

# APÉNDICE 1 : CADENAS DE CARACTERES

## ■ OPERACIONES CON CADENAS DE CARACTERES

En esta sección hacemos un repaso sobre algunas de las funciones para operar con cadenas de caracteres. La sintaxis para estas funciones es la misma que para las funciones que operan sobre listas, anteponiendo `String`

Algunas de las funciones para manipular cadenas de caracteres son las siguientes

`s1<>s2<>....` une cadenas  
`StringJoin[s1,s2,...]` une cadenas  
`StringLength[s]` da el número de caracteres de `s`  
`StringReverse[s]`

Algunas funciones para tomar trozos de cadenas

`StringTake[s,n]` toma los `n` primeros caracteres de `s`  
`StringTake[s,{n}]` toma el `n`-ésimo caracter de `s`  
`StringTake[s,{n,m}]` toma los caracteres desde el `n` hasta el `m`  
`StringDrop[s,n]` borra los `n` primeros caracteres de `s`  
`StringDrop[s,{n,m}]` borra los caracteres desde el `n` hasta el `m`

### ■Ejemplo

```
S2.7.2      alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
S2.7.2      ABCDEFGHIJKLMNOPQRSTUVWXYZ
S2.7.2      StringTake[alpha, 5]
S2.7.2      ABCDE
S2.7.2      StringTake[alpha, {5}]
S2.7.2      E
```

Podemos también insertar caracteres. Las funciones son las que utilizabamos para listas anteponiendo `String`

`StringInsert[s,c,n]` inserta `c` en `s` en la posición `n`  
`StringInsert[s,c,{n1,n2,...}]` inserta `c` en `s` en las posiciones `n1,n2,..`

Podemos también encontrar las posiciones de subcadenas en `s`

`StringPosition[s,sub]` da la posición de `sub` en `s`

### ■Ejemplo

```
S2.7.2      StringPosition["abcdabcbdaabcbabcd", "abc"]
            {{1, 3}, {5, 7}, {10, 12}, {13, 15}}
```

Podemos reemplazar partes de cadenas

**StringReplace**[s,{s1->sp1,s2->sp2,...}] reemplaza s1 por sp1,s2 pos sp2,...

Para algunas de estas funciones podemos especificar opciones:

IgnoreCase distingue entre mayúsculas y minúsculas. Su valor por defecto es False

#### ▪Ejemplo

```
StringPosition["abcdAbcdaabcabcd", "abc"]
{{1, 3}, {10, 12}, {13, 15}}
```

```
StringPosition["abcdAbcdaabcabcd", "abc", IgnoreCase -> True]
{{1, 3}, {5, 7}, {10, 12}, {13, 15}}
```

Podemos también ordenar cadenas alfabeticamente

**Sort**[{s1,s2,...}] ordena alfabeticamente s1,s2,...

#### ▪Ejemplo

```
Sort[{"hola", "Hola", "adios", "Adios"}]

{adios, Adios, hola, Hola}
```

### ■ CARACTERES ESPECIALES:NUEVA LINEA Y TABULACIÓN

**\n** se incluye una nueva linea en la cadena  
**\t** se incluye una tabulación en la cadena

#### ▪Ejemplo

```
Print["hola\nhola"]

hola
hola
```

```
Print["hola\thola"]

hola    hola
```

### ■ CODIGO ASCII

**ToCharacterCode**[string] escribe string en código ascii  
**FromCharacterCode**[{n1,n2,...}] construye la cadena a partir del código

#### ▪Ejemplo

```
ToCharacterCode["hola"]

{104, 111, 108, 97}
```

```
FromCharacterCode[%]

hola
```

```

ToCharacterCode["hola\nhola"]

{104, 111, 108, 97, 10, 104, 111, 108, 97}

FromCharacterCode[%]

hola
hola

```

## ■ LISTA DE CARACTERES DE UNA CADENA

**Characters[string]** devuelve la lista de caracteres de string  
**StringJoin[{n1,n2,...}]** construye la cadena a partir de la lista de caracteres

```

Characters["hola"]

{h, o, l, a}

StringJoin[%]

hola

```

De este modo, pasando de la cadena a la lista de caracteres, podemos usar sobre "strings" las instrucciones que actúan sobre listas. Después de la manipulación deseada, volvemos a la cadena de caracteres.

# APÉNDICE 2: GRÁFICOS 2D y 3D.

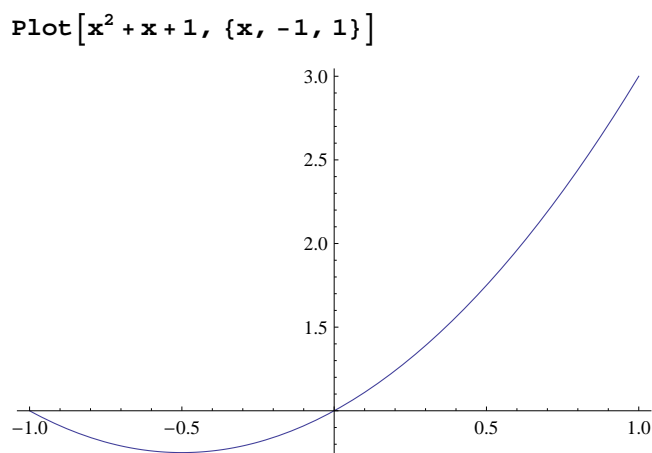
## GRÁFICAS 2D

### ■ EL COMANDO PLOT

El comando para dibujar gráficas de funciones en una variable es **Plot**. La sintaxis es la siguiente:

**Plot[f,{x,xmin,xmax}]** dibuja la función **f** que depende de **x** para los valores de **x** comprendidos entre **xmin** y **xmax**.

### ■ Ejemplo



*Mathematica* no sólo dibuja funciones acotadas, sino que también dibuja funciones con singularidades.