `Sqrt[x]` proporciona la raíz cuadrada de `x`.
12. `Log[b,x]` proporciona el logaritmo en base `b` de `x`.
13. `Log[x]` proporciona el logaritmo natural, es decir, en base el número e.

Han de tenerse en cuenta los siguientes puntos:

- En las funciones trigonométricas se supone siempre que el ángulo usado como argumento está expresado en radianes.
- Con objeto de no perder exactitud MATHEMATICA no evaluará una función en la que intervengan datos exactos y que dé como resultado un dato no exacto. Así por ejemplo

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

  En este ejemplo puesto que el número `1` es un dato exacto MATHEMATICA intenta obtener un resultado también exacto para `Sin[1]` lo cual no es posible dado que sen(1) es un número con infinitas cifras decimales por lo cual el programa opta finalmente por dejar la función sin evaluar. Este problema puede resolverse fácilmente usando la función de aproximación `N`:

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

- Los símbolos utilizados por MATHEMATICA para el número e son `E` y ⅇ (esta última forma se puede introducir utilizando la paleta o tecleando $\boxed{\text{escape}}$ ee $\boxed{\text{escape}}$).
- Los símbolos utilizados por MATHEMATICA para el número π son `Pi` o π (esta última forma se puede introducir utilizando la paleta o tecleando $\boxed{\text{escape}}$ pi $\boxed{\text{escape}}$).
- El símbolo matemático ∞ se representa en MATHEMATICA mediante la palabra clave `Infinity` o el propio símbolo ∞ (que se inserta desde la paleta o tecleando $\boxed{\text{escape}}$ inf $\boxed{\text{escape}}$).
- Cuando evaluamos una expresión en la que aparecen divisiones por 0 o que tiene límite infinito el programa dará como resultado:
  - **a)** ∞ o -∞ si la función tiene en ese punto límite $+\infty$ ó $-\infty$.
  - **b)** `ComplexInfinity` si en ese punto uno de los límites laterales es $+\infty$ y el otro es $-\infty$.

2. Funciones definidas por el usuario

MATHEMATICA nos proporciona numerosas funciones implementadas en su lenguaje, pero también nos ofrece las herramientas necesarias para que podemos construir nuevas funciones a la medida de nuestras necesidades.

Veamos a continuación cómo definir una función:

Comando: Declaración de función
Sintaxis:

`nomfuncion[x_]:=expresion`

Resultado: Declara una función cuyo nombre será `nombrefuncion` que tendrá variable `x` estará definida mediante la fórmula dada por `expresion`.

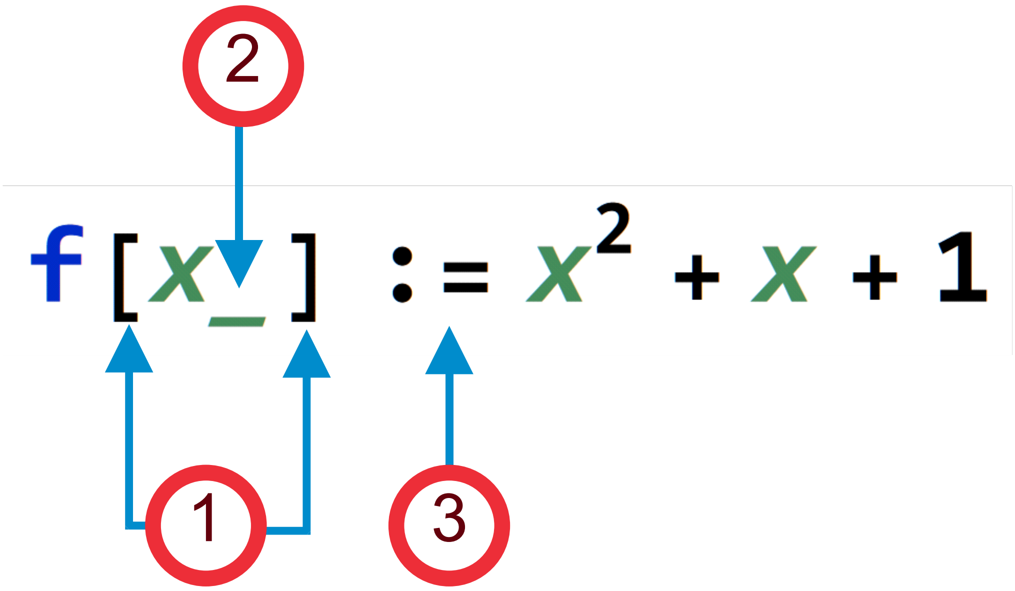
La definición o declaración de una función incluye varios elementos que no son habituales en la notación clásica y con los que, por tanto, hemos de tener especial cuidado. Estos son:

1. **Los corchetes:** Las variables de una función se encierran siempre entre corchetes y nunca entre paréntesis. Así escribiremos `f[x]` en lugar de `f(x)`.
2. **El carácter _ :** En la definición de una función debemos indicar cuáles son las variables mediante el carácter "_" (denominado blanck) que se obtiene pulsando la tecla en la que aparece el signo "–" junto con la tecla SHIFT. Debe tenerse en cuenta que debemos incluir el carácter "_" solamente en la definición y no cuando posteriormente utilicemos la función.
3. **El símbolo de := :** El símbolo "=" se emplea para la asignación de valores a una variable. Para la definición de funciones utilizaremos en su lugar el símbolo ":=" denominado "operador de asignación diferida". Su misión es similar a la del operador de asignación "=" que ya conocemos y asocia las instrucciones del cuerpo de la definición al símbolo o nombre de la función que aparece en la cabecera de la definición.

Veamos algunos ejemplos sencillos de definición de funciones:

Ejemplo 1:

Para definir la función f(x)= $x^2 + x + 1$,



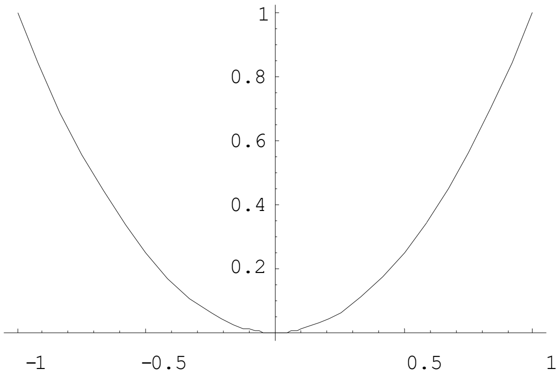
Ejemplo 2:

```
In[1]:=f[x_]:=Cos[x]-x²
In[2]:= f[0]
Out[2]= 1
```

Definimos la función:

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

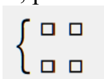
In[4]:= `Plot[f[x],{x,-1,1}];`



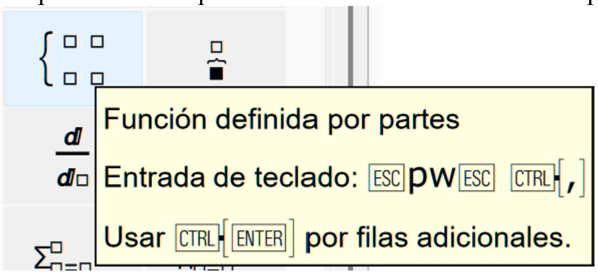
En instrucciones posterior podremos utilizar la función `f[x]` que hemos definido antes sin necesidad de volver a copiar su fórmula.

2.1. *Funciones definidas a trozos*

Para definir funciones a trozos, en la paleta "Asistente de clase", en el submenú desplegable "Composición tipográfica" que se encuentra bajando en la paleta, pulsamos el botón:



Si situamos el ratón sobre él podremos ver que la combinación de teclado correspondiente es



Es decir, pulsaremos escape pwescape para obtener el símbolo { y luego CTRL +, (CTRL juto con la coma) para insertar los cajas que contendrán la información de la función. Asimismo, tal y como vemos en el cuadró de diálogo amarillo, CTRL +ENTER, añadirá nuevas líneas dentro de las llaves.

Ejemplo 3:

Para definir la función

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

Comenzaremos tecleando la cabecera de la definición, tal y como hemos visto antes en la forma `f[x_]:=` luego escape pwescape para el símbolo { y CTRL +, para las cajas; de modo que se mostrará

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

Ahora completaremos en cada caja la información correspondiente (sin insertar los símbolos de coma ",")

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

Si la función que deseamos definir tuviera más líneas correspondientes a más trozos, simplemente pulsaríamos , CTRL +ENTER hasta obtener el número de líneas necesarias.

3. La función `Plot`

Mediante la representación gráfica de una función podemos estudiar de manera sencilla sus características detectando a simple vista los puntos en los que la función no está definida y su comportamiento en ellos. Una de las instrucciones que podemos utilizar en MATHEMATICA para obtener representaciones gráficas es la instrucción `Plot`.

Comando: `Plot`

Sintaxis:

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->{estilo1,…,estilok}`

donde `f[x]`, `f1[x]`,..., `fn[x]` son expresiones o funciones en las que únicamente aparece la variable `x`.

Resultado:

1) (primer formato) Proporciona la gráfica de la función `f[x]` para los valores de `x` que van desde el valor inicial `xi` hasta el valor final `xf`, es decir la gráfica de la función en el intervalo [`xi`,`xf`].

2) (segundo formato) Proporciona simultáneamente la gráfica de las funciónes `f1[x]`, `f2[x]`,..., `fn[x]` para `x` en [`xi`,`xf`].

3) (tercer formato) Se representa `f[x]`, apli:ando los estilos indicados. Algunas opciones son

- `RGBColor[rojo,verde,azul]`. Determina el color con el que se pintará la gráfica de la función, siendo `rojo,verde,azul` tres números entre 0 y 1 que indican la mezcla de los colores rojo, verde y azul que dará lugar al color resultante.
- `Dashing[{log1,long2,longk}]`. Permite trazar la gráfica de la función mediante trazos discontinuos cada uno de ellos de longitud `log1,long2,longk`.

4. La instrucción `Limit`

Mediante la instrucción `Limit` podemos calcular el límite de una amplia gama de funciones. Veamos ahora en el siguiente cuadro la manera de utilizar el comando `Limit`.

Comando: `Limit`
Sintaxis:

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

donde `f[x]` es una expresión en la que aparece la variable `x`.

Resultado:

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

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

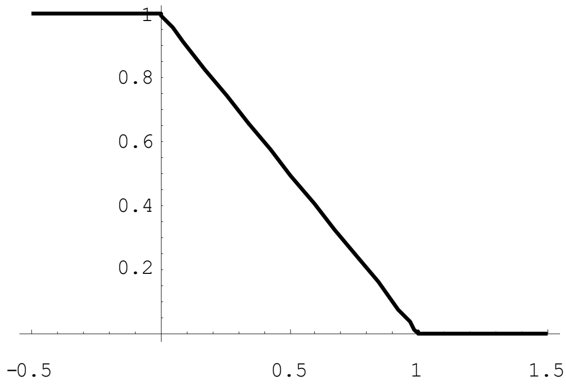
`Limit[f[x],x->x0,Direction->-1]` = $\lim_{x\to x_0^+} f(x)$

Ejemplo 4:

```
In[1]:=f[x_]:=x² eˣ;
In[2]:=Limit[f[x],x->∞]
Out[1]= 0
In[2]:=Limit[1/x,x->0,Direction->-1]
Out[2]= +∞
```

4.1. Ejercicios

1. Construir mediante dos métodos diferentes una función a trozos cuya función tenga por gráfica la siguiente:



2. Obténgase la gráfica de la función $f(x) = \begin{cases} \frac{x^5-1}{x^2+1}, & x < -1, \\ \cos(x), & -1 \le x < 0, \\ e^x, & 0 \le x < 1, \\ x^{\frac{1}{x-1}}, & 1 < x \end{cases}$

   y determínese, obteniendo la gráfica de f(x) en distintos intervalos, los puntos de discontinuidad, límites en los puntos de cambio de definición y en $\pm\infty$. Verifíquese utilizando la instrucción `Limit`[1] que los resultados obtenidos gráficamente son correctos.