













4D SVG

-  Introducción
-  Atributos
-  Colores y degradados
-  Dibujo
-  Documentos
-  Estructura y definiciones
-  Filtros
-  Texto
-  Utilitarios
-  4D SVG Constants
-  Anexos
-  Lista alfabética de los comandos

✚ **Introducción**

✚ Componente 4D SVG

✚ Herramientas de desarrollo

✚ Detalles de sintaxis

Componente 4D SVG

SVG (Scalable Vector Graphics) es un formato de archivo gráfico vectorial de dos dimensiones basado en XML. 4D incluye un motor de renderización integrado que se puede utilizar para mostrar los archivos SVG.

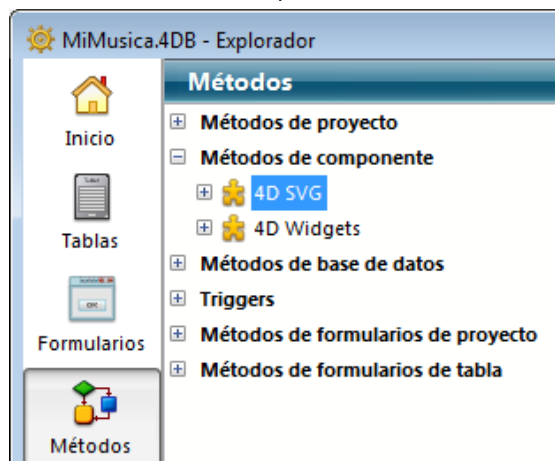
El lenguaje XML utilizado para la manipulación de imágenes SVG es particularmente rico y extenso. Con el fin de facilitarle sus primeros pasos, 4D ofrece el componente 4D SVG, que incluye numerosos comandos que se pueden utilizar para crear y manipular objetos gráficos comunes. Esta librería no pretende ser exhaustiva, sino satisfacer las necesidades más frecuentes de los desarrolladores 4D. Tenga en cuenta que las necesidades adicionales se pueden tratar con los comandos XML de 4D.

Instalación e implementación

El componente 4D SVG debe instalarse mínimo en 4D v11 SQL release 3 (versión 11.3). La base local debe funcionar en modo Unicode (el componente no puede utilizarse en las bases que funcionan en modo Compatibilidad ASCII).

Al igual que todos los componentes 4D, el componente 4D SVG se instala al copiar la carpeta del componente (4D SVG.4dbase) en la carpeta Components de la base. La carpeta Components de la base debe estar ubicada al mismo nivel que el archivo de estructura. Como los componentes se cargan al inicio, la base no debe lanzarse antes de copiar completamente todos los elementos.

Si el componente está instalado correctamente, el elemento de **4D SVG** aparece en la página Métodos de la base, en la sección "Métodos de componente":



Puede desplegar este elemento para poder ver todos los comandos del componente. Estos comandos se pueden utilizar en el editor de métodos de 4D como los comandos 4D o del plug-in estándar.

Tenga en cuenta que el componente 4D SVG permite beneficiarse de ventanas adicionales para la selección de comandos y la renderización del código SVG. Para obtener más información, consulte la sección [Herramientas de desarrollo](#).

Efectos SVG y Direct2D (Windows)

A partir de 4D v13, el motor de renderización Direct2D se utiliza por defecto bajo Windows. Dependiendo de su configuración de hardware y software, la utilización de este motor puede alterar la renderización de ciertos efectos SVG effects tales como sombras. En este caso, puede desactivar Direct2D en su aplicación con el comando **SET DATABASE PARAMETER**.

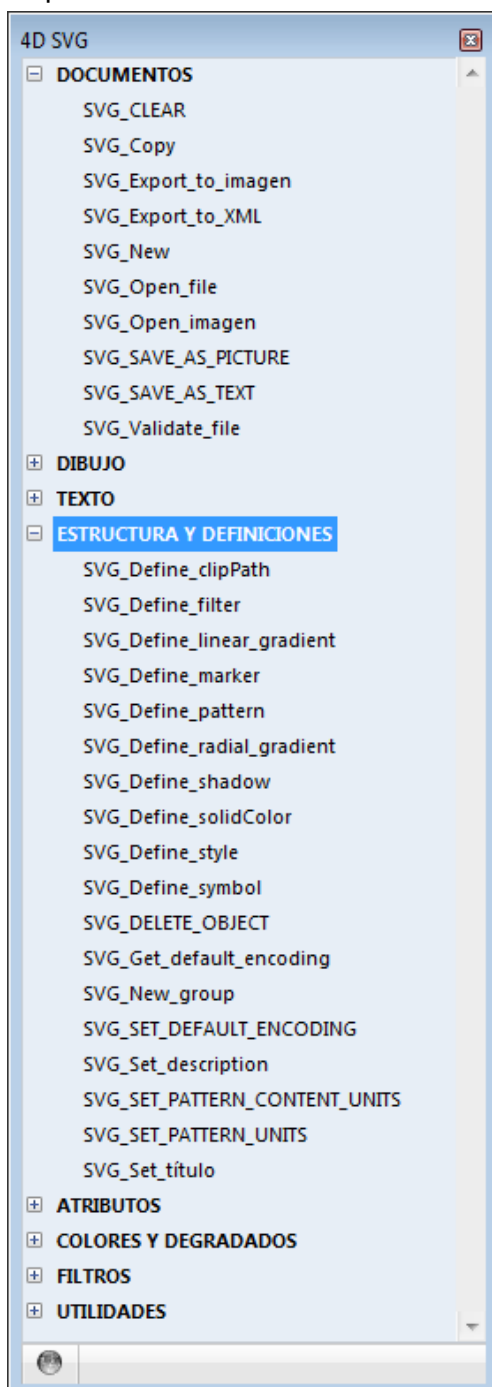
Herramientas de desarrollo

El componente 4D SVG ofrece un conjunto de herramientas para facilitar la entrada del código y la previsualización de los gráficos SVG:

- la paleta de sintaxis
- la paleta de colores
- el visualizador SVG.

Paleta de sintaxis

La paleta de sintaxis lista los comandos del componente 4D SVG agrupados por temas:



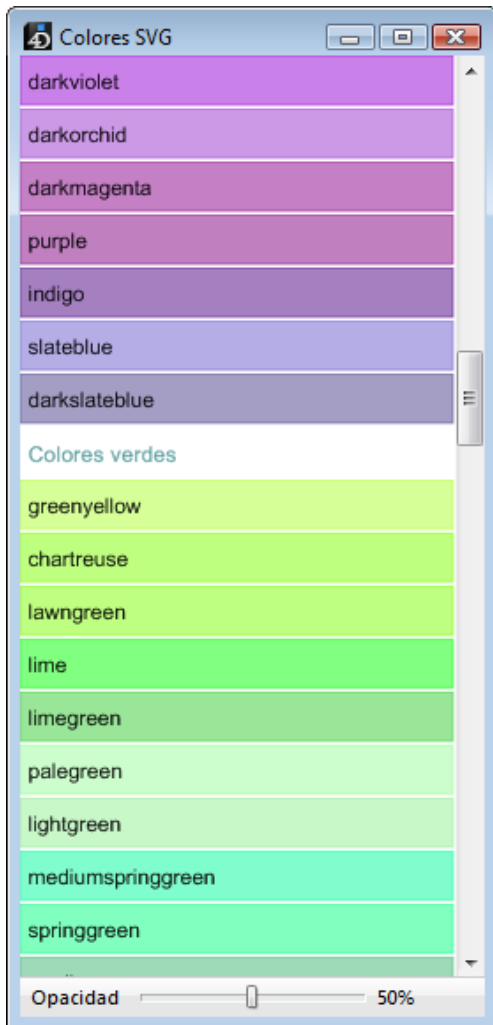
Esta paleta permite insertar los comandos del componente en el editor de métodos por arrastrar y soltar. El comando se pega a continuación en el método con sus parámetros. Los parámetros opcionales son precedidos por un guión.

Para mostrar la paleta de sintaxis, puede:

- Ejecutar el método **SVGTool_Display_syntax**, o
- Si tiene instalado 4D Pop haga clic en el botón **SVG** y elegir el comando **SVG Component syntax** en la paleta del componente 4D Pop, si lo está utilizando (ver a continuación).

Paleta de colores

La paleta de colores muestra el nombre y un ejemplo de cada color especificado en la norma SVG, así como un control deslizante que se puede utilizar para cambiar la opacidad:



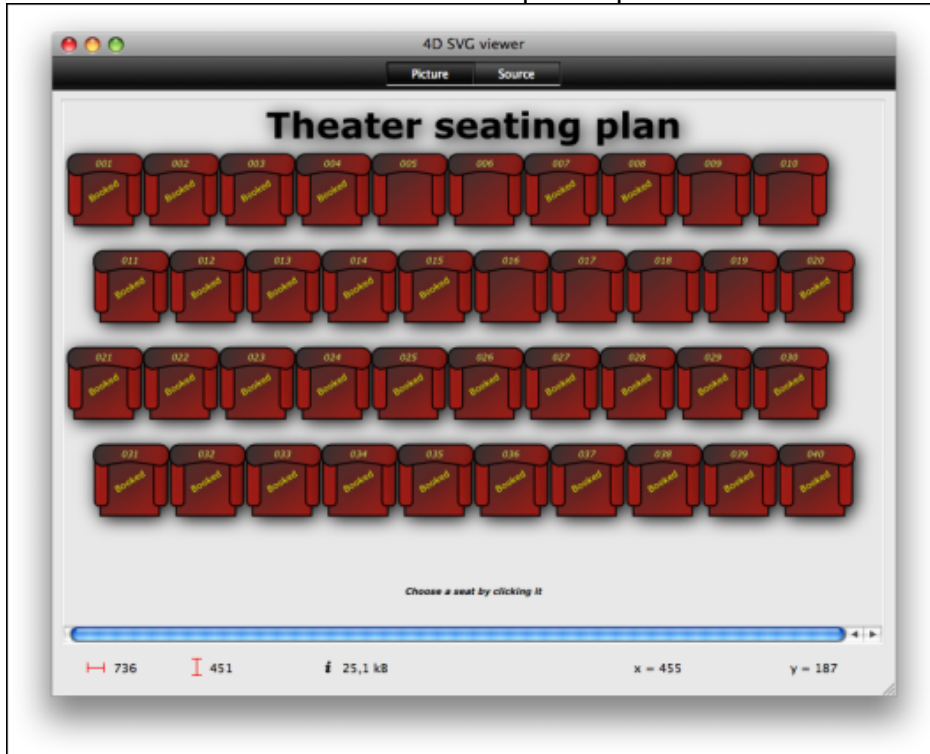
Puede utilizar esta paleta para insertar una referencia de color SVG por arrastrar y soltar en el editor de métodos de 4D. El color se inserta como una cadena que incluye el índice de opacidad (por ejemplo, "lavanda:30" para el color lavanda con una opacidad del 30%). Para mayor información sobre las referencias de colores, consulte la sección **Colores SVG**.

También puede arrastrar y soltar un color en el editor de formularios 4D. Esto crea un cuadrado de color en forma de imagen SVG estática.

Para mostrar la paleta de colores, sólo hay que ejecutar el método **SVGTool_Display_colors**.

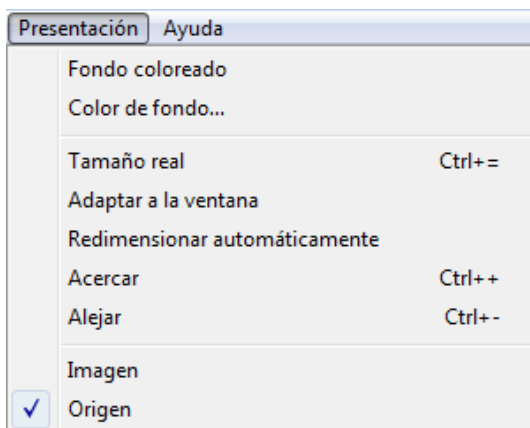
Visualizador SVG

4D SVG ofrece un visualizador SVG que es particularmente útil durante la fase de desarrollo:



La ventana del visualizador tiene dos páginas, accesibles a través de los botones **Imagen** y **Origen** o el menú **Presentación**:

- **Imagen:** esta página ofrece una zona de visualización en la que puede arrastrar y soltar o abrir un archivo imagen SVG (vía el menú **Archivo**). También puede mostrar una referencia [SVGTool_SHOW_IN_VIEWER](#).
- **Fuente:** esta página le permite ver el código XML asociado a la imagen. Puede seleccionar y copiar el código, pero no puede modificarlo. Cuando la ventana está en el primer plano, puede modificar varias opciones de visualización y guardar el archivo imagen en el disco utilizando el menú **Presentación**:



Nota: la página "imagen" tiene un menú contextual estándar.

Para mostrar la ventana de visualización, puede:

- ejecutar el método **SVGTool_Display_viewer**. En este caso, la ventana aparece vacía.
- llamar al método [SVGTool_SHOW_IN_VIEWER](#) pasando una referencia SVG válida para obtener una vista previa de la imagen de referencia (ver la descripción del comando).
- Haga clic en el botón **SVG** y elija el comando **SVG viewer** en la paleta del componente 4D Pop, si lo está utilizando (ver más abajo).

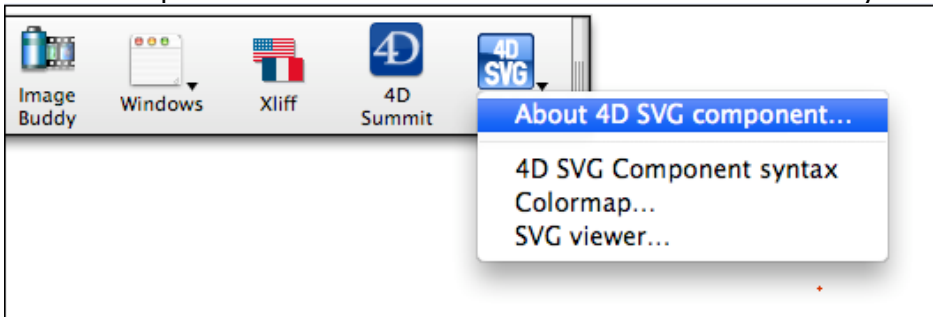
Integración a 4D Pop

4D Pop es un conjunto de componentes dedicados a la productividad del desarrollador y agrupados en una barra de herramientas que se integra en el entorno de desarrollo de 4D

Cuando se utiliza al mismo tiempo 4D Pop y 4D SVG, se agrega un nuevo botón a la barra de herramientas de 4D Pop:



Este botón permite el acceso directo a las herramientas de ayuda al desarrollo de 4D SVG:



Nota: 4DPop se incluye en las herramientas adicionales en el instalador completo de 4D.

✚ Detalles de sintaxis

SVG_Ref

La mayoría de los comandos del componente 4D SVG manipulan las estructuras a través de referencias de tipo **SVG_Ref**.

Un SVG_Ref es una expresión de tipo cadena de 16 caracteres que identifica de manera única una estructura SVG cargada en memoria. Puede ser un documento SVG cargado a través de los comandos *SVG_Copy*, *SVG_New*, *SVG_Open_picture* o *SVG_Open_file*, o de toda estructura SVG manipulada por programación (objeto, filtro, trazo, etc.).

Un SVG_Ref es una referencia XML. Todas las referencias SVG_Ref se pueden utilizar como parámetros *refElement* de los comandos XML DOM de 4D.

Una vez que no lo necesite, recuerde llamar al comando *SVG_CLEAR* con la referencia SVG_Ref con el fin de liberar memoria.



































Parámetros opcionales

A menos de que se especifique lo contrario, los argumentos numéricos opcionales se ignoran si el valor es igual a -1 y los argumentos textos se ignoran si la cadena pasada está vacía.

Coordenadas

A menos de que se especifique lo contrario, los parámetros de posición (x , y) y de dimensiones (*ancho*, *alto*, *radio*) se esperan en el sistema de coordenadas usuario actual.

Atributos

-  SVG_ADD_NAMESPACE
-  SVG_GET_ATTRIBUTES
-  SVG_Get_class
-  SVG_Get_fill_brush
-  SVG_Get_ID
-  SVG_SET_ATTRIBUTES
-  SVG_SET_ATTRIBUTES_BY_ARRAYS
-  SVG_SET_CLASS
-  SVG_SET_CLIP_PATH
-  SVG_SET_DIMENSIONS
-  SVG_SET_FILL_BRUSH
-  SVG_SET_FILL_RULE
-  SVG_SET_FILTER
-  SVG_SET_ID
-  SVG_SET_MARKER
-  SVG_SET_OPACITY
-  SVG_SET_ROUNDING_RECT
-  SVG_SET_SHAPE_RENDERING
-  SVG_SET_STROKE_BRUSH
-  SVG_SET_STROKE_DASHARRAY
-  SVG_SET_STROKE_LINECAP
-  SVG_SET_STROKE_LINEJOIN
-  SVG_SET_STROKE_MITERLIMIT
-  SVG_SET_STROKE_WIDTH
-  SVG_SET_TRANSFORM_FLIP
-  SVG_SET_TRANSFORM_MATRIX
-  SVG_SET_TRANSFORM_ROTATE
-  SVG_SET_TRANSFORM_SCALE
-  SVG_SET_TRANSFORM_SKEW
-  SVG_SET_TRANSFORM_TRANSLATE
-  SVG_SET_VIEWBOX
-  SVG_SET_VIEWPORT_FILL
-  SVG_SET_VISIBILITY
-  SVG_SET_XY

⚙️ SVG_ADD_NAMESPACE

SVG_ADD_NAMESPACE (objetoSVG ; prefijo {; URI})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia del objeto SVG
prefijo	Texto	→	Prefijo del namespace
URI	Texto	→	URI del namespace

Descripción

El método **SVG_ADD_NAMESPACE** añade un atributo XML namespace a la raíz del árbol DOM de la estructura SVG designada por el parámetro *objetoSVG*. Puede utilizar este método, más específicamente, para añadir un namespace a un fragmento de código SVG.

En *prefix*, pase una cadena con el prefijo del atributo namespace. Puede utilizar una de las siguientes constantes:

- "svgNS" para el namespace SVG estándar (<http://www.w3.org/2000/svg>)
- "xlinkNS" para el namespace estándar XLink (<http://www.w3.org/1999/xlink>)

En este caso, el parámetro *URI* no es necesario.

También puede pasar un prefijo de un namespace personalizado en el parámetro *prefijo* y su URI en el parámetro correspondiente. En este caso, el parámetro *URI* es obligatorio y si se omite, se genera un error.

Ejemplo

El siguiente código:

```
SVG_ADD_NAMESPACE($svgRef;"svgNS")
```

... añade el siguiente código a la raíz del objeto SVG:

```
<xmlns="http://www.w3.org/2000/svg">
```

⚙ SVG_GET_ATTRIBUTES

SVG_GET_ATTRIBUTES (objetoSVG ; puntArrayNoms ; puntArrayValores)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia SVG
puntArrayNoms	Puntero	→	Array cadena de los nombres de atributos
puntArrayValores	Puntero	→	Array cadena de los valores de atributos

Descripción

El comando *SVG_GET_ATTRIBUTES* llena los arrays a los que apunta *puntArrayNoms* y *puntArrayValores* respectivamente con los nombres y los valores de los atributos del elemento cuya referencia se pasa en el parámetro *objetoSVG*. Si *objetoSVG* no es válido o si este atributo no existe, se genera un error.

⚙️ SVG_Get_class

SVG_Get_class (objetoSVG {; nomClases}) -> Resultado

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia del objeto SVG
nomClases	Puntero	→	Puntero a array de nombres de clases
Resultado	Texto	↪	Nombre(s) de la(s) clase(s)

Descripción

El comando **SVG_Get_class** devuelve el nombre de la clase para una imagen SVG cuya referencia se pasa en el parámetro *objetoSVG*. Los nombres de las clases se devuelven como una cadena, con cada nombre separado por un espacio.

En el parámetro opcional *nomClases*, puede pasar un puntero a un array, cuyos elementos serán llenados con los nombres de las clases.

Ejemplo

```
// define 2 estilos
SVG_Define_style($Dom_SVG;".colored {fill: yellow; fill-opacity: 0.6; stroke: red; stroke-width: 8;
stroke-opacity: 0.6}")
SVG_Define_style($Dom_SVG;".blue {fill: blue}")

// crea un grupo y define un estilo por defecto
$Dom_g:=SVG_New_group($Dom_SVG)
SVG_SET_CLASS($Dom_g;"colored blue")

ARRAY TEXT($tTxt_Classes;0)
$tTxt_buffer:=SVG_Get_class($Dom_g;->$tTxt_classes)

// $tTxt_buffer = "colored blue"
// $tTxt_classes{1} = "colored"
// $tTxt_classes{2} = "blue"
```

⚙️ SVG_Get_fill_brush

SVG_Get_fill_brush (svgObject) -> Resultado

Parámetro	Tipo		Descripción
svgObject	Ref_SVG	→	Referencia objeto SVG
Resultado	Texto	↩	Color de relleno

Descripción

El comando **SVG_Get_fill_brush** devuelve el color de relleno de una imagen SVG cuya referencia se pasa en el parámetro *objetoSVG*. Si *objetoSVG* no tiene un atributo "fill", el comando devuelve una cadena vacía.

SVG_Get_ID

SVG_Get_ID (objetoSVG) -> Resultado

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia del elemento SVG
Resultado	Cadena	↩	Nombre del elemento

Descripción

El comando *SVG_Get_ID* devuelve el valor del atributo 'id' del elemento cuya referencia se pasa en el parámetro *objetoSVG*. Si *objetoSVG* no es válido o si este atributo no existe, se genera un error.

⚙ SVG_SET_ATTRIBUTES

SVG_SET_ATTRIBUTES (objetoSVG ; nomAtrib ; valorAtrib {; nomAtrib2 ; valorAtrib2 ; ... ; nomAtribN ; valorAtribN})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
nomAtrib	Cadena	→	Nombre del atributo a asignar
valorAtrib	Cadena	→	Valor del atributo

Descripción

El comando *SVG_SET_ATTRIBUTES* permite asignar uno o varios atributos personalizados a un objeto SVG de referencia *objetoSVG*. Si uno o más de estos atributos ya existe, sus valores se reemplazan por los pasados como parámetros.

Los atributos y sus valores se pasan como parámetros en pares.

Ejemplo

```
$svg:=SVG_New  
$object:=SVG_New_rect($svg;10;10;200;200;0;0;"black";"white";2)  
SVG_SET_ATTRIBUTES($object;"style";"fill:red; stroke:blue; stroke-width:3")
```

⚙️ SVG_SET_ATTRIBUTES_BY_ARRAYS

SVG_SET_ATTRIBUTES_BY_ARRAYS (objetoSVG ; puntArrayNoms ; puntArrayValores)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
puntArrayNoms	Puntero	→	Nombres de los atributos
puntArrayValores	Puntero	→	Valores sincronizados de los atributos

Descripción

El comando *SVG_SET_ATTRIBUTES_BY_ARRAYS* permite asignar uno o más atributos personalizados a un objeto SVG de referencia *objetoSVG*. Si uno o más de estos atributos ya existen, sus valores se sustituyen por los pasados como parámetros.

Los atributos y sus valores se pasan por intermedio de dos arrays, a los que apuntan *puntArrayNoms* y *puntArrayValores*.

Ejemplo

```
$svg:=SVG_New
$object:=SVG_New_rect($svg;10;10;200;200;0;0;"black";"white";2)
ARRAY TEXT($attributes;0)
ARRAY TEXT($values;0)
APPEND TO ARRAY($attributes;"fill")
APPEND TO ARRAY($values;"red")
APPEND TO ARRAY($attributes;"stroke")
APPEND TO ARRAY($values;"blue")
APPEND TO ARRAY($attributes;"stroke-width")
APPEND TO ARRAY($values;"3")
SVG_SET_ATTRIBUTES_BY_ARRAYS($object;->$attributes;->$values)
```


⚙ SVG_SET_CLASS

SVG_SET_CLASS (objetoSVG ; clase)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
clase	Texto	→	Nombre de la clase

Descripción

El comando *SVG_SET_CLASS* define la *clase* para el objeto pasado en *objetoSVG*. Se genera un error si *objetoSVG* no es una referencia válida.

Ver también: <http://www.w3.org/TR/SVG/styling.html#ClassAttribute>

Ejemplo

Consulte el ejemplo del comando *SVG_Define_style*.

⚙️ SVG_SET_CLIP_PATH

SVG_SET_CLIP_PATH (objetoSVG ; IDRecorte)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
IDRecorte	Texto	→	Nombre del trazo de recorte

Descripción

El comando `SVG_SET_CLIP_PATH` asigna el trazo de recorte llamado `IDRecorte` al objeto SVG designado por `objetoSVG`. Se genera un error si `objetoSVG` no es una referencia válida o si el trazo de recorte no está definido.

Ver también: <http://www.w3.org/TR/2001/REC-SVG-20010904/masking.html#EstablishingANewClippingPath>

Ejemplo 1

Definición de un trazo de recorte circular que se le asignará a una imagen:



```
//Definición de un trazo circular
$Dom_clipPath:=SVG_Define_clip_path($Dom_SVG;"theClip")
$Dom_circle:=SVG_New_circle($Dom_clipPath;150;100;100)

//Creación de un grupo
$Dom_g:=SVG_New_group($Dom_SVG)

//Inserción de una imagen
$Txt_path:=Get 4D folder(6)+"logo.svg"
READ PICTURE FILE($Txt_path;$Pic_buffer)
$Dom_picture:=SVG_New_embedded_image($Dom_g;$Pic_buffer)
SVG_SET_ID($Dom_picture;"MyPicture")

//Aplicación del recorte al grupo
SVG_SET_CLIP_PATH($Dom_g;"theClip")
```

Ejemplo 2

La misma imagen con un trazo de recorte rectangular con esquinas redondeadas:



```
//Definición de un trazo rectangular
$Dom_clipPath:=SVG_Define_clip_path($Dom_SVG;"theClip")
$Dom_rect:=SVG_New_rect($Dom_clipPath;5;10;320;240;10;10)

//Creación de un grupo
$Dom_g:=SVG_New_group($Dom_SVG)

//Inserción de una imagen
$txt_path:=Get 4D folder(6)+"logo.svg"
READ PICTURE FILE($txt_path;$Pic_buffer)
$Dom_picture:=SVG_New_embedded_image($Dom_g;$Pic_buffer)
SVG_SET_ID($Dom_picture;"MyPicture")

//Aplicación del recorte al grupo
SVG_SET_CLIP_PATH($Dom_g;"theClip")
```

⚙️ SVG_SET_DIMENSIONS

SVG_SET_DIMENSIONS (objetoSVG ; ancho {; alto {; Unidad}})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	➡	Referencia de un elemento SVG
ancho	Entero largo	➡	Dimensión en el eje X
alto	Entero largo	➡	Dimensión en el eje Y
Unidad	Cadena	➡	Unidad de medida

Descripción

El comando *SVG_SET_DIMENSIONS* permite fijar las dimensiones del objeto SVG de referencia *objetoSVG*. Si estos atributos ya existen, sus valores se sustituyen por los pasados como parámetros.

Si se pasa el parámetro *unidad*, se utilizará. Los valores esperados son: px, pt, pc, cm, mm, in, em, ex o %. Un valor *unidad* incorrecto genera un error. Si el parámetro se omite, se esperan los valores de los parámetros *ancho* y *alto* en el sistema de coordenadas usuario.

Ejemplo

```
$svg&NBSP;:=SVG_New ` Crear un nuevo documento
$object:=SVG_New_rect($svg;10;10;200;200;0;0;"black";"white";2)
SVG_SET_DIMENSIONS($object;-1;400) ` Nueva altura
```

⚙️ SVG_SET_FILL_BRUSH

SVG_SET_FILL_BRUSH (objetoSVG ; color)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
color	Cadena	→	Expresión de color

Descripción

El comando *SVG_SET_FILL_BRUSH* permite fijar el color de relleno para el objeto SVG de referencia *objetoSVG*. Si este atributo ya existe, su valor se sustituye por el valor pasado en el parámetro.

Para obtener más información sobre las referencias de colores, consulte la sección "**Colores SVG**".

Ejemplo

```
$svg:=SVG_New  
$object:=SVG_New_rect($svg;10;10;200;200;0;0;"black";"white";2)  
SVG_SET_FILL_BRUSH($object;"blue")
```

⚙️ SVG_SET_FILL_RULE

SVG_SET_FILL_RULE (objetoSVG ; modoRelleno)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
modoRelleno	Texto	→	Modo de relleno del objeto

Descripción

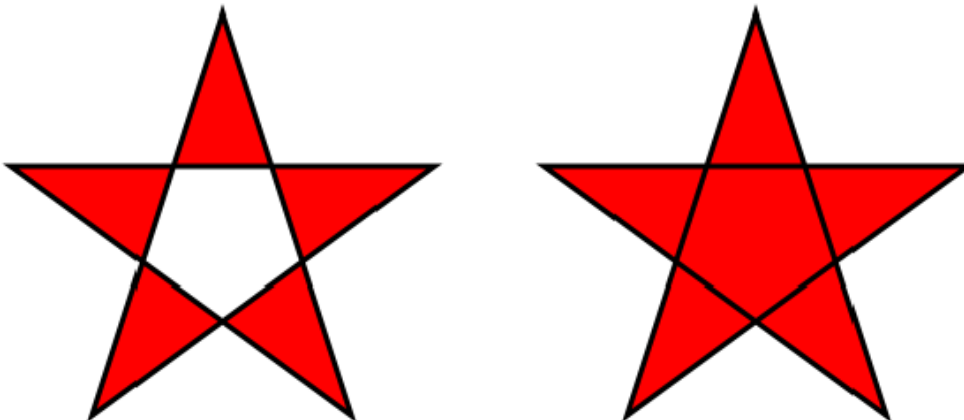
El comando `SVG_SET_FILL_RULE` permite especificar el modo de relleno del objeto SVG designado por `objetoSVG`. Se genera un error si `objetoSVG` no es una referencia válida.

El parámetro `modoRelleno` debe contener uno de los siguientes valores: "nonzero", "evenodd" o "inherit". De lo contrario, se genera un error.

Ver también: <http://www.w3.org/TR/SVG/painting.html#FillRuleProperty>

Ejemplo

Ilustración de los modos de relleno:



```
//Creación de un trazo con el modo de relleno 'evenodd'  
$Dom_path:=SVG_New_path($Dom_SVG;250;75)  
SVG_PATH_LINE_TO($Dom_path;323;301;131;161;369;161;177;301)  
SVG_PATH_CLOSE($Dom_path)  
SVG_SET_FILL_BRUSH($Dom_path;"red")  
SVG_SET_STROKE_WIDTH($Dom_path;3)  
SVG_SET_FILL_RULE($Dom_path;"evenodd")
```

```
//Creación de un objeto similar con el modo de relleno 'nonzero'  
$Dom_path:=SVG_New_path($Dom_SVG;250;75)  
SVG_PATH_LINE_TO($Dom_path;323;301;131;161;369;161;177;301)  
SVG_PATH_CLOSE($Dom_path)  
SVG_SET_FILL_BRUSH($Dom_path;"red")  
SVG_SET_STROKE_WIDTH($Dom_path;3)  
SVG_SET_FILL_RULE($Dom_path;"nonzero")  
//Movimiento horizontal  
SVG_SET_TRANSFORM_TRANSLATE($Dom_path;300)
```

⚙️ SVG_SET_FILTER

SVG_SET_FILTER (objetoSVG ; id)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
id	Cadena	→	Nombre del filtro

Descripción

El comando *SVG_SET_FILTER* permite asociar un filtro al objeto que tiene la referencia *objetoSVG*. Si *objetoSVG* no es una referencia válida, se genera un error. Si el atributo ya existe, su valor se reemplaza.

El parámetro *url* es el nombre del filtro a utilizar según lo definido por el comando *SVG_Define_filter*. Si este nombre no existe, se genera un error.

Ejemplo

Ver el ejemplo del comando *SVG_Define_filter*.

⚙️ SVG_SET_ID

SVG_SET_ID (objetoSVG ; id)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
id	Cadena	→	ID a asignar al objeto

Descripción

El comando *SVG_SET_ID* permite establecer la propiedad 'ID' del objeto SVG de referencia *objetoSVG*. Si este atributo ya existe, su valor se sustituye por el valor pasado en el parámetro.

El ID de un objeto se utiliza para referenciar un objeto. Esta referencia se recupera utilizando el comando *SVG_Get_ID*. El identificador también es utilizado por el comando 4D **SVG Find element ID by coordinates** (consulte la documentación de 4D).

Ejemplo

```
$svg:=SVG_New  
$object:=SVG_New_rect($svg;10;10;200;200;0;0;"black";" white";2)  
SVG_SET_ID($object;"border")
```


⚙️ SVG_SET_MARKER

SVG_SET_MARKER (objetoSVG ; id {; posicion})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
id	Cadena	→	Nombre del marcador
posicion	Cadena	→	Posición del marcador

Descripción

El comando **SVG_SET_MARKER** permite asociar un marcador al objeto de referencia *objetoSVG*. Si *objetoSVG* no es la referencia de un elemento 'line', 'path', 'polyline' o 'polygon', se genera un error. Si el atributo ya existe, su valor se reemplaza.

El parámetro *id* es el nombre del elemento marcador a utilizar según lo especificado por el comando **SVG_Define_marker**. Si este nombre no existe, se genera un error.

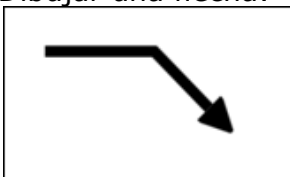
Para eliminar un marcador existente, pase la cadena "none" o una cadena vacía en el parámetro *id*.

El parámetro opcional *posicion* permite fijar la posición del marcador con respecto al objeto. Es posible poner diferentes marcadores (si se desea) al principio, final o en cualquier otro pico de un trazo. Los valores pueden ser los siguientes:

- *start* para poner un marcador al comienzo del trazo
 - *end* para poner un marcador al final del trazo
 - *middle* para poner un marcador en cada pico que no sea al principio y al final.
 - *all* para poner marcadores en todos los picos del trazo.
- Si se omite este parámetro, el marcador se pone al final del trazo.

Ejemplo 1

Dibujar una flecha:



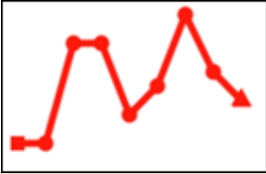
```
$SVG:=SVG_New
` Definir la flecha
$arrow:=SVG_Define_marker($SVG;"arrow";0;5;4;3;-1)
SVG_SET_VIEWBOX($arrow;0;0;10;10)
$path:=SVG_New_path($arrow;0;0)
SVG_SET_FILL_BRUSH($path;"black")
SVG_PATH_LINE_TO($path;10;5)
SVG_PATH_LINE_TO($path;0;10)
SVG_PATH_CLOSE($path)

$line:=SVG_New_path($SVG;100;75)
SVG_SET_STROKE_WIDTH($line;10)
SVG_PATH_LINE_TO($line;200;75)
SVG_PATH_LINE_TO($line;250;125)
```

```
`Poner una flecha al final del trazo
SVG_SET_MARKER($line;" arrow ")
```

Ejemplo 2

Dibujar un diagrama con marcadores diferentes al inicio y al final:



```
$SVG:=SVG_New
SVG_SET_DEFAULT_BRUSHES("red";"red")

`Definir un círculo para marcar los puntos
$point:=SVG_Define_marker($SVG;"pointMarker";2;2;3;3)
SVG_SET_VIEWBOX($point;0;0;4;4)
SVG_New_circle($point;2;2;1)

`Definir un cuadrado para el punto de inicio
$start:=SVG_Define_marker($SVG;"startMarker";1;1;2;2)
SVG_New_rect($start;0;0;2;2)

`Definir un triángulo para el punto final
$end:=SVG_Define_marker($SVG;"endMarker";5;5;3;3;60)
SVG_SET_VIEWBOX($end;0;0;10;10)
SVG_New_regular_polygon($end;10;3)

ARRAY LONGINT($tX;0)
ARRAY LONGINT($tY;0)
`Eje de las X
For($Lon_i;0;200;20)
  APPEND TO ARRAY($tX;$Lon_i+10)
End for
`Datos
APPEND TO ARRAY($tY;100)
APPEND TO ARRAY($tY;100)
APPEND TO ARRAY($tY;30)
APPEND TO ARRAY($tY;30)
APPEND TO ARRAY($tY;80)
APPEND TO ARRAY($tY;60)
APPEND TO ARRAY($tY;10)
APPEND TO ARRAY($tY;40)
APPEND TO ARRAY($tY;50)
APPEND TO ARRAY($tY;70)
$line:=SVG_New_polyline_by_arrays($SVG;->$tX;->$tY;"red";"none";5)
`Ordenar los marcadores:
SVG_SET_MARKER($line;"startMarker";"start")
SVG_SET_MARKER($line;"pointMarker";"middle")
SVG_SET_MARKER($line;"endMarker";"end")
```

⚙️ SVG_SET_OPACITY

SVG_SET_OPACITY (objetoSVG ; opacidadFondo {; grueso})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
opacidadFondo	Entero largo	→	Opacidad (%)
grueso	Entero largo	→	Opacidad (%)

Descripción

El comando **SVG_SET_OPACITY** permite definir la opacidad del relleno y la de la línea del objeto de referencia *objetoSVG*. Si estos atributos ya existen, sus valores se sustituyen por los pasados como parámetros.

Los valores esperados deben estar entre 0 y 100.

Ejemplo

```
$svg:=SVG_New `Crear un nuevo documento
$object:=SVG_New_rect($svg;10;10;200;100;0;0;"red";"blue")
SVG_SET_OPACITY($object;-1;50)&NBSP;&NBSP; `Fijar la opacidad del trazo en 50%
```

⚙️ SVG_SET_ROUNDING_RECT

SVG_SET_ROUNDING_RECT (objetoSVG ; redondX {; redondY})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	➡	Referencia de un elemento SVG
redondX	Entero largo	➡	Radio en el eje X
redondY	Entero largo	➡	Radio en eje Y

Descripción

El comando *SVG_SET_ROUNDING_RECT* permite fijar los radios de la elipse utilizada para redondear las esquinas de un rectángulo de referencia *objetoSVG*. Si estos atributos ya existen, sus valores se sustituyen por los pasados como parámetros. Si *objetoSVG* no es la referencia de un rectángulo, se genera un error.

Los valores se esperan en el sistema de coordenadas del usuario.

Ejemplo

```
$svg&NBSP;:=SVG_New ` Crear un nuevo documento  
$object:=SVG_New_rect($svg&NBSP;;10;10;200;100)  
SVG_SET_ROUNDING_RECT($object;20) ` Redondear las esquinas
```

⚙️ SVG_SET_SHAPE_RENDERING

SVG_SET_SHAPE_RENDERING (objetoSVG ; rendering)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
rendering	Texto	→	Tipo de renderización

Descripción

El comando *SVG_SET_SHAPE_RENDERING* permite fijar los compromisos que deben hacerse para la renderización de los elementos gráficos del objeto designado por *objetoSVG*. Si *objetoSVG* no es un objeto SVG, se genera un error.

El parámetro *renderizacion* debe contener uno de los siguientes valores: "auto", "optimizeSpeed", "crispEdges", "geometricPrecision" o "inherit". De lo contrario, se genera un error.

Ver también: <http://www.w3.org/TR/2001/REC-SVG-20010904/painting.html#ShapeRenderingProperty>

⚙️ SVG_SET_STROKE_BRUSH

SVG_SET_STROKE_BRUSH (objetoSVG ; color)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
color	Cadena	→	Expresión de color

Descripción

El comando *SVG_SET_STROKE_BRUSH* permite fijar el color utilizado para las líneas del objeto SVG de referencia *objetoSVG*. Si este atributo ya existe, su valor se sustituye por el valor pasado en parámetro.

Para más información sobre los colores, consulte la sección "[Colores SVG](#)".

Ejemplo

```
$svg:=SVG_New  
$object:=SVG_New_rect($svg;10;10;200;200;0;0;"black";"white";2)  
SVG_SET_STROKE_BRUSH($object;"red")
```

⚙ SVG_SET_STROKE_DASHARRAY

SVG_SET_STROKE_DASHARRAY (objetoSVG ; raya {; valor}{; valor2 ; ... ; valorN})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
raya	Real	→	Longitud de la primera raya
valor	Entero largo	→	Longitud de espacios y de rayas

Descripción

El comando *SVG_SET_STROKE_DASHARRAY* permite definir el patrón de guiones y espacios utilizados para delinear el trazo del objeto SVG designado por *objetoSVG*. Si *svgObject* no es una referencia SVG válida, se genera un error.

El valor total del parámetro *raya* indica la longitud de la primera raya del patrón de puntos. Si se omiten los parámetros *valor*, la línea punteada consistirá en una serie de guiones y espacios de la misma longitud.

El valor decimal del parámetro *raya*, si no es nulo, indica la distancia a partir de la cual comenzar las rayas.

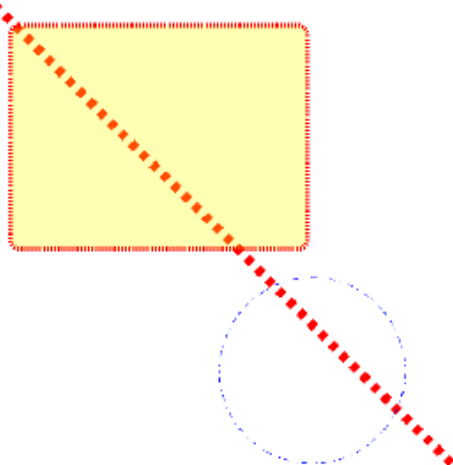
Si *raya* vale 0, el patrón de puntos se elimina.

Los parámetros *valor* especifican alternadamente la longitud de los espacios y las rayas que siguen la primera raya. Si se da un número impar de valores (incluyendo la primera raya), la lista de valores se repite hasta que se produce un número par de valores.

Ver también: <http://www.w3.org/TR/SVG/painting.html#StrokeProperties>

Ejemplo

Ilustraciones de una línea punteada:



```
//Línea
$Dom_line:=SVG_New_line($Dom_SVG;10;10;500;500)
SVG_SET_STROKE_WIDTH($Dom_line;10)
SVG_SET_STROKE_DASHARRAY($Dom_line;8,099)
SVG_SET_STROKE_BRUSH($Dom_line;"red")
```

```
//Rectángulo
```

```
$Dom_rect:=SVG_New_rect($Dom_SVG;25;30;320;240;10;10;"red";"yellow:30")
```

```
SVG_SET_STROKE_WIDTH($Dom_rect;5)
```

```
SVG_SET_STROKE_DASHARRAY($Dom_rect;2)
```

```
//Círculo
```

```
$Dom_circle:=SVG_New_circle($Dom_SVG;350;400;100;"blue";"none")
```

```
SVG_SET_STROKE_DASHARRAY($Dom_circle;2;4;6;8)
```


⚙️ SVG_SET_STROKE_LINECAP

SVG_SET_STROKE_LINECAP (objetoSVG ; modo)

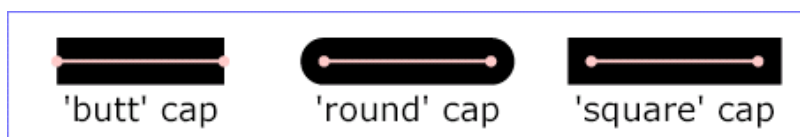
Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
modo	Cadena	→	Modo de renderización

Descripción

El comando `SVG_SET_STROKE_LINECAP` permite especificar la forma de los extremos del trazo del objeto SVG de referencia `objetoSVG`. Si este atributo ya existe, su valor se sustituye por el valor pasado como parámetro.

El parámetro `modo` debe contener una de las siguientes cadenas, gestionadas por el SVG:

- `butt` (por defecto): estándar
- `round`
- `square`
- `inherit`: heredar del objeto padre



Si el parámetro `modo` contiene cualquier otro valor, se genera un error.

⚙ SVG_SET_STROKE_LINEJOIN

SVG_SET_STROKE_LINEJOIN (objetoSVG ; modo)

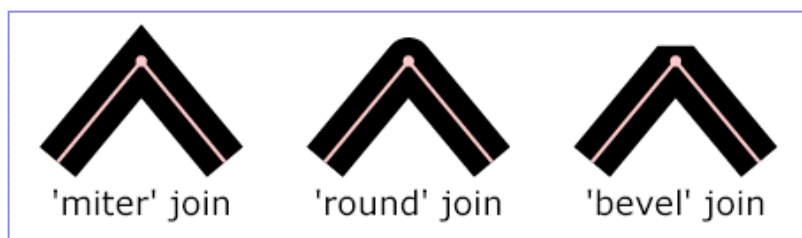
Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
modo	Cadena	→	Modo de renderización

Descripción

El comando `SVG_SET_STROKE_LINEJOIN` permite especificar la forma de los picos de los trazos del objeto SVG de referencia `objetoSVG`. Si este atributo ya existe, su valor se sustituye por el valor pasado como parámetro.

El parámetro `modo` debe contener uno de los siguientes valores, gestionado por SVG:

- `miter` (por defecto): estándar
- `round`
- `bevel`
- `inherit`: heredar del objeto padre



Si el parámetro `modo` contiene cualquier otro valor, se genera un error.

⚙ SVG_SET_STROKE_MITERLIMIT

SVG_SET_STROKE_MITERLIMIT (objetoSVG ; union)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
union	Entero largo	→	Valor de unión

Descripción

El comando `SVG_SET_STROKE_MITERLIMIT` permite definir el límite de la longitud de la unión en ángulo entre el trazo y el perfil del objeto SVG designado por *objetoSVG*. Si *objetoSVG* no es una referencia válida SVG, se genera un error.

Si el parámetro *union* es igual a `-1`, el valor será el valor por defecto (4). Si el parámetro *union* es igual a `0`, entonces la definición del atributo se elimina. Todo valor `<0`, provocará un error.

Ver también: <http://www.w3.org/TR/SVG/painting.html#StrokeProperties>

⚙️ SVG_SET_STROKE_WIDTH

SVG_SET_STROKE_WIDTH (objetoSVG ; anchoTrazo {; Unidad})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
anchoTrazo	Real	→	Ancho de línea
Unidad	Cadena	→	Unidad de medida

Descripción

El comando *SVG_SET_STROKE_WIDTH* permite establecer el grosor de las líneas del objeto SVG de referencia *objetoSVG*. Si este atributo ya existe, su valor se sustituye por el valor pasado en parámetro.

Pase en *anchoTrazo* el valor del grosor de las líneas. El parámetro opcional *unidad* permite especificar la unidad utilizada. Puede pasar uno de los siguientes valores: px, pt, pc, cm, mm, in, em, ex o %. Si se omite el parámetro *unidad*, el parámetro *anchoTrazo* se espera en el sistema de coordenadas del usuario.

Ejemplo

```
$svg:=SVG_New  
SVG_SET_STROKE_WIDTH(SVG_New_rect($svg;10;10;200;200;0;0;"black";"white";2);10)
```

⚙️ SVG_SET_TRANSFORM_FLIP

SVG_SET_TRANSFORM_FLIP (objetoSVG ; horizontal {; vertical})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	➡	Referencia de un elemento SVG
horizontal	Booleano	➡	Voltear horizontal
vertical	Booleano	➡	Voltear vertical

Descripción

El comando `SVG_SET_TRANSFORM_FLIP` permite aplicar un giro horizontal y/o vertical al objeto SVG de referencia `objetoSVG`.

Si el parámetro `horizontal` es igual a `True`, se aplica un giro *horizontal*.

Si el parámetro `vertical` es igual a `True`, se aplica un giro *vertical*.

Ejemplo

Efecto de girar un objeto texto:



```
svgRef:=SVG_New
SVG_SET_VIEWBOX(svgRef;0;0;400;200)
$tx:=SVG_New_text(svgRef;"4D";10;0;"";96)
SVG_SET_FONT_COLOR($tx;"blue") `Cambiar el color
```

`Efecto:

```
$tx:=SVG_New_text(svgRef;"4D";10;0;"";96) `Tomar el mismo texto
SVG_SET_FONT_COLOR($tx;"lightblue") `Cambiar el color
SVG_SET_TRANSFORM_FLIP($tx;Vrai) `Aplicar un giro vertical
SVG_SET_TRANSFORM_SKEW($tx;-10) `Inclinación
SVG_SET_TRANSFORM_TRANSLATE($tx;-17;-193) `Reposicionar
```

⚙️ SVG_SET_TRANSFORM_MATRIX

SVG_SET_TRANSFORM_MATRIX (objetoSVG ; a ; b { ; c ; d { ; e ; f } })

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
a	Entero largo	→	Elemento a de la matriz de transformación
b	Entero largo	→	Elemento b de la matriz de transformación
c	Entero largo	→	Elemento c de la matriz de transformación
d	Entero largo	→	Elemento d de la matriz de transformación
e	Entero largo	→	Elemento e de la matriz de transformación
f	Entero largo	→	Elemento f de la matriz de transformación

Descripción

El comando `SVG_SET_TRANSFORM_MATRIX` aplica una transformación matricial al objeto SVG de referencia `objetoSVG`.

Este tipo de transformación permite combinar las transformaciones tales como, por ejemplo, una rotación y una traducción.

Ejemplo

Writing with SVG is easy

```
SVG_SET_TRANSFORM_MATRIX($ID;0,707;-0,707;0,707;0,707;255,03;111,21)
```

` Equivale a aplicar las 3 transformaciones siguientes:

```
SVG_SET_TRANSFORM_TRANSLATE($ID;50;90)
```

```
SVG_SET_TRANSFORM_ROTATE($ID;-45)
```

```
SVG_SET_TRANSFORM_TRANSLATE($ID;130;160)
```

⚙️ SVG_SET_TRANSFORM_ROTATE

SVG_SET_TRANSFORM_ROTATE (objetoSVG ; angulo {; x ; y})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
angulo	Entero largo	→	Angulo de rotación
x	Entero largo	→	Coordenada del centro de rotación en el eje X
y	Entero largo	→	Coordenada del centro de rotación en el eje Y

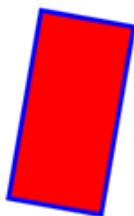
Descripción

El comando `SVG_SET_TRANSFORM_ROTATE` aplica una rotación del valor *angulo* al objeto SVG de referencia *objetoSVG*.

El ángulo de rotación que se espera en grados, la rotación se hace hacia la derecha.

Si no se pasan los parámetros opcionales *x* y *y*, la rotación se efectúa con respecto al origen del sistema de coordenadas usuario actual. Si estos parámetros se pasan, la rotación se efectúa con respecto a las coordenadas pasadas (*x*, *y*).

Ejemplo



```
svgRef:=SVG_New
`Dibujar un rectángulo rojo con un borde azul
$rec:=SVG_New_rect($svg;150;50;200;400;0;0;"blue";"red";10)
`Aplicar una rotación de 10° en el sentido de las manecillas del reloj con respecto al centro
SVG_SET_TRANSFORM_ROTATE($rec;370;175;225)
```

⚙ SVG_SET_TRANSFORM_SCALE

SVG_SET_TRANSFORM_SCALE (objetoSVG ; escalaX {; escalaY})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
escalaX	Real	→	Valor en el eje X
escalaY	Real	→	Valor en el eje Y

Descripción

El comando *SVG_SET_TRANSFORM_SCALE* aplica un cambio de escala horizontal y/o vertical a un objeto SVG de referencia *objetoSVG*.

Si el valor *escalaX* no es nulo, el objeto se agranda (valor > 1) o disminuye (0 < valor < 1) horizontalmente el número de unidades pasadas. El valor igual a 1 no cambia la escala del objeto.

Si se pasa el parámetro *escalaY*, el objeto se agranda (valor > 1) o disminuye (0 < valor < 1) verticalmente el número de unidades pasadas. El valor igual a 1 no cambia la escala del objeto. Si se omite este parámetro, su valor se supone que es igual a *escalaX*.

Ejemplo



```
$SVG:=SVG_New  
$Text:=SVG_New_text($SVG;"Hello world!";5)  
SVG_SET_TRANSFORM_SCALE($Text;3;12) `Zoom x*3 y*12
```


⚙️ SVG_SET_TRANSFORM_SKEW

SVG_SET_TRANSFORM_SKEW (objetoSVG ; horizontal {; vertical})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
horizontal	Entero largo	→	Valor de inclinación a lo largo del eje X
vertical	Entero largo	→	Valor de inclinación a lo largo del eje Y

Descripción

El comando *SVG_SET_TRANSFORM_SKEW* especifica una inclinación horizontal y/o vertical del objeto SVG de referencia *objetoSVG*.

Si el valor de *horizontal* no es nulo, el objeto se inclina horizontalmente en función del número de unidades pasadas, de lo contrario, se ignora.

Si el valor de *vertical* no es nulo, el objeto se inclina verticalmente en función del número de unidades pasadas.

Ejemplo



```
$svg :=SVG_New
  `Dibujo del fondo
SVG_New_rect($svg;0;0;270;160;10;10;"black";"gray")
  `Poner el texto...
$tx:=SVG_New_text($svg;"Hello world!";100;5;"";48)
  `en blanco
SVG_SET_FONT_COLOR($tx;"white")
  `Inclinación
SVG_SET_TRANSFORM_SKEW($tx;-50;10)
```

⚙ SVG_SET_TRANSFORM_TRANSLATE

SVG_SET_TRANSFORM_TRANSLATE (objetoSVG ; x {; y})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
x	Entero largo	→	Coordenada en el eje X
y	Entero largo	→	Coordenada en el eje Y

Descripción

El comando *SVG_SET_TRANSFORM_TRANSLATE* especifica una traducción horizontal y/o vertical del objeto SVG de referencia *objetoSVG*.

Si el valor de *x* no es nulo, el objeto se moverá horizontalmente el número de unidades pasadas, de lo contrario, se ignorará.

Si se pasa el parámetro *y*, el objeto se moverá verticalmente el número de unidades pasadas.

Ejemplo



```
svgRef:=SVG_New
` Dibujar un rectángulo rojo
$Object:=SVG_New_rect(svgRef;0;0;200;100;0;0;"black";"red")
` Dibujar un cuadrado en 0,0
$Object:=SVG_New_rect(svgRef;0;0;20;20)
` Mover el cuadrado a 150,50
SVG_SET_TRANSFORM_TRANSLATE($Object;150;50)
```

⚙️ SVG_SET_VIEWBOX

SVG_SET_VIEWBOX (objetoSVG ; x ; y ; ancho ; alto {; modo})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
x	Real	→	Posición X del rectángulo de visualización
y	Real	→	Posición Y del rectángulo de visualización
ancho	Real	→	Ancho del rectángulo de visualización
alto	Real	→	Altura del rectángulo de visualización
modo	Texto	→	Adaptación al rectángulo de visualización

Descripción

El comando *SVG_SET_VIEWBOX* permite definir el rectángulo de visualización del objeto SVG de referencia *objetoSVG*. Si este atributo ya existe, su valor se sustituye por el valor pasado en parámetro.

Los valores se esperan en el sistema de coordenadas usuario.

El parámetro opcional *modo* permite indicar si el gráfico debe adaptarse al tamaño del rectángulo de visualización. El valor esperado para *modo* debe ser reconocido por SVG: `'none'`, `'xMinYMin'`, `'xMidYMin'`, `'xMaxYMin'`, `'xMinYMid'`, `'xMidYMid'`, `'xMaxYMid'`, `'xMinYMax'`, `'xMidYMax'`, `'xMaxYMax'` y `'true'` (para `xMidYMid`).

Ejemplo

```
` Crear un documento SVG de 4x8cm
$svg:=SVG_New
SVG_SET_DIMENSIONS($SVG;4;8;"cm")
` Declarar el sistema de coordenadas usuario como 1 cm = 250 puntos usuario
SVG_SET_VIEWBOX($svg;0;0;1000;2000;"true")
```

⚙️ SVG_SET_VIEWPORT_FILL

SVG_SET_VIEWPORT_FILL (objetoSVG {; color {; opacidad}})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
color	Cadena	→	Color de relleno
opacidad	Entero largo	→	Porcentaje de opacidad

Descripción

El comando *SVG_SET_VIEWPORT_FILL* permite fijar el color de fondo de un documento SVG de referencia *objetoSVG*.

Si este atributo ya existe, su valor se sustituye por el valor pasado como parámetro. Si *objetoSVG* es un elemento SVG que no acepta este atributo, se genera un error.

El parámetro opcional *color* indica el color a utilizar para el fondo de la imagen. Si este parámetro se omite o contiene una cadena vacía, se utiliza el blanco. Para más información sobre colores, consulte la sección **Colores SVG**.

El parámetro opcional *opacidad* puede usarse para precisar el valor del porcentaje de opacidad a aplicar a este relleno. Si este parámetro se omite o si no se ha especificado opacidad para el documento, se utiliza el valor del 100%.

⚙️ SVG_SET_VISIBILITY

SVG_SET_VISIBILITY (objetoSVG {; ocultar})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
ocultar	Booleano	→	True = Mostrar, False = Ocultar

Descripción

El comando `SVG_SET_VISIBILITY` esconde o muestra el objeto SVG de referencia `objetoSVG`. Si `objetoSVG` no es la referencia de un objeto que se puede ocultar, se genera un error.

Si el parámetro opcional `ocultar` es True o se omite, el objeto se muestra. Si es False, el objeto se oculta.

Ejemplo

```
$svg&NBSP;:=SVG_New  
$object:=SVG_New_rect($svg;10;10;200;200;0;0;"black";" white";2)  
SVG_SET_VISIBILITY($object;False) ` El objeto se describe pero no será renderizado.
```

⚙ SVG_SET_XY

SVG_SET_XY (objetoSVG ; x {; y})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
x	Entero largo	→	Coordenada en el eje X
y	Entero largo	→	Coordenada en el eje Y

Descripción

El comando *SVG_SET_XY* permite fijar las coordenadas de la esquina superior izquierda del área rectangular donde el objeto SVG de referencia *objetoSVG* se ubica. Si estos atributos ya existen, sus valores se sustituyen por los pasados como parámetros. Si *objetoSVG* es un elemento SVG que no acepta este atributo, se genera un error.

Los valores se esperan en el sistema de coordenadas del usuario.

Ejemplo

```
$svg&NBSP;:=SVG_New `Crear un nuevo documento
$object:=SVG_New_image($svg;"#Pictures/logo4D.png") `Poner el logo
SVG_SET_XY($object;10;40) `Modificar la posición de la imagen
```

Colores y degradados

Colores SVG

-  SVG_Color_from_index
-  SVG_Color_grey
-  SVG_Color_RGB_from_CMYK
-  SVG_Color_RGB_from_HLS
-  SVG_Color_RGB_from_long
-  SVG_FADE_TO_GREY_SCALE
-  SVG_Filter_ColorMatrix
-  SVG_GET_COLORS_ARRAY
-  SVG_GET_DEFAULT_BRUSHES
-  SVG_Get_named_color_value
-  SVG_SET_BRIGHTNESS
-  SVG_SET_DEFAULT_BRUSHES
-  SVG_SET_HUE
-  SVG_SET_SATURATION

Definición de colores

SVG reconoce todas las sintaxis alternativas para los colores definidos en la norma CSS2. Los comandos del componente 4D SVG soportan todas estas sintaxis.

Un color puede expresarse en una de las formas siguientes:

- Formato RGB

Formato	Ejemplo
#rgb	#f00
#rrggbb	#ff0000
rgb(r,g,b)	rgb(255,0,0) rgb(100%, 0%, 0%)

- Formato clave "color"

SVG acepta una extensa lista de palabras claves de nombres de color, por ejemplo "red".

La lista de palabras claves, así como sus equivalentes RGB se encuentra en el [4D SVG Constants](#): y . También puede ver esta lista e insertar directamente los valores de colores vía la paleta de colores de 4D SVG. Para más información acerca de este punto, consulte [Herramientas de desarrollo](#)**Herramientas de desarrollo.**

None

Pase la palabra clave "**none**" en los parámetros *colorPrimerPlano* o *colorFondo* para no tener un relleno o un trazo en absoluto. La palabra clave "ninguno" se puede usar con la mayoría de los comandos SVG.

Opacidad

Es posible especificar la opacidad en las expresiones de color de los comandos del componente utilizando la sintaxis "color:opacidad", donde la opacidad es un número comprendido entre 0 (sin color) y 100 (color totalmente opaco. Por lo tanto "red:50" se interpretará como un rojo en 50% de opacidad.

Degradados

Los degradados son transiciones progresivas de color a lo largo de un vector. Estos degradados se definen con los comandos *SVG_Define_linear_gradient* y *SVG_Define_radial_gradient*. Una vez definidos, los gradientes se utilizan por referencia utilizando la sintaxis "url (#NombreDegradado)".

Del mismo modo, es posible definir un color personalizado asociado a una opacidad utilizando el comando *SVG_Define_solidColor*.

⚙️ SVG_Color_from_index

SVG_Color_from_index (indice) -> Resultado

Parámetro	Tipo		Descripción
indice	Entero largo	→	Número de color
Resultado	Campo Texto	↩	Color designado por índice

Descripción

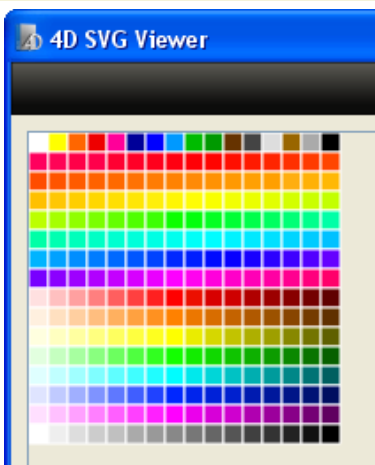
El método **SVG_Color_from_index** devuelve el color SVG correspondiente al color 4D definido en el parámetro *indice*.

El parámetro *indice* designa un número en la paleta de colores de 4D, donde los colores están numerados de 1 a 256. Para mayor información sobre este punto, consulte la descripción del comando de 4D **OBJECT SET COLOR**.

Ejemplo

En este ejemplo, recreamos la paleta de colores de 4D:

```
$Dom_svg:=SVG_New
$Lon_line:=0
For($Lon_ii;0;15;1)
  $Lon_column:=0
  For($Lon_i;1;16;1)
    $Txt_color:=SVG_Color_from_index(($Lon_ii*16)+$Lon_i)
    $Dom_rect:=SVG_New_rect($Dom_svg;$Lon_column;
    $Lon_line;11;11;0;0;"white";$Txt_color)
    $Lon_column:=$Lon_column+11
  End for
  $Lon_line:=$Lon_line+11
End for
SVGTool_SHOW_IN_VIEWER($Dom_svg)
```



⚙ SVG_Color_grey

SVG_Color_grey (porcentaje) -> Resultado

Parámetro	Tipo		Descripción
porcentaje	Entero	→	Intensidad del gris
Resultado	Cadena	↩	Cadena color

Descripción

El comando *SVG_Color_grey* devuelve una cadena que expresa un color gris con una intensidad *porcentaje*. La cadena devuelta es de la forma "RGB (rojo, verde, azul)" donde los 3 valores son iguales, sintaxis reconocida por los motores de renderización SVG.

Ejemplo

```
$txtColor:=SVG_Color_grey(60)
`$txtColor es igual a "#666666"
```

⚙ SVG_Color_RGB_from_CMYK

SVG_Color_RGB_from_CMYK (cian ; magenta ; amarillo ; negro {; formato}) -> Resultado

Parámetro	Tipo		Descripción
cian	Entero largo	→	Valor de cian
magenta	Entero largo	→	Valor de magenta
amarillo	Entero largo	→	Valor de amarillo
negro	Entero largo	→	Valor de negro
formato	Entero largo	→	Formato del color
Resultado	Texto	↩	Cadena color

Descripción

El comando *SVG_Color_RGB_from_CMYK* devuelve una cadena que expresa el color correspondiente a los cuatro parámetros de color, *cian*, *magenta*, *amarillo* y *negro*, pasados como argumentos. La cadena se devuelve por defecto de la forma "RGB (rojo, verde, azul)", sintaxis reconocida por los motores de renderización SVG.

cian, *magenta*, *amarillo* y *negro*, son enteros largos entre 0 y 100%.

El parámetro opcional *formato* permite especificar el formato deseado para la cadena color devuelta. Los valores son:

Valor	Formato
1 (por defecto)	rgb(r,g,b)
2	#rgb
3	#rrggbb
4	rgb(r%, g%, b%)

⚙ SVG_Color_RGB_from_HLS

SVG_Color_RGB_from_HLS (tono ; luminosidad ; saturación {; formato}) -> Resultado

Parámetro	Tipo		Descripción
tono	Entero largo	→	Valor del tono
luminosidad	Entero largo	→	Valor de la luminosidad
saturación	Entero largo	→	Valor de la saturación
formato	Entero largo	→	Formato del color
Resultado	Texto	↩	Cadena color

Descripción

El comando *SVG_Color_RGB_from_HLS* devuelve una cadena que expresa el color que corresponde a los parámetros *tono*, *luminosidad* y *saturacion* pasados como argumentos. La cadena se devuelve por defecto de la forma "RGB (rojo, verde, azul)", sintaxis reconocida por los motores de renderización SVG.

tono es un entero largo entre 0 y 360°.

luminosidad y *saturacion* son enteros largos entre 0 y 100%.

El parámetro opcional *formato* permite especificar el formato deseado para la cadena color devuelta. Los valores son:

Valor	Formato
1 (por defecto)	rgb(r,g,b)
2	#rgb
3	#rrggbb
4	rgb(r%, g%, b%)

⚙ SVG_Color_RGB_from_long

SVG_Color_RGB_from_long (color {; formato}) -> Resultado

Parámetro	Tipo		Descripción
color	Entero largo	→	Valor del color
formato	Entero	→	Formato del color
Resultado	Cadena	↩	Cadena color

Descripción

El comando *SVG_Color_RGB_from_long* devuelve una cadena que expresa el *color* pasado como argumento. La cadena devuelta es de esta forma "RGB (rojo, verde, azul)", sintaxis reconocida por los motores de renderización SVG.

El parámetro *color* es un entero largo de 4 bytes cuyo formato (0x00RRGGBB) se describe a continuación (los bytes se numeran del 0 al 3, de derecha a izquierda):

Byte Descripción

2	Componente rojo del color (0..255)
1	Componente verde del color (0..255)
0	Componente azul del color (0..255)

El parámetro opcional *formato* permite especificar el formato deseado para la cadena color devuelta. Los valores son:

Valor	Formato
1 (por defecto)	rgb(r,g,b)
2	#rgb
3	#rrggbb
4	rgb(r%, g%, b%)

Ejemplo

```
$txtColor:=SVG_Color_RGB_from_long($color)
```

```
`$txtColor vale "#ff8000" si $color vale 16744448 (orange)
```

⚙️ SVG_FADE_TO_GREY_SCALE

SVG_FADE_TO_GREY_SCALE (objetoSVG {; valor})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia del objeto SVG
valor	Real	→	Valor de gris

Descripción

El comando **SVG_FADE_TO_GREY_SCALE** se aplica a un filtro para transformar la escala de grises de la imagen SVG cuya referencia se pasa en el parámetro *objetoSVG*.

Puede pasar un valor de escala de grises que se aplicará en el parámetro opcional *valor*. Si no pasa este parámetro, la transformación se efectúa de acuerdo con la percepción visual de la luminancia (30% de rojo, 59% de verde y 11% de azul).

⚙️ SVG_Filter_ColorMatrix

SVG_Filter_ColorMatrix (objetoSVG {; in ; result} {; tipo {; valores}}) -> Resultado

Parámetro	Tipo	Descripción
objetoSVG	Ref_SVG	➔ Referencia del objeto SVG
in	Texto	➔ Identifica entrada para el filtro dado primitivo
result	Texto	➔ Ofrece una referencia para el resultado de salida de un filtro
tipo	Texto	➔ Indica el tipo de operación de la matriz
valores	Texto	➔ Valores numéricos de la matriz de transformación
Resultado	Ref_SVG	➔ Referencia para el objeto SVG con nuevos valores de color

Descripción

El comando **SVG_Filter_ColorMatrix** aplica una transformación de matriz de color a cada píxel de la imagen fuente pasada en el parámetro *objetoSVG* para producir un resultado con un nuevo conjunto de valores de color.

En el parámetro *in*, puede pasar una cadena que coincide con un valor "result" anterior o una de las seis siguientes palabras claves:

- **SourceGraphic**: el elemento objetivo (imagen, forma , grupo, etc.) que referencia el filtro. Esta palabra clave representa los elementos gráficos que eran la entrada original en el elemento "filtro".
- **SourceAlpha**: el canvas bajo el SourceGraphic. Esta palabra clave representa los elementos gráficos que eran la entrada original en el elemento "filtro".
- **BackgroundImage**: el canvas bajo el SourceGraphic. Esta palabra clave representa una instantánea de imagen del canvas bajo la región del filtro en el momento en que se invocó el elemento "filtro".
- **BackgroundAlpha**: el canal alfa del canvas bajo el SourceGraphic. Igual que BackgroundImage excepto que únicamente se utiliza el canal alfa.
- **FillPaint**: un pseudo-gráfico igual al tamaño de la región de filtro llenado con la propiedad de relleno del elemento objetivo. Esta palabra clave representa el valor de la propiedad 'relleno' en el elemento objetivo para el efecto filtro.
- **StrokePaint**: una pseudo-gráfico igual al tamaño de la región de filtro de llenado con la propiedad de trazo del elemento objetivo. Esta palabra clave representa el valor de la propiedad 'trazo' en el elemento objetivo para el efecto de filtro.

Si no se pasa ningún valor y este es el primer filtro primitiva, entonces el SourceGraphic se utiliza como su entrada. Si no se ofrece ningún valor y esto es un filtro posterior primitivo, a continuación, este filtro primitivo, utilizará el resultado del filtro anterior primitivo como su entrada.

En el parámetro *result*, pase una referencia para el resultado de la salida de un filtro que puede ser referenciado por el parámetro *in* en un uso posterior de este comando en el mismo elemento "filtro". Si no se ofrece ningún valor, la salida se encuentra disponible para su reutilización como la entrada implícita para la siguiente primitiva de filtro si ese filtro primitivo ofrece ningún valor para su parámetro *in*.

En el parámetro *tipo*, puede especificar el tipo de operación matriz al pasar una de las siguientes palabras:

- **saturate**: ajusta la saturación de todos los canales de color RGB utilizando un valor de número real de 0 a 1 que se pasa en el parámetro *valores*.
- **hueRotate**: rota el pixel hue de todos los canales de color RGB el ángulo especificado (en grados) en el parámetro *valores*,

- `luminanceToAlpha`: convierte los canales rojo, verde y azul en un valor de luminancia. Los canales RGB están definidos en negro (0,0,0.)
- `matrix`: define el color utilizando la lista de valores pasados en el parámetro *valores*. Permite el valor de cada canal en la salida a especificar de una combinación de su color existente y canales alfa.

Si no pasa un parámetro *tipo*, por defecto el efecto es como si un valor de matriz se hubiera especificado.

En el parámetro *valores*, pase valores numéricos basados en la palabra clave pasada en el parámetro *tipo*:

- Con la palabra clave "matrix": pase una lista de 20 valores de matriz, separados por espacios en blanco y/o una coma.
- Con la palabra clave "saturate": pase un valor de número real individual (0 a 1). El valor permitido de acuerdo con la especificación es de 0-1, pero muchos navegadores aceptan valores superiores > 1 para permitir el exceso de saturación.
- Con la palabra clave "hueRotate": pase un valor número real (para indicar los grados de rotación).
- Con la palabra clave "luminanceToAlpha": no pase un valor numérico. El atributo "valores" no se utiliza con este tipo, lo que descarta el canal alfa y lo reemplaza con valores iguales a la luminancia de la entrada.

Si no pasa un parámetro *valores*, el comportamiento por defecto depende de la palabra clave pasada en el parámetro *tipo*:

- Con la palabra clave "matrix": por defecto se utilizan los valores de la matriz de identidad
- Con la palabra clave "saturate": por defecto, el valor es 1 (sin cambios).
- Con la palabra clave "hueRotate": por defecto el valor es 0 (sin cambios).
- Con la palabra clave "luminanceToAlpha": por defecto, este parámetro no se utiliza.

Nota: bajo Windows, este comando requiere la desactivación previa de Direct2D (ver la constante [Direct2D disabled](#) en la descripción del comando **SET DATABASE PARAMETER**).

Ejemplo

No filter

Matrix

Saturate

HueRotate

Luminance

```
C_TEXT($Dom_filter;$Dom_node;$Dom_rect;$Dom_svg;$Txt_matrix)
```

```
SVG_SET_OPTIONS(SVG_Get_options?+5)
```

```
$Dom_svg:=SVG_New
```

```
$Dom_filter:=SVG_Define_filter($Dom_svg;"Matrix")
```

```
$Txt_matrix:=\
```

```
".33 .33 .33 0 0 "\
```

```
+".33 .33 .33 0 0 "\
```

```
+".33 .33 .33 0 0 "\
```

```
+".33 .33 .33 0 0"
```



```

$Dom_node:=SVG_Filter_ColorMatrix($Dom_filter;"SourceGraphic";"";"matrix";$Txt_matrix)

$Dom_filter:=SVG_Define_filter($Dom_svg;"Saturate")
$Dom_node:=SVG_Filter_ColorMatrix($Dom_filter;"SourceGraphic";"";"saturate";"1.5")
// otra sintaxis para valor
//$Dom_node:=SVG_Filter_ColorMatrix
($Dom_filter;"SourceGraphic";"";"saturate";String(1,5;"&xml"))

$Dom_filter:=SVG_Define_filter($Dom_svg;"HueRotate90")
$Dom_node:=SVG_Filter_ColorMatrix($Dom_filter;"SourceGraphic";"";"hueRotate";"90")

$Dom_filter:=SVG_Define_filter($Dom_svg;"LuminanceToAlpha")
$Dom_node:=SVG_Filter_ColorMatrix($Dom_filter;"SourceGraphic";"";"luminanceToAlpha")

$Dom_rect:=SVG_New_rect($Dom_svg;2;0;797;100;0;0;"none";"coral")

$Dom_rect:=SVG_New_rect($Dom_svg;2;100;797;100;0;0;"none";"coral")
SVG_SET_FILTER($Dom_rect;"Matrix")

$Dom_rect:=SVG_New_rect($Dom_svg;2;200;797;100;0;0;"none";"coral")
SVG_SET_FILTER($Dom_rect;"Saturate")

$Dom_rect:=SVG_New_rect($Dom_svg;2;300;797;100;0;0;"none";"coral")
SVG_SET_FILTER($Dom_rect;"HueRotate90")

$Dom_rect:=SVG_New_rect($Dom_svg;2;400;797;100;0;0;"none";"coral")
SVG_SET_FILTER($Dom_rect;"LuminanceToAlpha")

SVG_New_text($Dom_svg;"No filter";110;10;"Verdana";60;Bold;-1;"black")
SVG_New_text($Dom_svg;"Matrix";110;110;"Verdana";60;Bold;-1;"black")
SVG_New_text($Dom_svg;"Saturate";110;210;"Verdana";60;Bold;-1;"black")
SVG_New_text($Dom_svg;"HueRotate";110;310;"Verdana";60;Bold;-1;"black")
SVG_New_text($Dom_svg;"Luminance";110;410;"Verdana";60;Bold;-1;"black")

//Ver el resultado
SVGTool_SHOW_IN_VIEWER($Dom_svg)

//SVG_SAVE_AS_TEXT($Dom_svg;System folder(Desktop)+"export.svg")

//No olvide limpiar la memoria
SVG_CLEAR($Dom_svg)

```

⚙ SVG_GET_COLORS_ARRAY

SVG_GET_COLORS_ARRAY (punteroArrayNomcolor)

Parámetro	Tipo	Descripción
punteroArrayNomcolor	Puntero →	Puntero al array que recibe los nombres de los colores

Descripción

El comando *SVG_GET_COLORS_ARRAY* llena el array apuntado por el parámetro *punteroArrayNomcolor* con los nombres de los colores reconocidos por SVG.

Ver también: <http://www.w3.org/TR/SVG/types.html#ColorKeywords>

⚙️ SVG_GET_DEFAULT_BRUSHES

SVG_GET_DEFAULT_BRUSHES (linea {; fondo})

Parámetro	Tipo		Descripción
linea	Puntero	→	Variable Alfa
fondo	Puntero	→	Variable Alfa

Descripción

El comando `SVG_GET_DEFAULT_BRUSHES` devuelve, en la variable señalada por *linea*, el color por defecto actual para dibujar las líneas.

Si se pasa el parámetro opcional *fondo*, la variable señalada por este parámetro recibirá el color por defecto actual utilizado para los fondos.

Si no han sido modificados, estos colores son blanco y negro, respectivamente.

Ejemplo

Ver el ejemplo del comando `SVG_SET_DEFAULT_BRUSHES`.

⚙ SVG_Get_named_color_value

SVG_Get_named_color_value (colorName {; rgbComponent}) -> Resultado

Parámetro	Tipo		Descripción
colorName	Texto	→	nombre del color SVG
rgbComponent	Texto	→	"R", "G" o "B" para indicar el componente color
Resultado	Entero largo	↩	Devuelve el valor color

Descripción

El comando **SVG_Get_named_color_value** devuelve el valor del color SVG cuyo nombre se especifica en el parámetro *nomColor*.

En el parámetro opcional *componenteRgb*, puede pasar "R" (rojo), "G" (verde) o "B" (azul) para indicar el componente de color específico para el que desea obtener el valor. Si no pasa este parámetro, el comando devuelve el valor del color largo (completo).

⚙️ SVG_SET_BRIGHTNESS

SVG_SET_BRIGHTNESS (objetoSVG ; brillo {; brillo2 ; brillo3})

Parámetro	Tipo	Descripción
objetoSVG	Ref_SVG	➔ Referencia SVG
brillo	Real	➔ Valores entre 0 y 1 para oscurecer; > 1 para aclarar, aplicados globalmente o sólo al componente rojo del color
brillo2	Real	➔ Valor del brillo para el componente verde
brillo3	Real	➔ Valor del brillo para el componente azul

Descripción

El comando **SVG_SET_BRIGHTNESS** ajusta el brillo de una imagen SVG o contenedor cuya referencia se pasa en el parámetro *objetoSVG*.

En el parámetro *brillo*, pase un valor entre 0 y 1 para oscurecer el brillo, o un valor superior a 1 para iluminar. Cuando se pasa un solo parámetro *brillo*, el factor de brillo se aplica globalmente al objeto.

Si pasa los dos parámetros (opcionales) *brillo2* y *brillo3*, en este caso cada valor se aplica a cada componente color, es decir, *brillo* se aplica al componente "R" (rojo) , *brillo2* se aplica al componente "G" (verde) y *brillo3* se aplica al componente "B" (azul).

⚙️ SVG_SET_DEFAULT_BRUSHES

SVG_SET_DEFAULT_BRUSHES (línea {; fondo})

Parámetro	Tipo		Descripción
línea	Cadena	→	Color de línea
fondo	Cadena	→	Color de fondo

Descripción

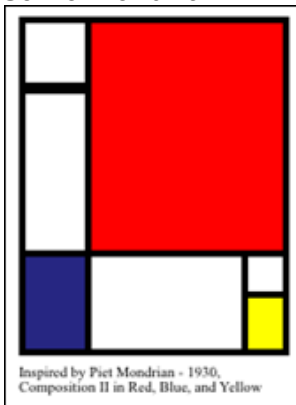
El comando `SVG_SET_DEFAULT_BRUSHES` permite fijar los colores por defecto utilizados por el componente SVG.

El parámetro *línea* contiene el nuevo color que se utilizará para las líneas. El parámetro opcional *fondo* contiene el nuevo color que se utilizará para el dibujo de fondos.

Puede pasar una cadena vacía en cualquiera de estos parámetros con el fin de restablecer el valor por defecto del componente, es decir, negro para las líneas y blanco para el fondo.

Ejemplo

Como Mondrian...



```
$svg:=SVG_New
  ` Definir los colores por defecto
  SVG_SET_DEFAULT_BRUSHES("black";"white")
  ` Líneas de 4 puntos de ancho
  SVG_SET_STROKE_WIDTH($svg;4)
  $g:=SVG_New_group($svg)
  SVG_New_rect($g;2;2;40;40)
  SVG_New_rect($g;2;45;40;100)
  SVG_SET_FILL_BRUSH(SVG_New_rect($g;2;144;40;60);"midnightblue")
  SVG_SET_FILL_BRUSH(SVG_New_rect($g;42;2;120;142);"red")
  SVG_New_rect($g;42;144;95;60)
  SVG_New_rect($g;137;144;25;25)
  SVG_SET_FILL_BRUSH(SVG_New_rect($g;137;169;25;35);"yellow")
  SVG_SET_TRANSFORM_TRANSLATE($g;10;10)
  ` Caption
  SVG_New_text($svg;"Inspirado por Piet Mondrian - 1930,\rComposition II in Red, Blue, and
  Yellow";10;220;"";9)
```

⚙️ SVG_SET_HUE

SVG_SET_HUE (objetoSVG ; tono)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia del objeto SVG
tono	Entero largo	→	Valor de tono

Descripción

El comando **SVG_SET_HUE** define un valor de tono para el objeto SVG designado por el parámetro *objetoSVG*. *objetoSVG* debe ser un contenedor SVG (svg, grupo, símbolo, patrón, marcador, etc.) o una imagen; de lo contrario, se devuelve un error.

En el parámetro *tono*, pase un valor entre 0 y 360.

⚙️ SVG_SET_SATURATION

SVG_SET_SATURATION (objetoSVG ; saturacion)





















Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia del objeto SVG
saturacion	Entero largo	→	Valor de saturación

Descripción

El método **SVG_SET_SATURATION** define un valor para el objeto SVG designado por el parámetro *objetoSVG*. *objetoSVG* debe ser un contenedor SVG (svg, grupo, símbolo, patrón, marcador, etc.) o una imagen; de lo contrario, se devuelve un error.

En el parámetro *saturacion*, pase un valor entre 0 y 100

Dibujo

-  SVG_Add_object
-  SVG_ADD_POINT
-  SVG_New_arc
-  SVG_New_circle
-  SVG_New_ellipse
-  SVG_New_ellipse_bounded
-  SVG_New_embedded_image
-  SVG_New_image
-  SVG_New_line
-  SVG_New_path
-  SVG_New_polygon
-  SVG_New_polygon_by_arrays
-  SVG_New_polyline
-  SVG_New_polyline_by_arrays
-  SVG_New_rect
-  SVG_New_regular_polygon
-  SVG_PATH_ARC
-  SVG_PATH_CLOSE
-  SVG_PATH_CURVE
-  SVG_PATH_LINE_TO
-  SVG_PATH_MOVE_TO
-  SVG_PATH_QCURVE
-  SVG_Use

⚙ SVG_Add_object

SVG_Add_object (objetoSVGObjetivo ; objetoSVGFuente) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGObjetivo	Ref_SVG	→	Referencia del elemento padre
objetoSVGFuente	Ref_SVG	→	Referencia del objeto a añadir
Resultado	Ref_SVG	↩	Referencia del objeto SVG

Descripción

El comando *SVG_Add_object* permite ubicar en el contenedor SVG designado por *objetoSVGFuente* un objeto SVG designado por *objetoSVGObjetivo* y devuelve su referencia. El contenedor SVG puede ser la raíz del documento o cualquier otra referencia a un objeto SVG que pueda contener este tipo de elemento.

SVG_ADD_POINT

SVG_ADD_POINT (objetoSVGPadre ; x ; y { ; x2 ; y2 ; ... ; xN ; yN })

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
x	Entero largo	→	Coordenada del nuevo punto en el eje X
y	Entero largo	→	Coordenada del nuevo punto en eje Y

Descripción

El comando *SVG_ADD_POINT* añade uno o más segmentos a la ruta de acceso referenciada por *objetoSVGPadre*. La ruta puede ser de tipo 'path' "polyline" o "polygon". Si *objetoSVGPadre* no es una referencia de un trazo de este tipo, se genera un error.

Si se pasan varios pares de coordenadas, los diferentes puntos se añadirán sucesivamente. En este caso, si el último par de coordenadas está incompleto (falta y), se ignorará.

Ejemplo

Ver los ejemplos del comando [SVG_New_polyline](#).

⚙ SVG_New_arc

SVG_New_arc (objetoSVGPadre ; x ; y ; radio ; inicio ; fin {; colorPrimerPlano {; colorFondo {; anchoTrazo}} }) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
x	Entero largo	→	Coordenada del centro en el eje X
y	Entero largo	→	Coordenada del centro en el eje Y
radio	Entero largo	→	Radio del círculo
inicio	Entero largo	→	Valor en grados del inicio del arco
fin	Entero largo	→	Valor en grados del final del arco
colorPrimerPlano	Cadena	→	Nombre del color o del degradé
colorFondo	Cadena	→	Nombre del color o del degradé
anchoTrazo	Real	→	Grueso del trazo
Resultado	Ref_SVG	↩	Referencia del arco

Descripción

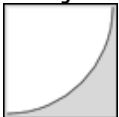
El comando *SVG_New_arc* crea un nuevo arco de círculo en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

Los parámetros opcionales *colorPrimerPlano* y *colorFondo* contienen respectivamente el nombre del color de la línea y del color de fondo. (Para más información sobre los colores, consulte los comandos del tema **Colores y degradés**).

El parámetro opcional *anchoTrazo* contiene el tamaño del trazo expresado en píxeles. Su valor por defecto es 1.

Ejemplo 1

Dibujo de un arco de 0° a 90° (color de relleno y borde, grueso del trazo por defecto):



```
svgRef:=SVG_New  
objectRef:=SVG_New_arc(svgRef;100;100;90;90;180)
```

Ejemplo 2

Dibujo de un arco de 90° a 180° de un círculo azul claro con un borde azul y un trazo de 2 puntos:



```
svgRef:=SVG_New  
objectRef:=SVG_New_arc(svgRef;100;100;90;180;270;"blue";"lightblue";2)
```

⚙ SVG_New_circle

SVG_New_circle (objetoSVGPadre ; x ; y ; radio {; colorPrimerPlano {; colorFondo {; anchoTrazo}} }) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
x	Entero largo	→	Coordenada del centro en el eje X
y	Entero largo	→	Coordenada del centro en el eje Y
radio	Entero largo	→	Radio del círculo
colorPrimerPlano	Cadena	→	Nombre del color o del degradé
colorFondo	Cadena	→	Nombre del color o del degradé
anchoTrazo	Real	→	Grueso del trazo
Resultado	Ref_SVG	↩	Referencia del círculo

Descripción

El comando *SVG_New_circle* crea un nuevo círculo en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

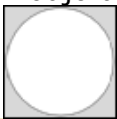
El círculo se ubica y dimensiona de acuerdo a las coordenadas del centro (*x*, *y*) y el *radio* se pasa como parámetro.

Los parámetros opcionales *colorPrimerPlano* y *colorFondo* contienen respectivamente el nombre del color de la línea y del color de fondo. (Para más información sobre los colores, consulte los comandos del tema [Colores y degradés](#)).

El parámetro opcional *anchoTrazo* contiene el tamaño del trazo expresado en píxeles. Su valor por defecto es 1.

Ejemplo 1

Dibujo de un círculo (color de relleno y de borde, grosor de línea por defecto):



```
svgRef:=SVG_New
objectRef:=SVG_New_circle(svgRef;100;100;90)
```

Ejemplo 2

Dibujo de un círculo azul claro con un borde azul y un grosor de línea de 2 puntos:



```
svgRef:=SVG_New
objectRef:=SVG_New_circle(svgRef;100;100;90;"blue";"lightblue";2)
```

⚙ SVG_New_ellipse

SVG_New_ellipse (objetoSVGPadre ; x ; y ; radioX ; radioY {; colorPrimerPlano {; colorFondo {; anchoTrazo}} }) -> Resultado

Parámetro	Tipo	Descripción
objetoSVGPadre	Ref_SVG	→ Referencia del elemento padre
x	Entero largo	→ Coordenada en el eje X del centro del elipse
y	Entero largo	→ Coordenada del centro de la elipse en el eje Y
radioX	Entero largo	→ Radio en el eje X
radioY	Entero largo	→ Radio en el eje Y
colorPrimerPlano	Cadena	→ Nombre del color o del degradé
colorFondo	Cadena	→ Nombre del color o del degradé
anchoTrazo	Real	→ Grosor del trazo
Resultado	Ref_SVG	↩ Referencia del elipse

Descripción

El comando *SVG_New_ellipse* crea una nueva elipse en el contenedor SVG designado por *objetoSVGPadre*. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

La elipse se ubica y dimensiona de acuerdo a los valores de *x*, *y*, *ancho* y *alto*.

Los parámetros opcionales *colorPrimerPlano* y *colorFondo* contienen respectivamente el nombre del color de la línea y del color de fondo. (Para más información sobre los colores, consulte los comandos del tema **Colores y degradés**).

El parámetro opcional *anchoTrazo* contiene el tamaño del trazo expresado en píxeles. Su valor por defecto es 1.

Ejemplo 1

Dibujo de una elipse (color de relleno y de borde, grosor de línea por defecto):



```
svgRef:=SVG_New  
objectRef:=SVG_New_ellipse(svgRef;100;50;90;40)
```

Ejemplo 2

Dibujo de una elipse azul claro con un borde azul y un grosor de línea de 2 puntos:



```
svgRef:=SVG_New  
objectRef:=SVG_New_ellipse(svgRef;100;50;90;40;"blue";"lightblue";2)
```

⚙ SVG_New_ellipse_bounded

SVG_New_ellipse_bounded (objetoSVGPadre ; x ; y ; ancho ; alto {; colorPrimerPlano {; colorFondo {; anchoTrazo}}}) -> Resultado

Parámetro	Tipo	Descripción
objetoSVGPadre	Ref_SVG	➔ Referencia del elemento padre
x	Entero largo	➔ Coordenada de la esquina superior izquierda en el eje X
y	Entero largo	➔ Coordenada de la esquina superior izquierda en el eje Y
ancho	Entero largo	➔ Ancho del rectángulo que rodea
alto	Entero largo	➔ Alto del rectángulo que rodea
colorPrimerPlano	Cadena	➔ Nombre del color o del degradé
colorFondo	Cadena	➔ Nombre del color o del degradé
anchoTrazo	Real	➔ Grosor del trazo
Resultado	Ref_SVG	➔ Referencia de la elipse

Descripción

El comando *SVG_New_ellipse_bounded* crea una nueva elipse en el contenedor SVG designado por *objetoSVGPadre*. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

La elipse creada cabe en el rectángulo definido por *x*, *y*, *ancho* y *alto*.

Los parámetros opcionales *colorPrimerPlano* y *colorFondo* contienen respectivamente el nombre del color de la línea y del color de fondo. (Para más información sobre los colores, consulte los comandos del tema **Colores y degradés**).

El parámetro opcional *anchoTrazo* contiene el tamaño del trazo expresado en píxeles. Su valor por defecto es 1.

Ejemplo 1

Dibujo de una elipse (color de relleno y de borde, grosor de línea por defecto):



```
svgRef:=SVG_New  
objectRef:=SVG_New_ellipse_bounded(svgRef;10;10;200;100)
```

Ejemplo 2

Dibujo de una elipse azul con un borde azul y un grosor de línea de 2 puntos:



```
svgRef:=SVG_New  
objectRef:=SVG_New_ellipse_bounded(svgRef;100;100;200;100;"blue";"lightblue";2)
```

⚙ SVG_New_embedded_image

SVG_New_embedded_image (objetoSVGPadre ; imagen {; x {; y}}{; codec}) -> Resultado

Parámetro	Tipo	Descripción
objetoSVGPadre	Ref_SVG	➔ Referencia del elemento padre
imagen	Imagen	➔ Imagen a incluir
x	Entero largo	➔ Coordenada de la esquina superior izquierda en el eje X
y	Entero largo	➔ Coordenada de la esquina superior derecha en el eje Y
codec	Texto	➔ Codec a utilizar
Resultado	Ref_SVG	➔ Referencia de la imagen

Descripción

El comando **SVG_New_embedded_image** permite incluir *imagen* en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

La imagen se codifica en base64 luego se incluye en el documento.

El parámetro *imagen* es una variable o un campo imagen 4D.

Los parámetros opcionales *x* e *y* permiten precisar la posición de la esquina superior derecha de la imagen en el contenedor SVG (valor por defecto: 0).

El parámetro opcional *codec* permite definir el codec a utilizar para la *imagen*. Por defecto, si este parámetro se omite, el codec es ".png".

Ejemplo

Incluir la imagen 'logo4D.png' ubicada en la carpeta 'Resources':



```
svgRef:=SVG_New
$Path:=Get 4D folder(Current resources folder)+"logo4D.png"
READ PICTURE FILE($Path;$Picture)
If(OK=1)
  objectRef:=SVG_New_embedded_image(svgRef;$Picture)
End if
```


⚙ SVG_New_image

SVG_New_image (objetoSVGPadre ; url {; x ; y {; ancho ; alto}}) -> Resultado

Parámetro	Tipo	Descripción
objetoSVGPadre	Ref_SVG	➔ Referencia del elemento padre
url	Cadena	➔ Dirección de la imagen
x	Entero largo	➔ Coordenada de la esquina superior izquierda en el eje X
y	Entero largo	➔ Coordenada de la esquina superior izquierda en el eje Y
ancho	Entero largo	➔ Ancho de la imagen
alto	Entero largo	➔ Alto de la imagen
Resultado	Ref_SVG	➔ Referencia de la imagen

Descripción

El comando *SVG_New_image* permite referenciar una imagen en la dirección *url* en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

El parámetro *url* especifica la ubicación de la imagen y puede tomar varias formas:

- Un **URL local** (ruta de acceso de la forma: `file://...`): en este caso, la imagen sólo se mostrará si el archivo es realmente accesible en el momento de la renderización del objeto. Esta URL local puede ser relativa (de la forma: `"#Imágenes/miImagen.png"`), en este caso el comando prefija la ruta de acceso con la de la carpeta **Resources** de la base local. Si los parámetros *ancho* y *alto* se omiten, serán calculados por el comando (lo cual debe evitarse ya que esto hace que la ejecución del comando sea más lenta). En el caso de una ruta relativa, si no es válida, se genera un error.
- Una **URL no local** (`http://miSitio.com/imagenes/miImagen.jpeg`). En este caso, no se lleva a cabo la verificación de la validez del enlace y un error se generará si se omiten los parámetros *ancho* y *alto*.
- Una **URL relativa** (`"../picture.png"`). Es particularmente importante en modo cliente/servidor cuando los archivos están en la carpeta `resources`. URLs relativos pueden comenzar por:
 - `"/"`, para indicar la ruta `"~/Resources/SVG/"`
 - `"./"`, para indicar la ruta `"~/Resources/"`
 - `"../"`, para indicar la carpeta de la base.

Los parámetros opcionales *x* e *y* permiten precisar la posición de la esquina superior derecha de la imagen en el contenedor SVG (valor por defecto: 0).

Los parámetros *ancho* y *alto* especifican el tamaño del rectángulo en el que se mostrará la imagen y así determinar el tamaño y el radio de aspecto de la imagen. Estos parámetros sólo son opcionales en el caso de una imagen referenciada por una ruta relativa en la carpeta **Resources** de la base local. Si *ancho* y/o *alto* es igual a 0, la imagen no se renderiza.

Ejemplo 1

Ubicar la imagen `'logo4D.png'` ubicado en la carpeta `'Pictures'` de la carpeta `'Resources'`:



```
svgRef:=SVG_New
objectRef:=SVG_New_image(svgRef;"#Pictures/logo4D.png")
```

Ejemplo 2

Ubicar la imagen '4dlogo.gif' accesible en el directorio 'pictures' del sitio '4d.com':



```
svgRef:=SVG_New
objectRef:=SVG_New_image(svgRef;"http://www.4d.com/pictures/4dlogo.gif";20;20;39;53)
```

Ejemplo 3

Estos son algunos ejemplos para acceder a imágenes utilizando las URLs relativas:

```
SVG_New_image($Dom_svg;"/images/picture.png";10;10)
// base es la carpeta "Resources"
// el código XML será xlink:href="../images/picture.png"
```

```
SVG_New_image($Dom_svg;"/picture.png";70;180)
// base es la carpeta de la base
// XML code will be xlink:href="picture.png"
```

```
SVG_New_image($Dom_svg;"/sample pictures/picture.png";110;90;100;100)
// base es la carpeta "SVG" en la carpeta "Resources"
// el código XML será xlink:href="sample%20pictures/picture.gif"
```

⚙ SVG_New_line

SVG_New_line (objetoSVGPadre ; inicioX ; inicioY ; finX ; finY {; color {; anchoTrazo}}) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
inicioX	Entero largo	→	Posición horizontal de inicio
inicioY	Entero largo	→	Posición vertical de inicio
finX	Entero largo	→	Posición horizontal de fin
finY	Entero largo	→	Posición vertical de fin
color	Cadena	→	Nombre del color o del degradé
anchoTrazo	Real	→	Grueso del trazo
Resultado	Ref_SVG	↩	Referencia de línea

Descripción

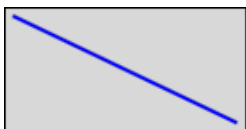
El comando *SVG_New_line* crea una nueva línea en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. El objeto se ubica de acuerdo con las coordenadas *inicioX*, *inicioY*, *finX* y *finY*. El contenedor SVG puede ser la raíz del documento o cualquier otra referencia a un objeto SVG que pueda contener este tipo de elemento.

El parámetro opcional *color* contiene el nombre del color de la línea. (Para más información sobre los colores, consulte los comandos del tema [Colores y degradés](#)).

El parámetro opcional *anchoTrazo* contiene el tamaño de la pluma expresado en píxeles. Su valor por defecto es 1.

Ejemplo

Dibujo de una línea azul de 3 píxeles:



```
svgRef:=SVG_New  
objectRef:=SVG_New_line(svgRef;10;10;200;100;"blue";3)
```

⚙ SVG_New_path

SVG_New_path (objetoSVGPadre ; x ; y {; colorPrimerPlano {; colorFondo {; anchoTrazo}} }) -
> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
x	Entero largo	→	Coordenada del inicio del trazo en el eje X
y	Entero largo	→	Coordenada del inicio del trazo en el eje Y
colorPrimerPlano	Cadena	→	Nombre del color o del degradé
colorFondo	Cadena	→	Nombre del color o del degradé
anchoTrazo	Real	→	Grueso del trazo
Resultado	Ref_SVG	↪	Referencia del trazo

Descripción

El comando *SVG_New_path* inicia un nuevo trazo en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

Un trazo representa el contorno de una forma. Un trazo se describe utilizando el concepto de punto actual. Por analogía con un dibujo en el papel, el punto actual puede compararse con la posición de la pluma. Este punto puede cambiar y el contorno de una forma (abierta o cerrada) se puede remontar al mover la pluma a lo largo de una línea recta o en curva.

Los trazos representan la geometría del contorno de un objeto definido de acuerdo con las instrucciones de los siguientes elementos: [SVG_PATH_MOVE_TO](#) (establece un nuevo punto actual), [SVG_PATH_LINE_TO](#) (dibuja una línea recta), [SVG_PATH_CURVE](#) (dibuja una curva con una curva de Bézier cúbica), [SVG_PATH_ARC](#) (dibuja un arco circular o elíptico) y [SVG_PATH_CLOSE](#) (cierra la forma actual al trazar una línea hasta el último inicio del trazo). Es posible tener trazos compuestos (es decir un trazo con varios subtrazos) que permiten efectos como un "agujero de dona" en los objetos.

Los parámetros *x* e *y* permiten especificar la posición del inicio del trazo en el contenedor SVG.

Los parámetros opcionales *colorPrimerPlano* y *colorFondo* contienen respectivamente el nombre del color de la línea y del color de fondo. (Para más información sobre los colores, consulte los comandos del tema [Colores y degradados](#)).

El parámetro opcional *anchoTrazo* contiene el tamaño del trazo expresado en píxeles. Su valor por defecto es 1.

Ejemplo 1

Dibujar una línea quebrada cerrada:



```
svgRef:=SVG_New
objectRef:=SVG_New_path(svgRef;20;20;"red";"none";5)
SVG_PATH_LINE_TO(objectRef;40)
SVG_PATH_LINE_TO(objectRef;40;40)
SVG_PATH_LINE_TO(objectRef;80;40;80;20;100;20;100;100;80;100;80;80;40;80;40;100;20;100)
SVG_PATH_CLOSE(objectRef)
```

Ejemplo 2

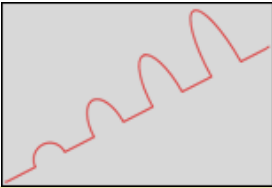
Dibujar una curva de Bezier:



```
svgRef:=SVG_New
objectRef:=SVG_New_path(svgRef;100;200;"aquamarine";"none";10)
SVG_PATH_CURVE(objectRef;250;200;100;100;250;100)
SVG_PATH_CURVE(objectRef;400;200;400;300)
```

Ejemplo 3

Comando de arco en los datos de trazo:



```
svgRef:=SVG_New
objectRef:=SVG_New_path(svgRef;20;300;"red";"none";2)
SVG_SET_OPTIONS(SVG_Get_options?-4)&NBSP;&NBSP; `Cambiar a coordenadas relativas
SVG_PATH_LINE_TO(objectRef;50;-25)
For($Lon_i;1;4;1)
  SVG_PATH_ARC(objectRef;25;25*$Lon_i;50;-25;-30)
  SVG_PATH_LINE_TO(objectRef;50;-25)
End for
```

Ejemplo 4

Trazo complejo (curva de Bezier cúbica):



```
`Creación de un nuevo elemento SVG
$txt_svg:=SVG_New(174,96;125,04;"4D Logo";"";True)

`Creación de un nuevo trazo
$txt_path:=SVG_New_path($txt_svg;150,665;13,021)
`Definición de los colores
SVG_SET_STROKE_BRUSH($txt_path;"#212a6f")
SVG_SET_FILL_BRUSH($txt_path;"#212a6f")
...
SVG_PATH_CURVE($txt_path;-9,683;-6,54;-20,842;-8,888;-33,06;-10,462)
SVG_PATH_CURVE($txt_path;-7,042;-0,915;-14,587;-0,877;-22,087;-0,877)
SVG_PATH_CURVE($txt_path;-1,725;0;-4,312;-0,405;-5,761;0,24)
SVG_PATH_CURVE($txt_path;-1,762;0;-5,092;-0,382;-6,479;0,24)
```

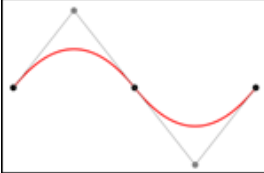
```

...
SVG_PATH_CURVE($Txt_path;181,489;70,216;177,236;30,976;150,665;13,021)
SVG_PATH_MOVE_TO($Txt_path;146,03;98,078)
...
SVG_PATH_CURVE($Txt_path;153,11;78,668;151,407;89,558;146,03;98,078)

```

Ejemplo 5

Curva de Bezier cuadrática:



```

` Creación de un nuevo elemento SVG
$svg:=SVG_New

` Inicialización del trazo negro y sin relleno
SVG_SET_DEFAULT_BRUSHES("");"none")

` Dibujar una curva de Bezier cuadrática en rojo
$qCurve:=SVG_New_path($svg;200;300)
SVG_SET_STROKE_BRUSH($qCurve;"red")
SVG_SET_STROKE_WIDTH($qCurve;5)
SVG_PATH_QCURVE($qCurve;400;50;600;300)
SVG_PATH_QCURVE($qCurve;1000;300)

` Puntos finales en negro
$g:=SVG_New_group($svg)
SVG_Set_description($g;"End points")
SVG_SET_DEFAULT_BRUSHES("black";"black")
SVG_New_circle($g;200;300;10)
SVG_New_circle($g;600;300;10)
SVG_New_circle($g;1000;300;10)

` Puntos y líneas de control en gris
$g:=SVG_New_group($svg)
SVG_Set_description($g;"Control points and lines from end points to control points")
SVG_SET_DEFAULT_BRUSHES(SVG_Color_grey(50);"none")
$path:=SVG_New_path($svg;200;300)
SVG_SET_STROKE_WIDTH($path;2)
SVG_PATH_LINE_TO($path;400;50;600;300;800;550;1000;300)
$gray:=SVG_Color_grey(50) ` grey 50%
SVG_SET_DEFAULT_BRUSHES($gray;$gray)
SVG_New_circle($g;400;50;10)
SVG_New_circle($g;800;550;10)

```

⚙ SVG_New_polygon

SVG_New_polygon (objetoSVGPadre {; puntos {; colorPrimerPlano {; colorFondo {; anchoTrazo}}}}) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
puntos	Cadena	→	Trazo
colorPrimerPlano	Cadena	→	Nombre del color o del degradé
colorFondo	Cadena	→	Nombre del color o del degradé
anchoTrazo	Real	→	Grueso del trazo
Resultado	Ref_SVG	↩	Referencia del polígono

Descripción

El comando *SVG_New_polygon* crea una nueva forma cerrada en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es una referencia válida, se genera un error.

El parámetro opcional *puntos* permite pasar los puntos del trazo de la línea del polígono como es esperado por el estándar SVG. Si se omite este parámetro o está vacío, los puntos se pueden definir con el comando *SVG_ADD_POINT*.

Los parámetros opcionales *colorPrimerPlano* y *colorFondo* contienen respectivamente el nombre del color de la línea y del color de fondo. (Para más información sobre los colores, consulte la sección **Colores y degradés**).

El parámetro opcional *anchoTrazo* contiene el tamaño de la pluma expresado en píxeles. Su valor por defecto es 1.

⚙ SVG_New_polygon_by_arrays

SVG_New_polygon_by_arrays (objetoSVGPadre ; punteroArrayX ; punteroArrayY { ; colorPrimerPlano { ; colorFondo { ; anchoTrazo } } }) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
punteroArrayX	Puntero	→	Coordenadas de los puntos en el eje X
punteroArrayY	Puntero	→	Coordenadas de los puntos en el eje Y
colorPrimerPlano	Cadena	→	Nombre del color o del degradé
colorFondo	Cadena	→	Nombre del color o del degradé
anchoTrazo	Real	→	Grueso del trazo
Resultado	Ref_SVG	↩	Referencia del polígono

Descripción

El comando *SVG_New_polygon_by_arrays* dibuja una forma cerrada que consiste en un conjunto serie de segmentos rectos conectados en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

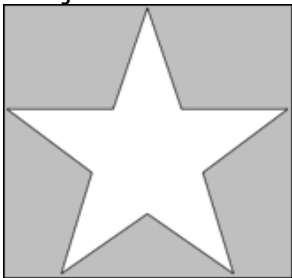
Todos los valores de coordenadas están en el sistema de coordenadas usuario.

Los parámetros opcionales *colorPrimerPlano* y *colorFondo* contienen respectivamente el nombre del color de la línea y del color de fondo. (Para más información sobre los colores, consulte los comandos del tema [Colores y degradés](#)).

El parámetro opcional *anchoTrazo* contiene el tamaño del trazo expresado en píxeles. Su valor por defecto es 1.

Ejemplo

Dibujo de una estrella (color del borde y grueso de la línea por defecto):



```
ARRAY LONGINT($tX;0)
ARRAY LONGINT($tY;0)
```

```
APPEND TO ARRAY($tX;129)
APPEND TO ARRAY($tY;10)
APPEND TO ARRAY($tX;158)
APPEND TO ARRAY($tY;96)
APPEND TO ARRAY($tX;248)
APPEND TO ARRAY($tY;96)
APPEND TO ARRAY($tX;176)
APPEND TO ARRAY($tY;150)
APPEND TO ARRAY($tX;202)
```


APPEND TO ARRAY(\$tY;236)
APPEND TO ARRAY(\$tX;129)
APPEND TO ARRAY(\$tY;185)
APPEND TO ARRAY(\$tX;56)
APPEND TO ARRAY(\$tY;236)
APPEND TO ARRAY(\$tX;82)
APPEND TO ARRAY(\$tY;150)
APPEND TO ARRAY(\$tX;10)
APPEND TO ARRAY(\$tY;96)
APPEND TO ARRAY(\$tX;100)
APPEND TO ARRAY(\$tY;96)

objectRef:=SVG_New_polygon_by_arrays(svgRef;->\$tX;->\$tY)

⚙ SVG_New_polyline

SVG_New_polyline (objetoSVGPadre {; puntos {; colorPrimerPlano {; colorFondo {; anchoTrazo}}}}) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
puntos	Cadena	→	Trazo
colorPrimerPlano	Cadena	→	Nombre del color o del degradé
colorFondo	Cadena	→	Nombre del color o del degradé
anchoTrazo	Real	→	Grueso del trazo
Resultado	Ref_SVG	↩	Referencia de línea

Descripción

El comando *SVG_New_polyline* crea una nueva línea quebrada abierta en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es una referencia válida, se genera un error.

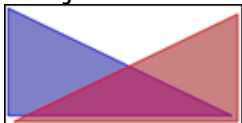
El parámetro opcional *puntos* permite pasar los puntos del trazo de la línea como lo espera el estándar SVG. Si se omite o está vacío este parámetro, los puntos pueden definirse con el comando *SVG_ADD_POINT*.

Los parámetros opcionales *colorPrimerPlano* y *colorFondo* contienen respectivamente el nombre del color de la línea y del color de fondo. (Para más información sobre los colores, consulte los comandos del tema **Colores y degradés**).

El parámetro opcional *anchoTrazo* contiene el tamaño del trazo expresado en píxeles. Su valor por defecto es 1.

Ejemplo

Dibujo de dos triángulos:



```
$polyline:=SVG_New_polyline($svg;"10,10 200,100 10,100 10,10";"blue";"blue:50")
$polyline:=SVG_New_polyline($svg;"";"red";"red:50")
SVG_ADD_POINT($polyline;205;15)
SVG_ADD_POINT($polyline;15;105)
SVG_ADD_POINT($polyline;205;105)
SVG_ADD_POINT($polyline;205;15)
```

⚙ SVG_New_polyline_by_arrays

SVG_New_polyline_by_arrays (objetoSVGPadre ; punteroArrayX ; punteroArrayY {; colorPrimerPlano {; colorFondo {; anchoTrazo}} }) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
punteroArrayX	Puntero	→	Coordenadas de los puntos en el eje X
punteroArrayY	Puntero	→	Coordenadas de los puntos en el eje Y
colorPrimerPlano	Cadena	→	Nombre del color o del degradé
colorFondo	Cadena	→	Nombre del color o del degradé
anchoTrazo	Real	→	Grueso del trazo
Resultado	Ref_SVG	↩	Referencia de línea

Descripción

El comando *SVG_New_polyline_by_arrays* traza una línea quebrada compuesta de segmentos rectos conectados entre sí en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

Por lo general, los elementos "polyline" diseñan formas abiertas, pero también pueden utilizarse para formas cerradas. En este caso, el último punto debe ser igual al primero.

Todos los valores de coordenadas están en el sistema de coordenadas usuario.

Los parámetros opcionales *colorPrimerPlano* y *colorFondo* contienen respectivamente el nombre del color de la línea y del color de fondo. (Para más información sobre los colores, consulte los comandos del tema [Colores y degradés](#)).

El parámetro opcional *anchoTrazo* contiene el tamaño del trazo expresado en píxeles. Su valor por defecto es 1.

Ejemplo 1

Dibujar un triángulo (color de borde y grueso de línea por defecto):



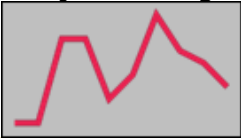
```
ARRAY LONGINT($tX;0)
ARRAY LONGINT($tY;0)
```

```
APPEND TO ARRAY($tX;10)
APPEND TO ARRAY($tY;10)
APPEND TO ARRAY($tX;200)
APPEND TO ARRAY($tY;100)
APPEND TO ARRAY($tX;10)
APPEND TO ARRAY($tY;100)
APPEND TO ARRAY($tX;10)
APPEND TO ARRAY($tY;10)
```

```
svgRef:=SVG_New
objectRef:=SVG_New_polyline_by_arrays(svgRef;->$tX;->$tY)
```

Ejemplo 2

Dibujar un diagrama de línea:



```
ARRAY LONGINT($tX;0)
ARRAY LONGINT($tY;0)
`Eje X
For($Lon_i;0;200;20)
  APPEND TO ARRAY($tX;$Lon_i)
End for
`Valores
APPEND TO ARRAY($tY;100)
APPEND TO ARRAY($tY;100)
APPEND TO ARRAY($tY;30)
APPEND TO ARRAY($tY;30)
APPEND TO ARRAY($tY;80)
APPEND TO ARRAY($tY;60)
APPEND TO ARRAY($tY;10)
APPEND TO ARRAY($tY;40)
APPEND TO ARRAY($tY;50)
APPEND TO ARRAY($tY;70)
```

```
objectRef:=SVG_New_polyline_by_arrays(svgRef;->$tX;->$tY;"crimson";"none";5)
```

⚙ SVG_New_rect

SVG_New_rect (objetoSVGPadre ; x ; y ; ancho ; alto {; redondX {; redondY {; colorPrimerPlano {; colorFondo {; anchoTrazo}}}}) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
x	Entero largo	→	X de la esquina superior izquierda
y	Entero largo	→	Y de la esquina superior izquierda
ancho	Entero largo	→	Ancho del rectángulo
alto	Entero largo	→	Alto del rectángulo
redondX	Entero largo	→	Curva horizontal
redondY	Entero largo	→	Curva vertical
colorPrimerPlano	Cadena	→	Nombre del color o del degradé
colorFondo	Cadena	→	Nombre del color o del degradé
anchoTrazo	Real	→	Grueso del trazo
Resultado	Ref_SVG	↩	Referencia del rectángulo

Descripción

El comando *SVG_New_rect* crea un nuevo rectángulo en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

El rectángulo se ubica y se dimensiona de acuerdo a los valores de *x*, *y*, *ancho* y *alto*.

Los parámetros opcionales *redondX* y *redondY* permiten redondear los ángulos de acuerdo a los valores indicados. Si el parámetro *redondY* se omite (o es -1), la curva será regular. Pase 1 en estos parámetros si quiere que sean ignorados por el comando.

Los parámetros opcionales *colorPrimerPlano* y *colorFondo* contienen respectivamente el nombre del color de la línea y del color de fondo. (Para más información sobre los colores, consulte los comandos del tema [Colores y degradés](#)).

El parámetro opcional *anchoTrazo* contiene el tamaño del trazo expresado en píxeles. Su valor por defecto es 1.

Ejemplo 1

Dibujo de un rectángulo (color de relleno y de borde, grueso de línea por defecto):



```
svgRef:=SVG_New  
objectRef:=SVG_New_rect(svgRef;10;10;200;100)
```

Ejemplo 2

Dibujo de un rectángulo azul con un borde rojo de 3 píxeles:



svgRef:= SVG_New

objectRef:=SVG_New_rect (svgRef;10;10;200;100;0;0;"red";"blue";3)

Ejemplo 3

Dibuje un rectángulo azul con bordes redondeados (color de relleno y del borde, grueso de línea por defecto):



svgRef:=SVG_New

objectRef:=SVG_New_rect(svgRef;10;10;100;100;20)

Ejemplo 4

Dibujo de un rectángulo azul claro con esquinas redondeadas y un borde azul (grueso de línea por defecto):



svgRef:=SVG_New

objectRef:=SVG_New_rect(svgRef;10;10;200;100;-1;50;"blue";"lightblue")

⚙ SVG_New_regular_polygon

SVG_New_regular_polygon (objetoSVGPadre ; ancho ; numLados {; x {; y {; colorPrimerPlano {; colorFondo {; anchoTrazo}}}}) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
ancho	Entero largo	→	Diámetro del círculo circundante
numLados	Entero largo	→	Número de lados
x	Entero largo	→	Coordenada centro en el eje X
y	Entero largo	→	Coordenada centro en el eje Y
colorPrimerPlano	Cadena	→	Nombre del color o del degradé
colorFondo	Cadena	→	Nombre del color o del degradé
anchoTrazo	Real	→	Grueso del trazo
Resultado	Ref_SVG	↩	Referencia del polígono

Descripción

El comando *SVG_New_regular_polygon* dibuja un polígono regular con el número de lados definido por *numLados* circunscrito en el círculo de diámetro *ancho* en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

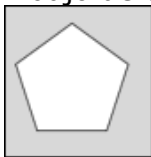
Los parámetros opcionales *x* e *y* permiten precisar el centro del círculo. Si se omiten, la figura se dibujará en la esquina superior izquierda del documento.

Los parámetros opcionales *colorPrimerPlano* y *colorFondo* contienen respectivamente el nombre del color de la línea y del color de fondo. (Para más información sobre los colores, consulte los comandos del tema **Colores y degradés**).

El parámetro opcional *anchoTrazo* contiene el tamaño del trazo expresado en píxeles. Su valor por defecto es 1.

Ejemplo 1

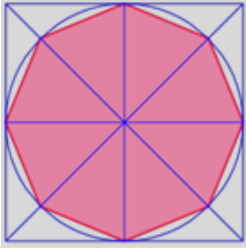
Dibujo de un pentágono (color de relleno y de borde, grueso de línea por defecto):



```
svgRef:=SVG_New  
objectRef:=SVG_New_regular_polygon(svgRef;100;5)
```

Ejemplo 2

Dibujo de un octágono, de su círculo circunscrito y de las líneas de trazo:



```
svgRef:=SVG_New
$width:=200
$sides:=8
objectRef:=SVG_New_regular_polygon(svgRef;$width;$sides;0;0;"crimson";"palevioletred";2)
```

```
$radius:=$width/2
```

```
objectRef:=SVG_New_rect(svgRef;0;0;$width;$width;0;0;"blue";"none")
```

```
objectRef:=SVG_New_line(svgRef;0;$radius;$width;$radius;"blue")
```

```
objectRef:=SVG_New_line(svgRef;$radius;0;$radius;$width;"blue")
```

```
objectRef:=SVG_New_line(svgRef;0;0;$width;$width;"blue")
```

```
objectRef:=SVG_New_line(svgRef;$width;0;0;$width;"blue")
```

```
objectRef:=SVG_New_circle(svgRef;$radius;$radius;$radius;"blue";"none")
```


SVG_PATH_ARC

SVG_PATH_ARC (objetoSVGPadre ; radioX ; radioY ; x ; y {; rotacion {; trazoArc}})

Parámetro	Tipo	Descripción
objetoSVGPadre	Ref_SVG	➔ Referencia de trazo
radioX	Entero largo	➔ Radio del elipse en el eje X
radioY	Entero largo	➔ Radio del elipse en el eje Y
x	Entero largo	➔ Coordenada del punto de destino en el eje X
y	Entero largo	➔ Coordenada del punto de destino en el eje Y
rotacion	Entero largo	➔ Valor de rotación
trazoArc	Entero largo	➔ Trazo del arco

Descripción

El comando `SVG_PATH_ARC` dibuja un arco elíptico, desde el punto actual al punto (x, y) , al final del trazo referenciado por `objetoSVGPadre`. Si `objetoSVGPadre` no es una referencia de un trazo (elemento 'path'), se genera un error.

El tamaño y la orientación de la elipse son definidos por dos radios (`radioX`, `radioY`) y un valor `rotacion` en el eje X que indica la rotación de la elipse en su conjunto con respecto al sistema de coordenadas actual.

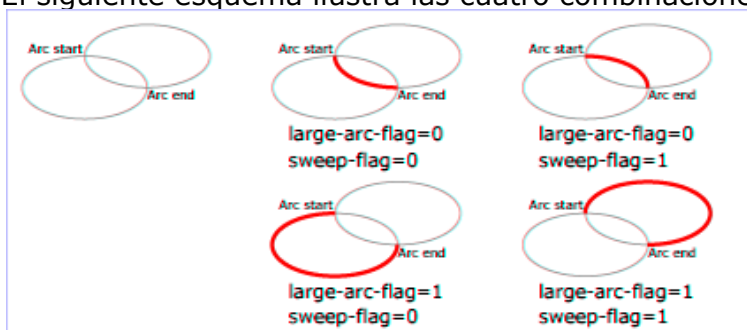
El parámetro opcional `trazoArc` permite aplicar una combinación de restricciones que van a determinar cómo se dibujará el arco. La restricción `large-arc-flag` permite elegir (o no) el más grande de los dos arcos posibles (mayor a 180°) y la restricción `sweep-flag` define la dirección del trazo (ángulo positivo o negativo).

Los siguientes valores, representan las cuatro posibles combinaciones de las dos restricciones:

- 0: `large-arc-flag = 0`, `sweep-flag = 1`
- 1: `large-arc-flag = 1`, `sweep-flag = 0`
- 2: `large-arc-flag = 0`, `sweep-flag = 0`
- 3: `large-arc-flag = 1`, `sweep-flag = 1`

Cuando `large-arc-flag` es igual a 1, se dibuja el arco más grande (y el más pequeño cuando es igual a 0). Cuando `sweep-flag` es igual a 1, el arco se dibuja en un ángulo positivo (y en un ángulo negativo cuando es igual a 0).

El siguiente esquema ilustra las cuatro combinaciones posibles:



Por defecto, el valor de `trazoArc` es 0 (`large-arc-flag=0`, `sweep-flag=1`)

Ejemplo

Ver los ejemplos del comando `SVG_New_path`.

⚙️ SVG_PATH_CLOSE

SVG_PATH_CLOSE (objetoSVGPadre)

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia de trazo

Descripción

El comando *SVG_PATH_CLOSE* cierra el subtrazo actual referenciado por *objetoSVGPadre* con el dibujo de una línea recta del punto actual al punto inicial. Si *objetoSVGPadre* no es la referencia de un trazo (elemento 'path'), se genera un error.

Ejemplo

Ver los ejemplos del comando *SVG_New_path*.

⚙ SVG_PATH_CURVE

SVG_PATH_CURVE (objetoSVGPadre {; controlInicioX ; controlInicioY} ; controlFinX ; controlFinY ; x ; y)

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia de trazo
controlInicioX	Entero largo	→	Coordenada del punto de control en el eje X
controlInicioY	Entero largo	→	Coordenada del punto de control en el eje Y
controlFinX	Entero largo	→	Coordenada del punto de control en el eje X
controlFinY	Entero largo	→	Coordenada del punto de control en el eje Y
x	Entero largo	→	Coordenada del punto de destino en el eje X
y	Entero largo	→	Coordenada del punto de destino en el eje Y

Descripción

El comando *SVG_PATH_CURVE* añade una curva de Bézier cúbica del punto actual al punto de coordenadas (x, y) al trazo referenciado por *objetoSVGPadre*. Si *objetoSVGPadre* no es una referencia de un trazo (elemento 'path'), se genera un error.

Los parámetros *controlInicioX* y *controlInicioY* permiten especificar la posición del punto de control al inicio de la curva. Si se omiten, el primer punto de control se supone que es el reflejo del segundo punto de control del comando anterior con respecto al punto actual.

Los parámetros *controlFinX* y *controlFinY* permiten especificar la posición del punto de control al final de la curva.

Ejemplo

Ver los ejemplos del comando [SVG_New_path](#).

⚙ SVG_PATH_LINE_TO

SVG_PATH_LINE_TO (objetoSVGPadre ; x {; y}{; x2 ; y2 ; ... ; xN ; yN})

Parámetro	Tipo	Descripción
objetoSVGPadre	Ref_SVG	➔ Referencia del elemento padre
x	Entero largo	➔ Coordenada en el eje X del nuevo punto
y	Entero largo	➔ Coordenada en el eje Y del nuevo punto

Descripción

El comando *SVG_PATH_LINE_TO* añade uno o más segmentos rectos al trazo referenciado por *objetoSVGPadre*. Si *objetoSVGPadre* no es una referencia de un trazo (elemento 'path'), se genera un error.

Los parámetros *x* e *y* permiten especificar la posición inicial del trazo en el contenedor SVG.

- Si sólo se pasa el parámetro *x*, la línea se trazará horizontalmente desde el punto actual (*xc*, *yc*) al punto (*x*, *yc*).
- Si se pasan *x* e *y*, una línea se dibujará desde el punto actual (*xc*, *yc*) al punto (*x*, *y*).
- Si se pasan varios pares de coordenadas, los diferentes puntos se sumarán sucesivamente. En este caso, si el último par de coordenadas está incompleto (falta *y*), se ignorará.

Ejemplo

Ver los ejemplos del comando *SVG_New_path*

⚙ SVG_PATH_MOVE_TO

SVG_PATH_MOVE_TO (objetoSVGPadre ; x ; y)

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia de trazo
x	Entero largo	→	Coordenada en el eje X
y	Entero largo	→	Coordenada en el eje Y

Descripción

El comando *SVG_PATH_MOVE_TO* comienza una nueva subtrazado en el punto de las coordenadas (x, y) dadas en el trazo referenciado por *objetoSVGPadre*. Si *objetoSVGPadre* no es una referencia de un trazo (elemento 'path'), se genera un error.

El efecto que produce es como si la "pluma" se levantara y se moviera a una nueva ubicación. El punto actual se convierte en el nuevo punto de partida que tendrá en cuenta el comando *SVG_PATH_CLOSE*.

Ejemplo

Consulte los ejemplos del comando *SVG_New_path*.

⚙️ SVG_PATH_QCURVE

SVG_PATH_QCURVE (objetoSVGPadre {; controlX ; controlY} ; x ; y)

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia de trazo
controlX	Entero largo	→	Coordenada del punto de control en el eje X
controlY	Entero largo	→	Coordenada del punto de control en el eje Y
x	Entero largo	→	Coordenada del punto de destino en el eje X
y	Entero largo	→	Coordenada del punto de destino en el eje Y

Descripción

El comando *SVG_PATH_CURVE* añade una curva Bezier cuadrática del punto actual al punto de coordenadas (x, y) a la línea referenciada por *objetoSVGPadre*. Si *objetoSVGPadre* no es una referencia de un trazo (elemento 'path'), se genera un error.

Los parámetros opcionales *controlX* y *controlY* se pueden utilizar para especificar la posición del punto de control al inicio de la curva. Si se omiten, el primer punto de control se supone que es un reflejo del segundo punto de control del comando anterior con respecto al punto actual.

Ejemplo

Ver los ejemplos del comando *SVG_New_path*.

⚙ SVG_Use

SVG_Use (objetoSVGPadre ; id {; x ; y ; ancho ; alto {; modo}}) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	➔	Referencia del elemento padre
id	Cadena	➔	Nombre del símbolo
x	Entero largo	➔	Posición X del rectángulo de visualización
y	Entero largo	➔	Posición Y del rectángulo de visualización
ancho	Entero largo	➔	Ancho del rectángulo de visualización
alto	Entero largo	➔	Alto del rectángulo de visualización
modo	Cadena	➔	Adaptación al rectángulo de visualización
Resultado	Ref_SVG	➔	Referencia del objeto SVG

Descripción

El comando *SVG_Use* pone una ocurrencia del símbolo en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG o si *id* no es el nombre de un objeto del documento SVG, se genera un error.

Un símbolo se utiliza para definir los objetos gráficos, nunca renderiza directamente, pero puede crear ser instanciado mediante el comando *SVG_Use*.

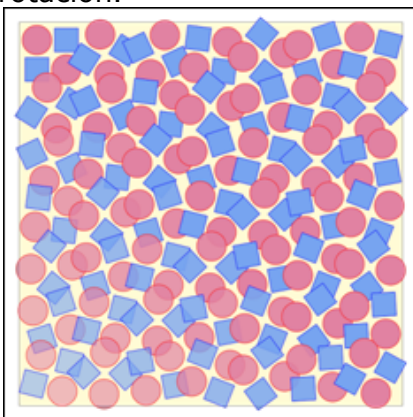
El parámetro *id* especifica el nombre del símbolo.

Los parámetros opcionales *x*, *y*, *ancho* y *alto* especifican el rectángulo del área de visualización (atributo 'Viewbox').

El parámetro opcional *modo* permite indicar si el gráfico debe adaptarse y cómo, al tamaño del rectángulo de visualización. (Ver el comando *SVG_New*).

Ejemplo

Definimos un gráfico compuesto de dos círculos rojos y dos cuadrados azules. Luego utilizamos este gráfico en un bucle para crear 36 ocurrencias con diferentes posiciones, opacidad y rotación.



```
$SVG:=SVG_New
  `Dibujo del fondo
SVG_New_rect($SVG;20;20;650;650;0;0;"gray";"lemonchiffon")
  `Definición de un símbolo compuesto por 2 cuadrados y 2 círculos
$Symbol:=SVG_Define_symbol($SVG;"MySymbol";0;0;110;110;"true")
SVG_New_circle($Symbol;30;30;25;"red";"palevioletred")
```

```
SVG_New_rect($Symbol;10;60;40;40;0;0;"blue";"cornflowerblue")
```

```
SVG_New_rect($Symbol;60;10;40;40;0;0;"blue";"cornflowerblue")
```

```
SVG_New_circle($Symbol;80;80;25;"red";"palevioletred")
```

```
`En un grupo...
```

```
$g:=SVG_New_group($SVG)
```

```
`...posicionado a 20 unidades de la esquina superior izquierda del documento...
```

```
SVG_SET_TRANSFORM_TRANSLATE($g;20;20)
```

```
`...poner 36 patrones variando la posición, opacidad y rotación
```

```
For($x;0;540;90)&NBSP;&NBSP; `6 columnas
```

```
For($y;0;540;90)&NBSP;&NBSP;&NBSP; `6 líneas
```

```
$use:=SVG_Use($g;"MySymbol";$x;$y;110;110)
```












```
SVG_SET_OPACITY($use;100-($y/12)+($x/12))
```

```
SVG_SET_TRANSFORM_ROTATE($use;($x*(18/50))+($y*(18/50));($x+55);($y+55))
```

```
End for
```

```
End for
```


Documentos

-  SVG_CLEAR
-  SVG_Copy
-  SVG_Export_to_picture
-  SVG_Export_to_XML
-  SVG_New
-  SVG_Open_file
-  SVG_Open_picture
-  SVG_SAVE_AS_PICTURE
-  SVG_SAVE_AS_TEXT
-  SVG_SET_DOCUMENT_VARIABLE
-  SVG_Validate_file

SVG_CLEAR

```
SVG_CLEAR {( objetoSVG )}
```

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia del objeto SVG

Descripción

El comando **SVG_CLEAR** libera la memoria ocupada por el objeto SVG designado por *objetoSVG*.

objetoSVG puede ser un objeto SVG raíz (creado con los comandos *SVG_New*, *SVG_Copy* o *SVG_Open_file*) o todo objeto SVG válido.

Si no se pasa *objetoSVG*, el comando libera todos los objetos SVG creados utilizando *SVG_New*, *SVG_Copy* o *SVG_Open_file*. Esta sintaxis es útil durante la fase de desarrollo en la cual una referencia SVG puede crearse, pero la memoria no fue liberada porque un error que impide terminar la ejecución del método. En un final de desarrollo, toda referencia SVG que ya no se utilice debe liberarse con el comando *SVG_CLEAR*.

SVG_Copy

SVG_Copy (objetoSVG) -> Resultado

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia del objeto SVG a copiar
Resultado	Ref_SVG	↩	Referencia del nuevo objeto SVG

Descripción

El comando *SVG_Copy* crea un nuevo documento SVG que es una copia del documento referenciado por *objetoSVG*.

El comando devuelve una cadena de 16 caracteres (*SVG_Ref*) que consiste de la referencia en memoria de la estructura virtual del documento. Esta referencia se debe utilizar con los otros comandos del componente.

Importante: una vez que ya no la necesite, no olvide de llamar al comando *SVG_CLEAR* con esta referencia para liberar la memoria.

Ejemplo

```
svgRef:=SVG_New
...
svgRef Copy:=SVG_Copy(svgRef)
```

⚙️ SVG_Export_to_picture

SVG_Export_to_picture (objetoSVG {; tipoExport}) -> Resultado

Parámetro	Tipo	Descripción
objetoSVG	Ref_SVG	➔ Referencia del objeto SVG
tipoExport	Entero largo	➔ 0 = No guardar la fuente de datos 1 = Copiar la fuente de datos 2 (por defecto) = Fuente de datos propia
Resultado	Imagen	➔ Imagen renderizada por el motor SVG

Descripción

El comando *SVG_Export_to_picture* devuelve la imagen descrita por la estructura SVG referenciada por *objetoSVG*.

El parámetro opcional *tipoExport* permite definir la forma en que la fuente de datos XML debe ser manejada por el comando. Para más información sobre este parámetro, consulte la descripción del comando **SVG EXPORT TO PICTURE** de 4D. Si se omite este parámetro, el valor por defecto es 1, [Copy XML Data Source](#).

Ejemplo

```
svgRef:=SVG_New(500;200;Test component)
...
MyPicture:=SVG_Export_to_picture(svgRef;0)

SVG_CLEAR(svgRef)
```

⚙️ SVG_Export_to_XML

SVG_Export_to_XML (objetoSVG) -> Resultado

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia del objeto SVG
Resultado	Texto	↩	Texto XML del documento SVG

Descripción

El comando *SVG_Export_to_XML* devuelve el texto XML de descripción de la estructura SVG referenciada por *objetoSVG*.

Ejemplo

```
svgRef:=SVG_New(500;200;Test component)
...
MyText:=SVG_Export_to_XML(svgRef)

SVG_CLEAR(svgRef)
```

```
SVG_New {( ancho ; alto {; titulo {; descripcion {; rectangulo {; visualizacion}}}} )} -> Resultado
```

Parámetro	Tipo		Descripción
ancho	Entero largo	→	Ancho del documento en puntos
alto	Entero largo	→	Alto del documento en puntos
titulo	Cadena	→	Texto del documento
descripcion	Cadena	→	Descripción
rectangulo	Booleano	→	Definir el rectángulo de visualización
visualizacion	Entero	→	Formato de visualización de la imagen
Resultado	Ref_SVG	↪	Referencia del objeto SVG

Descripción

El comando `SVG_New` crea un nuevo documento SVG y devuelve su número de referencia.

Los parámetros opcionales `ancho` y `alto` permiten delimitar el espacio del documento SVG. Estos dos parámetros deben expresarse en puntos usuario ('px'), si desea especificar otra unidad, debe utilizar el comando `SVG_SET_DIMENSIONS`.

Los parámetros opcionales `titulo` y `descripcion` permiten dar información sobre el contenido.

Si pasa `True` en el parámetro opcional `rectangulo`, el rectángulo de visualización (atributo "viewBox") se ajusta automáticamente al tamaño del documento creado.

Nota: es posible modificar las coordenadas del rectángulo de visualización del gráfico y para ajustar la adaptación de la imagen con más precisión utilizando el comando `SVG_SET_VIEWBOX`.

El parámetro opcional `visualizacion` puede utilizarse para indicar si el gráfico debe adaptarse al tamaño del documento. Puede pasar en este parámetro una de las siguientes constantes del tema `Formatos de visualización de imágenes de 4D`: `Scaled to fit_prop_centered` o `Scaled to Fit`.

El comando devuelve una cadena de 16 caracteres (`SVG_Ref`) que consiste de la referencia en memoria de la estructura virtual del documento. Esta referencia se debe utilizar con los otros comandos del componente.

Importante: una vez que ya no la necesite, no olvide de llamar al comando `SVG_CLEAR` con esta referencia para liberar la memoria.

Ejemplo

```
svgRef:=SVG_New
svgRef:=SVG_New(500;200)
svgRef:=SVG_New(900;700;"Prueba componente SVG";"Este es un ejemplo";True;Scaled to fit)
```

⚙️ SVG_Open_file

SVG_Open_file (rutaAcceso) -> Resultado

Parámetro	Tipo		Descripción
rutaAcceso	Cadena	→	Ruta de acceso del documento SVG a abrir
Resultado	Ref_SVG	↩	Referencia del documento abierto

Descripción

El comando *SVG_Open_file* analiza (y valida con la DTD) el documento SVG que se encuentra en el lugar designado por el parámetro *rutaAcceso* y devuelve una referencia SVG (cadena de 16 caracteres) para este documento.

Importante: una vez no lo necesite, no olvide llamar al comando *SVG_CLEAR* con esta referencia con el fin de liberar la memoria.

⚙️ SVG_Open_picture

SVG_Open_picture (imagen) -> Resultado

Parámetro	Tipo		Descripción
imagen	Imagen	→	Campo o variable tipo imagen 4D
Resultado	Ref_SVG	↩	Referencia del documento SVG

Descripción

El comando *SVG_Open_picture* analiza una imagen SVG y devuelve una referencia SVG para esta imagen. Si *imagen* no contiene una imagen SVG, el comando devuelve una cadena vacía.

Importante: una vez no lo necesite, no olvide llamar al comando *SVG_CLEAR* con esta referencia con el fin de liberar la memoria.

Ejemplo

```
READ PICTURE FILE("";$picture)
If(OK=1)
  $ref:=SVG_Open_picture($picture)
  ...
  SVG_CLEAR($ref)
End if
```


⚙️ SVG_SAVE_AS_PICTURE

SVG_SAVE_AS_PICTURE (objetoSVG ; doc {; codec})

Parámetro	Tipo	Descripción
objetoSVG	Ref_SVG	➔ Referencia del objeto SVG
doc	Cadena	➔ Nombre del documento o Ruta de acceso completa del documento
codec	Cadena	➔ Identificación de codec de imagen

Descripción

El comando **SVG_SAVE_AS_PICTURE** escribe el contenido del objeto SVG designado por *objetoSVG* en el archivo imagen designado por *nomArchivo*. Si *objetoSVG* no es un documento SVG, se genera un error.

En *nomArchivo*, puede pasar la ruta de acceso completa del archivo o sólo su nombre, en cuyo caso el archivo se creará junto al archivo de estructura de la base. Si pasa una cadena vacía ("") en *nomArchivo*, aparece la caja de diálogo Guardar archivo para que el usuario pueda especificar el nombre, la ubicación y el formato del archivo a crear.

El parámetro opcional *codec* permite especificar el formato en el cual guardar la imagen. Si parámetro se omite, la imagen se guarda en formato png.

El comando **SVG_SAVE_AS_PICTURE** modifica el valor de la variable (si la hay) designada por el comando **SVG_SET_DOCUMENT_VARIABLE**.

Ejemplo

```
svgRef:=SVG_New(500;200;"Sales statistics")
...
SVG_SAVE_AS_PICTURE(svgRef;"test.png") //Save
SVG_SAVE_AS_PICTURE(svgRef;"test.gif";".gif")
SVG_CLEAR(svgRef)
```

⚙️ SVG_SAVE_AS_TEXT

SVG_SAVE_AS_TEXT (objetoSVG {; doc})

Parámetro	Tipo	Descripción
objetoSVG	Ref_SVG →	Referencia del objeto SVG
doc	Cadena →	Nombre del documento o Ruta de acceso completa del documento

Descripción

El comando **SVG_SAVE_AS_TEXT** escribe el contenido del objeto SVG especificado por *objetoSVG* en el archivo disco especificado por *nomArchivo*. Si *objetoSVG* no es un documento SVG, se genera un error.

En *nomArchivo*, puede pasar la ruta de acceso completa del archivo o sólo su nombre, en cuyo caso el archivo se creará junto al archivo de estructura de la base. Si pasa una cadena vacía (""), en *nomArchivo*, aparece la caja de diálogo Guardar archivo para que el usuario pueda especificar el nombre, la ubicación y el formato del archivo a crear.

El comando **SVG_SAVE_AS_TEXT** modifica el valor de la variable (si la hay) designada por el comando **SVG_SET_DOCUMENT_VARIABLE**.

Ejemplo

```
svgRef:=SVG_New(500;200;"Estadísticas de ventas")
...
SVG_SAVE_AS_TEXT(svgRef;"test.svg") //El documento se guarda junto a la estructura
SVG_CLEAR(svgRef)
```

⚙️ SVG_SET_DOCUMENT_VARIABLE

SVG_SET_DOCUMENT_VARIABLE (puntero)

Parámetro	Tipo		Descripción
puntero	Puntero	→	Puntero a la variable a definir

Descripción

El método **SVG_SET_DOCUMENT_VARIABLE** permite definir un puntero a la variable de la base local que se actualiza después de cada llamada a **SVG_SAVE_AS_PICTURE** o **SVG_SAVE_AS_TEXT**. Debe llamar a este método sólo una vez por sesión (por ejemplo en un método de inicialización).

En el parámetro *puntero*, pase un puntero a la variable cuyo valor quiere seguir (generalmente, la variable sistema **Document**).

Para borrar el enlace, pase un puntero Nil en el parámetro *puntero*.

SVG_Validate_file



















SVG_Validate_file (rutaAcceso) -> Resultado

Parámetro	Tipo		Descripción
rutaAcceso	Cadena	→	Ruta de acceso del documento SVG a validar
Resultado	Booleano	↩	Verdadero si el documento corresponde a la DTD

Descripción

El comando *SVG_Validate_file* intenta validar el documento SVG almacenado en la *rutaAcceso* con la DTD (1.0). El comando devuelve True si el documento está bien y False de lo contrario.

Estructura y definiciones

-  SVG_Define_clip_path
-  SVG_Define_filter
-  SVG_Define_linear_gradient
-  SVG_Define_marker
-  SVG_Define_pattern
-  SVG_Define_radial_gradient
-  SVG_Define_shadow
-  SVG_Define_solidColor
-  SVG_Define_style
-  SVG_DEFINE_STYLE_WITH_ARRAYS
-  SVG_Define_symbol
-  SVG_DELETE_OBJECT
-  SVG_Get_default_encoding
-  SVG_New_group
-  SVG_SET_DEFAULT_ENCODING
-  SVG_Set_description
-  SVG_SET_PATTERN_CONTENT_UNITS
-  SVG_SET_PATTERN_UNITS
-  SVG_Set_title

⚙ SVG_Define_clip_path

SVG_Define_clip_path (objetoSVGPadre ; IDRecorte) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
IDRecorte	Texto	→	Nombre del trazo de recorte
Resultado	Ref_SVG	↩	Referencia del trazo de recorte

Descripción

El comando *SVG_Define_clip_path* define un nuevo trazo de recorte en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es (o no pertenece a) un documento SVG, se genera un error.

El parámetro *IDRecorte* designa el nombre del trazo de recorte. Este nombre se utiliza para asociar un trazo de recorte con un objeto. Si un elemento con el mismo nombre ya existe en el documento, se genera un error.

Ver también: <http://www.w3.org/TR/2001/REC-SVG-20010904/masking.html#EstablishingANewClippingPath>

Ejemplo

Consulte el ejemplo del comando *SVG_SET_CLIP_PATH*.

⚙ SVG_Define_filter

SVG_Define_filter (objetoSVGPadre ; id {; marcoX ; marcoX {; anchoMarco ; altoMarco {; unidadMarco ; unidadFiltro}}}) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
id	Cadena	→	Nombre del símbolo
marcoX	Entero largo	→	Coordenada en el eje X
marcoY	Entero largo	→	Coordenada en el eje Y
anchoMarco	Entero largo	→	Ancho del rectángulo objetivo
altoMarco	Entero largo	→	Alto del rectángulo objetivo
unidadMarco	Cadena	→	Sistema de coordenadas del marco
unidadFiltro	Cadena	→	Sistema de valores del filtro
Resultado	Ref_SVG	↩	Referencia del filtro

Descripción

El comando *SVG_Define_filter* define un nuevo filtro en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

Un filtro es una sucesión de operaciones gráficas que se aplicarán al elemento de destino. El elemento filtro nunca se renderiza directamente, se aplica a un objeto mediante el comando *SVG_SET_FILTER*.

El parámetro *id* especifica el nombre del marcador. El nombre se utiliza para asociar un filtro con un objeto. Si existe un elemento con el mismo nombre, se reemplazará.

Los parámetros opcionales *marcoX*, *marcoY*, *largoMarco* y *anchoMarco* definen una región rectangular del documento al que se aplica el filtro.

El parámetro opcional *unidadMarco* define el sistema de coordenadas de los 4 parámetros anteriores. Los valores esperados: "userSpaceOnUse" u "objectBoundingBox" (valor por defecto).

El parámetro opcional *unidadFiltro* define el sistema de coordenadas para las longitudes y las propiedades de definición del filtro. Los valores esperados son: "userSpaceOnUse" (valor por defecto) u "objectBoundingBox".

Ejemplo

En este ejemplo, queremos efectuar las siguientes operaciones:

- crear un rectángulo con fondo azul 50%
- crear un filtro blur 4% y aplicarlo a este rectángulo
- guardar el resultado en un archivo SVG en disco.

```
$Dom_SVG:=SVG_New
```

```
//creation of a rectangle with 50% blue background
```

```
$Dom_rect:=SVG_New_rect($Dom_SVG;50;50;50;50;0;0;"blue:50";"blue:50")
```

```
//creación de un rectángulo fondo blur 4%
```

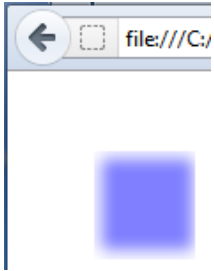
```
$Dom_filter:=SVG_Define_filter($Dom_SVG;"blur")
SVG_Filter_Blur($Dom_filter;4)
SVG_Filter_Offset($Dom_filter;4)

//aplicación de este filtro al rectángulo
SVG_SET_FILTER($Dom_rect;"blur")

//guardar el resultado en un archivo SVG
SVG_SAVE_AS_TEXT($Dom_SVG;System folder(Desktop)+"test.svg")

SVG_CLEAR($Dom_SVG)
```

Resultado:



⚙️ SVG_Define_linear_gradient

SVG_Define_linear_gradient (objetoSVGPadre ; id ; colorInicio ; colorFin {; rotacion {; spreadMethod {; x1 ; y1 ; x2 ; y2}{; iniciarOffsetColor ; finOffsetColor}} }) -> Resultado

Parámetro	Tipo	Descripción
objetoSVGPadre	Ref_SVG	➔ Referencia del elemento padre
id	Cadena	➔ Nombre del degradé
colorInicio	Cadena	➔ Color de inicio
colorFin	Cadena	➔ Color de fin
rotacion	Entero	➔ Rotación del vector de degradé
spreadMethod	Texto	➔ Modo de difusión (pad, reflect o repeat)
x1	Real	➔ Coordenada x1 del vector de degradé (Si se omite = 0)
y1	Real	➔ Coordenada y1 del vector de degradé (Si se omite = 0)
x2	Real	➔ Coordenada x2 del vector de degradé (Si se omite = 0)
y2	Real	➔ Coordenada y2 del vector de degradé (Si se omite = 0)
iniciarOffsetColor	Real	➔ Valor del porcentaje de offset para iniciar color
finOffsetColor	Real	➔ Valor del porcentaje de offset para finalizar color
Resultado	Cadena	➔ Referencia del degradé

Descripción

El comando **SVG_Define_linear_gradient** define un nuevo degradé lineal en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

Un degradé consiste en una transición de color progresiva continua de un color a otro a lo largo de un vector. Una vez definido, los degradés se llaman en un elemento gráfico dado. Debe indicar si el elemento debe llenarse o bordearse con el degradé llamado.

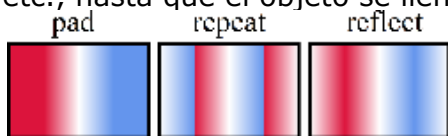
El parámetro *id* especifica el nombre del degradé. Si existe un elemento con el mismo nombre, se reemplazará. Este nombre se utilizará para llamar al degradé cada vez que una expresión color se espere utilizando la sintaxis "url (#ID)".

Los parámetros *colorInicio* y *colorFin* especifican los colores que se utilizan para iniciar y terminar el degradé.

El parámetro opcional *rotacion* define la posición y la dirección del vector de degradé (ver ejemplo).

El parámetro opcional *spreadMethod* permite definir el relleno cuando el degradé comienza o termina al interior de los límites del objeto *objetoSVGPadre*. Puede pasar en este parámetro una de las siguientes cadenas:

- "pad": utilizar los colores extremos del degradé para llenar el resto del área.
- "reflect": reflejar indefinidamente el motivo del degradé en inicio/fin luego fin/inicio luego inicio/fin, etc., hasta que se llena el objeto.
- "repeat", repetir indefinidamente el motivo del degradé en inicio/fin, inicio/fin, inicio/fin, etc., hasta que el objeto se llena.



Si este parámetro se omite, el efecto del valor "pad" se utiliza.

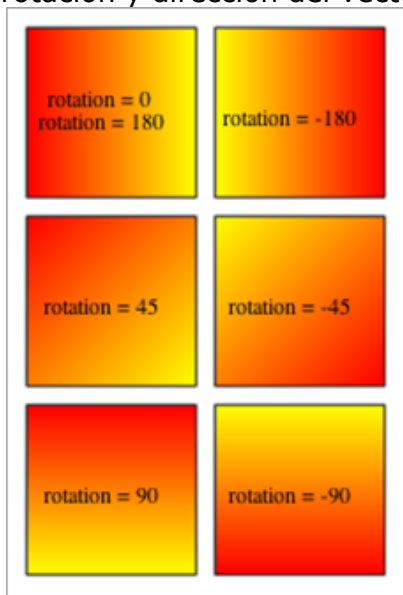
Los parámetros opcionales *x1*, *y1*, *x2* y *y2* permiten definir el vector del degradé. Este vector ofrece los puntos de inicio y final utilizados por el motor de renderización. Debe pasar en este parámetro los porcentajes expresados en forma de radios (0...1).

A partir de la v14, puede pasar los parámetros *inicioOffsetColor* y *finOffsetColor* opcionales para definir el valor porcentual, respectivamente, del color inicial y el color final offset. Puede pasar un valor real <1, o un valor entre 0 y 100 para establecer el porcentaje, es decir, "0.1" y/o "10" se interpretan como el 10%.

Al pasar un valor negativo se interpreta como 0% para el parámetro *inicioOffsetColor* y un 100% para el parámetro *finOffsetColor*. Si pasa un valor > 100, se interpreta como 100%.

Ejemplo 1

Dibujar 6 cuadrados donde cada uno utilice un servidor de pintura de degradé lineal que varia la rotación y dirección del vector de degradé:



```
$svg:=SVG_New
```

```
SVG_Define_linear_gradient($svg;"demoGradient_1";"red";"yellow")
SVG_New_rect($svg;10;10;90;90;0;0;"black";"url(#demoGradient_1)")
SVG_New_text($svg;"rotation = 0\rrotation = 180";50;40;"";-1;-1;Align_center)
```

```
SVG_Define_linear_gradient($svg;"demoGradient_2";"red";"yellow";-180)
SVG_New_rect($svg;110;10;90;90;0;0;"black";"url(#demoGradient_2)")
SVG_New_text($svg;"rotation = -180";150;50;"";-1;-1;Align_center)
```

```
SVG_Define_linear_gradient($svg;"demoGradient_3";"red";"yellow";45)
SVG_New_rect($svg;10;110;90;90;0;0;"black";"url(#demoGradient_3)")
SVG_New_text($svg;"rotation = 45";50;150;"";-1;-1;Align_center)
```

```
SVG_Define_linear_gradient($svg;"demoGradient_4";"red";"yellow";-45)
SVG_New_rect($svg;110;110;90;90;0;0;"black";"url(#demoGradient_4)")
SVG_New_text($svg;"rotation = -45";150;150;"";-1;-1;Align_center)
```

```
SVG_Define_linear_gradient($svg;"demoGradient_5";"red";"yellow";90)
SVG_New_rect($svg;10;210;90;90;0;0;"black";"url(#demoGradient_5)")
SVG_New_text($svg;"rotation = 90";50;250;"";-1;-1;Align_center)
```

```
SVG_Define_linear_gradient($svg;"demoGradient_6";"red";"yellow";-90)
SVG_New_rect($svg;110;210;90;90;0;0;"black";"url(#demoGradient_6)")
SVG_New_text($svg;"rotation = -90";150;250;"";-1;-1;Align_center)
```

```
//Guardar documento
SVG_SAVE_AS_TEXT($svg;"test.svg")
//Liberar memoria
SVG_CLEAR($svg)
```

Ejemplo 2

Ejemplo utilizando los parámetros *distanciaColorInicio* y *distanciaColorFin* (añadidos en v14):

```
$Dom_node:=SVG_Define_linear_gradient($Dom_svg;"clicked";"black:50";"black:20";-90;"reflect";0;80)
```

dará la definición:

```
<linearGradient id="clicked spreadMethod="reflect x1="0" x2="0" y1="1" y2="0"> <stop
offset="0%" stop-color="black" stop-opacity="0.5"/> <stop offset="80%" stop-color="black"
stop-opacity="0.2"/> </linearGradient>
```

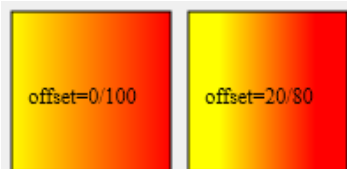
Ejemplo 3

Este ejemplo ilustra el efecto de los parámetros *iniciarOffsetColor* y *finOffsetColor*:

```
$svg:=SVG_New

SVG_Define_linear_gradient($svg;"demoGradient_1";"red";"yellow";-180;"reflect")
SVG_New_rect($svg;10;10;90;90;0;0;"black";"url(#demoGradient_1)")
SVG_New_text($svg;"offset=0/100";50;50;"";-1;-1;Align_center)

SVG_Define_linear_gradient($svg;"demoGradient_2";"red";"yellow";-180;"reflect";20;80)
SVG_New_rect($svg;110;10;90;90;0;0;"black";"url(#demoGradient_2)")
SVG_New_text($svg;"offset=20/80";150;50;"";-1;-1;Align_center)
//Save document
SVG_SAVE_AS_TEXT($svg;"test2.svg")
//Free up memory
SVG_CLEAR($svg)
```



⚙ SVG_Define_marker

SVG_Define_marker (objetoSVGPadre ; id {; x ; y {; ancho ; alto {; orientacion}} }) -> Resultado

Parámetro	Tipo	Descripción
objetoSVGPadre	Ref_SVG	→ Referencia del elemento padre
id	Cadena	→ Nombre del símbolo
x	Entero largo	→ Coordenada en el eje X del punto de referencia
y	Entero largo	→ Coordenada en el eje Y del punto de referencia
ancho	Entero largo	→ Ancho del marcador
alto	Entero largo	→ Alto del marcador
orientacion	Entero largo	→ Orientación del marcador
Resultado	Ref_SVG	↩ Referencia del marcador

Descripción

El comando *SVG_Define_marker* crea un marcador en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

Un objeto marcador se utiliza para dibujar una flecha o marcadores múltiples (los puntos de una curva, por ejemplo). Un marcador se adjunta a un elemento SVG con el comando [SVG_SET_MARKER](#).

El parámetro *id* especifica el nombre del marcador. El nombre se utilizará para asociar una marcador con un objeto. Si existe un elemento con el mismo nombre, se reemplazará.

Los parámetros opcionales *x* e *y* especifican las coordenadas del punto de referencia que tiene que coincidir exactamente con la posición del marcador.

Los parámetros opcionales *ancho* y *alto* especifican el ancho y alto del rectángulo de renderizado.

El parámetro opcional *orientacion* permite ajustar la orientación del marcador. Un valor entre 0 y 360 representa el ángulo entre el eje X del marcador y el del espacio usuario. Si este parámetro se omite o si su valor no está en el intervalo 0 - 360 la ubicación será calculada automáticamente por el motor de renderización en función del objeto.

Ejemplo

Consulte el ejemplo del comando [SVG_SET_MARKER](#).

⚙ SVG_Define_pattern

SVG_Define_pattern (objetoSVGPadre ; idPatron {; ancho {; alto {; x {; y {; unidad {; viewBox}}}}}) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
idPatron	Texto	→	Nombre del patrón
ancho	Entero largo	→	Ancho del patrón
alto	Entero largo	→	Alto del patrón
x	Entero largo	→	Posición x del patrón
y	Entero largo	→	Posición y del patrón
unidad	Texto	→	Unidad de longitud y posiciones
viewBox	Texto	→	Rectángulo de visualización
Resultado	Ref_SVG	↪	Referencia del patrón

Descripción

El comando *SVG_Define_pattern* define un nuevo patrón personalizado en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es (o no pertenece a) un documento SVG, se genera un error.

El parámetro *IDPatron* especifica el nombre del patrón. Este nombre se utiliza para asociar el patrón a un objeto. Si existe un elemento con el mismo nombre, se genera un error.

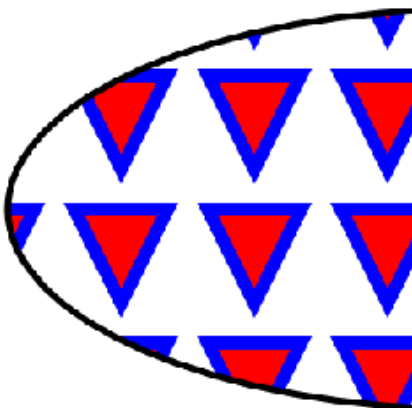
Los parámetros opcionales *ancho*, *alto*, *x*, *y*, *unidad* y *viewBox* definen el rectángulo de referencia del patrón, en otras palabras, la forma en que el mosaico del patrón se ubicará y espaciará.

El patrón se asociará como relleno o contorno al pasar la cadena "url (#id)" como valor cuando se espera una expresión color.

Ver también: <http://www.w3.org/TR/SVG/pservers.html#Patterns>

Ejemplo 1

Definición de un patrón y uso del patrón para el relleno de una elipse:



```
//Definición del patrón
$Dom_pattern:=SVG_Define_pattern($Dom_SVG;"MyPattern";100;100;0;0;"";"0 0 10 10")
$Dom_path:=SVG_New_path($Dom_pattern;0;0)

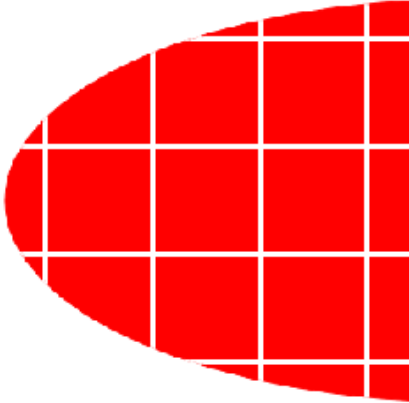
SVG_PATH_MOVE_TO($Dom_path;0;0)
SVG_PATH_LINE_TO($Dom_path;7;0)
SVG_PATH_LINE_TO($Dom_path;3,5;7)
```

```
SVG_PATH_CLOSE($Dom_path)
SVG_SET_FILL_BRUSH($Dom_path;"red")
SVG_SET_STROKE_BRUSH($Dom_path;"blue")
```

```
//Dibujo de una elipse llena con el patrón
$Dom_ellipse:=SVG_New_ellipse($Dom_SVG;400;200;350;150;"black";"url(#MyPattern)";5)
```

Ejemplo 2

Definición de un patrón y uso para el relleno y los contornos de una elipse:



```
//Definición del patrón
$Dom_pattern:=SVG_Define_pattern($Dom_SVG;"MyPattern ";80;80;0;0;"";"0 0 20 20")
$Dom_rect:=SVG_New_rect($Dom_pattern;0;0;20;20;0;0;"white";"red")
```

```
//Dibujo de una elipse
$Dom_ellipse:=SVG_New_ellipse($Dom_SVG;400;200;350;150)
```

```
//Uso del patrón para el relleno y los contornos
SVG_SET_FILL_BRUSH($Dom_ellipse;"url(#MyPattern)")
SVG_SET_STROKE_BRUSH($Dom_ellipse;"url(#MyPattern)")
```

⚙ SVG_Define_radial_gradient

```
SVG_Define_radial_gradient ( objetoSVGPadre ; id ; colorInicio ; colorFin {; cx ; cy ; r {; fx ; fy}} ) -> Resultado
```

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	➔	Referencia del elemento padre
id	Cadena	➔	Nombre del degradé
colorInicio	Cadena	➔	Color de inicio
colorFin	Cadena	➔	Color de fin
cx	Entero	➔	Coordenada del centro de colorFin en el eje X
cy	Entero	➔	Coordenada del centro de colorFin en el eje Y
r	Entero	➔	Radio de colorFin
fx	Entero	➔	Coordenada del centro de colorInicio en el eje X
fy	Entero	➔	Coordenada del centro de colorInicio en el eje Y
Resultado	Cadena	➔	Referencia del degradé

Descripción

El comando *SVG_Define_radial_gradient* define un nuevo degradé radial en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

Un degradé consiste en una transición de color progresiva continua de un color a otro a lo largo de un vector. Una vez definido, los degradés se llaman en un elemento gráfico dado. Debe indicar si el elemento debe llenarse o bordearse con el degradé llamado.

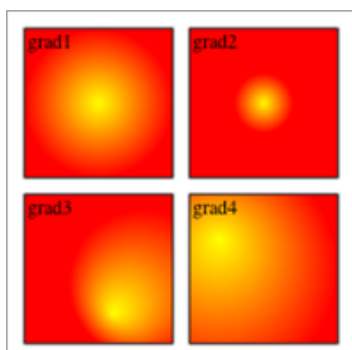
El parámetro *id* especifica el nombre del degradé. Si existe un elemento con el mismo nombre, se reemplazará. Este nombre se utilizará para llamar al degradé cada vez que una expresión color se espere utilizando la sintaxis "url (#ID)".

Los parámetros *colorInicio* y *colorFin* especifican los colores que se utilizan para iniciar y terminar el degradé.

Los parámetros opcionales *cx*, *cy* y *r* especifican, en porcentaje, el círculo limite externo de colorFin del degradé. Sus valores deben estar entre 0 y 100.

Los parámetros opcionales *fx* y *fy* especifican, en porcentaje, el punto focal del degradé. El *colorInicio* comienza en el punto [*fx*, *fy*]. Sus valores deben estar entre 0 y 100. Si estos argumentos se omiten, este punto coincide con [*cx*, *cy*].

Ejemplo



`$svg:=SVG_New`

`SVG_Define_radial_gradient($svg;"grad1";"yellow";"red")`
`SVG_New_rect($svg;10;10;90;90;0;0;"black";"url(#grad1)")`
`SVG_New_text($svg;"grad1";12;10)`

`SVG_Define_radial_gradient($svg;"grad2";"yellow";"red";50;50;20;50;50)`
`SVG_New_rect($svg;110;10;90;90;0;0;"black";"url(#grad2)")`
`SVG_New_text($svg;"grad2";112;10)`

`SVG_Define_radial_gradient($svg;"grad3";"yellow";"red";80;60;50;60;80)`
`SVG_New_rect($svg;10;110;90;90;0;0;"black";"url(#grad3)")`
`SVG_New_text($svg;"grad3";12;110)`

`SVG_Define_radial_gradient($svg;"grad4";"yellow";"red";20;50;80;20;30)`
`SVG_New_rect($svg;110;110;90;90;0;0;"black";"url(#grad4)")`
`SVG_New_text($svg;"grad4";112;110)`

` Guardar el documento

`SVG_SAVE_AS_TEXT($svg;"test.svg")`

` Liberar la memoria

`SVG_CLEAR($svg)`

⚙ SVG_Define_shadow

SVG_Define_shadow (objetoSVGPadre ; id {; desviación {; offsetX {; offsetY}} }) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	➔	Referencia del elemento padre
id	Cadena	➔	Nombre del filtro
desviación	Entero largo	➔	Valor de dispersión de la sombra
offsetX	Entero largo	➔	Offset en el eje X
offsetY	Entero largo	➔	Offset en el eje Y
Resultado	Ref_SVG	➔	Referencia del filtro

Descripción

El comando *SVG_Define_shadow* establece un nuevo filtro de sombra en el contenedor de SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

Este filtro se aplicará a los objetos para los cuales se quiere una sombra utilizando el comando *SVG_SET_FILTER*.

El parámetro *id* especifica el nombre del filtro. Este nombre se utiliza para asociar un filtro a un objeto. Si existe un elemento con el mismo nombre, se reemplazará.

El parámetro opcional *desviacion* define la intensidad de la dispersión de la sombra. Valor por defecto: 4.

Los parámetros opcionales *offsetX* y *offsetY* definen respectivamente el offset horizontal y vertical de la sombra con respecto al objeto. Valor por defecto: 4.

Ejemplo

Declaración de un filtro que permite generar una sombra sobre un texto:



```
$svg:=SVG_New  
  
$text:=SVG_New_text($svg;"SVG";52;76-45;"Verdana";45)  
SVG_SET_FONT_COLOR($text;"red")  
` Definir el filtro  
SVG_Define_shadow($svg;"myShadow")  
` y aplicarlo al texto  
SVG_SET_FILTER($text;"myShadow")
```

⚙ SVG_Define_solidColor

SVG_Define_solidColor (objetoSVGPadre ; id ; color {; opacidad}) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
id	Cadena	→	Nombre del color
color	Cadena	→	Color expression
opacidad	Entero largo	→	Opacidad
Resultado	Ref_SVG	↩	Referencia del color

Descripción

El comando *SVG_Define_solidColor* define un nuevo color personalizado en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

El parámetro *id* especifica el nombre del color. El nombre se utilizará para asociar un color a un objeto. Si existe un elemento con el mismo nombre, se reemplazará.

El parámetro *color* es una expresión color reconocida por SVG (ver [Colores y degradados](#)).

El parámetro opcional *opacidad* puede usarse para especificar una opacidad (de 0 a 100) para este color. Si el parámetro se omite, la opacidad será del 100%.

El conjunto de colores de esta manera se asociará con el relleno o el trazo de pintura pasando la cadena "url (# ID)" como el valor cuando se espera una expresión de color.

Ejemplo

```
` Definir un azul al 50%
SVG_Define_solidColor($svg;"MyColor";"blue";50)
...
SVG_New_rect($svg;0;0;20;20;0;0;"url(#MyColor)";"url(#MyColor)")
...
$line:=SVG_New_line(10;10;100;100)
SVG_SET_STROKE_BRUSH($line;"url(#MyColor)")
```

⚙ SVG_Define_style

SVG_Define_style (objetoSVGPadre ; estilo {; tipo {; media}}) -> Resultado

Parámetro	Tipo	Descripción
objetoSVGPadre	Ref_SVG	➔ Referencia del elemento padre
estilo	Texto	➔ Definición del estilo O Ruta de acceso del archivo a utilizar
tipo	Texto	➔ Tipo de contenido
media	Texto	➔ Descriptor de media
Resultado	Ref_SVG	➔ Referencia del estilo

Descripción

El comando *SVG_Define_style* se utiliza para definir una nueva hoja de estilo en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es (o no pertenece a) un documento SVG, se genera un error.

El parámetro *estilo* permite la incorporación de hojas de estilo directamente en un contenido SVG:

- Si el parámetro *estilo* contiene una ruta de acceso válida a un archivo CSS, la definición de estilo se hace utilizando un mecanismo de referencia de hojas de estilo externas. La ruta, si comienza con el carácter # o por la cadena "file:", expresa una ruta relativa cuya raíz es la carpeta "Recursos" de la base.
- El parámetro *estilo* también puede ser un URL de tipo "http:// ...", en este caso, la hoja de estilo será referenciada como recurso externo.
- También, puede pasar en el parámetro *estilo* una URL relativa a la subcarpeta "Resources/SVG/" de la base. Esto es particularmente útil en modo cliente/servidor, cuando los archivos se guardan en la carpeta "Resources". Los URLs relativos pueden comenzar por:
 - "/", para indicar la ruta "~/Resources/SVG/"
 - "./", para indicar la ruta "~/Resources/"
 - "../", para indicar la carpeta de la base.

Para ejemplos de URLs relativas, consulte el comando **SVG_New_image**.

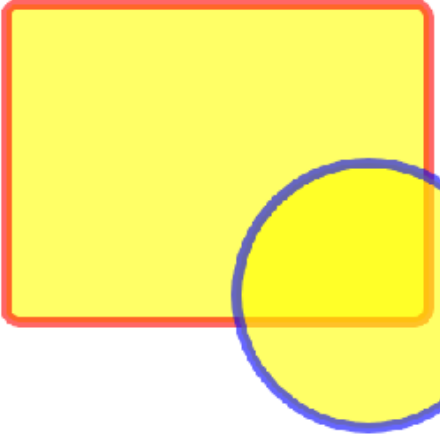
El parámetro opcional *tipo* especifica el lenguaje de la hoja de estilo del contenido del elemento. El valor por defecto es "text/css".

El parámetro opcional *media* especifica la media de destino para la información de estilo. Si omite este parámetro, el valor por defecto es "all". Si el valor no se incluye en la lista de tipos de medias reconocidas por CSS2, se genera un error.

Ver también: <http://www.w3.org/TR/SVG/styling.html#StyleElement>

Ejemplo 1

Definición de un estilo incorporado y superposición de uno de los elementos:



```
//Definición del estilo
$Txt_style=".colored {fill: yellow; fill-opacity: 0.6; stroke: red;stroke-width:8; stroke-opacity: 0.6}"
SVG_Define_style($Dom_SVG;$Txt_style)

//Creación de un grupo y afectación de un estilo por defecto
$Dom_g:=SVG_New_group($Dom_SVG)
SVG_SET_CLASS($Dom_g;"colored")

//Dibujo de un rectángulo
$Dom_rect:=SVG_New_rect($Dom_g;25;30;320;240;10;10;"";"")

//Dibujo de un círculo y superposición del estilo con un color de contorno personalizado
$Dom_circle:=SVG_New_circle($Dom_g;300;250;100;"";"")
SVG_SET_STROKE_BRUSH($Dom_circle;"blue")
```

Ejemplo 2

Referenciación del archivo "mystyle.css" ubicado en la carpeta "dev" de la carpeta "Resources":



```
//Definición del estilo
SVG_Define_style($Dom_svg;"#dev/monstyle.css")

//Creación de un grupo y asignación de un estilo por defecto
$Dom_g:=SVG_New_group($Dom_SVG)
SVG_SET_CLASS($Dom_g;"colored")

//Dibujo de un rectángulo
$Dom_rect:=SVG_New_rect($Dom_g;25;30;320;240;10;10;"";"")
```

mystyle.css file:

```
.colored {fill: red; fill-opacity: 0.6; stroke: blue; stroke-width:8; stroke-opacity: 0.6}
```

⚙️ SVG_DEFINE_STYLE_WITH_ARRAYS

```
SVG_DEFINE_STYLE_WITH_ARRAYS ( objetoSVG ; puntArrayNoms ; puntArrayValores { ;  
className { ; tipo { ; media { ; titulo } } } )
```

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	➡	Referencia del objeto SVG
puntArrayNoms	Puntero	➡	Puntero al array de los nombres de estilos
puntArrayValores	Puntero	➡	Puntero al array de los valores de estilos
className	Texto	➡	Nombre de la clase del estilo CSS
tipo	Texto	➡	Tipo del contenido
media	Texto	➡	Descriptor de media
titulo	Texto	➡	Nombre del estilo

Descripción

El comando **SVG_DEFINE_STYLE_WITH_ARRAYS** define los estilos del objeto SVG (utilizando arrays) designado por el parámetro *objetoSVG*.

- Si el parámetro *objetoSVG* designa el elemento raíz, los estilos se definen como elementos "style" incluidos en la sección "defs" (Internal Style Sheet). En este caso, el parámetro *nomclass* es obligatorio (si falta, devuelve un error). Luego puede asignar la hoja de estilo *className* a los objetos SVG pasando su nombre al método **SVG_SET_CLASS** (ver el ejemplo 1).
- Si el parámetro *objetoSVG* designa un elemento SVG diferente al elemento raíz, el estilo está definido como un atributo de estilo para este elemento (Inline Style Sheet) (ver ejemplo 2).

El parámetro opcional *tipo* especifica el lenguaje de la hoja de estilo del contenido del elemento. El valor por defecto es "text/css".

El parámetro opcional *media* indica la media de destino para la información de estilo. Si omite este parámetro, el valor por defecto utilizado es "all". Si el valor no está incluido en la lista de medias reconocidas por CSS2, se genera un error.

El parámetro opcional *titulo* añade un atributo de tipo "title".

Ejemplo 1

Ejemplo de definición de estilos internos:

```
ARRAY TEXT($arrnames;0)  
ARRAY TEXT($arrvalues;0)  
APPEND TO ARRAY($arrnames;"fill")  
APPEND TO ARRAY($arrvalues;"black")  
APPEND TO ARRAY($arrnames;"font-family")  
APPEND TO ARRAY($arrvalues;"'Lucida Grande' Verdana")  
APPEND TO ARRAY($arrnames;"font-size")  
APPEND TO ARRAY($arrvalues;"20px")  
APPEND TO ARRAY($arrnames;"text-align")  
APPEND TO ARRAY($arrvalues;"center")  
  
$svg:=SVG_New  
SVG_DEFINE_STYLE_WITH_ARRAYS($svg;->$arrnames;->$arrvalues;"title")
```

```
$object:=SVG_New_textArea($svg;"Hello World!";10;10;200;310)
SVG_SET_CLASS($object;"title")
```

Este método genera el siguiente código:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?> <svg
xmlns="http://www.w3.org/2000/svg"> <defs id="4D"> <style
type="text/css">.title{fill:red;font-family:'Lucida Grande' Verdana;font-size:20px;text-align:center;}
</style> </defs> <textArea class="title" height="310" width="200" x="10" y="10">Hello
World!</textArea> </svg>
```

Ejemplo 2

Ejemplo de definición de estilos inline:

```
ARRAY TEXT($arrnames;0)
ARRAY TEXT($arrvalues;0)
APPEND TO ARRAY($arrnames;"fill")
APPEND TO ARRAY($arrvalues;"black")
APPEND TO ARRAY($arrnames;"font-family")
APPEND TO ARRAY($arrvalues;"'Lucida Grande' Verdana")
APPEND TO ARRAY($arrnames;"font-size")
APPEND TO ARRAY($arrvalues;"20px")
APPEND TO ARRAY($arrnames;"text-align")
APPEND TO ARRAY($arrvalues;"center")

$svg:=SVG_New
$object:=SVG_New_textArea($svg;"Hello World!";10;10;200;310)
SVG_DEFINE_STYLE_WITH_ARRAYS($object;->$arrnames;->$arrvalues)
```

Este método genera el siguiente código:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?> <svg
xmlns="http://www.w3.org/2000/svg"> <textArea height="310" style="fill:red;font-
family:'Lucida Grande' Verdana;font-size:20px;text-align:center;" width="200" x="10" y="10">Hello
World!</textArea> </svg>
```

⚙ SVG_Define_symbol

SVG_Define_symbol (objetoSVGPadre ; id {; x {; y {; ancho {; alto {; modo}}}} }) ->
Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
id	Cadena	→	Nombre del símbolo
x	Entero largo	→	Posición X del rectángulo de visualización
y	Entero largo	→	Posición Y del rectángulo de visualización
ancho	Entero largo	→	Ancho del rectángulo de visualización
alto	Entero largo	→	Alto del rectángulo de visualización
modo	Cadena	→	Adaptación al rectángulo de visualización
Resultado	Ref_SVG	↩	Referencia del símbolo

Descripción

El comando *SVG_Define_symbol* crea un símbolo en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

Un objeto símbolo se utiliza para definir los objetos gráficos que pueden ser instancias utilizando el comando *SVG_Use*.

El parámetro *id* especifica el nombre del símbolo.

Los parámetros opcionales *x*, *y*, *ancho* y *alto* especifican el rectángulo del área de visualización (atributo 'Viewbox').

El parámetro opcional *modo* permite indicar si el gráfico debe adaptarse y cómo, al tamaño del rectángulo de visualización. Para obtener más información sobre este punto, consulte la descripción del comando *SVG_New*.

El parámetro opcional *opacidad* puede usarse para especificar una opacidad (de 0 a 100) para este color. Si el parámetro se omite, la opacidad será del 100%.

El conjunto de colores de esta manera se asociará con el relleno o el trazo de pintura pasando la cadena "url (# ID)" como el valor cuando se espera una expresión de color.

Ejemplo

Consulte la descripción del comando *SVG_Use*.

⚙ SVG_DELETE_OBJECT

SVG_DELETE_OBJECT (objetoSVG)

Parámetro	Tipo	Descripción
objetoSVG	Ref_SVG →	Referencia de un elemento SVG

Descripción

El comando *SVG_DELETE_OBJECT* borra el objeto SVG designado por *objetoSVG* del documento al que pertenece. Se genera un error si *objetoSVG* no es una referencia válida.

⚙ SVG_Get_default_encoding

SVG_Get_default_encoding -> Resultado

Parámetro	Tipo		Descripción
Resultado	Texto		Conjunto de caracteres

Descripción

El comando *SVG_Get_default_encoding* devuelve la codificación utilizada durante la creación de un nuevo documento.

⚙ SVG_New_group

SVG_New_group (objetoSVGPadre {; id {; url {; objetivo}}}) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
id	Cadena	→	Nombre del grupo
url	Cadena	→	Enlace externo
objetivo	Cadena	→	Objetivo del enlace
Resultado	Ref_SVG	↩	Referencia del grupo

Descripción

El comando *SVG_New_group* crea un grupo en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un grupo o un documento SVG válido, se genera un error.

El grupo (elemento "g") permite agrupar varios elementos gráficos enlazados, que heredará las propiedades del grupo.

El parámetro opcional *id* permite asignar un nombre al grupo. Los grupos con nombre son necesarios para varios propósitos, tales como la animación y los objetos reutilizables.

El parámetro opcional *url* permite asociar un enlace externo. Se puede hacer clic a continuación en los objetos del grupo (similar al elemento 'a' de HTML).

El parámetro opcional *objetivo* especifica el nombre del objetivo en el que el documento se abrirá cuando se activa el enlace. Los valores esperados son los de la especificación HTML a los que se suman los valores "new" para abrir en una nueva ventana y 'none', que es equivalente a no procesar este atributo.

Nota: los enlaces externos se ignoran cuando el SVG se muestra en un objeto imagen (variable o campo) de un formulario 4D. La gestión de las referencias externas es efectuada por el motor de renderización. En estas condiciones, el resultado puede depender de la plataforma y el software de visualización.

Ejemplo 1

Creación de un grupo de líneas, todas del mismo color:



```
$SVG:=SVG_New
$group:=SVG_New_group($SVG)
  `Asignar un color a los elementos del grupo
SVG_SET_STROKE_BRUSH($group;"firebrick")
$newobject:=SVG_New_line($group;100;300;300;100;"";5)
$newobject:=SVG_New_line($group;300;300;500;100;"";10)
$newobject:=SVG_New_line($group;500;300;700;100;"";15)
$newobject:=SVG_New_line($group;700;300;900;100;"";20)
$newobject:=SVG_New_line($group;900;300;1100;100;"";25)
```

Ejemplo 2

Creación de un texto en el que se puede hacer clic:

www.w3.org

```
$SVG:=SVG_New
```

```
$group:=SVG_New_group($SVG;"w3Link";"http://www.w3.org";"new")
```

```
$newobject:=SVG_New_text($group;"www.w3.org";10;10;"arial";12;Underline;Align left;"blue")
```

⚙️ SVG_SET_DEFAULT_ENCODING

SVG_SET_DEFAULT_ENCODING {(codificacion)}

Parámetro	Tipo		Descripción
codificacion	Cadena	→	Conjunto de caracteres

Descripción

El comando *SVG_SET_DEFAULT_ENCODING* permite definir la codificación que se utilizará durante las próximas creaciones de documentos.

Si se omite el parámetro *codificacion*, el comando restablece el conjunto de caracteres por defecto: "UTF-8".

⚙ SVG_Set_description

SVG_Set_description (objetoSVGPadre ; descripcion) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
descripcion	Cadena	→	Texto del comentario
Resultado	Ref_SVG	↩	Referencia de la descripción

Descripción

El comando *SVG_Set_description* define un texto para el elemento SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un elemento SVG, se genera un error.

Una descripción por lo general se utiliza para insertar un comentario o un texto explicativo en el código SVG.

Ejemplo

```
$SVG:=SVG_New
$g:=SVG_group($SVG)
SVG_Set_title($g;"Ventas de la empresa por región")
SVG_Set_description($g;"Diagrama en barra de las ventas de la empresa por región.")
...
```

⚙️ SVG_SET_PATTERN_CONTENT_UNITS

SVG_SET_PATTERN_CONTENT_UNITS (patronObjeto ; sisCoord)

Parámetro	Tipo		Descripción
patronObjeto	Ref_SVG	→	Referencia del patrón a modificar
sisCoord	Texto	→	Sistema de coordenadas a utilizar

Descripción

El comando `SVG_SET_PATTERN_CONTENT_UNITS` define el sistema de coordenadas para los contenidos de los patrones designados por *patronObjeto*. Si *patronObjeto* no es un patrón, se genera un error.

El parámetro *sisCoord* especifica el nombre del sistema que se utilizará. Debe ser igual a "userSpaceOnUse" o "objectBoundingBox", de lo contrario se genera un error.

Ver también: <http://www.w3.org/TR/SVG/pservers.html#Patterns>

⚙️ SVG_SET_PATTERN_UNITS

SVG_SET_PATTERN_UNITS (patronObjeto ; sisCoord)

Parámetro	Tipo		Descripción
patronObjeto	Ref_SVG	→	Referencia del patrón a modificar
sisCoord	Texto	→	Sistema de coordenadas a utilizar

Descripción

El comando *SVG_SET_PATTERN_UNITS* define el sistema de coordenadas para los atributos x, y, ancho y alto del patrón designado por *patronObjeto*. Si *patronObjeto* no es un patrón, se genera un error.

El parámetro *sisCoord* especifica el nombre del sistema que se utilizará. Debe ser igual a "userSpaceOnUse" o "objectBoundingBox", de lo contrario se genera un error.

Ver también: <http://www.w3.org/TR/SVG/pservers.html#Patterns>

⚙ SVG_Set_title

SVG_Set_title (objetoSVGPadre ; titulo) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	→	Referencia del elemento padre
titulo	Cadena	→	Texto del título
Resultado	Ref_SVG	↩	Referencia del título

Descripción


El comando *SVG_Set_title* define un título para el elemento SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un elemento SVG, se genera un error.

Un título es un dato textual que no está incluido en la imagen renderizada pero se usa para la estructuración de documentos complejos. Algunos motores de renderización SVG utilizan el texto de este elemento para mostrar un mensaje de ayuda cuando el ratón se mueve sobre el objeto.

Ejemplo

```
$SVG:=SVG_New
$rec:=SVG_New_rect($SVG;20;20;650;650;0;0;"gray";"lemonchiffon")
SVG_Set_title($rec;"Rectángulo de fondo")
$Symbol:=SVG_Define_symbol($SVG;"MySymbol";0;0;110;110;"true")
SVG_Set_title($Symbol;" Definición de un símbolo compuesto por 2 cuadrados y 2 círculos ")
...
```

Filtros

-  Filtros SVG
-  SVG_Filter_Blend
-  SVG_Filter_Blur
-  SVG_Filter_Offset

Filtros SVG

Los comandos del tema "**Filtros**" permiten definir los efectos de filtro que se aplican a los elementos SVG. Un efecto de filtro consiste en una sucesión de operaciones gráficas, aplicadas a un gráfico fuente, que produce un gráfico modificado.

El resultado del efecto de filtro se renderiza en el dispositivo objetivo en lugar del gráfico fuente original.

Un filtro se define utilizando el comando *SVG_Define_filter*, que se encuentra en el tema "**Estructura y definiciones**" y se aplica utilizando el comando *SVG_SET_FILTER*, que se encuentra en el tema "**Atributos**". Los comandos del tema "**Filtros**" permiten construir las operaciones de filtrado o "primitivas de filtro".

⚙️ SVG_Filter_Blend

SVG_Filter_Blend (refFiltro ; imagen ; imagenFondo {; modo {; nombre}}) -> Resultado

Parámetro	Tipo		Descripción
refFiltro	Ref_SVG	➔	Referencia de filtro
imagen	Cadena	➔	Imagen fuente
imagenFondo	Cadena	➔	Imagen de fondo fuente
modo	Cadena	➔	Modo de mezcla
nombre	Cadena	➔	Objetivo de la primitiva de filtro
Resultado	Ref_SVG	➔	Referencia de la primitiva

Descripción

El comando *SVG_Filter_Blend* define un filtro de composición para el filtro *refFiltro* y devuelve su referencia. Si *refFiltro* no es una referencia de filtro, se genera un error.

Este filtro se compone de dos fuentes, *imagenFondo* e *imagen*, con la ayuda de los modos de mezcla que utiliza actualmente el software de imágenes.

El parámetro opcional *modo* permite definir el modo de combinación de los píxeles utilizados para la mezcla (ver la especificación). Su valor debe ser: "normal" (valor por defecto), "multiply", "screen", "darken" o "lighten".

El parámetro opcional *nombre* es el nombre asignado al resultado de esta primitiva de filtro.

Nota: a partir de 4D v14 R5, este comando funciona en Windows con Direct2D activado en modo software (ver la constante [Direct2D disabled](#) en la descripción del comando **SET DATABASE PARAMETER**).

Ejemplo

En un formulario, mostramos dos imágenes SVG idénticas, luego creamos un filtro "blend" y lo asignamos a la derecha. Este filtro es una combinación de los filtros "offset" y "blur":

```
$root:=SVG_New(400;400;"filters test") //definición de la primera imagen (izquierda)
$rect:=SVG_New_rect($root;10;10;380;100;0;0;"darkblue";"white";1)
SVG_SET_FILL_BRUSH($root;"orange")
$textAreaRef:=SVG_New_textArea($root;"Hello World!";10;10;380;100;"arial";60;Normal;Align
center)
<>pict1:=SVG_Export_to_picture($root) //visualización de la primera imagen

$root2:=SVG_New(400;400;"filters test") //definición de la imagen derecha idéntica

//creación del filtro
$filter:=SVG_Define_filter($root2;"MyShadow")
$vGraph:=True //aplicación de la capa gráfica - pase False para aplicar la capa alpha
If($vGraph)
    $ref1:=SVG_Filter_Blur($filter;2;"sourceGraphic";"blurResult") //"blurResult" se utilizará como
"entrada" del filtro offset
Else
    $ref1:=SVG_Filter_Blur($filter;2;"sourceAlpha";"blurResult") //"blurResult" se utilizará como
"input" del filtro offset
```

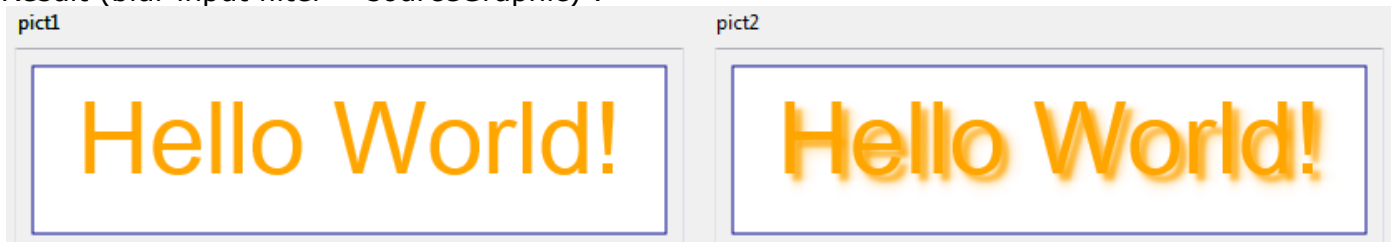
End if

```
//Adición del filtro offset
$ref2:=SVG_Filter_Offset($filter;5;5;"blurResult";"alphaBlurOffset")
//Adding blend filter
$ref3:=SVG_Filter_Blend($filter;"sourceGraphic";"alphaBlurOffset";"normal";"finalFilter";)

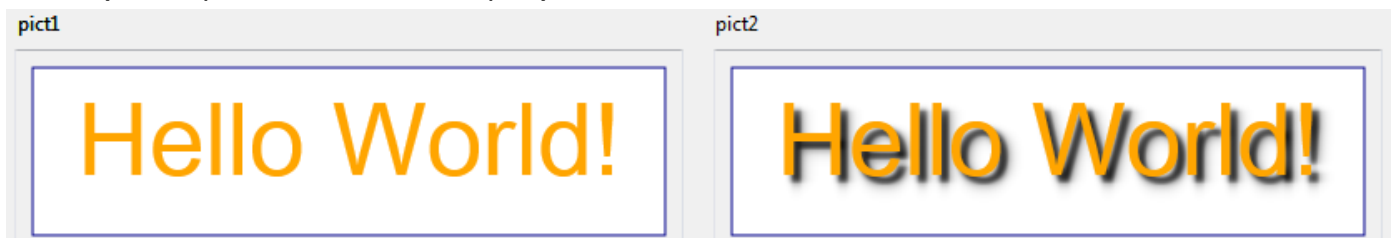
$rect2:=SVG_New_rect($root2;10;10;380;100;0;0;"darkblue";"white";1)
SVG_SET_FILL_BRUSH($root2;"orange")
$textAreaRef2:=SVG_New_textArea($root2;"Hello World!";10;10;380;100;"arial";60;Normal;Align
center)

SVG_SET_FILTER($textAreaRef2;"MyShadow") //aplicación del filtro final
<>pict2:=SVG_Export_to_picture($root2) //visualización de la segunda imagen
```

Result (blur input filter = sourceGraphic) :



Result (blur input filter = sourceAlpha) :



⚙ SVG_Filter_Blur

SVG_Filter_Blur (refFiltro ; desviación {; entrada {; nombre}}) -> Resultado

Parámetro	Tipo	Descripción
refFiltro	Ref_SVG	➔ Referencia de filtro
desviación	Real	➔ Desviación estándar para la operación de volver borroso
entrada	Cadena	➔ Fuerza de la primitiva de filtro
nombre	Cadena	➔ Objetivo de la primitiva de filtro
Resultado	Ref_SVG	➔ Referencia de la primitiva

Descripción

El comando *SVG_Filter_Blur* define un desenfoque Gaussiano para el filtro *refFiltro* y devuelve su referencia. Si *refFiltro* no es una referencia de filtro, se genera un error.

El parámetro *desviación* permite definir la desviación estándar para la operación de desenfoque. Si el número es un número entero, la misma desviación se aplicará a los ejes X e Y. Si el número incluye una parte decimal, la parte entera representa la desviación que se aplicará al eje X y la parte decimal representa la desviación que se aplicará al eje Y.

El parámetro opcional *entrada* identifica la fuente gráfica de la primitiva de filtro. Puede pasar:

- "sourceGraphic", indicando que el gráfico es la fuente del filtro (por defecto),
- o "sourceAlpha", que indica que el canal alpha es la fuente del filtro.

El parámetro opcional *nombre* es el nombre asignado al resultado de esta primitiva de filtro.

Nota: a partir de 4D v14 R5, este comando funciona en Windows con Direct2D activado en modo software (ver la constante [Direct2D disabled](#) en la descripción del comando **SET DATABASE PARAMETER**).

Ejemplo

En un formulario, mostramos dos imágenes SVG idénticas luego creamos un filtro "blur" y lo asignamos a la imagen a la derecha:

```
$root:=SVG_New(400;400;"filters test") //definición de primera imagen (izquierda)
$rect:=SVG_New_rect($root;10;10;380;100;0;0;"darkblue";"white";1)
SVG_SET_FILL_BRUSH($root;"orange")
$textAreaRef:=SVG_New_textArea($root;"Hello World!";10;10;380;100;"arial";60;Normal;Align
center)
<>pict1:=SVG_Export_to_picture($root) //visualización de la primera imagen

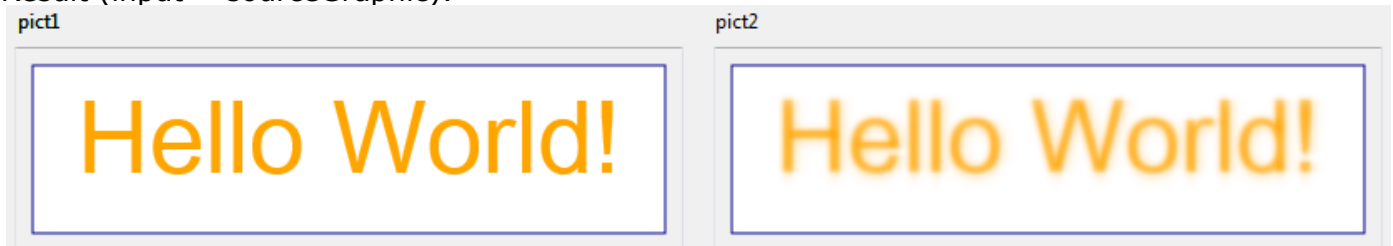
$root2:=SVG_New(400;400;"filters test") //definición de la imagen de la derecha idéntica

//creación del filtro
$filter1:=SVG_Define_filter($root2;"blur")
// filter definition
$vGraph:=True //aplicación en la capa gráfica - pase False para aplicar la capa alpha
If($vGraph)
    SVG_Filter_Blur($filter1;Deviation{Deviation};"sourceGraphic")
Else
    SVG_Filter_Blur($filter1;Deviation{Deviation};"sourceAlpha")
```

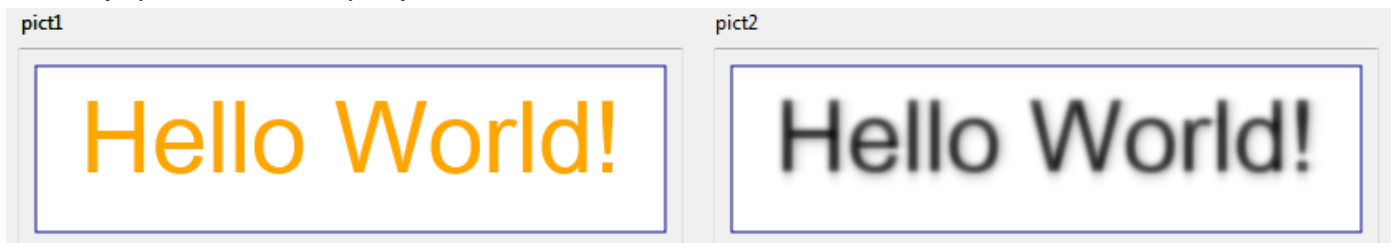
End if

```
$rect2:=SVG_New_rect($root2;10;10;380;100;0;0;"darkblue";"white";1) //definición de la imagen  
derecha idéntica  
SVG_SET_FILL_BRUSH($root2;"orange")  
$textAreaRef2:=SVG_New_textArea($root2;"Hello World!";10;10;380;100;"arial";60;Normal;Align  
center)  
  
SVG_SET_FILTER($textAreaRef2;"blur") //aplicación del filtro  
<>pict2:=SVG_Export_to_picture($root2) //visualización de la segunda imagen
```

Result (input = sourceGraphic):



Result (input = sourceAlpha):



⚙ SVG_Filter_Offset

SVG_Filter_Offset (refFiltro ; dx {; dy {; entrada {; nombre}} }) -> Resultado

Parámetro	Tipo		Descripción
refFiltro	Ref_SVG	➔	Referencia de filtro
dx	Entero largo	➔	Offset en eje X
dy	Entero largo	➔	Offset en eje Y
entrada	Cadena	➔	Fuente de la primitiva de filtro
nombre	Cadena	➔	Objetivo de la primitiva de filtro
Resultado	Ref_SVG	➔	Referencia de la primitiva

Descripción

El comando *SVG_Filter_Offset* define un offset para el filtro *refFiltro* y devuelve su referencia. Si *refFiltro* no es una referencia de filtro, se genera un error.

El parámetro *dx* es el valor del desplazamiento horizontal.

El parámetro opcional *dy* es el valor del desplazamiento vertical.

El parámetro opcional *entrada* identifica la fuente gráfica de la primitiva de filtro. Puede pasar:

- "sourceGraphic", indicando el gráfico como fuente del filtro (por defecto),
- o "sourceAlpha", que indica el canal alfa del gráfico como fuente del filtro.

El parámetro opcional *nombre* es el nombre asignado al resultado al resultado de esta primitiva de filtro.

Nota: a partir de 4D v14 R5, este comando trabaja bajo Windows con Direct2D activado en modo software (ver la constante [Direct2D disabled](#) en la descripción del comando **SET DATABASE PARAMETER**).

Ejemplo

En un formulario, mostramos dos imágenes SVG idénticas luego creamos un filtro "offset" y lo asignamos a la imagen de la derecha:

```
$root:=SVG_New(400;400;"filters test") //definición de la primera imagen (izquierda)
$rect:=SVG_New_rect($root;10;10;380;100;0;0;"darkblue";"white";1)
SVG_SET_FILL_BRUSH($root;"orange")
$textAreaRef:=SVG_New_textArea($root;"Hello World!";10;10;380;100;"arial";60;Normal;Align
center)
<>pict1:=SVG_Export_to_picture($root) //visualización de la primera imagen

$root2:=SVG_New(400;400;"filters test") //definición de la imagen de la derecha idéntica
$rect2:=SVG_New_rect($root2;10;10;380;100;0;0;"darkblue";"white";1)
SVG_SET_FILL_BRUSH($root2;"orange")
$textAreaRef2:=SVG_New_textArea($root2;"Hello World!";10;10;380;100;"arial";60;Normal;Align
center)

$filter:=SVG_Define_filter($root2;"Offset") //creación del filtro
SVG_Filter_Offset($filter;10;20)
SVG_SET_FILTER($textAreaRef2;"Offset") //aplicación del filtro

<>pict2:=SVG_Export_to_picture($root2) //visualización de la segunda imagen
```


Resultado:

















pict1

Hello World!

pict2

Hello World!

Texto

-  SVG_APPEND_TEXT_TO_TEXTAREA
-  SVG_Get_text
-  SVG_New_text
-  SVG_New_textArea
-  SVG_New_tspan
-  SVG_New_vertical_text
-  SVG_SET_FONT_COLOR
-  SVG_SET_FONT_FAMILY
-  SVG_SET_FONT_SIZE
-  SVG_SET_FONT_STYLE
-  SVG_SET_TEXT_ANCHOR
-  SVG_SET_TEXT_KERNING
-  SVG_SET_TEXT_LETTER_SPACING
-  SVG_SET_TEXT_RENDERING
-  SVG_SET_TEXT_WRITING_MODE
-  SVG_SET_TEXTAREA_TEXT

⚙ SVG_APPEND_TEXT_TO_TEXTAREA

SVG_APPEND_TEXT_TO_TEXTAREA (objetoSVG ; textoAdicional)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento texto
textoAdicional	Texto	→	Texto a agregar

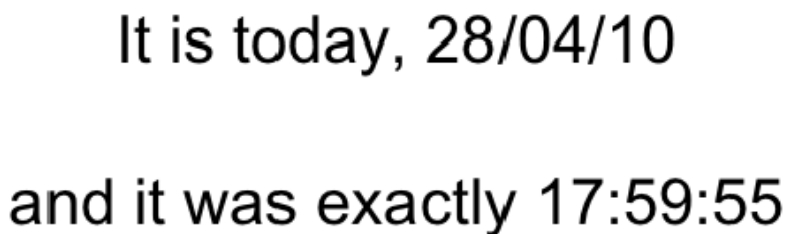
Descripción

El comando `SVG_APPEND_TEXT_TO_TEXTAREA` permite añadir texto al contenido textual del objeto texto designado por `objetoSVG`. Si `objetoSVG` no es un objeto "textArea", se genera un error.

Los caracteres retorno de línea son reemplazados automáticamente por los elementos "`<tbreak/>`".

Ejemplo

Añadiendo el siguiente texto:



It is today, 28/04/10
and it was exactly 17:59:55

```
//Mostrar los contornos utilizando el elemento 'rect'  
$Dom_rect:=SVG_New_rect($Dom_SVG;10;10;500;200;0;0;"blue:50";"none")  
  
//Crear el texto  
$Dom_text:=SVG_New_textArea($Dom_SVG;"It is today, ";10;30;500;200;"Arial";36;0;3)  
  
//Añadir la fecha y 2 CR  
SVG_APPEND_TEXT_TO_TEXTAREA($Dom_text;String(Current date)+"\r\r")  
  
//Finalmente, añadir la hora actual  
SVG_APPEND_TEXT_TO_TEXTAREA($Dom_text;"and it was exactly "+String(Current time))
```

⚙ SVG_Get_text

SVG_Get_text (objetoSVG) -> Resultado

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento texto
Resultado	Texto	↩	Contenido de texto

Descripción

El comando *SVG_Get_text* devuelve el contenido textual del elemento designado por *objetoSVG*. Si *objetoSVG* no es una referencia de objeto texto ('text', 'textArea' o 'tspan'), se genera un error.

En el caso de un objeto textArea, los elementos <tbreak/> se convierten en CR.

⚙️ SVG_New_text

SVG_New_text (objetoSVGPadre ; texto {; x {; y {; fuente | estiloFuente {; talla {; estilo {; alineación {; color {; rotacion {; interlineado {; estiramiento}}}}}}}} }) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	➔	Referencia del elemento padre
texto	Texto	➔	Texto a insertar
x	Real	➔	Coordenada en el eje X
y	Real	➔	Coordenada en el eje Y
fuentes estiloFuente	Texto	➔	Nombre de la fuente o definición de estilo
talla	Entero largo	➔	Tamaño de los caracteres en puntos
estilo	Entero largo	➔	Estilo de los caracteres
alineación	Entero largo	➔	Alineación
color	Cadena	➔	Color del texto
rotacion	Real	➔	Angulo de rotación del texto
interlineado	Real	➔	Espacio entre líneas en puntos
estiramiento	Real	➔	Factor de estiramiento horizontal
Resultado	Ref_SVG	➔	Referencia del objeto texto SVG

Descripción

El comando **SVG_New_text** inserta el texto en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

Nota: a partir de 4D v15, el comando **SVG_New_text** soporta el texto multiestilo simple (el texto puede contener los diferentes estilos, pero sin atributos SPAN anidados). Ver el ejemplo 5.

Los parámetros opcionales *x* e *y* permiten especificar la posición en el eje X e Y de la esquina superior del primer carácter del texto. Este punto está situado de manera diferente según el valor de la alineación: a la izquierda para una alineación a la izquierda, a la derecha para una alineación a la derecha o en el centro cuando el texto está centrado.

El comando **SVG_New_text** acepta dos sintaxis diferentes para la definición de los caracteres:

- Puede pasar diferentes valores en los parámetros *fuentes*, *talla*, *estilo* y *alineación*: *fuentes* y *talla* permiten especificar la fuente y el tamaño en puntos a utilizar. Cuando no se pasan estos parámetros, el texto se escribirá en **Times New Roman 12 pts**. El parámetro opcional *estilo* da el estilo de caracteres a utilizar. En este parámetro, debe pasar uno de los siguientes estilos o una combinación de varios de estos valores (o también puede utilizar las constantes 4D correspondientes en el tema **Estilos de fuente**):

0 = Normal
1 = Negrita
2 = Itálica
4 = Subrayado
8 = Tachado

El parámetro opcional *alineación* permite especificar el tipo de alineación a aplicar al texto dibujado. Puede pasar uno de los siguientes valores:

2 = Alineado a la izquierda
3 = Centrado
4 = Alineación a la derecha

El parámetro opcional *color* contiene el nombre del color de la fuente. (Para más información sobre los colores, consulte la sección **Colores y degradados**).

El parámetro opcional *rotacion* permite precisar la rotación a aplicar al texto.
El parámetro opcional *interlineado* permite especificar el valor del interlineado si el texto tiene más de una línea. Valor por defecto = 1.
El parámetro opcional *estiramiento* permite aplicar un factor de estiramiento horizontal (valor > 1) o de condensación (valor entre 0 y 1) al texto.

- O puede pasar una definición de estilo en el parámetro *defEstilo* (en lugar del parámetro *fuelle*) y luego omite los siguientes parámetros. Por ejemplo, puede pasar:

```
SVG_New_textArea($Dom_svg;"Hola mundo";x;y;vWidth;vHeight;style_definition)
```

... donde el parámetro *style_definition* contiene una definición de estilo completa. Si pasa, por ejemplo, "{font-size:48px;fill:red;}", esta definición se añade como un atributo de estilo de esta forma:

```
style="font-size:48px;fill:red;"
```

En este caso, los parámetros adicionales se ignoran.

Ejemplo 1

Texto simple utilizando las propiedades de texto por defecto:

```
$SVG:=SVG_New  
$textID:=SVG_New_text($SVG;"Hello world!")
```

Ejemplo 2

Texto en azul, itálica, subrayado y alineado a la derecha:

```
$SVG:=SVG_New  
$text:="Hello world!\rBonjour le monde!\rHola Mundo!"  
$size:=48  
$font:="helvetica"  
$textID:=SVG_New_text($SVG;$text;400;10;$font;$size;Italic+Underline;Align right;"blue")
```

Ejemplo 3

Texto vertical:

```
$SVG:=SVG_New  
$textID:=SVG_New_text($SVG;$text;-250;0;"";48;-1;-1;"red";-90)
```

Ejemplo 4

Texto condensado o expandido:



Hello world (condensed)
Hello world (normal)
Hello world (stretched)

```
$SVG:=SVG_New
$textID:=SVG_New_text($SVG;"Hello world (condensed)";0;0;"",-1;-1;-1;"blue";0;1;0,8)
$textID:=SVG_New_text($SVG;"Hello world (normal)";0;24)
$textID:=SVG_New_text($SVG;"Hello world (stretched)";0;48;"",-1;-1;-1;"red";0;1;2)
```

Ejemplo 5

Mostrar texto multiestilo:

```
C_TEXT($Dom_svg;$Dom_text;$Txt_buffer)
//definición del texto multiestilo
$Txt_buffer:="<SPAN STYLE=\"font-size:18pt\">Hello </SPAN>"+\
"<SPAN STYLE=\"font-size:24pt;font-weight:bold;color:#D81E05\">World</SPAN>"+\
"<SPAN STYLE=\"font-size:36pt\">!</SPAN><BR/>"+\
"<SPAN STYLE=\"font-size:19pt;font-style:italic\">It's </SPAN>"+\
"<SPAN STYLE=\"font-size:24pt\">Monday</SPAN>"
$Dom_svg:=SVG_New

//título
SVG_SET_FONT_COLOR(SVG_New_text($Dom_svg;"_____ svg_Newtext
_____";10;30);"blue")
//text
$Dom_text:=SVG_New_text($Dom_svg;$Txt_buffer;50;50)

SVGTool_SHOW_IN_VIEWER($Dom_svg)
SVG_CLEAR($Dom_svg)
```

⚙ SVG_New_textArea

SVG_New_textArea (objetoSVGPadre ; texto {; x {; y {; ancho {; alto {; fuente | estiloFuente {; talla {; estilo {; alineacion}}}}}} }) -> Resultado

Parámetro	Tipo	Descripción
objetoSVGPadre	Ref_SVG	➔ Referencia del elemento padre
texto	Texto	➔ Texto a insertar
x	Entero largo	➔ Coordenada en el eje X
y	Entero largo	➔ Coordenadas en el eje Y
ancho	Entero largo	➔ Ancho del área de texto
alto	Entero largo	➔ Alto del área de texto
fuentes estiloFuente	Texto	➔ Nombre de la fuente o definición de estilo
talla	Entero	➔ Talla de los caracteres en puntos
estilo	Entero	➔ Estilo de los caracteres
alineacion	Entero	➔ Alineación
Resultado	Ref_SVG	➔ Referencia del objeto texto SVG

Descripción

El comando **SVG_New_textArea** inserta un área de texto en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

El elemento "textarea" es recomendado por la norma SVG Tiny 1.2 e implementado en 4D v11 SQL a partir de la versión 11.3 (ver <http://www.w3.org/TR/SVGMobile12/text.html#TextAreaElement>). Este elemento implementa un área de texto que, a diferencia del elemento "texto", se encarga de automatizar los retornos de línea cuando el texto supera el ancho requerido.

Notas:

- En el elemento "textArea", <tbreak/> los retornos de línea son reemplazados por los elementos.
- A partir de 4D v15, el comando **SVG_New_textArea** soporta el texto multi estilo simple (el texto puede contener diferentes estilos, pero los atributos SPAN no deben anidarse). Ver el ejemplo 2.

Los parámetros opcionales *x* e *y* permiten especificar la posición en los ejes X e Y de la esquina superior izquierda del área.

Los parámetros opcionales *ancho* y *alto* especifican el tamaño del área en el espacio de coordenadas usuario. Si no se pasa uno de estos parámetros, el área texto se adaptará automáticamente a su contenido.

El comando **SVG_New_textArea** acepta dos nuevas sintaxis diferentes para la definición de los caracteres:

Puede pasar varios valores en los parámetros *fuentes*, *talla*, *estilo* y *alineacion*: *fuentes* y *talla* permiten especificar el tipo de letra y tamaño, en puntos, a utilizar. Cuando no se pasan estos parámetros, el texto será escrito en **Times New Roman 12 pts**.

El parámetro opcional *estilo* ofrece información sobre el estilo de caracteres a utilizar. En el parámetro *estilo*, debe pasar uno de los siguientes valores o una combinación de varios de ellos (o puede utilizar las constantes 4D correspondientes en el tema **Estilos de fuente** theme):

Los parámetros opcionales *fuentes* y *talla* permiten especificar el tipo de letra y tamaño, en puntos, a utilizar. Cuando no se pasan estos parámetros, el texto será escrito en Times New Roman 12 pts.

- 0 = Normal
- 1 = Negrita
- 2 = Itálica

- 4 = Subrayado
- 8 = Tachado

El parámetro opcional *alineacion* permite especificar el tipo de alineación que se aplicará al texto dibujado. Puede pasar uno de los siguientes valores:

- 1 = Alineación por defecto (izquierda)
- 2 = Alineado a la izquierda
- 3 = Centrado
- 4 = Alineación a la derecha
- 5 = Justificado

- O puede pasar una definición de estilo en el parámetro *defEstilo* (en lugar del parámetro *fuelle*) y luego omitir los siguientes parámetros. Por ejemplo, puede pasar:

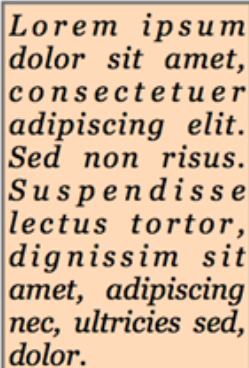
```
SVG_New_textArea($Dom_svg;"Hola mundo";x;y;vWidth;vHeight;style_definition)
```

... donde el parámetro *style_definition* contiene una definición de estilo completa. Si pasa, por ejemplo "{font-size:48px;fill:red;}", esta definición se añade como un atributo de estilo de la forma:

```
style="font-size:48px;fill:red;"
```

En este caso, se ignoran los parámetros adicionales.

Ejemplo 1



*Lorem ipsum
dolor sit amet,
consectetur
adipiscing elit.
Sed non risus.
Suspendisse
lectus tortor,
dignissim sit
amet, adipiscing
nec, ultricies sed,
dolor.*

```
$svg:=SVG_New
```

```
` Posicionar un rectángulo de borde
```

```
$rec:=SVG_New_rect($svg;5;5;210;320;0;0;"#777";"peachpuff";3)
```

```
` El texto
```

```
$txt:="Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus  
tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor."
```

```
$txtArea:=SVG_New_textArea($svg;$txt;10;10;200;310;"Georgia";25;italic;5)
```

```
` Guardar el documento
```

```
SVG_SAVE_AS_TEXT($svg;"test.svg")
```

Ejemplo 2

Mostrar texto multi estilo:

```
C_TEXT($Dom_svg;$Dom_text;$Txt_buffer)
//definición del texto multi estilo
$Txt_buffer:="<SPAN STYLE=\"font-size:18pt\">Hello </SPAN>"+\
"<SPAN STYLE=\"font-size:24pt;font-weight:bold;color:#D81E05\">World</SPAN>"+\
"<SPAN STYLE=\"font-size:36pt\">!</SPAN><BR/>"+\
"<SPAN STYLE=\"font-size:19pt;font-style:italic\">It's </SPAN>"+\
"<SPAN STYLE=\"font-size:24pt\">Monday</SPAN>"
$Dom_svg:=SVG_New

//título
SVG_SET_FONT_COLOR(SVG_New_text($Dom_svg;"_____ SVG_New_textArea
_____";10;30;"";-1);"blue")
//textArea
$Dom_text:=SVG_New_textArea($Dom_svg;$Txt_buffer;50;50)

SVGTTool_SHOW_IN_VIEWER($Dom_svg)
SVG_CLEAR($Dom_svg)
```

⚙ SVG_New_tspan

SVG_New_tspan (objetoSVGPadre ; texto {; x {; y {; fuente | estiloFuente {; talla {; estilo {; alineacion {; color}}}}}}) -> Resultado

Parámetro	Tipo	Descripción
objetoSVGPadre	Ref_SVG	➔ Referencia del elemento padre
texto	Texto	➔ Texto a insertar
x	Entero largo	➔ Coordenada en el eje X
y	Entero largo	➔ Coordenada en el eje Y
fuentes estiloFuente	Texto	➔ Nombre de la fuente o Definición de estilo
talla	Entero	➔ Tamaño de los caracteres en puntos
estilo	Entero	➔ Estilo de los caracteres
alineacion	Entero	➔ Alineación
color	Cadena	➔ Color del texto
Resultado	Ref_SVG	➔ Referencia del objeto texto SVG

Descripción

El comando **SVG_New_tspan** crea un nuevo elemento en el elemento 'text' o 'tspan' designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es una referencia de un elemento 'text' o 'tspan', se genera un error.

Los diferentes parámetros opcionales se describen con el comando **SVG_New_text**. Si se omiten ciertos parámetros opcionales, sus valores se heredan de los elementos padres.

Ejemplo 1

En un texto, es posible crear párrafos que hereden las propiedades del padre.



```
$SVG:=SVG_New
` Crea un nuevo texto en Arial, azul y alineado a la izquierda
$textID:=SVG_New_text($SVG;"";0;0;"arial";-1;-1;Align left;"blue")
` Párrafos anidados con indentación y cambio de tamaño y de estilo
$textID:=SVG_New_tspan($textID;"TITLE 1";10;10;"";24;Bold+Underline)
$textID:=SVG_New_tspan($textID;"Title 2";20;42;"";12;Bold)
$textID:=SVG_New_tspan($textID;"Title 3";30;60;"";10;Bold+Italic)
$textID:=SVG_New_tspan($textID;"Title 4";40;78;"";8;Italic)
```

Ejemplo 2

Cambiar una propiedad mientras permanece en un elemento "texto", acá el tamaño del texto.

Writing with SVG is **easy**

```
$textID:=SVG_New_text($SVG;"Writing ";10;10;"arial";12)
SVG_SET_FONT_SIZE(SVG_New_tspan($textID;"with ");14)
SVG_SET_FONT_SIZE(SVG_New_tspan($textID;"SVG ");18)
```

```
SVG_SET_FONT_SIZE(SVG_New_tspan($textID;"is ");24)  
SVG_SET_FONT_SIZE(SVG_New_tspan($textID;"easy ");36)
```

⚙ SVG_New_vertical_text

SVG_New_vertical_text (objetoSVGPadre ; texto {; x {; y {; fuente {; talla {; estilo {; alineacion {; color {; rotacion}}}}}} }) -> Resultado

Parámetro	Tipo		Descripción
objetoSVGPadre	Ref_SVG	➡	Referencia del elemento padre
texto	Texto	➡	Texto a insertar
x	Entero largo	➡	Coordenada en el eje X
y	Entero largo	➡	Coordenada en el eje Y
fuentes	Cadena	➡	Nombre de la fuente
talla	Entero	➡	Tamaño de los caracteres en puntos
estilo	Entero	➡	Estilo de los caracteres
alineacion	Entero	➡	Alineación
color	Cadena	➡	Color del texto
rotacion	Entero largo	➡	Angulo de rotación del texto
Resultado	Ref_SVG	↩	Referencia del objeto texto SVG

Descripción

El comando *SVG_New_vertical_text* inserta el texto verticalmente en el contenedor SVG designado por *objetoSVGPadre* y devuelve su referencia. Si *objetoSVGPadre* no es un documento SVG, se genera un error.

Los parámetros opcionales *x* e *y* permiten especificar la posición en el eje X e Y de la esquina inferior izquierda del primer carácter del texto.

Los parámetros opcionales *fuentes* y *talla* permiten especificar la fuente y el tamaño en puntos a utilizar. Cuando no se pasan estos parámetros, el texto será escrito en Times New Roman 12 pts.

El parámetro opcional *estilo* da el estilo de caracteres a utilizar. En este parámetro, debe pasar uno de los siguientes estilos, o una combinación de varios de estos valores:

- 0 = Normal
- 1 = Negrita
- 2 = Itálica
- 4 = Subrayado
- 8 = Tachado

El parámetro opcional *alineacion* permite especificar el tipo de alineación a aplicar al texto dibujado. Puede pasar uno de los siguientes valores:

- 2 = Alineado a la izquierda
- 3 = Centrado
- 4 = Alineación a la derecha

El parámetro opcional *color* contiene el nombre del color de la fuente. (Para mayor información sobre colores, consulte la sección "[Colores y degradados](#)").

El parámetro opcional *rotacion* permite precisar la rotación a aplicar al texto.

Ejemplo



```
$SVG:=SVG_New
```

```
$textID:=SVG_New_text($SVG;"Hello world";10;12)
```

```
$textID:=SVG_New_vertical_text($SVG;"Hello world";22;3;"";-1;-1;Center;"blue")
```

⚙️ SVG_SET_FONT_COLOR

SVG_SET_FONT_COLOR (objetoSVG ; color)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
color	Cadena	→	Color del texto

Descripción

El comando **SVG_SET_FONT_COLOR** permite especificar el color de la fuente para el objeto SVG de referencia *objetoSVG*. Si *objetoSVG* no hace referencia a un elemento válido, se genera un error.

El parámetro *color* contiene el nombre del color a utilizar. (Para más información sobre los colores, consulte la sección "**Colores SVG**").

⚙ SVG_SET_FONT_FAMILY

SVG_SET_FONT_FAMILY (objetoSVG ; fuente {; fuente2 ; ... ; fuenteN})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
fuentes	Cadena	→	Nombre de fuente

Descripción

El comando `SVG_SET_FONT_FAMILY` permite especificar la fuente para el objeto SVG de referencia `objetoSVG`. Si `objetoSVG` no hace referencia a un elemento válido, se genera un error.

El parámetro `fuentes` contiene el nombre de la fuente a utilizar. Cuando se pasan varios nombres, el comando crea automáticamente la lista de fuentes y/o de familias genéricas.

Ejemplo

Paso de varios nombres de fuentes:

```
SVG_SET_FONT_FAMILY(svgObject;"Lucida grande";"Sans-serif")  
// crear la lista: " 'Lucida grande' 'Sans-serif'"
```


⚙ SVG_SET_FONT_SIZE

SVG_SET_FONT_SIZE (objetoSVG ; talla)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
talla	Entero	→	Tamaño de los caracteres en puntos

Descripción

El comando *SVG_SET_FONT_SIZE* permite especificar el *tamaño* de la fuente para el objeto SVG de referencia *objetoSVG*. Si *objetoSVG* no referencia un elemento válido, se genera un error.

El parámetro *talla* contiene el tamaño de la fuente expresado en puntos.

⚙️ SVG_SET_FONT_STYLE

SVG_SET_FONT_STYLE (objetoSVG ; estilo)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
estilo	Entero	→	Estilo de caracteres

Descripción

El comando *SVG_SET_FONT_STYLE* permite especificar el estilo del texto para el objeto SVG de referencia *objetoSVG*. Si *objetoSVG* no hace referencia a un elemento válido, se genera un error.

En el parámetro *estilo*, debe pasar uno de los siguientes valores o una combinación de varios valores:

- 0 = Normal
- 1 = Negrita
- 2 = Itálica
- 4 = Subrayado
- 8 = Tachado

⚙️ SVG_SET_TEXT_ANCHOR

SVG_SET_TEXT_ANCHOR (objetoSVG ; alineacion)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
alineacion	Entero	→	Alineación

Descripción

El comando *SVG_SET_TEXT_ANCHOR* permite modificar la alineación del objeto SVG de referencia *objetoSVG*. Si *objetoSVG* no hace referencia a un elemento válido, se genera un error.

En el parámetro *alineacion*, debe pasar uno de los siguientes valores:

- 1 = Alineación por defecto (izquierda)
- 2 = Alineado a la izquierda
- 3 = Centrado
- 4 = Alineación a la derecha
- 5 = Justificar (sólo para el objeto TextArea)

SVG_SET_TEXT_KERNING

SVG_SET_TEXT_KERNING (objetoSVG ; kerning {; unidad})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento texto
kerning	Real	→	Espacio de las letras
unidad	Texto	→	Unidad del valor de espacio

Descripción

El comando *SVG_SET_TEXT_KERNING* permite modificar el espacio entre los caracteres del objeto texto designado por *objetoSVG*. Si *objetoSVG* no es un objeto texto SVG, se genera un error.

El parámetro opcional *unidad* permite especificar la unidad del valor de kerning. El valor por defecto es "%".

Si *kerning* es -1, el valor de kerning está en 'auto'.

Nota: en Windows, la implementación se limita al texto de izquierda a derecha y de arriba a abajo (desactivada para el texto de derecha a izquierda) y a los elementos "texto" y "tspan"; bajo Mac OS, el soporte no es limitado.

Ver también: <http://www.yoyodesign.org/doc/w3c/svg1/text.html#KerningProperty>

Ejemplo

Ejemplos de variaciones de kerning:

```
Hello world !
Hello world !
Hello world !
Hello world !
Hello world !
Hello world !
Hello world !
Hello world !
Hello world !
Hello world !
```

```
//Referencia
$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;40;"";36)
$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;80;"";36)
SVG_SET_TEXT_KERNING($Dom_text;0,5)
$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;120;"";36)
SVG_SET_TEXT_KERNING($Dom_text;1)
$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;160;"";36)
SVG_SET_TEXT_KERNING($Dom_text;1,5)
$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;200;"";36)
SVG_SET_TEXT_KERNING($Dom_text;2)
$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;240;"";36)
SVG_SET_TEXT_KERNING($Dom_text;1,5)
```

```
$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;280;"";36)
SVG_SET_TEXT_KERNING($Dom_text;1)
$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;320;"";36)
SVG_SET_TEXT_KERNING($Dom_text;0,5)
$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;360;"";36)
SVG_SET_TEXT_KERNING($Dom_text;0)
```

⚙ SVG_SET_TEXT_LETTER_SPACING

SVG_SET_TEXT_LETTER_SPACING (objetoSVG ; espacio {; unidad})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento texto
espacio	Real	→	Espacio de las letras
unidad	Texto	→	Unidad de valor

Descripción

El comando *SVG_SET_TEXT_LETTER_SPACING* se utiliza para modificar el espacio de las letras del objeto texto designado por *objetoSVG* además de la separación, debido a la propiedad 'kerning'. Si *objetoSVG* no es un objeto texto SVG, se genera un error.

El parámetro opcional *unidad* permite especificar la unidad del valor de los espacios. El valor por defecto es "%".

Si *espacio* es igual a -1, el valor del espacio se fija en "normal".

Ver también: <http://www.w3.org/TR/SVG/text.html#LetterSpacingProperty>

Ejemplo

Ejemplos de variaciones de espacio:

```
Hello world !
Hello world !
H e l l o   w o
Hello world !
Hello world !
Hello world !
H e l l o   w o r
H   e   l   l   o
```

```
//Referencia
$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;40;"";36)

$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;80;"";36)
SVG_SET_TEXT_LETTER_SPACING($Dom_text;1)

$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;120;"";36)
SVG_SET_TEXT_LETTER_SPACING($Dom_text;1;"em")

$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;160;"";36)
SVG_SET_TEXT_LETTER_SPACING($Dom_text;1;"px")

$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;200;"";36)
SVG_SET_TEXT_LETTER_SPACING($Dom_text;1;"pt")

$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;240;"";36)
SVG_SET_TEXT_LETTER_SPACING($Dom_text;1;"pc")

$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;280;"";36)
SVG_SET_TEXT_LETTER_SPACING($Dom_text;1;"mm")

$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;320;"";36)
```

```
SVG_SET_TEXT_LETTER_SPACING($Dom_text;1;"cm")
$Dom_text:=SVG_New_text($Dom_SVG;"Hello world !";20;360;"";36)
SVG_SET_TEXT_LETTER_SPACING($Dom_text;1;"in")
```

⚙️ SVG_SET_TEXT_RENDERING

SVG_SET_TEXT_RENDERING (objetoSVG ; rende)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento texto
rende	Texto	→	Valor de renderización

Descripción

El comando *SVG_SET_TEXT_RENDERING* permite definir los tradeoff a utilizar para la renderización del texto del objeto texto designado por *objetoSVG*. Si *objetoSVG* no es un objeto texto SVG, se genera un error.

El parámetro *rende* puede tener uno de los siguientes valores: "auto", "optimizeSpeed", "optimizeLegibility", "geometricPrecision" o "inherit". De lo contrario, se genera un error.

Ver también: <http://www.w3.org/TR/2001/REC-SVG-20010904/painting.html#TextRenderingProperty>

⚙ SVG_SET_TEXT_WRITING_MODE

SVG_SET_TEXT_WRITING_MODE (objetoSVG ; modoEscrit)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento texto
modoEscrit	Texto	→	Sentido de la escritura

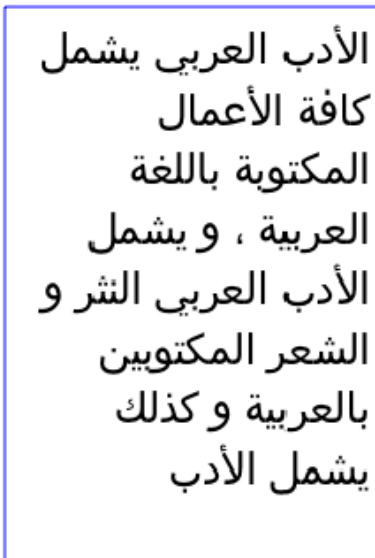
Descripción

El comando *SVG_SET_TEXT_WRITING_MODE* permite definir si la dirección de escritura del texto objeto designado por *objetoSVG* será de izquierda a derecha, de derecha a izquierda o de abajo hacia arriba. Si *objetoSVG* no es un objeto texto SVG, se genera un error.

El parámetro *modoEscritura* puede tener uno de los siguientes valores: "lr-tb", "rl-tb", "tb-rl", "lr", "rl", "tb" or "inherit". De lo contrario, se genera un error.

Ejemplo

Escritura de derecha a izquierda:



```
//Marco
SVG_New_rect($Dom_SVG;5;5;210;310;0;0;"blue";"none")

//Texto
$Dom_text:=SVG_New_textArea($Dom_SVG;$Txt_sample;10;10;200;300;"Baghdad 'Arial Unicode
MS";25)
SVG_SET_TEXT_WRITING_MODE($Dom_text;"rl")
```

⚙ SVG_SET_TEXTAREA_TEXT




















SVG_SET_TEXTAREA_TEXT (objetoSVG ; elTexto)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento texto
elTexto	Texto	→	Texto a definir

Descripción

El comando *SVG_SET_TEXTAREA_TEXT* permite fijar/reemplazar el contenido textual del objeto texto designado por *objetoSVG*. Si *objetoSVG* no es un objeto "textArea", se genera un error. Los caracteres retorno de línea se reemplazan automáticamente por los elementos "<tbreak/>".

Utilitarios

-  SVG_ABOUT
-  SVG_Count_elements
-  SVG_ELEMENTS_TO_ARRAYS
-  SVG_Estimate_weight
-  SVG_Find_ID
-  SVG_Get_options
-  SVG_Get_root_reference
-  SVG_Get_version
-  SVG_Is_reference_valid
-  SVG_Post_comment
-  SVG_Read_element_type
-  SVG_Read_last_error
-  SVG_References_array
-  SVG_ROTATION_CENTERED
-  SVG_SCALING_CENTERED
-  SVG_Set_error_handler
-  SVG_SET_OPTIONS
-  SVGTool_SET_VIEWER_CALLBACK
-  SVGTool_SHOW_IN_VIEWER

⚙️ SVG_ABOUT

SVG_ABOUT

Este comando no requiere parámetros

Descripción

El comando *SVG_ABOUT* muestra un diálogo con el logo 4D SVG e indica el número de versión del componente:



⚙ SVG_Count_elements

SVG_Count_elements (objetoSVG) -> Result

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia SVG
Result	Entero largo	↩	Número de objetos

Descripción

El comando *SVG_Count_elements* devuelve el número de objetos gráficos en el *objetoSVG* pasado en parámetro. Un grupo cuenta como un objeto. Para conocer el número de objetos gráficos en un grupo, pase su referencia al comando. Si *objetoSVG* no es válido, se genera un error.

⚙ SVG_ELEMENTS_TO_ARRAYS

```
SVG_ELEMENTS_TO_ARRAYS ( objetoSVG ; punteroArrayRef {; punteroArrayTipos {;
puntArrayNoms}} )
```

Parámetro	Tipo		Descripción
objetoSVG	Alpha	→	Referencia SVG
punteroArrayRef	Puntero	→	Array cadena de las referencias de objetos
punteroArrayTipos	Puntero	→	Array cadena de los tipos de objetos
puntArrayNoms	Puntero	→	Array cadena de los id de objetos

Descripción

El comando *SVG_ELEMENTS_TO_ARRAYS* llena el array apuntado por *punteroArrayRefs* con las referencias de los objetos gráficos de primer nivel para la referencia SVG pasada en *objetoSVG*.

Si se pasa el puntero opcional *punteroArrayTipos*, el array se llena con los tipos de los objetos.

Si se pasa el puntero opcional *punteroArrayNoms*, el array se llena con los identificadores de los objetos.

Un grupo cuenta como un objeto. Para obtener esta información para los objetos gráficos de un grupo, pase su referencia al comando.

Si *objetoSVG* no es válido o si este atributo no existe, se genera un error.

⚙ SVG_Estimate_weight

SVG_Estimate_weight (objetoSVG) -> Resultado

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un documento SVG
Resultado	Real	↩	Tamaño en bytes del documento SVG

Descripción

El comando *SVG_Estimate_weight* devuelve el tamaño en bytes del árbol SVG cuya referencia se pasa en el parámetro *objetoSVG*. Si *objetoSVG* no es una referencia válida, se genera un error.

⚙️ SVG_Find_ID

SVG_Find_ID (objetoSVG ; nombre) -> Resultado

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia del objeto SVG
nombre	Cadena	→	ID del elemento SVG
Resultado	Ref_SVG	↩	Referencia del elemento

Descripción

El comando *SVG_Find_ID* devuelve la referencia del elemento cuyo ID se pasa en el parámetro *nom*, que pertenece a la estructura SVG cuyo elemento raíz se pasa en el parámetro *objetoSVG*. Si el elemento no se encuentra, se genera un error.

⚙ SVG_Get_options

SVG_Get_options -> Resultado

Parámetro	Tipo	Descripción
Resultado	Entero largo	Opciones

Descripción

El comando `SVG_Get_options` devuelve un entero largo que representa un array de 32 bits donde cada bit puede representar una opción del componente. Puede utilizar los Operadores en los bits de 4D para probar (??) el estado de una opción y activarla (+?) o desactivarla (-?).

Las siguientes opciones están disponibles:

Bit	Opción	Por defecto
1	Asignar un ID automáticamente cuando se crea un elemento	0 (desactivado)
2	Cerrar automáticamente los objetos	0 (desactivado)
3	Crear objetos con un fondo	1 (activado)
4	Coordenadas absolutas para trazos	1 (activado)
5	Crear código más legible	0 (desactivado)
6	Pitido cuando se produce un error	1 (activado)
7	No mostrar los errores de 4D	0 (desactivado)
8	Imágenes transparentes	1 (activado)
9	Utilizar origen trigonométrico	0 (desactivado)
10	Reemplazar automáticamente Arial	1 (activado)
11	Utilizar la renderización 'crispEdges' por defecto para un nuevo diseño	0 (desactivado)
12	Controlar los parámetros	1 (activado)
13	Keep extra spaces	0 (desactivado)
14	Rotación centrada	0 (desactivado)

Asignar un ID automáticamente al crear un elemento[#/table]

- *Asignar un ID automáticamente cuando se crea un elemento*
Si esta opción está activa, cuando el componente crea un nuevo elemento, añade y llena sistemáticamente un atributo 'id' para el objeto creado, si esto no se ha especificado.
- *Cerrar automáticamente los objetos*
Si esta opción está activa, los objetos creados con los comandos `SVG_New_arc` y `SVG_New_polyline_by_arrays` se cerrarán automáticamente.
- *Crear objetos con un fondo*
Si esta opción está activa, los objetos cerrados se crearán con un color de fondo, de lo contrario, el fondo será transparente.
- *Coordenadas absolutas para trazos*
Durante el dibujo de trazos con los comandos `SVG_PATH_MOVE_TO`, `SVG_PATH_LINE_TO`, `SVG_PATH_CURVE` y `SVG_PATH_ARC`, las coordenadas pasadas se interpretarán como absolutas si esta opción está activa, en caso contrario serán consideradas como relativas.
- *Crear código más legible*
Esta opción permite crear un código indentado y con buenos espacios, pero sin embargo difícil de manejar, su activación es especialmente útil durante la fase de depuración.
- *Pitido cuando se produce un error*
Cuando se produce un error y ningún método de gestión de errores de la base local se ha

instalado con el comando `SVG_Set_error_handler`, se emite un pito si esta opción está activa.

- *No mostrar los errores de 4D*
Esta opción activa por defecto bloquea la visualización de errores 4D mediante la instalación de un método de gestión de errores propio al componente 4D SVG. Puede preferir no utilizar esta gestión interna y permitir a 4D mostrar estos mensajes. Esto puede ser útil durante la depuración, por ejemplo.
- *Imágenes transparentes*
Por defecto, las imágenes SVG creadas con el comando `SVG_New` son transparentes. Al deshabilitar esta opción, las imágenes tendrán un fondo blanco.
- *Utilizar origen trigonométrico*
Por defecto, SVG ubica el origen en la parte superior de la escala de grados (medianoche). Esta opción le permite pasar las coordenadas de acuerdo a los puntos de referencia trigonométricos (3h o 15min). La conversión se realiza de inmediato.
- *Reemplazar automáticamente Arial*
Por defecto, 4D SVG reemplaza la fuente 'Arial' por 'Arial Unicode MS', 'Arial' para mejorar la compatibilidad con los caracteres no romanos (japonés por ejemplo). En algunos casos, puede desactivar este funcionamiento. Esta opción permite no reemplazar las fuentes Arial.
- *Utilizar la renderización 'crispEdges' por defecto para un nuevo diseño*
El atributo `crispEdges` (ver `SVG_SET_SHAPE_RENDERING`) puede forzarse por defecto gracias a esta opción.
- *Control de los parámetros*
Por defecto, 4D SVG controla la validez de los parámetros pasados a los comandos. Una vez la fase de desarrollo termina, puede desactivar esta opción con el fin de acelerar sensiblemente la ejecución del código.
- *Mantener espacios extras* (Nuevo en v14)
Permite múltiples espacios adyacentes para mostrar en objetos texto.
- *Rotación centrada* (Nuevo en v14)
Si se activa esta opción, el comando `SVG_SET_TRANSFORM_ROTATE` intenta realizar una rotación centrada cuando no se pasa el tercer y cuarto parámetro. El centro de rotación se calcula sobre la base de la `x`, `y`, atributos de ancho y alto del objeto. Si el objeto referenciado no tiene estos atributos, la rotación se realiza alrededor del punto (0,0).

Ejemplo

Ver el comando `SVG_SET_OPTIONS`.

⚙️ SVG_Get_root_reference

SVG_Get_root_reference (objetoSVG) -> Resultado

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia del objeto SVG
Resultado	Ref_SVG	↩	Referencia del elemento raíz

Descripción

El comando **SVG_Get_root_reference** devuelve la referencia del elemento raíz del objeto SVG cuya referencia se pasó en el parámetro *objetoSVG*.

SVG_Get_version

SVG_Get_version -> Resultado

Parámetro	Tipo		Descripción
Resultado	Cadena		Número de versión

Descripción

El comando *SVG_Get_version* devuelve la versión del componente en forma alfanumérica. La cadena devuelta siempre incluye el número de versión y subversión (por ejemplo "11.0" para la versión 11 y "11.3" para la tercera actualización de la versión 11). Cuando se trata de una versión beta, se especifica el número de distribución, anteponiendo la letra "B" (por ejemplo "11.3B1" para la beta 1 de la versión 11.3)

⚙ SVG_Is_reference_valid

SVG_Is_reference_valid (objetoSVG) -> Resultado

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
Resultado	Booleano	↩	True si la referencia pertenece a un elemento SVG

Descripción

El comando *SVG_Is_reference_valid* devuelve **True** si la referencia pasada en el parámetro *objetoSVG* es la de un elemento de un árbol SVG. Si el elemento no pertenece a un árbol SVG, el comando devuelve **False**. Si *objetoSVG* no es una referencia válida, se genera un error.

⚙️ SVG_Post_comment

SVG_Post_comment (objetoSVG ; comentario) -> Resultado

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia del objeto SVG
comentario	Texto	→	Texto a añadir como comentario
Resultado	Ref_SVG	↩	Referencia del comentario

Descripción

El método **SVG_Post_comment** añade el texto pasado en *comentario* en el objeto SVG designado por el parámetro *objetoSVG*.

El método devuelve la referencia SVG del comentario.

Ejemplo

El siguiente código:

```
[#code4D]C_TEXT($comment)
$comment:="Modified on "+String(Current date)
$ref:= SVG_Post_comment ($svg; $comment )
```

.. añade en el objeto SVG \$svg:

```
<!--Modificado el 12/12/2011-->[#/codeXML]
```

⚙ SVG_Read_element_type

SVG_Read_element_type (objetoSVG) -> Resultado

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de un elemento SVG
Resultado		↩	Tipo de elemento

Descripción

El comando *SVG_Read_element_type* devuelve el tipo de elemento cuya referencia se pasa en el parámetro *objetoSVG*.

Si *objetoSVG* no es una referencia válida o si este atributo no existe, se genera un error.

⚙️ SVG_Read_last_error

SVG_Read_last_error -> Resultado

Parámetro	Tipo	Descripción
Resultado	Entero largo	Número del último error

Descripción

El comando *SVG_Read_last_error* devuelve el número del último error ocurrido durante la ejecución de un comando del componente 4D SVG y reinicializa este error.

El número de error devuelto puede ser específico a un comando del componente o un error generado por 4D. Los errores generados por el componente son:

- 8850 Insufficient number of parameters
- 8851 Invalid parameter type
- 8852 Invalid reference
- 8853 Incorrect value for attribute
- 8854 The element does not accept this command
- 8855 Invalid (ID not found in document) object name (symbol, marker, filter, etc.)
- 8856 DTD file not found
- 8857 Incorrect value for a parameter
- 8858 Unknown error

Ejemplo 1

Dado el método "SVG_error_mgmt" del ejemplo del comando *SVG_Set_error_handler*:

```
` Instalación del método de gestión de errores
$ Error_Method_Txt:=SVG_Set_error_handler("SVG_error_mgmt")
` de ahora en adelante se ejecutará el método SVG_error_mgmt en caso de error

` Creación de un nuevo documento SVG
$SVG:=SVG_New(1200;900;"SVG Component Test";"";True)
SVG_SET_VIEWBOX($SVG;0;0;1500;1000)

If(SVG_Read_last_error=0)

    ...

Else
    ` El método SVG_error_mgmt ha sido llamado y recibió el número de error
End if

` Desinstalación del método de gestión de errores
SVG_Set_error_handler
```

Ejemplo 2

Dado el siguiente método *SVG_error_mgmt*:

C_LONGINT(\$1)

C_TEXT(\$2)

` Conservar el error y el contexto

errorNumber:=\$1

commandName:=\$2

` Definir la variable sistema OK en 0

OK:=0

Este método puede utilizarse de la siguiente forma:

` Instalación del método de gestión de errores

\$ Error_Method_Txt:=SVG_Set_error_handler("SVG_error_mgmt")

` Creación de un nuevo documento SVG

\$SVG:=SVG_New(1200;900;" SVG Component Test ","";True)

SVG_SET_VIEWBOX(\$SVG;0;0;1500;1000)

If(OK=1)

...

Else

**ALERT("Error No."+String(errorNumber)+" durante la ejecución del comando
\""+commandName+"\"")**

End if

` Desinstalación del método de gestión de errores

SVG_Set_error_handler

⚙️ SVG_References_array

SVG_References_array (punteroArrayRef) -> Resultado

Parámetro	Tipo		Descripción
punteroArrayRef	Puntero	→	Array cadena de las referencia de documentos
Resultado	Entero largo	↩	Número de referencias

Descripción

El comando *SVG_References_array* devuelve la lista de referencias de documentos SVG actuales en el array apuntado por *punteroArrayRef*. Como resultado, el comando devuelve el número de referencias encontradas.

SVG_References_array es útil durante la depuración. Cada vez que un documento SVG se crea con los comandos *SVG_New*, *SVG_Copy* o *SVG_Open_file*, el componente añade la referencia devuelta por el comando en un array interno. Cuando un documento SVG se libera utilizando el comando *SVG_CLEAR*, el componente elimina la referencia del array.

⚙️ SVG_ROTATION_CENTERED

SVG_ROTATION_CENTERED (objetoSVG ; angulo)

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia objeto SVG
angulo	Real	→	Ángulo de rotación

Descripción

El comando **SVG_ROTATION_CENTERED** realiza una rotación centrada para el objeto SVG cuya referencia se pasa en el parámetro *objetoSVG*. Este tipo de rotación sólo puede aplicarse a los objetos que tienen atributos x, y, ancho y alto.

En el parámetro *angulo*, se pasa el ángulo de rotación a realizar.

⚙️ SVG_SCALING_CENTERED

SVG_SCALING_CENTERED (objetoSVG ; escala {; x ; y})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia del objeto SVG
escala	Real	→	Valor de la escala
x	Real	→	Eje X
y	Real	→	Eje Y

Descripción

El comando **SVG_SCALING_CENTERED** realiza una escala centrada de la imagen SVG cuya referencia se pasa en el parámetro *objetoSVG*.

En el parámetro *escala*, pase un valor escala positivo (>1). Si pasa 1, la escala se define en 100%.

En los parámetros *x* y *y* opcionales, puede pasar, respectivamente, las coordenadas del eje x y del eje y para el punto central. Si no pasa estos parámetros, el punto central se determina en base a los atributos "x", "y", "ancho" y "alto" del objeto (si las hay). Si este tipo de transformación se aplica a un objeto que no tiene estos atributos, y los parámetros opcionales *x* y *y* se omiten, se devuelve una cadena vacía.

⚙️ SVG_Set_error_handler

SVG_Set_error_handler {{ metodo }} -> Resultado

Parámetro	Tipo		Descripción
metodo	Cadena	→	Nombre del método de la base local a instalar
Resultado	Cadena	↩	Nombre del método previamente instalado

Descripción

El comando *SVG_Set_error_handler* permite instalar el *metodo* de la base local que se llamará en caso de error y devuelve el nombre del método previamente instalado.

Los comandos del componente 4D SVG efectúan un mínimo de verificaciones cuando se les llama: número mínimo de parámetros, validez de las referencias, para el elemento al cual se aplica un comando. El componente controla los errores de una manera estructurada y permite que la base local recupere los posibles errores.

Cuando el funcionamiento por defecto no se ha modificado, si se produce un error se emite un pitido y el comando se interrumpe.

La base local puede recuperar el número de error y el nombre del comando que falla en uno de estos métodos. Para ello, la base local debe crear un método que recibirá el número de error como primer parámetro y el nombre del comando como segundo parámetro.

Este método, si es instalado por el comando *SVG_Set_error_handler*, se llama cuando se produce un error, en este caso, no es generado un sonido por el código del componente.

Si *metodo* se omite o es una cadena vacía, el método se desinstala y el comportamiento por defecto se reactiva.

Nota: el método de la base local que será llamado por el componente debe tener la propiedad "Compartido entre componentes y base local".

Ejemplo

Instalación del método *SVG_error_mgmt* (método de base local) como método de gestión de errores:

```
$error:=SVG_Set_error_handler("SVG_error_mgmt")
```

Código del método:

```
` Método de gestión de errorre SVG_error_mgmt  
ALERT("Error No."+String($1)+" durante la ejecución del comando \"+$2+"\")
```

⚙️ SVG_SET_OPTIONS

SVG_SET_OPTIONS {(opciones)}

Parámetro	Tipo	Descripción
opciones	Entero largo	→ Opciones del componente 4D SVG

Descripción

El comando `SVG_SET_OPTIONS` permite fijar las opciones del componente 4D SVG con el entero largo `opciones`. Para mayor información sobre el contenido de `opciones`, consulte la descripción del comando `SVG_Get_options`.

Como todas las opciones se fijan simultáneamente, este comando debe estar precedido de una llamada al comando `SVG_Get_options`, seguido por el uso de los de 4D.

Si no se pasa el parámetro `opcion`, todas las opciones se vuelven a su valor por defecto (ver el comando `SVG_Get_options`).

Ejemplo 1

Crear un código más legible:

```
$Options :=SVG_Get_options
$Options :=$Options ?+5 `activar la opción
SVG_SET_OPTIONS($Options)
```

Ejemplo 2

Dibujar un gráfico pie:

```
$svg:=SVG_New

`Activar el cierre automático de los objetos
SVG_SET_OPTIONS(SVG_Get_options?+2)

SVG_New_arc($svg;100;100;90;0;105;"gray";"lightcoral";1)
SVG_New_arc($svg;100;100;90;105;138;"gray";"lightskyblue";1)
SVG_New_arc($svg;100;100;90;138;230;"gray";"lightgreen";1)
SVG_New_arc($svg;100;100;90;230;270;"gray";"lightsteelblue";1)
SVG_New_arc($svg;100;100;90;270;360;"gray";"lightyellow";1)
```

Ejemplo 3

Visualización de múltiples espacios en los objetos texto utilizando la opción `Keep extra spaces option (13)` (añadida en v14):

```
$Txt_buffer:="abc    def"
$Dom_text:=SVG_New_textArea($Dom_svg;$Txt_buffer;50;50)
```

se muestra como "abc def"

```
SVG_SET_OPTIONS(SVG_Get_options?+13) // conservar espacios en objetos texto
$txt_buffer="abc    def"
$Dom_text:=SVG_New_textArea($Dom_svg,$txt_buffer;50;50)
```

se muestra como "abc def"

⚙️ SVGTool_SET_VIEWER_CALLBACK

SVGTool_SET_VIEWER_CALLBACK (*nomMetodo*)

Parámetro	Tipo	Descripción
<i>nomMetodo</i>	Texto	→ Nombre del método proyecto 4D

Descripción

El comando **SVGTool_SET_VIEWER_CALLBACK** permite instalar *nomMetodo* como el método proyecto que se llama cuando los eventos On Clicked y On Mouse Move ocurren en la imagen mostrada por el comando *SVGTool_SHOW_IN_VIEWER*.

Este método recibe un parámetro texto que es el id del elemento en el que se hace clic en o sobre el cual el ratón se mueve según lo dispuesto por el comando 4D **SVG Find element ID by coordinates**.

Este parámetro debe ser declarado en el método proyecto *nomMetodo* de la base local mediante su inserción en la línea C_TEXT(\$1).

Debe asignar la propiedad "Compartido entre componentes y base local" para el método *nomMetodo*.

Si el método no existe o no se comparte, el error -10508 es generado por 4D.

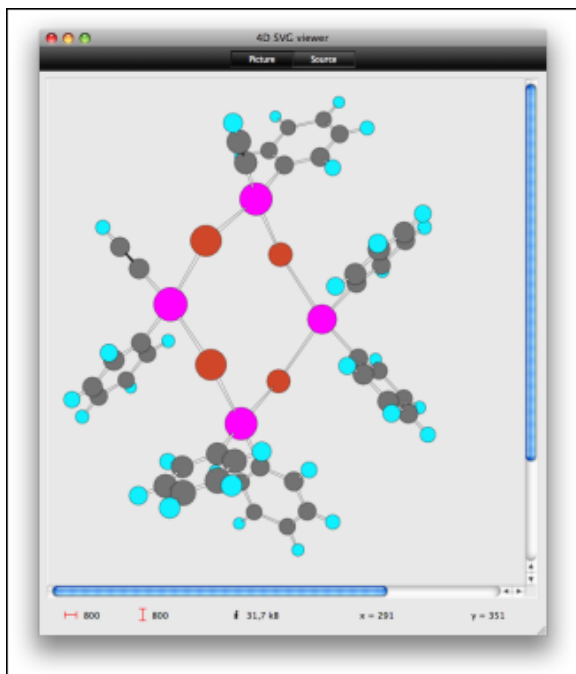
⚙ SVGTool_SHOW_IN_VIEWER

SVGTool_SHOW_IN_VIEWER (objetoSVG {; fuentes})

Parámetro	Tipo		Descripción
objetoSVG	Ref_SVG	→	Referencia de la imagen a mostrar
fuentes	Cadena	→	Abre directamente el viewer SVG en la página fuente

Descripción

El comando `SVGTool_SHOW_IN_VIEWER` muestra la imagen SVG definida por `objetoSVG` en una ventana del Visualizador SVG. Esta herramienta se ofrece con el componente `SVG`:




Si pasa el parámetro opcional `fuentes` (añadido en v14), el visualizador se abre directamente en la página fuente.








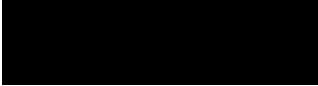










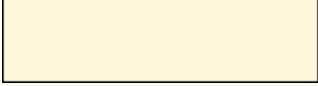



Para mayor información sobre el Visualizador SVG, consulte la sección [Herramientas de desarrollo](#).

4D SVG Constants

 Colores SVG (Nombres)









 SVG Colors (RGB)

Colores SVG (Nombres)


Constante	Tipo	Valor	Comentario
color SVG aliceblue	Cadena	aliceblue	
color SVG antiquewhite	Cadena	antiquewhite	
color SVG aqua	Cadena	aqua	
color SVG aquamarine	Cadena	aquamarine	
color SVG azure	Cadena	azure	
color SVG beige	Cadena	beige	
color SVG bisque	Cadena	bisque	
color SVG black	Cadena	black	
color SVG blanchedalmond	Cadena	blanchedalmond	
color SVG blue	Cadena	blue	
color SVG blueviolet	Cadena	blueviolet	
color SVG brown	Cadena	brown	
color SVG burlywood	Cadena	burlywood	
color SVG cadetblue	Cadena	cadetblue	
color SVG chartreuse	Cadena	chartreuse	
color SVG chocolate	Cadena	chocolate	
color SVG coral	Cadena	coral	
color SVG cornflowerblue	Cadena	cornflowerblue	
color SVG cornsilk	Cadena	cornsilk	
color SVG crimson	Cadena	crimson	
color SVG cyan	Cadena	cyan	
color SVG darkblue	Cadena	darkblue	

Constante	Tipo	Valor	Comentario
color SVG darkcyan	Cadena	darkcyan	
color SVG darkgoldenrod	Cadena	darkgoldenrod	
color SVG darkgray	Cadena	darkgray	
color SVG darkgreen	Cadena	darkgreen	
color SVG darkgrey	Cadena	darkgrey	
color SVG darkkhaki	Cadena	darkkhaki	
color SVG darkmagenta	Cadena	darkmagenta	
color SVG darkolivegreen	Cadena	darkolivegreen	
color SVG darkorange	Cadena	darkorange	
color SVG darkorchid	Cadena	darkorchid	
color SVG darkred	Cadena	darkred	
color SVG darksalmon	Cadena	darksalmon	
color SVG darkseagreen	Cadena	darkseagreen	
color SVG darkslateblue	Cadena	darkslateblue	
color SVG darkslategray	Cadena	darkslategray	
color SVG darkslategrey	Cadena	darkslategrey	
color SVG darkturquoise	Cadena	darkturquoise	
color SVG darkviolet	Cadena	darkviolet	
color SVG deeppink	Cadena	deeppink	
color SVG deepskyblue	Cadena	deepskyblue	
color SVG dimgray	Cadena	dimgray	
color SVG dimgrey	Cadena	dimgrey	

Constante	Tipo	Valor	Comentario
color SVG dodgerblue	Cadena	dodgerblue	
color SVG firebrick	Cadena	firebrick	
color SVG floralwhite	Cadena	floralwhite	
color SVG forestgreen	Cadena	forestgreen	
color SVG fuchsia	Cadena	fuchsia	
color SVG gainsboro	Cadena	gainsboro	
color SVG ghostwhite	Cadena	ghostwhite	
color SVG gold	Cadena	gold	
color SVG goldenrod	Cadena	goldenrod	
color SVG gray	Cadena	gray	
color SVG green	Cadena	green	
color SVG greenyellow	Cadena	greenyellow	
color SVG grey	Cadena	grey	
color SVG honeydew	Cadena	honeydew	
color SVG hotpink	Cadena	hotpink	
color SVG indianred	Cadena	indianred	
color SVG indigo	Cadena	indigo	
color SVG ivory	Cadena	ivory	
color SVG khaki	Cadena	khaki	
color SVG lavender	Cadena	lavender	
color SVG lavenderblush	Cadena	lavenderblush	
color SVG lawngreen	Cadena	lawngreen	








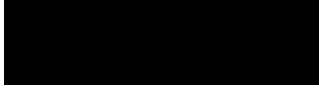










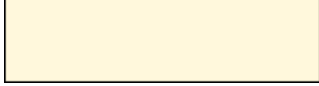



Constante	Tipo	Valor	Comentario
color SVG lemonchiffon	Cadena	lemonchiffon	
color SVG lightblue	Cadena	lightblue	
color SVG lightcoral	Cadena	lightcoral	
color SVG lightcyan	Cadena	lightcyan	
color SVG lightgoldenrodyellow	Cadena	lightgoldenrodyellow	
color SVG lightgray	Cadena	lightgray	
color SVG lightgreen	Cadena	lightgreen	
color SVG lightgrey	Cadena	lightgrey	
color SVG lightpink	Cadena	lightpink	
color SVG lightsalmon	Cadena	lightsalmon	
color SVG lightseagreen	Cadena	lightseagreen	
color SVG lightskyblue	Cadena	lightskyblue	
color SVG lightslategray	Cadena	lightslategray	
color SVG lightslategrey	Cadena	lightslategrey	
color SVG lightsteelblue	Cadena	lightsteelblue	
color SVG lightyellow	Cadena	lightyellow	
color SVG lime	Cadena	lime	
color SVG limegreen	Cadena	limegreen	
color SVG linen	Cadena	linen	
color SVG magenta	Cadena	magenta	
color SVG maroon	Cadena	maroon	
color SVG mediumaquamarine	Cadena	mediumaquamarine	



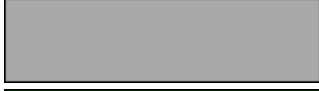

















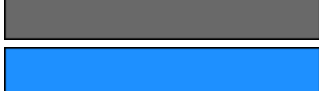

Constante	Tipo	Valor	Comentario
color SVG mediumblue	Cadena	mediumblue	
color SVG mediumorchid	Cadena	mediumorchid	
color SVG mediumpurple	Cadena	mediumpurple	
color SVG mediumseagreen	Cadena	mediumseagreen	
color SVG mediumslateblue	Cadena	mediumslateblue	
color SVG mediumspringgreen	Cadena	mediumspringgreen	
color SVG mediumturquoise	Cadena	mediumturquoise	
color SVG mediumvioletred	Cadena	mediumvioletred	
color SVG midnightblue	Cadena	midnightblue	
color SVG mintcream	Cadena	mintcream	
color SVG mistyrose	Cadena	mistyrose	
color SVG moccasin	Cadena	moccasin	
color SVG navajowhite	Cadena	navajowhite	
color SVG navy	Cadena	navy	
color SVG oldlace	Cadena	oldlace	
color SVG olive	Cadena	olive	
color SVG olivedrab	Cadena	olivedrab	
color SVG orange	Cadena	orange	
color SVG orangered	Cadena	orangered	
color SVG orchid	Cadena	orchid	
color SVG palegoldenrod	Cadena	palegoldenrod	
color SVG palegreen	Cadena	palegreen	












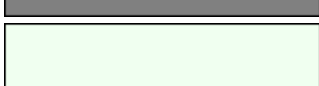




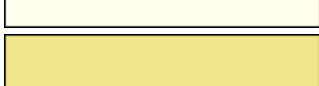
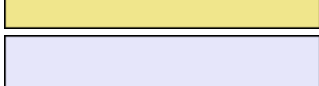
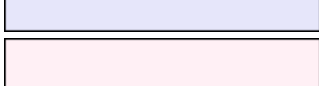



Constante	Tipo	Valor	Comentario
color SVG paleturquoise	Cadena	paleturquoise	
color SVG palevioletred	Cadena	palevioletred	
color SVG papyawhip	Cadena	papyawhip	
color SVG peachpuff	Cadena	peachpuff	
color SVG peru	Cadena	peru	
color SVG pink	Cadena	pink	
color SVG plum	Cadena	plum	
color SVG powderblue	Cadena	powderblue	
color SVG purple	Cadena	purple	
color SVG red	Cadena	red	
color SVG rosybrown	Cadena	rosybrown	
color SVG royalblue	Cadena	royalblue	
color SVG saddlebrown	Cadena	saddlebrown	
color SVG salmon	Cadena	salmon	
color SVG sandybrown	Cadena	sandybrown	
color SVG seagreen	Cadena	seagreen	
color SVG seashell	Cadena	seashell	
color SVG sienna	Cadena	sienna	
color SVG silver	Cadena	silver	
color SVG skyblue	Cadena	skyblue	
color SVG slateblue	Cadena	slateblue	
color SVG slategray	Cadena	slategray	














Constante	Tipo	Valor	Comentario
color SVG slategrey	Cadena	slategrey	
color SVG snow	Cadena	snow	
color SVG springgreen	Cadena	springgreen	
color SVG steelblue	Cadena	steelblue	
color SVG tan	Cadena	tan	
color SVG teal	Cadena	teal	
color SVG thistle	Cadena	thistle	
color SVG tomato	Cadena	tomato	
color SVG turquoise	Cadena	turquoise	
color SVG violet	Cadena	violet	
color SVG wheat	Cadena	wheat	
color SVG white	Cadena	white	
color SVG whitesmoke	Cadena	whitesmoke	
color SVG yellow	Cadena	yellow	
color SVG yellowgreen	Cadena	yellowgreen	



















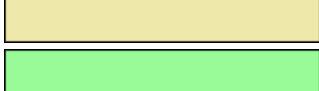



SVG Colors (RGB)

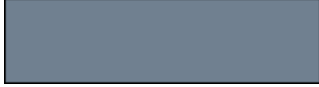
Constante	Tipo	Valor	Comentario
color RGB aliceblue	Entero largo	15792383	
color RGB antiquewhite	Entero largo	16444375	
color RGB aqua	Entero largo	65535	
color RGB aquamarine	Entero largo	8388564	
color RGB azure	Entero largo	15794175	
color RGB beige	Entero largo	16119260	
color RGB bisque	Entero largo	16770244	
color RGB black	Entero largo	0	
color RGB blanchedalmond	Entero largo	16772045	
color RGB blue	Entero largo	255	
color RGB blueviolet	Entero largo	9055202	
color RGB brown	Entero largo	10824234	
color RGB burlywood	Entero largo	14596231	
color RGB cadetblue	Entero largo	6266528	
color RGB chartreuse	Entero largo	8388352	
color RGB chocolate	Entero largo	13789470	
color RGB coral	Entero largo	16744272	
color RGB cornflowerblue	Entero largo	6591981	
color RGB cornsilk	Entero largo	16775388	
color RGB crimson	Entero largo	14423100	
color RGB cyan	Entero largo	65535	
color RGB darkblue	Entero largo	139	









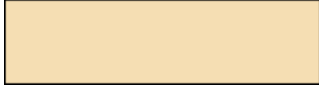




Constante	Tipo	Valor	Comentario
color RGB darkcyan	Entero largo	35723	
color RGB darkgoldenrod	Entero largo	12092939	
color RGB darkgray	Entero largo	11119017	
color RGB darkgreen	Entero largo	25600	
color RGB darkgrey	Entero largo	11119017	
color RGB darkkhaki	Entero largo	12433259	
color RGB darkmagenta	Entero largo	9109643	
color RGB darkolivegreen	Entero largo	5597999	
color RGB darkorange	Entero largo	16747520	
color RGB darkorchid	Entero largo	10040012	
color RGB darkred	Entero largo	9109504	
color RGB darksalmon	Entero largo	15308410	
color RGB darkseagreen	Entero largo	9419919	
color RGB darkslateblue	Entero largo	4734347	
color RGB darkslategray	Entero largo	3100495	
color RGB darkslategrey	Entero largo	3100495	
color RGB darkturquoise	Entero largo	52945	
color RGB darkviolet	Entero largo	9699539	
color RGB deeppink	Entero largo	16716947	
color RGB deepskyblue	Entero largo	49151	
color RGB dimgray	Entero largo	6908265	
color RGB dodgerblue	Entero largo	2003199	

Constante	Tipo	Valor	Comentario
color RGB firebrick	Entero largo	11673122	
color RGB forestgreen	Entero largo	2263842	
color RGB fuchsia	Entero largo	16711935	
color RGB gainsboro	Entero largo	14474460	
color RGB ghostwhite	Entero largo	16316671	
color RGB gold	Entero largo	16766720	
color RGB goldenrod	Entero largo	14329120	
color RGB gray	Entero largo	8421504	
color RGB green	Entero largo	32768	
color RGB greenyellow	Entero largo	11403055	
color RGB grey	Entero largo	8421504	
color RGB honeydew	Entero largo	15794160	
color RGB hotpink	Entero largo	16738740	
color RGB indianred	Entero largo	13458524	
color RGB indigo	Entero largo	4915330	
color RGB ivory	Entero largo	16777200	
color RGB khaki	Entero largo	15787660	
color RGB lavender	Entero largo	15132410	
color RGB lavenderblush	Entero largo	16773365	
color RGB lawngreen	Entero largo	8190976	
color RGB lemonchiffon	Entero largo	16775885	
color RGB lightblue	Entero largo	11393254	

Constante	Tipo	Valor	Comentario
color RGB lightcoral	Entero largo	1576136	
color RGB lightcyan	Entero largo	14745599	
color RGB lightgoldenrodyellow	Entero largo	16448210	
color RGB lightgray	Entero largo	13882323	
color RGB lightgreen	Entero largo	9498256	
color RGB lightgrey	Entero largo	13882323	
color RGB lightpink	Entero largo	16758465	
color RGB lightsalmon	Entero largo	16752762	
color RGB lightseagreen	Entero largo	2142890	
color RGB lightskyblue	Entero largo	8900346	
color RGB lightslategray	Entero largo	7833753	
color RGB lightslategrey	Entero largo	7833753	
color RGB lightsteelblue	Entero largo	11584734	
color RGB lightyellow	Entero largo	16777184	
color RGB lime	Entero largo	65280	
color RGB limegreen	Entero largo	3329330	
color RGB linen	Entero largo	16445670	
color RGB magenta	Entero largo	16711935	
color RGB maroon	Entero largo	8388608	
color RGB medianaquamarine	Entero largo	6737322	
color RGB mediumblue	Entero largo	205	
color RGB mediumorchid	Entero largo	12211667	

Constante	Tipo	Valor	Comentario
color RGB mediumpurple	Entero largo	9662683	
color RGB mediumseagreen	Entero largo	3978097	
color RGB mediumslateblue	Entero largo	8087790	
color RGB mediumspringgreen	Entero largo	64154	
color RGB mediumturquoise	Entero largo	4772300	
color RGB mediumvioletred	Entero largo	13047173	
color RGB midnightblue	Entero largo	1644912	
color RGB mintcream	Entero largo	16121850	
color RGB mistyrose	Entero largo	16770273	
color RGB moccasin	Entero largo	16770273	
color RGB navajowhite	Entero largo	16768685	
color RGB navy	Entero largo	128	
color RGB oldlace	Entero largo	16643558	
color RGB olive	Entero largo	8421376	
color RGB olivedrab	Entero largo	7048739	
color RGB orange	Entero largo	16753920	
color RGB orangered	Entero largo	16729344	
color RGB orchid	Entero largo	14315734	
color RGB palegoldenrod	Entero largo	15657130	
color RGB palegreen	Entero largo	10025880	
color RGB paleturquoise	Entero largo	11529966	
color RGB palevioletred	Entero largo	14381203	

Constante	Tipo	Valor	Comentario
color RGB papyawhip	Entero largo	16773077	
color RGB peachpuff	Entero largo	16767673	
color RGB peru	Entero largo	13468991	
color RGB pink	Entero largo	16761035	
color RGB plum	Entero largo	14524637	
color RGB powderblue	Entero largo	11591910	
color RGB purple	Entero largo	8388736	
color RGB red	Entero largo	16711680	
color RGB rosybrown	Entero largo	12357519	
color RGB royalblue	Entero largo	4286945	
color RGB saddlebrown	Entero largo	9127187	
color RGB salmon	Entero largo	16416882	
color RGB sandybrown	Entero largo	16032864	
color RGB seagreen	Entero largo	3050327	
color RGB seashell	Entero largo	16774638	
color RGB sienna	Entero largo	10506797	
color RGB silver	Entero largo	12632256	
color RGB skyblue	Entero largo	8900331	
color RGB slateblue	Entero largo	6970061	
color RGB slategray	Entero largo	7372944	
color RGB slategrey	Entero largo	7372944	
color RGB snow	Entero largo	16775930	

Constante	Tipo	Valor	Comentario
color RGB springgreen	Entero largo	65407	
color RGB steelblue	Entero largo	4620980	
color RGB tan	Entero largo	13808780	
color RGB teal	Entero largo	32896	
color RGB thistle	Entero largo	14204888	
color RGB tomato	Entero largo	16737095	
color RGB turquoise	Entero largo	4251856	
color RGB violet	Entero largo	15631086	
color RGB wheat	Entero largo	16113331	
color RGB white	Entero largo	16777215	
color RGB whitesmoke	Entero largo	16119285	
color RGB yellow	Entero largo	16776960	
color RGB yellowgreen	Entero largo	10145074	

☰ **Anexos**

✚ Anexo B, Enlaces externos

Anexo B, Enlaces externos

Esta es una lista de enlaces útiles relativos a la norma SVG:

Presentación

<http://www.w3.org/Graphics/SVG/>

Especificación

<http://www.w3.org/TR/SVG12/> (Status: Working Draft 13 April 2005)

<http://www.yoyodesign.org/doc/w3c/svg1/> (French translation (Version 1.0))

Tutoriales

<http://www.w3.org/Consortium/Offices/Presentations/SVG/1.svg>

Comunidad

<http://svg.org/>

<http://svgfr.org>

<http://www.openclipart.org/>

<http://www.gosvg.net/>