

4D SVG

Composant pour
Windows® et Mac OS®



Composant 4D SVG

Version 11.4 pour Windows® et Mac OS®

Copyright © 2009 4D SAS
Tous droits réservés

Les informations contenues dans ce manuel peuvent faire l'objet de modifications sans préavis et ne sauraient en aucune manière engager 4D SAS. La fourniture du logiciel décrit dans ce manuel est régie par un octroi de licence dont les termes sont précisés par ailleurs dans la licence électronique figurant sur le support du Logiciel et de la Documentation y afférente. Le logiciel et sa Documentation ne peuvent être utilisés, copiés ou reproduits sur quelque support que ce soit et de quelque manière que ce soit, que conformément aux termes de cette licence.

Aucune partie de ce manuel ne peut être reproduite ou recopiée de quelque manière que ce soit, électronique ou mécanique, y compris par photocopie, enregistrement, archivage ou tout autre procédé de stockage, de traitement et de récupération d'informations, pour d'autres buts que l'usage personnel de l'acheteur, et ce exclusivement aux conditions contractuelles, sans la permission explicite de 4D SAS.

4D, 4D Write, 4D View, 4D Server ainsi que le logo 4D sont des marques enregistrées de 4D SAS.

Windows et Microsoft sont des marques enregistrées de Microsoft Corporation.

Apple, Macintosh, Mac OS, QuickTime sont des marques enregistrées ou des noms commerciaux de Apple Computer, Inc.

Mac2Win Software Copyright © 1990-2009 est un produit de Altura Software, Inc.

Correcteur orthographique, © Copyright SYNAPSE Développement, Toulouse, France, 1994-2009.

ACROBAT © Copyright 1987-2009, Secret Commercial Adobe Systems Inc. Tous droits réservés. ACROBAT est une marque enregistrée d'Adobe Systems Inc.

Tous les autres noms de produits ou appellations sont des marques déposées ou des noms commerciaux appartenant à leurs propriétaires respectifs.

www.Mcours.com
Site N°1 des Cours et Exercices Email: contact@mcours.com

Sommaire

1. Introduction..... 9

Composant 4D SVG.....	11
Outils de développement.....	13
Précisions de syntaxe.....	17

2. Attributs..... 19

SVG_GET_ATTRIBUTES.....	21
SVG_Get_ID.....	22
SVG_SET_ATTRIBUTES.....	23
SVG_SET_ATTRIBUTES_BY_ARRAYS.....	24
SVG_SET_DIMENSIONS.....	25
SVG_SET_FILL_BRUSH.....	26
SVG_SET_FILTER.....	27
SVG_SET_ID.....	28
SVG_SET_MARKER.....	29
SVG_SET_OPACITY.....	32
SVG_SET_ROUNDING_RECT.....	33
SVG_SET_STROKE_BRUSH.....	34
SVG_SET_STROKE_LINECAP.....	35
SVG_SET_STROKE_LINEJOIN.....	36
SVG_SET_STROKE_WIDTH.....	37
SVG_SET_TRANSFORM_FLIP.....	38
SVG_SET_TRANSFORM_MATRIX.....	40
SVG_SET_TRANSFORM_ROTATE.....	41
SVG_SET_TRANSFORM_SCALE.....	42
SVG_SET_TRANSFORM_SKEW.....	43
SVG_SET_TRANSFORM_TRANSLATE.....	44
SVG_SET_VIEWBOX.....	45
SVG_SET_VIEWPORT_FILL.....	46
SVG_SET_VISIBILITY.....	47
SVG_SET_XY.....	48

3. Couleurs et dégradés..... 49

Couleurs SVG.....	51
SVG_Color_grey.....	52
SVG_Color_RGB_from_long.....	53
SVG_GET_DEFAULT_BRUSHES.....	54
SVG_SET_DEFAULT_BRUSHES.....	55

4. Dessin..... 57

SVG_ADD_POINT.....	59
SVG_New_arc.....	60
SVG_New_circle.....	62
SVG_New_ellipse.....	64
SVG_New_ellipse_bounded.....	66
SVG_New_embedded_image.....	68
SVG_New_image.....	70
SVG_New_line.....	72
SVG_New_path.....	73
SVG_New_polygon.....	77
SVG_New_polygon_by_arrays.....	78
SVG_New_polyline.....	80
SVG_New_polyline_by_arrays.....	82
SVG_New_rect.....	85
SVG_New_regular_polygon.....	87
SVG_PATH_ARC.....	89
SVG_PATH_CLOSE.....	91
SVG_PATH_CURVE.....	92
SVG_PATH_LINE_TO.....	93
SVG_PATH_MOVE_TO.....	94
SVG_PATH_QCURVE.....	95
SVG_Use.....	96

5. Documents..... 99

SVG_CLEAR.....	101
SVG_Copy.....	102
SVG_Export_to_picture.....	103
SVG_Export_to_XML.....	104
SVG_New.....	105
SVG_Open_file.....	107
SVG_Open_picture.....	108
SVG_SAVE_AS_PICTURE.....	109
SVG_SAVE_AS_TEXT.....	110
SVG_Validate_file.....	111

6. Filtres..... 113

Filtres SVG.....	115
SVG_Filter_Blend.....	116
SVG_Filter_Blur.....	117
SVG_Filter_Offset.....	118

7. Structure et Définitions..... 119

SVG_Define_filter.....	121
SVG_Define_linear_gradient.....	123
SVG_Define_marker.....	126
SVG_Define_radial_gradient.....	128
SVG_Define_shadow.....	130
SVG_Define_solidColor.....	132
SVG_Define_symbol.....	133
SVG_New_group.....	134
SVG_Set_description.....	136
SVG_Set_title.....	137

8. Texte..... 139

SVG_New_text.....	141
SVG_New_textArea.....	144
SVG_New_tspan.....	147
SVG_New_vertical_text.....	149
SVG_SET_FONT_COLOR.....	151
SVG_SET_FONT_FAMILY.....	152
SVG_SET_FONT_SIZE.....	153
SVG_SET_FONT_STYLE.....	154
SVG_SET_TEXT_ANCHOR.....	155

9. Utilitaires..... 157

SVG_Count_elements.....	159
SVG_ELEMENTS_TO_ARRAYS.....	160
SVG_Estimate_weight.....	161
SVG_Find_ID.....	162
SVG_Get_options.....	163
SVG_Get_version.....	165
SVG_Is_reference_valid.....	166
SVG_Read_element_type.....	167
SVG_Read_last_error.....	168
SVG_References_array.....	170
SVG_Set_error_handler.....	171
SVG_SET_OPTIONS.....	173
SVGTool_SHOW_IN_VIEWER.....	175

10. Annexes..... 177

Annexe A, Tableau de couleurs.....	179
Annexe B, Liens externes.....	181

Index des commandes.....183

1

Introduction

Le SVG (*Scalable Vector Graphics*) est un format de fichier graphique vectoriel bidimensionnel basé sur le XML. 4D comporte un moteur de rendu intégré permettant d'afficher les fichiers SVG.

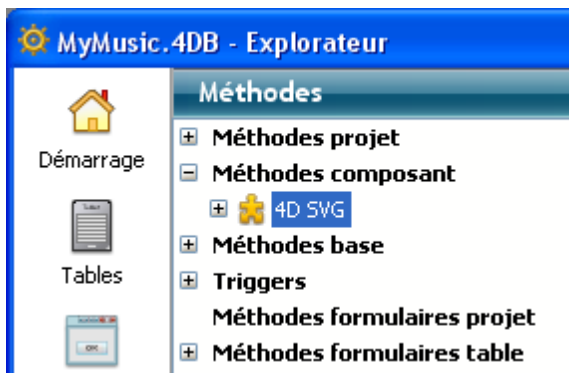
Le langage XML dédié à la manipulation des images SVG est particulièrement riche et étendu. Afin d'en simplifier l'accès et la prise en main, 4D propose le composant 4D SVG. Ce composant comporte de nombreuses commandes permettant la création et la manipulation d'objets graphiques usuels. Le but de cette bibliothèque n'est pas d'être exhaustif mais de répondre aux besoins les plus courants des développeurs 4D. A noter que tous les besoins spécifiques supplémentaires pourront être traités avec les commandes XML de 4D.

Installation et mise en oeuvre

Le Composant 4D SVG doit être installé au minimum dans 4D v11 SQL release 3 (version 11.3).

Comme tout composant 4D, le Composant 4D SVG s'installe par la copie du dossier du composant (**4D SVG.4dbase**) dans le dossier **Components** de la base. Le dossier Components de la base doit être situé au même niveau que le fichier de structure. Les composants étant chargés au démarrage, la base ne doit pas être lancée avant la copie complète de tous les éléments.

Si le composant est correctement installé, l'élément **4D SVG** apparaît dans la page Méthodes de la base, rubriques "Méthodes composant" :



Vous pouvez déployer cet élément afin de visualiser l'ensemble des commandes du composant. Les commandes s'utilisent dans l'éditeur de méthodes de 4D comme des commandes 4D ou de plug-in standard.

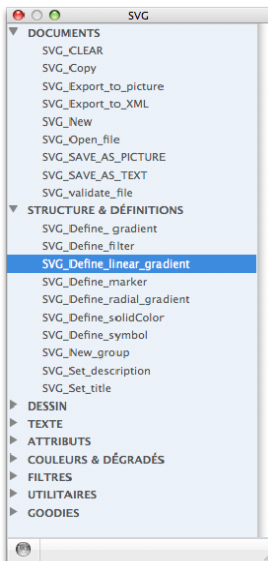
A noter que le Composant 4D SVG vous permet de bénéficier de fenêtres supplémentaires pour la sélection des commandes et le rendu du code SVG. Pour plus d'informations, reportez-vous à la section Outils de développement.

Le composant 4D SVG propose un ensemble d'outils destinés à faciliter la saisie du code et la prévisualisation des graphiques SVG :

- la palette de syntaxe
- la palette de couleurs
- le visualisateur SVG

Palette de syntaxe

La palette de syntaxe liste les commandes du composant 4D SVG regroupées par thèmes :



La palette permet d'insérer les commandes du composant dans l'éditeur de méthodes par simple glisser-déposer. La commande est alors collée dans la méthode avec ses paramètres. Les paramètres optionnels sont préfixés d'un trait de soulignement.

Pour afficher la palette de syntaxe, vous pouvez soit :

- exécuter la méthode `SVGTool_Display_syntax`,
- cliquer sur le bouton **SVG** et choisir la commande **Syntaxe du composant SVG** dans la palette du composant 4D Pop si vous l'utilisez (cf. ci-dessous).

Palette de couleurs

La palette de couleurs affiche le nom et un échantillon de chaque couleur définie dans la norme SVG, ainsi qu'un curseur permettant de faire varier le taux d'opacité :



Vous pouvez utiliser cette palette pour insérer par glisser-déposer une référence de couleur SVG dans l'éditeur de méthodes de 4D. La couleur est insérée sous forme de chaîne incluant éventuellement le taux d'opacité (par exemple "lavender:30" pour la couleur lavande et une opacité de 30 %). Pour plus d'informations sur les références de couleurs, reportez-vous à la section Couleurs SVG.

Vous pouvez également glisser-déposer une couleur dans l'éditeur de formulaires de 4D. Cette action crée un carré de couleur sous forme d'image SVG statique.

Pour afficher la palette de couleurs, il vous suffit d'exécuter la méthode SVGTool_Display_colors.

Visualisateur SVG

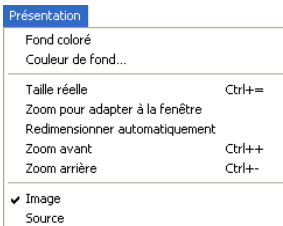
4D SVG propose un visualisateur SVG, particulièrement utile en phase de développement :



La fenêtre du visualisateur comporte deux pages, accessibles via les boutons **Image** et **Source** ou le menu **Présentation** :

- **Image** : cette page présente une zone de visualisation dans laquelle vous pouvez glisser-déposer ou ouvrir un fichier image SVG (via le menu **Fichier**). Vous pouvez également y afficher une référence SVG valide à l'aide de la commande `SVGTool_SHOW_IN_VIEWER`.
- **Source** : cette page permet de visualiser le code XML associé à l'image. Vous pouvez sélectionner et copier le code, mais vous ne pouvez pas le modifier.

Lorsque la fenêtre est au premier plan, vous pouvez modifier plusieurs options d'affichage et enregistrer le fichier image sur disque via le menu **Présentation** :



Note : La page "Image" dispose d'un menu contextuel standard.

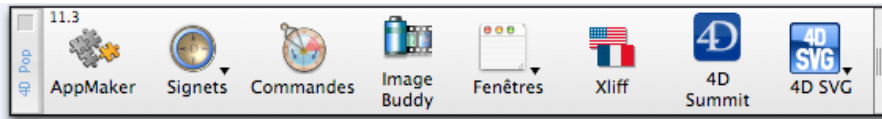
Pour afficher la fenêtre du visualisateur, vous pouvez soit :

- exécuter la méthode `SVGTool_Display_viewer`. Dans ce cas, la fenêtre s'affiche vide.
- appeler la méthode `SVGTool_SHOW_IN_VIEWER` en lui passant une référence SVG valide afin de prévisualiser l'image référencée (voir la description de la commande)
- cliquer sur le bouton **SVG** et choisir la commande **Visualisateur SVG** dans la palette du composant 4D Pop si vous l'utilisez (cf. ci-dessous).

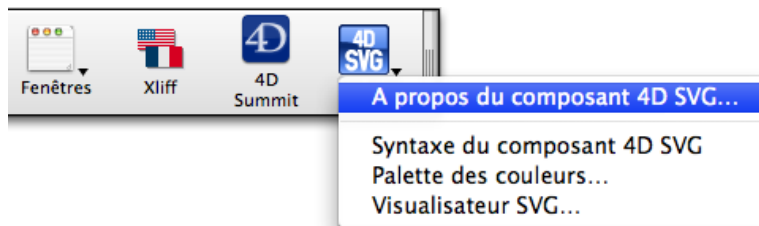
Intégration à 4D Pop

4D Pop est un ensemble de composants dédiés à la productivité du développeur et regroupés dans une barre d'outils qui s'intègre dans l'environnement de développement de 4D (voir <http://www.4d.fr/products/4dpop.html>).

Lorsque vous utilisez conjointement 4D Pop et 4D SVG, un nouveau bouton est disponible dans la barre d'outils de 4D Pop :



Ce bouton permet l'accès direct aux outils d'aide au développement de 4D SVG :



Ref_SVG

La plupart des commandes du composant 4D SVG manipulent les structures SVG via des références de type **Ref_SVG**.

Une Ref_SVG est une expression de type Chaîne de 16 caractères. Elle identifie de façon unique une structure SVG chargée en mémoire. Il peut s'agir d'un document SVG chargé via les commandes SVG_Copy, SVG_New, SVG_Open_picture ou SVG_Open_file, ou de toute structure SVG manipulée par programmation (objet, filtre, tracé, etc.).

Une Ref_SVG est une référence XML. Toutes les références Ref_SVG peuvent être utilisées comme paramètres refÉlément des commandes XML DOM de 4D.

Une fois que vous n'en avez plus besoin, n'oubliez pas d'appeler la commande SVG_CLEAR avec les références Ref_SVG afin de libérer la mémoire.

Paramètres optionnels

Sauf mention contraire, les arguments numériques optionnels sont ignorés si leur valeur est égale à -1 et les arguments texte sont ignorés si la chaîne passée est vide.

Coordonnées

Sauf mention contraire, les paramètres de position (x, y) et de dimensions (largeur, hauteur, rayon) sont attendus dans le système de coordonnées utilisateur courant.

2

Attributs



SVG_GET_ATTRIBUTES (objetSVG; pointTabNoms; pointTabValeurs)

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence SVG
pointTabNoms	Pointeur	→ Tableau chaîne des libellés d'attributs
pointTabValeurs	Pointeur	→ Tableau chaîne des valeurs d'attributs

Description

La commande SVG_GET_ATTRIBUTES remplit les tableaux pointés par pointTabNoms et pointTabValeurs respectivement des noms et des valeurs des attributs de l'élément dont la référence est passée dans objetSVG. Si objetSVG n'est pas valide ou si cet attribut n'existe pas, une erreur est générée.

Référence

SVG_SET_ATTRIBUTES, SVG_SET_ATTRIBUTES_BY_ARRAYS.

SVG_Get_ID (objetSVG) → Chaîne

Paramètre	Type		Description
objetSVG	Ref_SVG	→	Référence d'élément SVG
Résultat	Chaîne	←	Nom de l'élément

Description

La commande SVG_Get_ID retourne la valeur de l'attribut 'id' de l'élément dont la référence est passée en paramètre.

Si objetSVG n'est pas valide ou si cet attribut n'existe pas, une erreur est générée.

Référence

SVG_Find_ID, SVG_SET_ID.

SVG_SET_ATTRIBUTES (objetSVG; nomAttribut; valeurAttribut{; nomAttribut2; valeurAttribut2; ...; nomAttributN; valeurAttributN})

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
nomAttribut	Chaîne	→ Nom de l'attribut à fixer
valeurAttribut	Chaîne	→ Valeur de l'attribut

Description

La commande SVG_SET_ATTRIBUTES permet d'affecter un ou plusieurs attribut(s) personnalisé(s) à un objet SVG de référence objetSVG. Si le ou les attribut(s) existaient déjà, leurs valeurs sont remplacées par celles passées en paramètre.

Les attributs et leurs valeurs sont passés par couples de paramètres.

Exemple

```
$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")
```

Référence

SVG_GET_ATTRIBUTES, SVG_SET_ATTRIBUTES_BY_ARRAYS.

SVG_SET_ATTRIBUTES_BY_ARRAYS (objetSVG; pointTabNoms; pointTabValeurs)

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
pointTabNoms	Pointeur	→ Noms des attributs
pointTabValeurs	Pointeur	→ Valeurs synchronisées des attributs

Description

La commande SVG_SET_ATTRIBUTES_BY_ARRAYS permet d'affecter un ou plusieurs attribut(s) personnalisé(s) à un objet SVG de référence objetSVG. Si le ou les attribut(s) existaient déjà, leurs valeurs sont remplacées par celles passées en paramètres.

Les attributs et leurs valeurs sont passés par l'intermédiaire de deux tableaux, sur lesquels pointent pointTabNoms et pointTabValeurs.

Exemple

```

$svg:=SVG_New
$object:=SVG_New_rect ($svg; 10; 10; 200; 200; 0; 0; "black";"white"; 2)
TABLEAU TEXTE($attributes; 0)
TABLEAU TEXTE ($values; 0)
AJOUTER A TABLEAU($attributes; "fill")
AJOUTER A TABLEAU($values; "red")
AJOUTER A TABLEAU($attributes; "stroke")
AJOUTER A TABLEAU($values; "blue")
AJOUTER A TABLEAU($attributes; "stroke-width")
AJOUTER A TABLEAU($values; "3")
SVG_SET_ATTRIBUTES_BY_ARRAYS ($object; -> $attributes; -> $values)

```

Référence

SVG_GET_ATTRIBUTES, SVG_SET_ATTRIBUTES.

SVG_SET_DIMENSIONS (objetSVG; largeur{; hauteur{; unité}})

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
largeur	Numérique →	Dimension sur l'axe x
hauteur	Numérique →	Dimension sur l'axe y
unité	Chaîne →	Unité

Description

La commande SVG_SET_DIMENSIONS permet de fixer les dimensions de l'objet SVG de référence objetSVG.

Si ces attributs existaient déjà, leurs valeurs sont remplacées par celles passées en paramètres.

Si le paramètre unité est passé, il sera utilisé. Les valeurs attendues sont : *px*, *pt*, *pc*, *cm*, *mm*, *in*, *em*, *ex* ou *%*. Une valeur d'unité incorrecte génère une erreur. Si le paramètre est omis, les valeurs de largeur et hauteur sont attendues dans le système de coordonnées utilisateur.

Exemple

```
$svg :=SVG_New `Créer un nouveau document
$object:=SVG_New_rect ($svg; 10; 10; 200; 200; 0; 0; "black";"white"; 2)
SVG_SET_DIMENSIONS ($object; -1; 400) `Nouvelle hauteur
```

SVG_SET_FILL_BRUSH (objetSVG; couleur)

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
couleur	Chaîne	→ Expression couleur

Description

La commande SVG_SET_FILL_BRUSH permet de fixer la couleur de remplissage de l'objet SVG de référence objetSVG. Si cet attribut existait déjà, sa valeur est remplacée par celle passée en paramètre.

Pour plus d'informations sur les références de couleurs, reportez-vous à la section "Couleurs SVG".

Exemple

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

Référence

Couleurs SVG, SVG_SET_STROKE_BRUSH.

SVG_SET_FILTER (objetSVG; url)

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'élément SVG
url	Chaîne	→ Nom du filtre

Description

La commande SVG_SET_FILTER permet d'associer un filtre à l'objet de référence objetSVG. Si objetSVG n'est pas une référence valide, une erreur est générée. Si l'attribut existait déjà, sa valeur est remplacée.

Le paramètre url est le nom du filtre à utiliser, tel que défini avec la commande SVG_Define_filter. Si ce nom n'existe pas, une erreur est générée.

Exemple

Reportez-vous à l'exemple de la commande SVG_Define_filter.

Référence

Filtres SVG, SVG_Define_filter.

SVG_SET_ID (objetSVG; id)

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
id	Chaîne	→ Identifiant à attribuer à l'objet

Description

La commande SVG_SET_ID permet de fixer la propriété 'ID' de l'objet SVG de référence objetSVG. Si cet attribut existait déjà, sa valeur est remplacée par celle passée en paramètre.

L'ID d'un objet est utilisée pour référencer un objet. Cette référence peut ensuite être retrouvée grâce à la commande SVG_Get_ID. L'ID est également utilisée par la commande de 4D SVG Chercher ID element par coordonnees (voir la documentation de 4D).

Exemple

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

Référence

SVG_Find_ID, SVG_Get_ID.

SVG_SET_MARKER (objetSVG; url{; position})

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'élément SVG
url	Chaîne	→ Nom du marqueur
position	Chaîne	→ Position du marqueur

Description

La commande SVG_SET_MARKER permet d'associer un marqueur à l'objet de référence objetSVG. Si objetSVG n'est pas la référence d'un élément 'line', 'path', 'polyline' ou 'polygon', une erreur est générée.

Si l'attribut existait déjà, sa valeur est remplacée.

Le paramètre url est le nom d'un élément marqueur à utiliser tel que défini avec la commande SVG_Define_marker. Si ce nom n'existe pas, une erreur est générée.

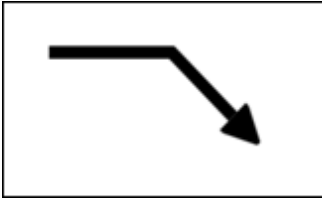
Le paramètre optionnel position permet de fixer la position du marqueur par rapport à l'objet. Il est possible de placer un marqueur différent ou non au début, à la fin ou à tous les autres sommets d'un tracé. Les valeurs peuvent être :

- start pour placer un marqueur en début de tracé
- end pour placer un marqueur à la fin du tracé
- middle pour placer un marqueur à tous les sommets autres que le début et la fin.
- all pour placer un marqueur sur tous les sommets d'un tracé.

Si ce paramètre est omis, le marqueur est placé à la fin du tracé.

Exemples

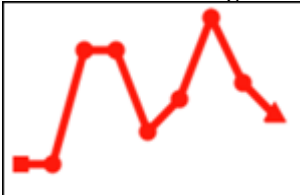
1. Dessiner une flèche :



```
$SVG:= SVG_New
  `Définir la flèche
$arrow:=SVG_Define_marker ($SVG;"fleche";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 ($line;250;125)
  `Mettre la flèche à la fin du tracé
SVG_SET_MARKER ($line;"fleche")
```

2. Dessiner un diagramme avec des marqueurs différents au début et à la fin :



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

  `Définir un cercle pour marquer les points
$point:=SVG_Define_marker ($SVG;"pointMarker";2;2;3;3)
SVG_SET_VIEWBOX ($point;0;0;4;4)
SVG_New_circle ($point;2;2;1)
```



```
`Définir un carré pour le point de début
$start:=SVG_Define_marker ($SVG;"startMarker";1;1;2;2)
SVG_New_rect ($start;0;0;2;2)
```

```
Définir un triangle pour le point de fin
$end:=SVG_Define_marker ($SVG;"endMarker";5;5;3;3;60)
SVG_SET_VIEWBOX ($end;0;0;10;10)
SVG_New_regular_polygon ($end;10;3)
```

```
TABLEAU ENTIER LONG($tX;0)
```

```
TABLEAU ENTIER LONG($tY;0)
```

```
`Axe des x
```

```
Boucle ($Lon_i;0;200;20)
```

```
  AJOUTER A TABLEAU($tX;$Lon_i+10)
```

```
Fin de boucle
```

```
`Données
```

```
AJOUTER A TABLEAU($tY;100)
```

```
AJOUTER A TABLEAU($tY;100)
```

```
AJOUTER A TABLEAU($tY;30)
```

```
AJOUTER A TABLEAU($tY;30)
```

```
AJOUTER A TABLEAU($tY;80)
```

```
AJOUTER A TABLEAU($tY;60)
```

```
AJOUTER A TABLEAU($tY;10)
```

```
AJOUTER A TABLEAU($tY;40)
```

```
AJOUTER A TABLEAU($tY;50)
```

```
AJOUTER A TABLEAU($tY;70)
```

```
$line:=SVG_New_polyline_by_arrays ($SVG;->$tX;->$tY;"red";"none";5)
```

```
`Disposer les marqueurs :
```

```
SVG_SET_MARKER ($line;"startMarker";"start")
```

```
SVG_SET_MARKER ($line;"pointMarker";"middle")
```

```
SVG_SET_MARKER ($line;"endMarker";"end")
```

Référence

SVG_Define_marker.

SVG_SET_OPACITY (objetSVG; opacitéFond{; opacitéTrait})

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
opacitéFond	Entier long →	Opacité (%)
opacitéTrait	Entier long →	Opacité (%)

Description

La commande SVG_SET_OPACITY permet de fixer l'opacité du remplissage et du trait de l'objet de référence objetSVG.

Si ces attributs existaient déjà, leurs valeurs sont remplacées par celles passées en paramètres.

Les valeurs attendues sont comprises entre 0 et 100.

Exemple

```
$svg :=SVG_New `Créer un nouveau document
$object:=SVG_New_rect ($svg ;10;10;200;100;0;0;"red";"blue")
SVG_SET_OPACITY ($object; -1; 50) `Fixer l'opacité du trait à 50%
```

Référence

Couleurs SVG.

SVG_SET_ROUNDING_RECT (objetSVG; rayonX{; rayonY})

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un rectangle SVG
rayonX	Numérique →	Rayon sur l'axe x
rayonY	Numérique →	Rayon sur l'axe y

Description

La commande SVG_SET_ROUNDING_RECT permet de fixer les rayons de l'ellipse utilisée pour l'arrondi des coins d'un rectangle de référence objetSVG. Si ces attributs existaient déjà, leurs valeurs sont remplacées par celles passées en paramètres. Si objetSVG n'est pas la référence d'un rectangle, une erreur est générée.

Les valeurs sont attendues dans le système de coordonnées utilisateur.

Exemple

```
$svg :=SVG_New ` Créer un nouveau document
$object:=SVG_New_rect ($svg ;10;10;200;100)
SVG_SET_ROUNDING_RECT ($object; 20) `Arrondir les angles
```

Référence

SVG_New_rect, SVG_SET_STROKE_LINEJOIN.

SVG_SET_STROKE_BRUSH (objetSVG; couleur)

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
couleur	Chaîne	→ Expression couleur

Description

La commande SVG_SET_STROKE_BRUSH permet de fixer la couleur utilisée pour les traits de l'objet SVG de référence objetSVG. Si cet attribut existait déjà sa valeur, est remplacée par celle passée en paramètre.

Pour plus d'informations sur les couleurs, reportez-vous à la section "Couleurs SVG".

Exemple

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

Référence

Couleurs SVG, SVG_GET_DEFAULT_BRUSHES, SVG_SET_FILL_BRUSH.

SVG_SET_STROKE_LINECAP (objetSVG; mode)

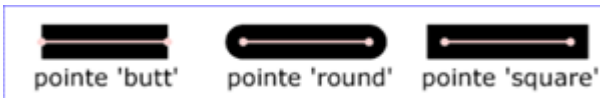
Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
mode	Chaîne	→ Mode de rendu

Description

La commande SVG_SET_STROKE_LINECAP permet de spécifier la forme des extrémités des tracés de l'objet SVG de référence objetSVG. Si cet attribut existait déjà, sa valeur est remplacée par celle passée en paramètre.

Le paramètre mode doit contenir l'une des chaînes suivantes, gérées par le SVG :

- butt (défaut) : standard
- round : arrondi
- square : carré
- inherit : hériter de l'objet parent



Si le paramètre mode contient une autre valeur, une erreur est générée.

Référence

SVG_SET_STROKE_LINEJOIN.

SVG_SET_STROKE_LINEJOIN (objetSVG; mode)

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
mode	Chaîne	→ Mode de rendu

Description

La commande SVG_SET_STROKE_LINEJOIN permet de spécifier la forme des sommets des tracés de l'objet SVG de référence objetSVG. Si cet attribut existait déjà, sa valeur est remplacée par celle passée en paramètre.

Le paramètre mode doit contenir l'une des chaînes suivantes, gérées par le SVG :

- miter (défaut) : standard
- round : arrondi
- bevel : biseau
- inherit : hériter de l'objet parent



Si le paramètre mode contient une autre valeur, une erreur est générée.

Référence

SVG_SET_ROUNDING_RECT, SVG_SET_STROKE_LINECAP.

SVG_SET_STROKE_WIDTH (objetSVG; tailleDuCrayon{; unité})

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
tailleDuCrayon	Numérique →	Épaisseur du tracé
unité	Chaîne →	Unité

Description

La commande SVG_SET_STROKE_WIDTH permet de fixer l'épaisseur des lignes de l'objet SVG de référence objetSVG. Si cet attribut existait déjà, sa valeur est remplacée par celle passée en paramètre.

Passez dans le paramètre tailleDuCrayon la valeur d'épaisseur des tracés. Le paramètre optionnel unité permet de définir l'unité utilisée. Vous pouvez passer l'une des valeurs suivantes : *em*, *ex*, *px*, *pt*, *pc*, *cm*, *mm*, *in* ou *%*. Si le paramètre unité est omis, la valeur tailleDuCrayon est attendue dans le système de coordonnées utilisateur.

Exemple

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

Référence

SVG_SET_STROKE_BRUSH.

SVG_SET_TRANSFORM_FLIP (objetSVG; horizontal{; vertical})

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
horizontal	Booléen	→ Miroir horizontal
vertical	Booléen	→ Miroir vertical

Description

La commande SVG_SET_TRANSFORM_FLIP permet d'appliquer un miroir horizontal et/ou vertical à l'objet SVG de référence objetSVG.

Si le paramètre horizontal est égal à Vrai, un miroir horizontal est appliqué.

Si le paramètre vertical est égal à Vrai, un miroir vertical est appliqué

Exemple

Effet de miroir sur un objet texte :



```
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") `Changer la couleur
```

`Effet :
\$tx:=**SVG_New_text** (svgRef;"4D";10;0;"";96) `Reprendre le même texte
SVG_SET_FONT_COLOR (\$tx;"lightblue") `Changer la couleur
SVG_SET_TRANSFORM_FLIP (\$tx;Vrai) `Appliquer un miroir vertical
SVG_SET_TRANSFORM_SKEW (\$tx;-10) `Inclinaison
SVG_SET_TRANSFORM_TRANSLATE (\$tx;-17;-193) `Repositionner

Référence

SVG_SET_TRANSFORM_SKEW.

SVG_SET_TRANSFORM_MATRIX (objetSVG; a; b; c; d; e; f))

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
a	Numérique →	Elément a de la matrice de transformation
b	Numérique →	Elément b de la matrice de transformation
c	Numérique →	Elément c de la matrice de transformation
d	Numérique →	Elément d de la matrice de transformation
e	Numérique →	Elément e de la matrice de transformation
f	Numérique →	Elément f de la matrice de transformation

Description

La commande SVG_SET_TRANSFORM_MATRIX applique une transformation matricielle à l'objet SVG de référence objetSVG.

Ce type de transformation permet de combiner des transformations telles que, par exemple, une rotation et une translation.

Exemple

Writing with SVG is easy

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

Équivaut à appliquer les 3 transformations suivantes :

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

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

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

Référence

SVG_SET_TRANSFORM_ROTATE.



SVG_SET_TRANSFORM_ROTATE (objetSVG; angle{; x; y})

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
angle	Numérique →	Angle de rotation
x	Numérique →	Coordonnée du centre de rotation sur l'axe x
y	Numérique →	Coordonnée du centre de rotation sur l'axe y

Description

La commande SVG_SET_TRANSFORM_ROTATE applique une rotation de la valeur angle à l'objet SVG de référence objetSVG.

L'angle de rotation est attendu en degrés, la rotation s'effectue dans le sens horaire.

Si les paramètres optionnels x et y ne sont pas passés, la rotation s'effectue par rapport à l'origine du système de coordonnées utilisateur courant. Si ces paramètres sont passés, la rotation s'effectue par rapport au point de coordonnées (x, y).

Exemple



```
svgRef := SVG_New
  `Dessiner un rectangle rouge avec une bordure bleue
$rec:=SVG_New_rect ($svg;150;50;200;400;0;0;"blue";"red";10)
  `Appliquer une rotation de 10° dans le sens horaire par rapport au centre
SVG_SET_TRANSFORM_ROTATE ($rec;370;175;225)
```

Référence

SVG_SET_TRANSFORM_FLIP.

SVG_SET_TRANSFORM_SCALE (objetSVG; échelleX{; échelleY})

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
échelleX	Numérique →	Valeur sur l'axe x
échelleY	Numérique →	Valeur sur l'axe y

Description

La commande SVG_SET_TRANSFORM_SCALE applique un changement d'échelle horizontale et/ou verticale à l'objet SVG de référence objetSVG.

Si la valeur échelleX est non nulle, l'objet est agrandi (valeur >1) ou diminué (valeur < 1) horizontalement du nombre d'unités passé. La valeur 1 équivaut à ne pas changer l'échelle de l'objet.

Si le paramètre échelleY est passé, l'objet est agrandi (valeur >1) ou diminué (valeur < 1) verticalement du nombre d'unités passé. La valeur 1 équivaut à ne pas changer l'échelle de l'objet. Si ce paramètre est omis, sa valeur est supposée égale à échelleX.

Exemple



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

Référence

SVG_SET_TRANSFORM_MATRIX.

SVG_SET_TRANSFORM_SKEW (objetSVG; horizontal{; vertical})

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
horizontal	Numérique →	Valeur de l'inclinaison le long de l'axe x
vertical	Numérique →	Valeur de l'inclinaison le long de l'axe y

Description

La commande SVG_SET_TRANSFORM_SKEW spécifie une inclinaison horizontale et/ou verticale de l'objet SVG de référence objetSVG.

Si la valeur de horizontal est non nulle, l'objet sera incliné horizontalement du nombre d'unités passé, sinon elle est ignorée.

Si la valeur de vertical est non nulle, l'objet sera incliné verticalement du nombre d'unité passé.

Exemple



```
$svg := SVG_New
  `Dessin du fond
SVG_New_rect ($svg; 0; 0; 270; 160; 10; 10; "black"; "gray")
  `Placer le texte...
$tx:=SVG_New_text ($svg;"Hello world!"; 100; 5; ""; 48)
  `en blanc
SVG_SET_FONT_COLOR ($tx; "white")
  `Inclinaison
SVG_SET_TRANSFORM_SKEW ($tx; -50; 10)
```

Référence

SVG_SET_TRANSFORM_FLIP.

SVG_SET_TRANSFORM_TRANSLATE (objetSVG; x{; y})

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
x	Numérique →	Coordonnée sur l'axe x
y	Numérique →	Coordonnée sur l'axe y

Description

La commande SVG_SET_TRANSFORM_TRANSLATE spécifie une translation horizontale et/ou verticale de l'objet SVG de référence objetSVG.

Si la valeur x est non nulle, l'objet sera déplacé horizontalement du nombre d'unités passé, sinon elle est ignorée.

Si le paramètre y est fourni, l'objet sera déplacé verticalement du nombre d'unité passé.

Exemple



```
svgRef := SVG_New
  `Dessiner un rectangle rouge
$Object:=SVG_New_rect (svgRef;0;0;200;100;0;0;"black";"red")
  `Dessiner un carré en 0,0
$Object:=SVG_New_rect (svgRef;0;0;20;20)
  `Déplacer le carré en 150,50
SVG_SET_TRANSFORM_TRANSLATE ($Object;150;50)
```

Référence

SVG_SET_TRANSFORM_ROTATE.

SVG_SET_VIEWBOX (objetSVG; x; y; largeur; hauteur{; mode})

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
x	Entier long →	Position X du rectangle de visualisation
y	Entier long →	Position Y du rectangle de visualisation
largeur	Entier long →	Largeur du rectangle de visualisation
hauteur	Entier long →	Hauteur du rectangle de visualisation
mode	Texte →	Adaptation au rectangle de visualisation

Description

La commande SVG_SET_VIEWBOX permet de définir le rectangle de visualisation de l'objet SVG de référence objetSVG. Si cet attribut existait déjà, sa valeur est remplacée par celle passée en paramètre.

Les valeurs sont attendues dans le système de coordonnées utilisateur.

Le paramètre optionnel mode permet d'indiquer si le graphique doit s'adapter, et comment, à la taille du rectangle de visualisation. La valeur attendue pour mode doit être une de celles reconnues par le SVG : 'none', 'xMinYMin', 'xMidYMin', 'xMaxYMin', 'xMinYMid', 'xMidYMid', 'xMaxYMid', 'xMinYMax', 'xMidYMax', 'xMaxYMax' ou 'true' (équivalent à xMidYMid).

Exemple

```

`Créer un document SVG de 4x8cm
$svg:=SVG_New
SVG_SET_DIMENSIONS($SVG;4;8;"cm")
Déclarer le système de coordonnées utilisateur, ici 1cm = 50 points utilisateur
SVG_SET_VIEWBOX ($svg; 0; 0; 1000; 2000; "true")

```

Référence

SVG_SET_VIEWPORT_FILL.

SVG_SET_VIEWPORT_FILL (objetSVG{; couleur{; opacité}})

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'élément SVG
couleur	Chaîne →	Couleur de remplissage
opacité	Entier long →	Pourcentage d'opacité

Description

La commande SVG_SET_VIEWPORT_FILL permet de fixer la couleur de fond d'un document SVG de référence objetSVG.

Si cet attribut existait déjà, sa valeur est remplacée par celle passée en paramètre. Si objetSVG est un élément SVG qui n'accepte pas cet attribut, une erreur est générée.

Le paramètre optionnel couleur indique la couleur à utiliser pour le fond de l'image. Si ce paramètre est omis ou contient une chaîne vide, c'est le blanc qui est utilisé. Pour plus d'informations sur les couleurs reportez-vous à la section Couleurs SVG.

Le paramètre optionnel opacité permet de préciser la valeur en pourcentage de l'opacité appliqué à ce remplissage. Si ce paramètre est omis et si aucune opacité n'était définie pour le document, la valeur 100% est utilisée.

Référence

Couleurs SVG.

SVG_SET_VISIBILITY (objetSVG{; cacher})

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
cacher	Booléen	→ Vrai = montrer, Faux = cacher

Description

La commande SVG_SET_VISIBILITY cache ou montre l'objet SVG de référence objetSVG. Si objetSVG n'est pas la référence d'un objet pouvant être caché, une erreur est générée.

Si le paramètre optionnel cacher est égal à Vrai ou omis, l'objet est montré. S'il est égal à Faux, l'objet est caché.

Exemple

```
$svg :=SVG_New  
$object:=SVG_New_rect ($svg; 10; 10; 200; 200; 0; 0; "black";" white"; 2)  
SVG_SET_VISIBILITY ($object; Faux) `L'objet est décrit mais ne sera pas rendu.
```

SVG_SET_XY (objetSVG; x{; y})

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un élément SVG
x	Numérique →	Coordonnée sur l'axe x
y	Numérique →	Coordonnée sur l'axe y

Description

La commande SVG_SET_XY permet de fixer les coordonnées du coin supérieur gauche de la région rectangulaire dans laquelle l'objet SVG de référence objetSVG se place. Si ces attributs existaient déjà, leurs valeurs sont remplacées par celles passées en paramètres. Si objetSVG est un élément SVG qui n'accepte pas cet attribut, une erreur est générée.

Les valeurs sont attendues dans le système de coordonnées utilisateur.

Exemple

```
$svg :=SVG_New `Créer un nouveau document
$object:=SVG_New_image ($svg;"#Pictures/logo4D.png") `Placer le logo
SVG_SET_XY ($object; 10; 40) `Modifier la position de l'image
```


3

Couleurs et dégradés



Définitions de couleurs

Le SVG reconnaît toutes les syntaxes alternatives pour les couleurs définies dans la norme CSS2. Les commandes du composant 4D SVG prennent en charge toutes ces syntaxes.

Une couleur peut être exprimée sous l'une des formes suivantes :

- *Format RVB*

Format	Exemple
#rvb	#f00
#rrrvvbb	#ff0000
rvb(r,v,b)	rvb(255, 0, 0)
	rvb(100%, 0%, 0%)

- *Format mot-clé "couleur"*

Le SVG admet une liste étendue de mots-clés de noms de couleur, par exemple "red".

La liste des mots-clés ainsi que leur correspondance RVB figure dans l'Annexe A, Tableau de couleurs. Vous pouvez également visualiser cette liste et insérer directement des valeurs de couleurs via la Palette de couleurs de 4D SVG. Pour plus d'informations sur ce point, reportez-vous à la section Outils de développement.

Opacité

Il est possible de spécifier l'opacité dans les expressions couleurs des commandes du composant en utilisant la syntaxe "couleur:opacité" où opacité est un nombre compris entre 0 (pas de couleur) et 100 (couleur complètement opaque). Ainsi "red:50" sera interprété comme un rouge à 50% d'opacité.

Dégradés

Les dégradés sont des transitions progressives de couleur le long d'un vecteur. Ces dégradés sont définis avec les commandes SVG_Define_linear_gradient et SVG_Define_radial_gradient. Une fois définis, les dégradés sont utilisés par référence en utilisant la syntaxe "url(#NomDuDégradé)".

De même, il est possible de définir une couleur personnalisée associée à une opacité avec la commande SVG_Define_solidColor.

SVG_Color_grey (pourcentage) → Chaîne

Paramètre	Type		Description
pourcentage	Entier	→	Intensité du gris
Résultat	Chaîne	←	Chaîne couleur

Description

La commande `SVG_Color_grey` retourne une chaîne exprimant une couleur grise d'intensité pourcentage. La chaîne retournée est de la forme "rgb(rouge, vert, bleue)" où les 3 valeurs sont égales, syntaxe reconnue par les moteurs de rendu SVG.

Exemple

```
$txtColor:= SVG_Color_grey (60)  
`$txtColor vaut "rgb(102,102,102)"
```

Référence

SVG_Color_RGB_from_long.

SVG_Color_RGB_from_long (couleur{; format}) → Chaîne

Paramètre	Type	Description
couleur	Entier long →	Valeur de la couleur
format	Entier →	Format de la couleur
Résultat	Chaîne ←	Chaîne couleur

Description

La commande `SVG_Color_RGB_from_long` retourne une chaîne exprimant la couleur passée en paramètre sous la forme "RGB(rouge, vert, bleue)", syntaxe reconnue par les moteurs de rendu SVG.

couleur est un entier long de 4 octets dont le format (0x00RRGGBB) est décrit ci-dessous (les octets sont numérotés de 0 à 3 de la droite vers la gauche) :

Octet	Description
2	Composante rouge de la couleur (0..255)
1	Composante verte de la couleur (0..255)
0	Composante bleue de la couleur (0..255)

Le paramètre optionnel `format` permet de spécifier le format désiré pour la chaîne couleur retournée. Les valeurs sont :

Valeur	Format
1 (défaut)	rgb(r,g,b)
2	#rgb
3	#rrggb
4	rgb(r%, g%, b%)

Exemple

```
$txtColor:=SVG_Color_RGB_from_long ($color)
`$txtColor vaut "rgb(255,128,0)" si $color vaut 16744448 (orange)
```

Référence

SVG_Color_grey.

SVG_GET_DEFAULT_BRUSHES (ligne{; fond})

Paramètre	Type	Description
ligne	Pointeur	→ Variable chaîne
fond	Pointeur	→ Variable chaîne

Description

La commande SVG_GET_DEFAULT_BRUSHES renvoie dans la variable pointée par ligne la couleur par défaut courante pour le dessin des lignes.

Si le paramètre optionnel fond est passé, la variable pointée par ce paramètre recevra la couleur par défaut courante utilisée pour les fonds.

Si elles n'ont pas été modifiées, ces couleurs sont respectivement le noir et le blanc.

Exemple

Reportez-vous à l'exemple de la commande SVG_SET_DEFAULT_BRUSHES.

Référence

SVG_SET_DEFAULT_BRUSHES, SVG_SET_STROKE_BRUSH.

SVG_SET_DEFAULT_BRUSHES (ligne{; fond})

Paramètre	Type	Description
ligne	Chaîne →	Couleur de ligne
fond	Chaîne →	Couleur de fond

Description

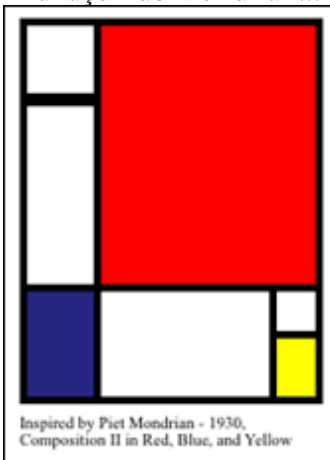
La commande SVG_SET_DEFAULT_BRUSHES permet de fixer les couleurs par défaut utilisées par le composant SVG.

Le paramètre ligne contient la nouvelle couleur qui sera utilisée pour les lignes. Le paramètre optionnel fond contient la nouvelle couleur à utiliser pour le dessin des fonds.

Passez une chaîne vide dans l'un ou l'autre de ces paramètres pour réinitialiser la valeur par défaut du composant, c'est-à-dire le noir et le blanc pour, respectivement, les lignes et le fond.

Exemple

A la façon de Mondrian...



```
$svg:=SVG_New  
  `Fixer le couleurs par défaut  
  SVG_SET_DEFAULT_BRUSHES ("black";"white")
```

```

`Lignes de 4 points d'épaisseur
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)
`Légende
SVG_New_text ($svg;"Inspired by Piet Mondrian - 1930,\rComposition II in Red, Blue, and
Yellow";10;220;"";9)

```

Référence

SVG_GET_DEFAULT_BRUSHES, SVG_SET_STROKE_BRUSH.

4

Dessin

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

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de tracé
x	Numérique →	Coordonnée du nouveau point sur l'axe x
y	Numérique →	Coordonnée du nouveau point sur l'axe y

Description

La commande SVG_ADD_POINT ajoute un ou plusieurs segments au tracé référencé par objetSVGParent. Le tracé peut être de type 'path', 'polyline' ou 'polygon'. Si objetSVGParent n'est pas la référence d'un tracé de ce type, une erreur est générée.

Si plusieurs couples de coordonnées (x, y) sont passés, les différents points seront ajoutés successivement. Dans ce cas, si le dernier couple de coordonnées est incomplet (y manquant), il sera ignoré.

Exemples

Reportez-vous aux exemples de la commande SVG_New_path.

Référence

SVG_New_path, SVG_New_polygon, SVG_New_polyline.

SVG_New_arc (objetSVGParent; x; y; rayon; début; fin; coulPremierPlan; coulArrièrePlan; tailleDuCrayon}}) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de l'élément parent
x	Numérique →	Coordonnée du centre sur l'axe x
y	Numérique →	Coordonnée du centre sur l'axe y
rayon	Numérique →	Rayon du cercle
début	Numérique →	Valeur en degré du début de l'arc
fin	Numérique →	Valeur en degré de la fin de l'arc
coulPremierPlan	Chaîne →	Nom de la couleur ou du dégradé
coulArrièrePlan	Chaîne →	Nom de la couleur ou du dégradé
tailleDuCrayon	Numérique →	Épaisseur du tracé
Résultat	Ref_SVG ←	Référence de l'arc

Description

La commande SVG_New_arc crée un nouvel arc de cercle dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

Les paramètres optionnels coulPremierPlan et coulArrièrePlan contiennent respectivement le nom de la couleur de la ligne et de la couleur de fond (pour plus d'informations sur les couleurs, reportez-vous aux commandes du thème Couleurs et dégradés).

Le paramètre optionnel tailleDuCrayon contient la taille du crayon exprimée en pixels. Sa valeur par défaut est 1.

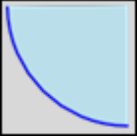
Exemples

1. Dessin d'un arc de 0° à 90° (couleur de remplissage et de bordure, épaisseur du trait par défaut) :



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

2. Dessin d'un arc de cercle de 90° à 180° bleu clair avec un bord bleu et une épaisseur du trait de 2 points :



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

Référence

SVG_PATH_ARC.

SVG_New_circle (objetSVGParent; x; y; rayon; coulPremierPlan; coulArrièrePlan; tailleDuCrayon))) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de l'élément parent
x	Numérique →	Coordonnée du centre sur l'axe x
y	Numérique →	Coordonnée du centre sur l'axe y
rayon	Numérique →	Rayon du cercle
coulPremierPlan	Chaîne →	Nom de la couleur ou du dégradé
coulArrièrePlan	Chaîne →	Nom de la couleur ou du dégradé
tailleDuCrayon	Numérique →	Épaisseur du tracé
Résultat	Ref_SVG ←	Référence du cercle

Description

La commande SVG_New_circle crée un nouveau cercle dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

Le cercle est positionné et dimensionné d'après les coordonnées du centre (x et y) et le rayon passés en paramètres.

Les paramètres optionnels coulPremierPlan et coulArrièrePlan contiennent respectivement le nom de la couleur de la ligne et de la couleur du fond (pour plus d'informations sur les couleurs, reportez-vous aux commandes du thème Couleurs et dégradés).

Le paramètre optionnel tailleDuCrayon contient la taille du crayon exprimée en pixels. Sa valeur par défaut est 1.

Exemples

1. Dessin d'un cercle (couleur de remplissage et de bordure, épaisseur du trait par défaut) :



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

2. Dessin d'un cercle bleu clair avec un bord bleu et une épaisseur du trait de 2 points :



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

Référence

SVG_New_ellipse.

SVG_New_ellipse (objetSVGParent; x; y; rayonX; rayonY; coulPremierPlan{; coulArrièrePlan{; tailleDuCrayon}}) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de l'élément parent
x	Numérique →	Coordonnée sur l'axe x du centre de l'ellipse
y	Numérique →	Coordonnée du centre de l'ellipse sur l'axe y
rayonX	Numérique →	Rayon sur l'axe x
rayonY	Numérique →	Rayon sur l'axe y
coulPremierPlan	Chaîne →	Nom de la couleur ou du dégradé
coulArrièrePlan	Chaîne →	Nom de la couleur ou du dégradé
tailleDuCrayon	Numérique →	Épaisseur du tracé
Résultat	Ref_SVG ←	Référence de l'ellipse

Description

La commande SVG_New_ellipse crée une nouvelle ellipse dans le conteneur SVG désigné par objetSVGParent. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

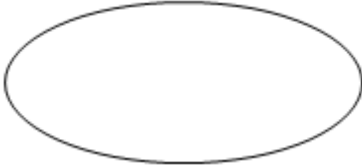
L'ellipse est positionnée et dimensionnée selon les valeurs de x, y, largeur et hauteur.

Les paramètres optionnels coulPremierPlan et coulArrièrePlan contiennent respectivement le nom de la couleur de la ligne et de la couleur de fond (pour plus d'informations sur les couleurs, reportez-vous aux commandes du thème Couleurs et dégradés).

Le paramètre optionnel tailleDuCrayon contient la taille du crayon exprimée en pixels. Sa valeur par défaut est 1.

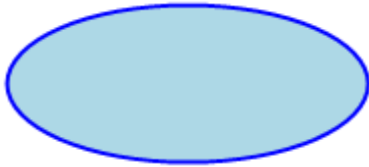
Exemples

1. Dessin d'une ellipse (couleur de remplissage et de bordure, épaisseur du trait par défaut) :



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

2. Dessin d'une ellipse bleu clair avec un bord bleu et une épaisseur de trait de 2 points :



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

Référence

SVG_New_circle, SVG_New_ellipse_bounded.

SVG_New_ellipse_bounded (objetSVGParent; x; y; largeur; hauteur{; coulPremierPlan{; coulArrièrePlan{; tailleDuCrayon}}}) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de l'élément parent
x	Numérique →	Coordonnée du coin supérieur gauche sur l'axe x
y	Numérique →	Coordonnée du coin supérieur gauche sur l'axe y
largeur	Numérique →	Largeur du rectangle englobant
hauteur	Numérique →	Hauteur du rectangle englobant
coulPremierPlan	Chaîne →	Nom de la couleur ou du dégradé
coulArrièrePlan	Chaîne →	Nom de la couleur ou du dégradé
tailleDuCrayon	Numérique →	Épaisseur du tracé
Résultat	Ref_SVG ←	Référence de l'ellipse

Description

La commande SVG_New_ellipse_bounded crée une nouvelle ellipse dans le conteneur SVG désigné par objetSVGParent. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

L'ellipse créée s'inscrit dans le rectangle défini par x, y, largeur et hauteur.

Les paramètres optionnels coulPremierPlan et coulArrièrePlan contiennent respectivement le nom de la couleur de la ligne et de la couleur de fond (pour plus d'informations sur les couleurs, reportez-vous aux commandes du thème Couleurs et dégradés).

Le paramètre optionnel tailleDuCrayon contient la taille du crayon exprimée en pixels. Sa valeur par défaut est 1.

Exemples

1. Dessin d'une ellipse (couleur de remplissage et de bordure, épaisseur du trait par défaut) :



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

2. Dessin d'une ellipse bleu clair avec un bord bleu et une épaisseur du trait de 2 points :



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

Référence

SVG_New_ellipse.

SVG_New_embedded_image (objetSVGParent; image{; x{; y})) → Ref_SVG

Paramètre	Type		Description
objetSVGParent	Ref_SVG	→	Référence de l'élément parent
image	Image	→	Image à inclure
x	Numérique	→	Coordonnée du coin supérieur gauche sur l'axe x
y	Numérique	→	Coordonnée du coin supérieur gauche sur l'axe y
Résultat	Ref_SVG	←	Référence de l'image

Description

La commande SVG_New_embedded_image permet d'inclure l'image dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

L'image sera encodée en base64 puis incluse dans le document.

Le paramètre image doit contenir une variable ou un champ image 4D.

Les paramètres optionnels x et y permettent de préciser la position du coin supérieur de l'image dans le conteneur SVG (valeur par défaut : 0).

Exemple

Inclure l'image 'logo4D.png' située dans le dossier 'Resources' :



```
svgRef:= SVG_New
$Path:=Dossier 4D(Dossier Resources courant)+"logo4D.png"
LIRE FICHIER IMAGE($Path; $Picture)
Si (OK=1)
    objectRef:=SVG_New_embedded_image (svgRef; $Picture)
Fin de si
```

Référence

SVG_New_image, SVG_Open_picture.

SVG_New_image (objetSVGParent; url{; x; y{; largeur; hauteur}}) → Ref_SVG

Paramètre	Type		Description
objetSVGParent	Ref_SVG	→	Référence de l'élément parent
url	Chaîne	→	Adresse de l'image
x	Numérique	→	Coordonnée du coin supérieur gauche sur l'axe x
y	Numérique	→	Coordonnée du coin supérieur gauche sur l'axe y
largeur	Numérique	→	Largeur de l'image
hauteur	Numérique	→	Hauteur de l'image
Résultat	Ref_SVG	←	Référence de l'image

Description

La commande SVG_New_image permet de référencer une image située à l'adresse url dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

Le paramètre url spécifie l'emplacement de l'image et peut prendre plusieurs formes :

- Un **URL local** (chemin d'accès de la forme "file://..."). Dans ce cas, l'image ne sera affichée que si le fichier est effectivement accessible au moment du rendu de l'objet. Cet URL local peut être relatif (de la forme "#Images/monImage.png"), dans ce cas la commande préfixera le chemin d'accès de celui du dossier **Resources** de la base hôte. Si les paramètres largeur et hauteur sont omis, ils seront calculés par la commande (dans ce cas l'exécution de la commande sera plus lente). Si le chemin relatif n'est pas valide, une erreur est générée.
- Un **URL non local** ("http://monSite.com/images/monImage.jpeg"). Dans ce cas, aucune vérification n'est effectuée sur la validité du lien et une erreur sera générée si les paramètres largeur et hauteur sont omis.

Les paramètres optionnels x et y permettent de préciser la position du coin supérieur gauche de l'image dans le conteneur SVG (valeur par défaut : 0).

Les paramètres largeur et hauteur spécifient la taille du rectangle dans lequel sera affichée l'image et déterminent donc la taille et le ratio d'aspect de l'image. Ces paramètres ne sont optionnels que dans le cas d'une image référencée par un chemin relatif dans le dossier **Resources** de la base hôte. Si largeur et/ou hauteur vaut 0, l'image n'est pas rendue.

Exemples

1. Placer l'image 'logo4D.png' située dans le dossier 'Images' du dossier 'Resources' :



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

2. Placer l'image '4dlogo.gif' accessible dans le répertoire 'images' du site '4d.fr' :



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

Référence

SVG_New_embedded_image, SVG_Open_picture.

SVG_New_line (objetSVGParent; débutX; débutY; finX; finY{; couleur{; tailleDuCrayon}}) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de l'élément parent
débutX	Numérique →	Position horizontale de début
débutY	Numérique →	Position verticale de début
finX	Numérique →	Position horizontale de fin
finY	Numérique →	Position verticale de fin
couleur	Chaîne →	Nom de la couleur ou du dégradé
tailleDuCrayon	Numérique →	Épaisseur du tracé
Résultat	Ref_SVG ←	Référence de ligne

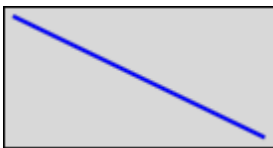
Description

La commande SVG_New_line crée une nouvelle ligne dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. L'objet est positionné d'après les coordonnées débutX, débutY, finX et finY. Le conteneur SVG peut être la racine du document ou toute autre référence à un objet SVG pouvant contenir ce type d'élément.

Le paramètre optionnel couleur contient le nom de la couleur de la ligne (pour plus d'informations sur les couleurs, reportez-vous à la section Couleurs SVG). Le paramètre optionnel tailleDuCrayon définit la taille du crayon exprimée en pixels (valeur par défaut : 1).

Exemple

Dessin d'une ligne bleue de trois pixels d'épaisseur :



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

Référence

SVG_New_polyline, SVG_PATH_LINE_TO.

SVG_New_path (objetSVGParent; x; y; coulPremierPlan{; coulArrièrePlan{; tailleDuCrayon}}) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de l'élément parent
x	Numérique →	Coordonnée du début du tracé sur l'axe x
y	Numérique →	Coordonnée du début du tracé sur l'axe y
coulPremierPlan	Chaîne →	Nom de la couleur ou du dégradé
coulArrièrePlan	Chaîne →	Nom de la couleur ou du dégradé
tailleDuCrayon	Numérique →	Épaisseur du tracé
Résultat	Ref_SVG ←	Référence du tracé

Description

La commande SVG_New_path débute un nouveau tracé dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

Un tracé représente le contour d'une forme. Un tracé se décrit en faisant appel au concept de "point courant". Par analogie avec un dessin sur du papier, le point courant peut être assimilé à la position du crayon. Celle-ci peut changer et le contour d'une forme (ouverte ou fermée) peut être tracé en faisant glisser le crayon selon une ligne droite ou courbe.

Les tracés représentent la géométrie du contour d'un objet, définis selon les instructions des éléments SVG_PATH_MOVE_TO (établit un nouveau point courant), SVG_PATH_LINE_TO (dessine une droite), SVG_PATH_CURVE (dessine une courbe à l'aide d'une courbe de Bézier cubique), SVG_PATH_ARC (dessine un arc circulaire ou elliptique) et SVG_PATH_CLOSE (clôt la forme courante en dessinant une ligne jusqu'au dernier début de tracé). Il est possible d'avoir des tracés composés (c'est-à-dire un tracé avec plusieurs sous-tracés) qui permettent des effets comme un "trou de donut" dans des objets.

Les paramètres x et y permettent de préciser la position du début du tracé dans le conteneur SVG.

Les paramètres optionnels coulPremierPlan et coulArrièrePlan contiennent respectivement le nom de la couleur de la ligne et de la couleur de fond (pour plus d'informations sur les couleurs, reportez-vous à la section Couleurs SVG).

Le paramètre optionnel `tailleDuCrayon` contient la taille du crayon exprimée en pixels. Sa valeur par défaut est 1.

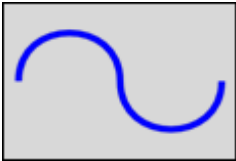
Exemples

1. Dessiner une ligne brisée fermée :



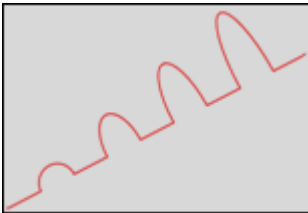
```
svgRef:= SVG_New  
objectRef:=SVG_New_path (svgRef;20;20;"red";"none";5)  
SVG_PATH_LINE_TO (objectRef;40)  
SVG_PATH_LINE_TO (objectRef;40;40)  
SVG_PATH_LINE_TO (objectRef;80;40;80;20;100;20;100;100;80;100;80;80;40;80;40;100;  
20;100)  
SVG_PATH_CLOSE (objectRef)
```

2. Dessiner une courbe de Bézier :



```
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)
```

3. Commandes d'arc dans des données de tracé :



```
svgRef:= SVG_New  
objectRef:=SVG_New_path (svgRef;20;300;"red";"none";2)  
SVG_SET_OPTIONS (SVG_Get_options ?-4) `Passer en coordonnées relatives  
SVG_PATH_LINE_TO (objectRef;50;-25)
```

```

Boucle($Lon_i;1;4;1)
  SVG_PATH_ARC (objectRef; 25;25*$Lon_i; 50;-25;-30)
  SVG_PATH_LINE_TO (objectRef;50;-25)
Fin de boucle

```

4. Tracé complexe (courbe de Bézier cubique) :



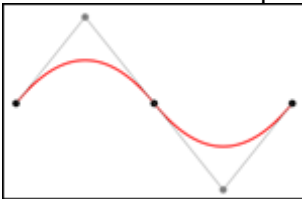
```

`Création d'un nouvel élément SVG
$txt_svg:=SVG_New (174,96; 125,04; "Logo4D"; ""; Vrai)

`Création d'un nouveau tracé
$txt_path:=SVG_New_path ($txt_svg;150,665;13,021)
`Définition des couleurs
SVG_SET_STROKE_BRUSH ($txt_path;"rgb(33,42,111)")
SVG_SET_FILL_BRUSH ($txt_path;"rgb(33,42,111)")
...
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)

```

5. Courbe de Bézier quadratique :



```

`Création d'un nouvel élément SVG
$txt_svg:=SVG_New

```

Initialisation du trait noir et pas de remplissage
SVG_SET_DEFAULT_BRUSHES ("","none")

Dessin d'une courbe de Bézier quadratique en rouge
\$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)

Points finaux en noir
\$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)

Points et lignes de contrôle 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)

Référence

SVG_PATH_CLOSE, SVG_PATH_LINE_TO.

SVG_New_polygon (objetSVGParent{; points{; coulPremierPlan{; coulArrièrePlan{; tailleDuCrayon{}}}) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de l'élément parent
points	Chaîne →	Tracé
coulPremierPlan	Chaîne →	Nom de la couleur ou du dégradé
coulArrièrePlan	Chaîne →	Nom de la couleur ou du dégradé
tailleDuCrayon	Numérique →	Épaisseur du tracé
Résultat	Ref_SVG ←	Référence du polygone

Description

La commande SVG_New_polygon crée une nouvelle forme fermée dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas une référence valide, une erreur est générée.

Le paramètre optionnel points permet de passer les points du tracé de la ligne tels qu'attendus par la norme SVG. Si ce paramètre est omis ou vide, les points pourront être définis avec la commande SVG_ADD_POINTS.

Les paramètres optionnels coulPremierPlan et coulArrièrePlan contiennent respectivement le nom de la couleur de la ligne et de la couleur de fond (pour plus d'informations sur les couleurs, reportez-vous aux commandes du thème Couleurs et dégradés).

Le paramètre optionnel tailleDuCrayon contient la taille du crayon exprimée en pixels. Sa valeur par défaut est 1.

Référence

SVG_New_polygon_by_arrays, SVG_New_rect, SVG_New_regular_polygon.

SVG_New_polygon_by_arrays (objetSVGParent; pointeurTabX; pointeurTabY{; coulPremierPlan{; coulArrièrePlan{; tailleDuCrayon}}}) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de l'élément parent
pointeurTabX	Pointeur →	Coordonnées des points sur l'axe x
pointeurTabY	Pointeur →	Coordonnées des points sur l'axe y
coulPremierPlan	Chaîne →	Nom de la couleur ou du dégradé
coulArrièrePlan	Chaîne →	Nom de la couleur ou du dégradé
tailleDuCrayon	Numérique →	Épaisseur du tracé
Résultat	Ref_SVG ←	Référence du polygone

Description

La commande SVG_New_polygon_by_arrays dessine une forme fermée consistant en un jeu de segments de droites reliés dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

Toutes les valeurs de coordonnées sont dans le système de coordonnées utilisateur.

Les paramètres optionnels coulPremierPlan et coulArrièrePlan contiennent respectivement le nom de la couleur de la ligne et de la couleur de fond (pour plus d'informations sur les couleurs, reportez-vous aux commandes du thème Couleurs et dégradés).

Le paramètre optionnel tailleDuCrayon contient la taille du crayon exprimée en pixels. Sa valeur par défaut est 1.

Exemple

Dessiner une étoile (couleur de bordure et épaisseur du trait par défaut) :



```
TABLEAU ENTIER LONG($tX;0)
TABLEAU ENTIER LONG($tY;0)
```

```
AJOUTER A TABLEAU($tX;129)
AJOUTER A TABLEAU($tY;10)
AJOUTER A TABLEAU($tX;158)
AJOUTER A TABLEAU($tY;96)
AJOUTER A TABLEAU($tX;248)
AJOUTER A TABLEAU($tY;96)
AJOUTER A TABLEAU($tX;176)
AJOUTER A TABLEAU($tY;150)
AJOUTER A TABLEAU($tX;202)
AJOUTER A TABLEAU($tY;236)
AJOUTER A TABLEAU($tX;129)
AJOUTER A TABLEAU($tY;185)
AJOUTER A TABLEAU($tX;56)
AJOUTER A TABLEAU($tY;236)
AJOUTER A TABLEAU($tX;82)
AJOUTER A TABLEAU($tY;150)
AJOUTER A TABLEAU($tX;10)
AJOUTER A TABLEAU($tY;96)
AJOUTER A TABLEAU($tX;100)
AJOUTER A TABLEAU($tY;96)
```

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

Référence

SVG_New_polygon, SVG_New_regular_polygon.

SVG_New_polyline (objetSVGParent{; points{; coulPremierPlan{; coulArrièrePlan{; tailleDuCrayon}}}) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de l'élément parent
points	Chaîne →	Tracé
coulPremierPlan	Chaîne →	Nom de la couleur ou du dégradé
coulArrièrePlan	Chaîne →	Nom de la couleur ou du dégradé
tailleDuCrayon	Numérique →	Épaisseur du tracé
Résultat	Ref_SVG ←	Référence de la ligne

Description

La commande SVG_New_polyline crée une nouvelle ligne brisée ouverte dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas une référence valide, une erreur est générée.

Le paramètre optionnel points permet de passer les points du tracé de la ligne tels qu'attendus par la norme SVG. Si ce paramètre est omis ou vide, les points pourront être définis avec la commande SVG_ADD_POINT.

Les paramètres optionnels coulPremierPlan et coulArrièrePlan contiennent respectivement le nom de la couleur de la ligne et de la couleur de fond (pour plus d'informations sur les couleurs, reportez-vous à la section Couleurs SVG).

Le paramètre optionnel tailleDuCrayon contient la taille du crayon exprimée en pixels. Sa valeur par défaut est 1.



Exemple

Dessin de deux triangles :



```
$polyline:=SVG_New_polyline($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)
```

Référence

SVG_New_polyline_by_arrays.

SVG_New_polyline_by_arrays (objetSVGParent; pointeurTabX; pointeurTabY; coulPremierPlan; coulArrièrePlan; tailleDuCrayon}}) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG	→ Référence de l'élément parent
pointeurTabX	Pointeur	→ Coordonnées des points sur l'axe x
pointeurTabY	Pointeur	→ Coordonnées des points sur l'axe y
coulPremierPlan	Chaîne	→ Nom de la couleur ou du dégradé
coulArrièrePlan	Chaîne	→ Nom de la couleur ou du dégradé
tailleDuCrayon	Numérique	→ Epaisseur du tracé
Résultat	Ref_SVG	← Référence de la ligne

Description

La commande SVG_New_polyline_by_arrays dessine une ligne brisée composée de segments de droite reliés entre eux dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

Typiquement, les éléments 'polyline' définissent des formes ouvertes mais peuvent être utilisés pour des formes fermées. Dans ce cas, le dernier point défini doit être égal au premier.

Toutes les valeurs de coordonnées sont situées dans le système de coordonnées utilisateur.

Les paramètres optionnels coulPremierPlan et coulArrièrePlan contiennent respectivement le nom de la couleur de la ligne et de la couleur de fond (pour plus d'informations sur les couleurs, reportez-vous à la section Couleurs SVG).

Le paramètre optionnel tailleDuCrayon contient la taille du crayon exprimée en pixels. Sa valeur par défaut est 1.

Exemples

1. Dessiner un triangle (couleur de bordure, épaisseur du trait par défaut) :



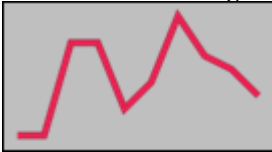
```
TABLEAU ENTIER LONG($tX;0)
TABLEAU ENTIER LONG($tY;0)
```

```
AJOUTER A TABLEAU($tX;10)
AJOUTER A TABLEAU($tY;10)
AJOUTER A TABLEAU($tX;200)
AJOUTER A TABLEAU($tY;100)
AJOUTER A TABLEAU($tX;10)
AJOUTER A TABLEAU($tY;100)
AJOUTER A TABLEAU($tX;10)
AJOUTER A TABLEAU($tY;10)
```

```
svgRef:= SVG_New
```

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

2. Dessiner un diagramme en ligne :



```
TABLEAU ENTIER LONG($tX;0)
TABLEAU ENTIER LONG($tY;0)
  `Axe des X
Boucle($Lon_i;0;200;20)
  AJOUTER A TABLEAU($tX;$Lon_i)
Fin de boucle
  `Valeurs
AJOUTER A TABLEAU($tY;100)
AJOUTER A TABLEAU($tY;100)
AJOUTER A TABLEAU($tY;30)
AJOUTER A TABLEAU($tY;30)
AJOUTER A TABLEAU($tY;80)
```

AJOUTER A TABLEAU(\$tY;60)
AJOUTER A TABLEAU(\$tY;10)
AJOUTER A TABLEAU(\$tY;40)
AJOUTER A TABLEAU(\$tY;50)
AJOUTER A TABLEAU(\$tY;70)

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

Référence

SVG_New_polyline.

SVG_New_rect (objetSVGParent; x; y; largeur; hauteur; arrondiX; arrondiY; coulPremierPlan; coulArrièrePlan; tailleDuCrayon}})) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de l'élément parent
x	Numérique →	X du coin supérieur gauche
y	Numérique →	Y du coin supérieur gauche
largeur	Numérique →	Largeur du rectangle
hauteur	Numérique →	Hauteur du rectangle
arrondiX	Numérique →	Arrondi horizontal
arrondiY	Numérique →	Arrondi vertical
coulPremierPlan	Chaîne →	Nom de la couleur ou du dégradé
coulArrièrePlan	Chaîne →	Nom de la couleur ou du dégradé
tailleDuCrayon	Numérique →	Épaisseur du tracé
Résultat	Ref_SVG ←	Référence du rectangle

Description

La commande SVG_New_rect crée un nouveau rectangle dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Le rectangle est positionné et dimensionné selon les valeurs de x, y, largeur et hauteur. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

Les paramètres optionnel arrondiX et arrondiY permettent d'arrondir les angles suivant les valeurs indiqués. Si le paramètre arrondiY est omis (ou vaut -1), l'arrondi sera régulier. Passez -1 dans ces paramètres pour qu'ils soient ignorés par la commande.

Les paramètres optionnels coulPremierPlan et coulArrièrePlan contiennent respectivement le nom de la couleur de la ligne et de la couleur de fond (pour plus d'informations sur les couleurs, reportez-vous à la section Couleurs SVG).

Le paramètre optionnel tailleDuCrayon définit la taille du crayon exprimée en pixels. Sa valeur par défaut est 1.

Exemples

1. Dessin d'un rectangle (couleur de remplissage et de bordure, épaisseur du trait par défaut) :



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

2. Dessin d'un rectangle bleu muni d'une bordure rouge de trois pixels :



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

3. Dessin d'un carré aux bords arrondis (couleur de remplissage et de bordure, épaisseur du trait par défaut) :



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

4. Dessin d'un rectangle bleu clair avec des bouts arrondis, un bord bleu (épaisseur du trait par défaut) :



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

Référence

SVG_New_polygon, SVG_SET_ROUNDING_RECT.

SVG_New_regular_polygon (objetSVGParent; largeur; nbCôtés{; x{; y{; coulPremierPlan{; coulArrièrePlan{; tailleDuCrayon}}}}) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de l'élément parent
largeur	Numérique →	Diamètre du cercle inscrit
nbCôtés	Numérique →	Nombre de côtés
x	Numérique →	Centre sur l'axe x du cercle circonscrit
y	Numérique →	Centre sur l'axe y du cercle circonscrit
coulPremierPlan	Chaîne →	Nom de la couleur ou du dégradé
coulArrièrePlan	Chaîne →	Nom de la couleur ou du dégradé
tailleDuCrayon	Numérique →	Épaisseur du tracé
Résultat	Ref_SVG ←	Référence du polygone

Description

La commande SVG_New_regular_polygon dessine un polygone régulier ayant un nombre de côtés défini par nbCôtés inscrit dans le cercle de diamètre largeur dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

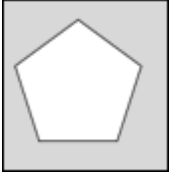
Les paramètres optionnels x et y permettent de préciser le centre du cercle. S'ils sont omis, la figure sera dessinée dans le coin supérieur gauche du document.

Les paramètres optionnels coulPremierPlan et coulArrièrePlan contiennent respectivement le nom de la couleur de la ligne et de la couleur de fond (pour plus d'informations sur les couleurs, reportez-vous à la section Couleurs SVG).

Le paramètre optionnel tailleDuCrayon contient la taille du crayon exprimée en pixels. Sa valeur par défaut est 1.

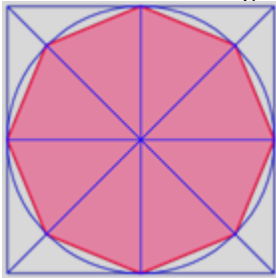
Exemples

1. Dessin d'un pentagone (couleur de remplissage et de bordure, épaisseur du trait par défaut) :



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

2. Dessin d'un octogone, de son cercle circonscrit et des lignes de tracé :



```
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")
```

Référence

SVG_New_polygon.

SVG_PATH_ARC (objetSVGParent; rayonX; rayonY; x; y{; rotation{; cheminArc{}}

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de tracé (élément path)
rayonX	Numérique →	Rayon de l'ellipse sur l'axe x
rayonY	Numérique →	Rayon de l'ellipse sur l'axe y
x	Numérique →	Coordonnée du point d'arrivée sur l'axe x
y	Numérique →	Coordonnée du point d'arrivée sur l'axe y
rotation	Numérique →	Valeur de rotation
cheminArc	Entier long →	Tracé de l'arc

Description

La commande SVG_PATH_ARC dessine un arc elliptique, du point courant jusqu'au point (x, y) à la fin du tracé référencé par objetSVGParent. Si objetSVGParent n'est pas la référence d'un tracé (élément 'path'), une erreur est générée.

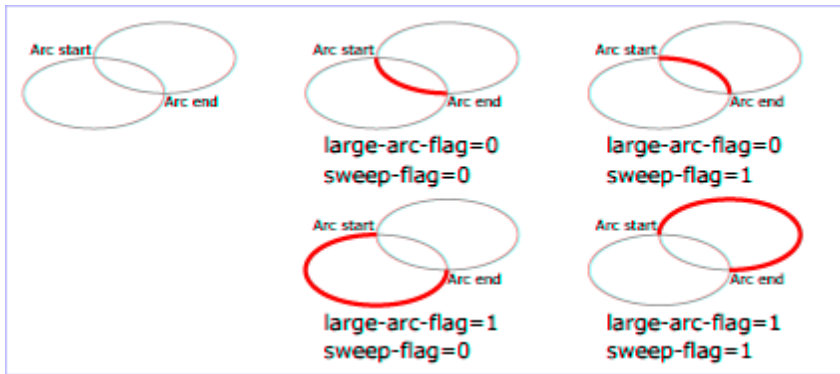
La taille et l'orientation de l'ellipse sont définies par deux rayons (rayonX, rayonY) et une valeur de rotation sur l'axe x, qui indique la rotation de l'ellipse dans son ensemble par rapport au système de coordonnées courant.

Le paramètre optionnel cheminArc permet d'appliquer une combinaison de contraintes dans le but de définir la manière dont l'arc devra être dessiné : la contrainte *large-arc-flag* permet de choisir (ou non) le plus grand des deux arcs possibles (supérieur à 180°) et la contrainte *sweep-flag* définit la direction du tracé (angle positif ou négatif). Vous pouvez passer dans cheminArc l'une des valeurs suivantes, représentant les quatre combinaisons possibles des deux contraintes :

- 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

Lorsque *large-arc-flag* vaut 1, l'arc le plus grand est dessiné. Le plus petit est dessiné lorsque *large-arc-flag* vaut 0. Lorsque *sweep-flag* vaut 1, l'arc est dessiné avec un angle positif. L'arc est dessiné avec un angle négatif lorsque *sweep-flag* vaut 0.

Le schéma suivant illustre les combinaisons possibles :



Par défaut, la valeur de cheminArc est 0 (*large-arc-flag* = 0 et *sweep-flag* = 1).

Exemple

Reportez-vous aux exemples de la commande `SVG_New_path`.

Référence

`SVG_New_arc`.



SVG_PATH_CLOSE (objetSVGParent)

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de tracé (élément path)

Description

La commande SVG_PATH_CLOSE ferme le sous-tracé courant référencé par objetSVGParent avec le dessin d'une ligne droite, du point courant jusqu'au point initial. Si objetSVGParent n'est pas la référence d'un tracé (élément 'path'), une erreur est générée.

Exemple

Reportez-vous aux exemples de la commande SVG_New_path.

Référence

SVG_New_path.

SVG_PATH_CURVE (objetSVGParent{; contrôleDébutX; contrôleDébutY; }contrôleFinX; contrôleFinY; x; y)

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de tracé (élément path)
contrôleDébutX	Numérique →	Coordonnée du point de contrôle sur l'axe x
contrôleDébutY	Numérique →	Coordonnée du point de contrôle sur l'axe y
contrôleFinX	Numérique →	Coordonnée du point de contrôle sur l'axe x
contrôleFinY	Numérique →	Coordonnée du point de contrôle sur l'axe y
x	Numérique →	Coordonnée du point d'arrivée sur l'axe x
y	Numérique →	Coordonnée du point d'arrivée sur l'axe y

Description

La commande SVG_PATH_CURVE ajoute une courbe de Bézier cubique du point courant jusqu'au point de coordonnées (x, y) au tracé référencé par objetSVGParent. Si objetSVGParent n'est pas la référence d'un tracé (élément 'path'), une erreur est générée.

Les paramètres optionnels contrôleDébutX et contrôleDébutY permettent de spécifier la position du point de contrôle en début de courbe. S'ils sont omis, le premier point de contrôle est censé être le reflet du second point de contrôle de la commande précédente par rapport au point courant.

Les paramètres contrôleFinX et contrôleFinY permettent de spécifier la position du point de contrôle en fin de courbe.

Exemple

Reportez-vous aux exemples de la commande SVG_New_path.

Référence

SVG_New_path, SVG_PATH_QCURVE.

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

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de tracé (élément path)
x	Numérique →	Coordonnée sur l'axe x du nouveau point
y	Numérique →	Coordonnée sur l'axe y du nouveau point

Description

La commande SVG_PATH_LINE_TO ajoute un ou plusieurs segments de droite au tracé référencé par objetSVGParent. Si objetSVGParent n'est pas la référence d'un tracé (élément 'path'), une erreur est générée.

Les paramètres x et y permettent de préciser la position du début du tracé dans le conteneur SVG.

- Si seul le paramètre x est fourni, la ligne sera tracée horizontalement depuis le point courant (xc, yc) jusqu'au point (x, yc).
- Si x et y sont passés, une ligne sera tracée depuis le point courant (xc, yc) jusqu'au point (x, y).
- Si plusieurs couples de coordonnées sont passés, les différents points seront ajoutés successivement. Dans ce cas, si le dernier couple de coordonnées est incomplet (y manquant), il sera ignoré.

Exemples

Reportez-vous aux exemples de la commande SVG_New_path.

Référence

SVG_New_path.

SVG_PATH_MOVE_TO (objetSVGParent; x; y)

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de tracé (élément path)
x	Numérique →	Coordonnée sur l'axe x
y	Numérique →	Coordonnée sur l'axe y

Description

La commande SVG_PATH_MOVE_TO commence un nouveau sous-tracé au point de coordonnées (x, y) donné dans le tracé référencé par objetSVGParent. Si objetSVGParent n'est pas la référence d'un tracé (élément 'path'), une erreur est générée.

L'effet produit équivaut à soulever le "crayon" et le déplacer vers un nouvel emplacement. Le point courant devient le nouveau point de début qui sera pris en compte par la commande SVG_PATH_CLOSE.

Exemples

Reportez-vous aux exemples de la commande SVG_New_path.

Référence

SVG_New_path.

SVG_PATH_QCURVE (objetSVGParent{; contrôleX; contrôleY; }x; y)

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de tracé (élément path)
contrôleX	Numérique →	Coordonnée du point de contrôle sur l'axe x
contrôleY	Numérique →	Coordonnée du point de contrôle sur l'axe y
x	Numérique →	Coordonnée du point d'arrivée sur l'axe x
y	Numérique →	Coordonnée du point d'arrivée sur l'axe y

Description

La commande SVG_PATH_QCURVE ajoute une courbe de Bézier quadratique du point courant jusqu'au point de coordonnées (x, y) au tracé référencé par objetSVGParent. Si objetSVGParent n'est pas la référence d'un tracé (élément 'path'), une erreur est générée.

Les paramètres optionnels contrôleX et contrôleY permettent de spécifier la position du point de contrôle en début de courbe. S'ils sont omis, le premier point de contrôle est censé être le reflet du second point de contrôle de la commande précédente par rapport au point courant.

Exemple

Reportez-vous aux exemples de la commande SVG_New_path.

Référence

SVG_PATH_CURVE.

SVG_Use (objetSVGParent; id{; x; y; largeur; hauteur{; mode})) → Ref_SVG

Paramètre	Type		Description
objetSVGParent	Ref_SVG	→	Référence de l'élément parent
id	Chaîne	→	Nom du symbole
x	Numérique	→	Position X du rectangle de visualisation
y	Numérique	→	Position Y du rectangle de visualisation
largeur	Numérique	→	Largeur du rectangle de visualisation
hauteur	Numérique	→	Hauteur du rectangle de visualisation
mode	Chaîne	→	Adaptation au rectangle de visualisation
Résultat	Ref_SVG	←	Référence de l'objet SVG

Description

La commande SVG_Use place une occurrence du symbole id dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG ou si id n'est pas le nom d'un objet du document SVG, une erreur est générée.

Un symbole est utilisé pour définir des objets graphiques, il n'est jamais rendu directement mais peut être instancié à l'aide de la commande SVG_Use.

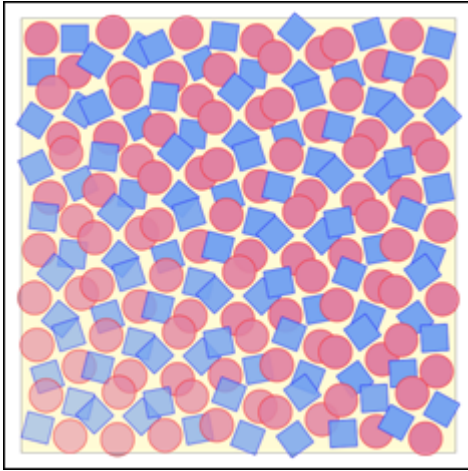
Le paramètre id spécifie le nom du symbole.

Les paramètres optionnels x, y, largeur et hauteur spécifient le rectangle de la zone de visualisation ('viewBox').

Le paramètre optionnel mode permet d'indiquer si le graphique doit s'adapter, et comment, à la taille du rectangle de visualisation. (cf. commande SVG_New).

Exemple

Nous souhaitons définir un graphique composé de deux ronds rouges et deux carrés bleus. L'idée est d'utiliser ce graphique en faisant varier sa position, son opacité et sa rotation dans une boucle de création de 36 occurrences.



```

$SVG:=SVG_New
  `Dessin du fond
SVG_New_rect ($SVG;20;20;650;650;0;0;"gray";"lemonchiffon")
  `Définition d'un symbole composé de 2 carrés et 2 cercles
$Symbol:=SVG_Define_symbol ($SVG;"MySymbol";0;0;110;110;"true")
SVG_New_circle ($Symbol;30;30;25;"red";"palevioletré")
SVG_New_rect ($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";"palevioletré")
  `Dans un groupe...
$g:=SVG_New_group ($SVG)
  `...positionné à 20 unités du coin supérieur gauche du document...
SVG_SET_TRANSFORM_TRANSLATE ($g;20;20)
  `...placer 36 motifs en faisant varier la position, l'opacité et la rotation
Boucle ($x;0;540;90) `6 colonnes
  Boucle ($y;0;540;90) `6 lignes
    $use:=SVG_Use ($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))
  Fin de boucle
Fin de boucle

```

Référence

SVG_Define_symbol.

5

Documents



SVG_CLEAR {(objetSVG)}

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'objet SVG

Description

The SVG_CLEAR commande libère l'espace mémoire occupé par l'objet SVG désigné par objetSVG.

Si objetSVG n'est pas un objet SVG racine créé avec les commandes SVG_New, SVG_Copy ou SVG_Open_file, une erreur est générée.

Si le paramètre objetSVG n'est pas passé, la commande libère tous les arbres SVG créés par l'intermédiaire des commandes SVG_New, SVG_Copy ou SVG_Open_file. Cette syntaxe est utile durant la phase de développement, dans le cas où une référence SVG peut être créée mais l'espace mémoire non libéré, une erreur ayant empêché de terminer l'exécution de la méthode. Dans un développement final, toute référence SVG qui n'est plus utilisée doit être libérée avec la commande SVG_CLEAR.

Référence

SVG_Copy, SVG_New, SVG_Open_file.

SVG_Copy (objetSVG) → Ref_SVG

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence de l'objet SVG à copier
Résultat	Ref_SVG ←	Référence du nouvel objet SVG

Description

La commande SVG_Copy crée un nouveau document SVG qui est une copie du document référencé par objetSVG.

La commande retourne une chaîne de 16 caractères (Ref_SVG) constituant la référence en mémoire de la structure virtuelle du nouveau document. Cette référence devra être utilisée avec les autres commandes du composant pour désigner le document.

Important : Une fois que vous n'en avez plus besoin, n'oubliez pas d'appeler la commande SVG_CLEAR avec cette référence afin de libérer la mémoire.

Exemples

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

Référence

SVG_CLEAR, SVG_New.

SVG_Export_to_picture (objetSVG{; typeExport}) → Image

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'objet SVG
typeExport	Entier long →	0=Ne pas stocker la source de données 1 (défaut)=Copier la source de données, 2=Prendre possession de la source de données
Résultat	Image ←	Image rendue par le moteur SVG

Description

La commande SVG_Export_to_picture commande retourne l'image décrite par la structure SVG référencée par objetSVG.

Le paramètre facultatif typeExport vous permet de définir la manière dont la source de données XML doit être prise en charge par la commande. Pour plus d'informations sur ce paramètre, reportez-vous à la description de la commande SVG EXPORTER VERS IMAGE de 4D. Si ce paramètre est omis, la valeur par défaut est 1, Copier source données XML.

Exemple

```

svgRef:=SVG_New(500;200; "Test composant")
...
MyPicture:= SVG_Export_to_picture (svgRef;0)

SVG_CLEAR (svgRef )

```

Référence

SVG_Export_to_XML, SVG_Open_picture, SVG_SAVE_AS_PICTURE.

SVG_Export_to_XML (objetSVG) → Texte

Paramètre	Type		Description
objetSVG	Ref_SVG	→	Référence d'objet SVG
Résultat	Texte	←	Texte XML du document SVG

Description

La commande SVG_Export_to_XML retourne le texte XML de description de la structure SVG référencée par objetSVG.

Exemple

```
svgRef :=SVG_New (500; 200; "Test composant")
...
MyText:= SVG_Export_to_XML (svgRef)
SVG_CLEAR (svgRef)
```

Référence

SVG_Export_to_picture, SVG_SAVE_AS_TEXT.

SVG_New {(largeur; hauteur; titre; description; rectangle; affichage)}} → Ref_SVG

Paramètre	Type	Description
largeur	Numérique →	Largeur du document en points
hauteur	Numérique →	Hauteur du document en points
titre	Chaîne →	Titre du document
description	Chaîne →	Description
rectangle	Booléen →	Définir le rectangle de visualisation
affichage	Entier →	Format d'affichage de l'image
Résultat	Ref_SVG ←	Référence de l'objet SVG

Description

La commande SVG_New crée un nouveau document SVG et retourne sa référence.

Les paramètres optionnels largeur et hauteur permettent de délimiter l'espace du document SVG. Ces deux paramètres doivent être exprimés en points utilisateur ('px'). Si vous souhaitez utiliser une autre unité, utilisez la commande SVG_SET_DIMENSIONS.

Les paramètres optionnels titre et description permettent d'associer au documents des informations de contenu.

Si vous passez Vrai dans le paramètre optionnel rectangle, le rectangle de visualisation (attribut 'viewBox') est automatiquement fixé à la taille du document créé.

Note : Il est possible de modifier les coordonnées du rectangle de visualisation du graphique et de régler plus précisément l'adaptation de l'image à ce dernier avec la commande SVG_SET_VIEWBOX.

Le paramètre optionnel affichage permet d'indiquer si le graphique doit s'adapter à la taille du document. Vous pouvez passer dans ce paramètre l'une des constantes du thème "Formats d'affichage des images" de 4D suivantes : Proportionnelle centrée ou Non tronquée.

La commande retourne une chaîne de 16 caractères (Ref_SVG) constituant la référence en mémoire de la structure virtuelle du document. Cette référence devra être utilisée avec les autres commandes du composant pour désigner le document.

Important : Une fois que vous n'en avez plus besoin, n'oubliez pas d'appeler la commande SVG_CLEAR avec cette référence afin de libérer la mémoire.

Exemple

```
svgRef:=SVG_New  
svgRef:=SVG_New(500;200)  
svgRef:= SVG_New (900; 700; "Test composant"; "Ceci est un exemple"; Vrai;  
Non tronquée)
```

Référence

SVG_CLEAR, SVG_Copy, SVG_Open_file.

SVG_Open_file (chemin) → Ref_SVG

Paramètre	Type	Description
chemin	Chaîne →	Chemin d'accès du document SVG à ouvrir
Résultat	Ref_SVG ←	Référence du document ouvert

Description

La commande SVG_Open_file analyse (et valide avec la DTD) le document SVG se trouvant à l'emplacement désigné par le paramètre chemin et retourne une référence SVG (chaîne de 16 caractères) pour ce document.

Important : Une fois que vous n'en avez plus besoin, n'oubliez pas d'appeler la commande SVG_CLEAR avec cette référence afin de libérer la mémoire.

Référence

SVG_CLEAR, SVG_Copy, SVG_New.

SVG_Open_picture (image) → Ref_SVG

Paramètre	Type		Description
image	Image	→	Variable ou champ image 4D
Résultat	Ref_SVG	←	Référence du document SVG

Description

La commande `SVG_Open_picture` analyse une image SVG et retourne une référence SVG pour cette image. Si `image` ne contient pas une image SVG, la commande retourne une chaîne vide.

Important : Une fois que vous n'en avez plus besoin, n'oubliez pas d'appeler la commande `SVG_CLEAR` avec cette référence afin de libérer la mémoire.

Exemple

```
LIRE FICHIER IMAGE("";$image)
Si (OK=1)
  $ref:=SVG_Open_picture ($image)
  ...
  SVG_CLEAR ($ref)
Fin de si
```

Référence

`SVG_Export_to_picture`.

SVG_SAVE_AS_PICTURE (objetSVG; nomFichier{; codec})

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'objet SVG
nomFichier	Chaîne	→ Nom du document ou Chemin d'accès complet du document
codec	Chaîne	→ Identifiant de codec d'image

Description

La commande SVG_SAVE_AS_PICTURE écrit le contenu de l'objet SVG désigné par objetSVG dans le fichier image désigné par nomFichier. Si objetSVG n'est pas un document SVG, une erreur est générée.

Vous pouvez passer dans nomFichier le chemin d'accès complet du fichier, ou uniquement son nom, auquel cas le fichier sera créé à côté du fichier de structure de la base. Si vous passez une chaîne vide (""), la boîte de dialogue standard d'enregistrement de fichiers apparaît, permettant à l'utilisateur de désigner le nom, l'emplacement et le format du fichier à créer.

Le paramètre optionnel codec vous permet de définir le format dans lequel l'image doit être sauvegardée. Si ce paramètre est omis, l'image est sauvegardée au format png.

Exemple

```
svgRef:=SVG_New(500;200; "Statistiques des ventes")
...
SVG_SAVE_AS_PICTURE (svgRef ;"test.png") `Sauvegarde
SVG_SAVE_AS_PICTURE (svgRef ;"test.gif" ;".gif")
SVG_CLEAR(svgRef )
```

Référence

SVG_Export_to_picture.

SVG_SAVE_AS_TEXT (objetSVG; nomFichier)

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'objet SVG
nomFichier	Chaîne	→ Nom du document ou Chemin d'accès complet du document

Description

La commande SVG_SAVE_AS_TEXT écrit le contenu de l'objet SVG désigné par objetSVG dans le fichier disque désigné par nomFichier. Si objetSVG n'est pas un document SVG, une erreur est générée.

Vous pouvez passer dans nomFichier le chemin d'accès complet du fichier ou uniquement son nom, auquel cas le fichier sera créé à côté du fichier de structure de la base. Si vous passez une chaîne vide ("") dans nomFichier, la boîte de dialogue standard d'enregistrement de fichiers apparaît, permettant à l'utilisateur de désigner le nom, l'emplacement et le format du fichier à créer.

Exemple

```
svgRef:=SVG_New(500;200; "Statistiques des ventes")
...
SVG_SAVE_AS_TEXT (svgRef ;"test.svg")
  `Le document est sauvegardé à côté de la structure
SVG_CLEAR (svgRef)
```

Référence

SVG_Export_to_XML.

SVG_Validate_file (cheminAccès) → Booléen

Paramètre	Type	Description
cheminAccès	Chaîne →	Chemin d'accès du document SVG à valider
Résultat	Booléen ←	Vrai si le document correspond à la DTD

Description

La commande `SVG_Validate_file` tente de valider le document SVG stocké sur disque désigné par le paramètre `cheminAccès` avec la DTD (1.0). La commande retourne Vrai si le document est bien formé et Faux sinon.

6

Filtres

Les commandes du thème "Filtres" permettent de définir des effets de filtre applicables aux éléments SVG. Un effet de filtre consiste en une succession d'opérations graphiques, appliquées sur un graphique source, dont le produit est un graphique modifié.

Le résultat de l'effet de filtre est rendu sur l'appareil cible à la place du graphique source original.

Un filtre est défini grâce à la commande `SVG_Define_filter`, placée dans le thème "Structure et Définitions" et appliqué grâce à la commande `SVG_SET_FILTER`, placée dans le thème "Attributs". Les commandes du thème "Filtres" permettent de construire les opérations de filtrage ou "primitives de filtre".

SVG_Filter_Blend (refFiltre; image; imageFond{; mode{; nom}) → Ref_SVG

Paramètre	Type	Description
refFiltre	Ref_SVG	→ Référence de filtre
image	Chaîne	→ Image source
imageFond	Chaîne	→ Image de fond source
mode	Chaîne	→ Mode de mélange
nom	Chaîne	→ Cible de la primitive de filtre
Résultat	Ref_SVG	← Référence de la primitive

Description

La commande SVG_Filter_Blend définit un filtre de composition pour le filtre refFiltre et retourne sa référence. Si refFiltre n'est pas une référence de filtre, une erreur est générée.

Ce filtre compose les deux sources, imageFond et image, à l'aide de modes de mélange couramment employés par les logiciels d'imagerie.

Le paramètre facultatif mode permet de définir le mode de combinaison des pixels utilisé pour le mélange (cf. spécification SVG). Sa valeur doit être : "normal" (valeur par défaut), "multiply", "screen", "darken" ou "lighten".

Le paramètre optionnel nom est le nom éventuellement assigné au résultat de cette primitive de filtre.

Référence

SVG_Filter_Blur, SVG_Filter_Offset.

SVG_Filter_Blur (refFiltre; déviation{; entrée{; nom}}) → Ref_SVG

Paramètre	Type		Description
refFiltre	Ref_SVG	→	Référence de filtre
déviatiion	Numérique	→	Déviatiion standard pour l'opération de flou
entrée	Chaîne	→	Source de la primitive de filtre
nom	Chaîne	→	Cible de la primitive de filtre
Résultat	Ref_SVG	←	Référence de primitive

Description

La commande SVG_Filter_Blur définit un flou gaussien pour le filtre refFiltre et retourne sa référence. Si refFiltre n'est pas une référence de filtre, une erreur est générée.

Le paramètre déviatiion permet de définir la déviatiion standard pour l'opération de flou. Si le nombre est entier, la même déviatiion sera appliquée sur les axes X et Y. Si le nombre comporte une partie décimale, la partie entière représente la déviatiion à appliquer sur l'axe X et la partie décimale la déviatiion à appliquer sur l'axe Y.

Le paramètre optionnel entrée identifie la source graphique de la primitive de filtre.

Le paramètre optionnel nom est le nom éventuellement assigné au résultat de cette primitive de filtre.

Référence

SVG_Filter_Blend, SVG_Filter_Offset.

SVG_Filter_Offset (refFiltre; dx{; dy{; entrée{; nom}}}) → Ref_SVG

Paramètre	Type	Description
refFiltre	Ref_SVG →	Référence de filtre
dx	Numérique →	Décalage sur l'axe x
dy	Numérique →	Décalage sur l'axe y
entrée	Chaîne →	Source de la primitive de filtre
nom	Chaîne →	Cible de la primitive de filtre
Résultat	Ref_SVG ←	Référence de primitive

Description

La commande SVG_Filter_Offset définit un décalage pour le filtre refFiltre et retourne sa référence. Si refFiltre n'est pas une référence de filtre, une erreur est générée.

Le paramètre dx est la valeur du décalage horizontal.

Le paramètre optionnel dy est la valeur du décalage vertical.

Le paramètre optionnel entrée identifie la source graphique de la primitive de filtre.

Le paramètre optionnel nom est le nom éventuellement assigné au résultat de cette primitive de filtre.

Référence

SVG_Filter_Blend, SVG_Filter_Blur.

7

Structure et Définitions

SVG_Define_filter (objetSVGParent; id{; canevasX; canevasY{; largeurCanevas; hauteurCanevas{; unitéCanevas{; unitéFiltre}}})) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de l'élément parent
id	Chaîne →	Nom du symbole
canevasX	Numérique →	Coordonnée sur l'axe x
canevasY	Numérique →	Coordonnée sur l'axe y
largeurCanevas	Numérique →	Largeur du rectangle cible
hauteurCanevas	Numérique →	Hauteur du rectangle cible
unitéCanevas	Chaîne →	Système de coordonnées du canevas
unitéFiltre	Chaîne →	Système des valeurs du filtre
Résultat	Ref_SVG ←	Référence du filtre

Description

La commande SVG_Define_filter définit un nouveau filtre dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

Un filtre est une succession d'opérations graphiques qui seront appliquées sur l'élément cible. L'élément filtre n'est jamais rendu directement, il sera appliqué à un objet grâce à la commande SVG_SET_FILTER.

Le paramètre id spécifie le nom du filtre. Le nom sera utilisé pour associer un filtre à un objet. Si un élément de même nom existait, il est remplacé.

Les paramètres optionnels canevasX, canevasY, largeurCanevas et hauteurCanevas définissent une région rectangulaire du document sur laquelle ce filtre s'applique.

Le paramètre optionnel unitéCanevas définit le système de coordonnées pour les quatre paramètres précédents. Les valeurs attendues sont "userSpaceOnUse" ou "objectBoundingBox" (valeur par défaut).

Le paramètre optionnel `unitéFiltre` définit le système de coordonnées pour les longueurs et les propriétés de définition du filtre. Les valeurs attendues sont "userSpaceOnUse" (valeur par défaut) ou "objectBoundingBox".

Référence

SVG_Filter_Blend, SVG_Filter_Blur, SVG_Filter_Offset, SVG_SET_FILTER.

SVG_Define_linear_gradient (objetSVGParent; id; couleurDébut; couleurFin{; rotation}) → Chaîne

Paramètre	Type		Description
objetSVGParent	Ref_SVG	→	Référence de l'élément parent
id	Chaîne	→	Nom du dégradé
couleurDébut	Chaîne	→	Couleur de démarrage
couleurFin	Chaîne	→	Couleur de fin
rotation	Entier	→	Rotation du vecteur de dégradé
Résultat	Chaîne	←	Référence du dégradé

Description

La commande SVG_Define_linear_gradient définit un nouveau dégradé linéaire dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

Un dégradé consiste en une transition de couleur progressive continue au long d'un vecteur, d'une couleur à l'autre. Une fois définis, les dégradés sont appelés sur un élément graphique donné. Vous devez indiquer si l'élément doit être rempli ou liseré avec le dégradé.

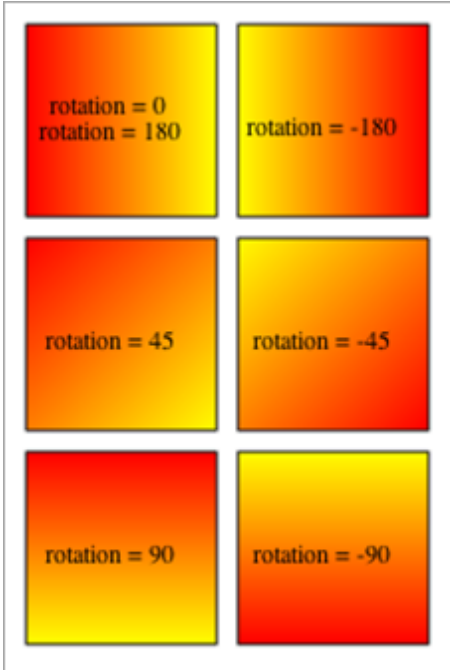
Le paramètre id spécifie le nom du dégradé. Si un élément de même nom existait, il est remplacé. Ce nom sera utilisé pour désigner le dégradé à chaque fois qu'une expression couleur sera attendue via la syntaxe "url(#ID)".

Les paramètres couleurDébut et couleurFin spécifient les couleurs utilisées pour débiter et terminer le dégradé.

Le paramètre optionnel rotation définit la position et le sens du vecteur de dégradé (cf. exemple)

Exemple

Dessiner 6 carrés pleins utilisant chacun un serveur de peinture de dégradé linéaire en variant la rotation et le sens du vecteur de dégradé



```
$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;Centré)
```

```
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; Centré)
```

```
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; Centré)
```

```
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; Centré)
```

```
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; Centré)
```

```
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; Centré)
```

```
  `Sauvegarder le document
SVG_SAVE_AS_TEXT ($svg;"test.svg")
  `Libérer la mémoire
SVG_CLEAR ($svg)
```

Référence

Couleurs SVG, SVG_Define_radial_gradient.

SVG_Define_marker (objetSVGParent; id{; x; y{; largeur; hauteur{; orientation}}}) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de l'élément parent
id	Chaîne →	Nom du symbole
x	Numérique →	Coordonnée sur l'axe x du point de référence
y	Numérique →	Coordonnée sur l'axe y du point de référence
largeur	Numérique →	Largeur du marqueur
hauteur	Numérique →	Hauteur du marqueur
orientation	Numérique →	Orientation du marqueur
Résultat	Ref_SVG ←	Référence du marqueur

Description

La commande SVG_Define_marker crée un marqueur dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

Un objet marqueur est utilisé pour le dessin de flèches ou de marqueurs multiples (les points d'une courbe par exemple). Un marqueur sera attaché à un élément SVG avec la commande SVG_SET_MARKER.

Le paramètre id spécifie le nom du marqueur. Le nom sera utilisé pour associer un marqueur à un objet. Si un élément de même nom existait, il est remplacé.

Les paramètres optionnels x et y spécifient les coordonnées du point de référence qui doit s'aligner exactement sur la position du marqueur.

Les paramètres optionnels largeur et hauteur précisent la largeur et la hauteur du rectangle de rendu.

Le paramètre optionnel orientation permet de régler l'orientation du marqueur. Une valeur comprise entre 0 et 360 représente l'angle entre l'axe x du marqueur et celui de l'espace utilisateur. Si ce paramètre est omis ou si sa valeur n'est pas dans l'intervalle [0 - 360] le placement sera calculé automatiquement par le moteur de rendu en fonction de l'objet.

Exemple

Reportez-vous à l'exemple de la commande `SVG_SET_MARKER`.

Référence

`SVG_SET_MARKER`.

SVG_Define_radial_gradient (objetSVGParent; id; couleurDébut; couleurFin{; cx; cy; r{; fx; fy}}) → Chaîne

Paramètre	Type		Description
objetSVGParent	Ref_SVG	→	Référence de l'élément parent
id	Chaîne	→	Nom du dégradé
couleurDébut	Chaîne	→	Couleur de démarrage
couleurFin	Chaîne	→	Couleur de fin
cx	Entier	→	Coordonnée du centre de couleurFin sur l'axe x
cy	Entier	→	Coordonnée du centre de couleurFin sur l'axe y
r	Entier	→	Rayon de couleurFin
fx	Entier	→	Coordonnée du centre de couleurDébut sur l'axe x
fy	Entier	→	Coordonnée du centre de couleurDébut sur l'axe y
Résultat	Chaîne	←	Référence du dégradé

Description

La commande SVG_Define_radial_gradient définit un nouveau dégradé radial dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

Un dégradé consiste en une transition de couleur progressive continue au long d'un vecteur, d'une couleur à l'autre. Une fois définis, les dégradés sont appelés sur un élément graphique donné. Vous devez indiquer si l'élément doit être rempli ou liseré avec le dégradé.

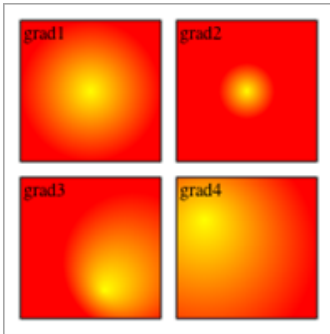
Le paramètre id spécifie le nom du dégradé. Si un élément de même nom existait, il est remplacé. Ce nom sera utilisé pour désigner le dégradé à chaque fois qu'une expression couleur sera attendue via la syntaxe "url(#ID)".

Les paramètres couleurDébut et couleurFin spécifient les couleurs utilisées pour débiter et terminer le dégradé.

Les paramètres optionnels cx, cy et r définissent, en pourcentage, le cercle limite externe de couleurFin du dégradé. Leurs valeurs doivent être comprises entre 0 et 100.

Les paramètres optionnels *fx* et *fy* définissent, en pourcentage, le foyer du dégradé. La couleurDébut commence au point [*fx*,*fy*]. Leurs valeurs doivent être comprises entre 0 et 100. Si ces arguments sont omis, ce point coïncide avec [*cx*,*cy*].

Exemple



```
$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)
```

```
`Sauvegarder le document
SVG_SAVE_AS_TEXT ($svg;"test.svg")
`Libérer la mémoire
SVG_CLEAR ($svg)
```

Référence

Couleurs SVG, SVG_Define_linear_gradient.

SVG_Define_shadow (objetSVGParent; id{; déviation{; offsetX{; offsetY{}}}) → Ref_SVG

Paramètre	Type		Description
objetSVGParent	Ref_SVG	→	Référence de l'élément parent
id	Chaîne	→	Nom du filtre
déviation	Numérique	→	Valeur de dispersion de l'ombre
offsetX	Numérique	→	Décalage sur l'axe x
offsetY	Numérique	→	Décalage sur l'axe y
Résultat	Ref_SVG	←	Référence du filtre

Description

La commande SVG_Define_shadow définit un nouveau filtre d'ombrage dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

Un fois défini, un filtre est appliqué aux objets souhaités via la commande SVG_SET_FILTER.

Le paramètre id spécifie le nom du filtre. Le nom sera utilisé pour associer un filtre à un objet. Si un élément de même nom existait, il est remplacé.

Le paramètre optionnel déviation définit l'intensité de la dispersion de l'ombre. La valeur par défaut est 4.

Les paramètres optionnels offsetX et offsetY définissent respectivement le décalage horizontal et vertical de l'ombre par rapport à l'objet. La valeur par défaut est 4.

Exemple

Déclaration d'un filtre permettant de générer une ombre sous un texte :



```
$svg:=SVG_New
```

```
$text:=SVG_New_text ($svg;"SVG";52;76-45;"Verdana";45)
```

```
SVG_SET_FONT_COLOR ($text;"red")
```

```
  `Définir le filtre
```

```
SVG_Define_shadow ($svg;"myShadow")
```

```
  `et l'appliquer au texte
```

```
SVG_SET_FILTER ($text;"myShadow")
```

Référence

SVG_SET_FILTER.

SVG_Define_solidColor (objetSVGParent; id; couleur{; opacité}) → Ref_SVG

Paramètre	Type		Description
objetSVGParent	Ref_SVG	→	Référence de l'élément parent
id	Chaîne	→	Nom de la couleur
couleur	Chaîne	→	Expression couleur
opacité	Entier long	→	Opacité
Résultat	Ref_SVG	←	Référence de la couleur

Description

La commande SVG_Define_solidColor définit une nouvelle couleur personnalisée dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

Le paramètre id spécifie le nom de la couleur. Le nom sera utilisé pour associer une couleur à un objet. Si un élément de même nom existait, il est remplacé.

Le paramètre couleur est une expression couleur reconnue par le SVG (cf. Couleurs SVG).

Le paramètre optionnel opacité permet de préciser une opacité (de 0 à 100) pour cette couleur. Si ce paramètre est omis, l'opacité est de 100 %.

Pour utiliser la couleur ainsi définie comme peinture de remplissage ou de contour, passez la chaîne "url(#id)" comme valeur lorsqu'une expression couleur est attendue.

Exemple

```
SVG_Define_solidColor($svg; "MaCouleur"; "blue"; 50) `Définir un bleu à 50 %
...
SVG_New_rect ($svg; 0; 0; 20; 20; 0; 0; "url(#MaCouleur)" ; "url(#MaCouleur)")
...
$line:=SVG_New_line (10; 10 ;100 ;100)
SVG_SET_STROKE_BRUSH ($line; "url(#MaCouleur)")
```

Référence

Couleurs SVG.

SVG_Define_symbol (objetSVGParent; id{; x{; y{; largeur{; hauteur{; mode}}}}) → Ref_SVG

Paramètre	Type		Description
objetSVGParent	Ref_SVG	→	Référence de l'élément parent
id	Chaîne	→	Nom du symbole
x	Numérique	→	Position X du rectangle de visualisation
y	Numérique	→	Position Y du rectangle de visualisation
largeur	Numérique	→	Largeur du rectangle de visualisation
hauteur	Numérique	→	Hauteur du rectangle de visualisation
mode	Chaîne	→	Adaptation au rectangle de visualisation
Résultat	Ref_SVG	←	Référence du symbole

Description

La commande SVG_Define_symbol crée un symbole dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

Un objet symbole est utilisé pour définir des objets graphiques qui pourront être instanciés en utilisant la commande SVG_Use.

Le paramètre id spécifie le nom du symbole.

Les paramètres optionnels x, y, largeur et hauteur spécifient le rectangle de la zone de visualisation ('viewBox').

Le paramètre optionnel mode permet d'indiquer si le graphique doit s'adapter, et comment, à la taille du rectangle de visualisation. Pour plus d'informations sur ce point, reportez-vous à la description de la commande SVG_New.

Exemple

Reportez-vous à la description de la commande SVG_Use.

Référence

SVG_Use.

SVG_New_group (objetSVGParent{; id{; url{; cible{}}}) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG	→ Référence de l'élément parent
id	Chaîne	→ Nom du groupe
url	Chaîne	→ Lien externe
cible	Chaîne	→ Cible du lien
Résultat	Ref_SVG	← Référence du groupe

Description

La commande SVG_New_group crée un groupe dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

Un groupe (élément 'g') permet de regrouper plusieurs éléments graphiques reliés, qui hériteront des propriétés du groupe.

Le paramètre optionnel id permet d'attribuer un nom au groupe. Les groupes nommés sont nécessaires pour plusieurs finalités telles que l'animation et les objets réutilisables.

Le paramètre optionnel url permet d'associer un lien externe. Les objets du groupe sont alors cliquables (analogue à l'élément 'a' du HTML).

Le paramètre optionnel cible spécifie le nom de la cible dans laquelle le document doit s'ouvrir quand le lien est activé. Les valeurs attendues sont celles du HTML auxquelles s'ajoutent la chaîne "new" pour l'ouverture dans une nouvelle fenêtre et "none" qui équivaut à ne pas traiter cet attribut.

Note : Les liens externes sont ignorés lorsque le SVG est affiché dans un objet image (variable ou champ) d'un formulaire 4D. La gestion des références externes est effectuée par le moteur de rendu. Dans ces conditions, le résultat peut dépendre de la plate-forme et du logiciel de visualisation.

Exemples

1. Création d'un groupe de lignes toutes de la même couleur :



```
$SVG:=SVG_New
$group:=SVG_New_group ($SVG)
  `Attribuer une couleur aux éléments du groupe
SVG_SET_STROKE_BRUSH ($group; "firebrick")
SVG_New_line ($group; 100; 300; 300; 100; ""; 5)
SVG_New_line ($group; 300; 300; 500; 100; ""; 10)
SVG_New_line ($group; 500; 300; 700; 100; ""; 15)
SVG_New_line ($group; 700; 300; 900; 100; ""; 20)
SVG_New_line ($group; 900; 300; 1100; 100; ""; 25)
```

2. Création d'un texte cliquable :



```
$SVG:=SVG_New
$group:=SVG_New_group ($SVG;"w3Link";"http://www.w3.org";"new")
SVG_New_text ($group;"www.w3.org";10;10;"arial";12;Souligné ;Aligné à gauche ;"blue")
```

SVG_Set_description (objetSVGParent; description) → Ref_SVG

Paramètre	Type		Description
objetSVGParent	Ref_SVG	→	Référence de l'élément parent
description	Chaîne	→	Texte du commentaire
Résultat	Ref_SVG	←	Référence de la description

Description

La commande SVG_Set_description définit un texte pour l'élément SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un élément SVG, une erreur est générée.

Une description est généralement utilisée pour insérer un commentaire ou un texte explicatif dans le code SVG.

Exemple

```
$SVG:=SVG_New
$g:=SVG_group ($SVG)
SVG_Set_title($g;"Ventes de la société par région")
SVG_Set_description($g;"Diagramme en barre des ventes de la société par région.")
...
```

Référence

SVG_Set_title.

SVG_Set_title (objetSVGParent; titre) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG	→ Référence de l'élément parent
titre	Chaîne	→ Texte du titre
Résultat	Ref_SVG	← Référence du titre

Description

La commande SVG_Set_title définit un titre pour l'élément SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un élément SVG, une erreur est générée.

Un titre est une donnée textuelle qui n'est pas incluse dans le rendu de l'image mais est utile pour structurer des documents complexes. Certains moteurs de rendu SVG utilisent le texte de cet élément pour afficher une infobulle au moment du survol de l'objet.

Exemple

```
$SVG:=SVG_New
$rec:=SVG_New_rect ($SVG;20;20;650;650;0;0;"gray";"lemonchiffon")
SVG_Set_title($rec;"Rectangle de fond")
$Symbol:=SVG_Define_symbol ($SVG;"MySymbol";0;0;110;110;"true")
SVG_Set_title($Symbol;"Définition d'un symbole comprenant 2 carrés et 2 cercles ")
...
```

Référence

SVG_Set_description.

8

Texte



SVG_New_text (objetSVGParent; texte{; x{; y{; police{; taille{; style{; alignement{; couleur{; rotation{; interligne{; étirement{}}}}}}}}) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de l'élément parent
texte	Texte →	Texte à insérer
x	Numérique →	Coordonnée sur l'axe x
y	Numérique →	Coordonnée sur l'axe y
police	Chaîne →	Nom de la police
taille	Entier →	Taille des caractères en points
style	Entier →	Style des caractères
alignement	Entier →	Alignement
couleur	Chaîne →	Couleur du texte
rotation	Numérique →	Angle de rotation du texte
interligne	Numérique →	Interlignage en point
étirement	Numérique →	Facteur d'étirement horizontal
Résultat	Ref_SVG ←	Référence de l'objet texte SVG

Description

La commande SVG_New_text insère le texte dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

Les paramètres optionnels x et y permettent de préciser le positionnement sur l'axe x et sur l'axe y du coin supérieur du premier caractère du texte. Ce point est différemment situé selon la valeur de l'alignement : à gauche pour un alignement à gauche, à droite pour un alignement à droite ou au centre lorsque le texte est centré.

Les paramètres optionnels police et taille permettent de spécifier la police et la taille, en points, à utiliser. Lorsque ces paramètres sont omis, le texte est écrit en *Times New Roman 12 pts*.

Le paramètre optionnel `style` précise le style de caractères à utiliser. Vous devez passer dans le paramètre `style` l'une des valeurs suivantes ou une combinaison de ces valeurs (vous pouvez également utiliser les constantes 4D correspondantes dans le thème "Styles de caractères") :

- 0 = Normal
- 1 = Gras
- 2 = Italique
- 4 = Souligné
- 8 = Barré

Le paramètre optionnel `alignement` permet de spécifier le type d'alignement appliqué au texte dessiné. Vous devez passer une des valeurs suivantes (vous pouvez également utiliser les constantes 4D correspondantes dans le thème "Alignement objet") :

- 2 = Aligné à gauche
- 3 = Centré
- 4 = Aligné à droite

Le paramètre optionnel `couleur` contient le nom de la couleur de la police. Pour plus d'informations sur les couleurs, reportez-vous au chapitre "Couleurs et dégradés".

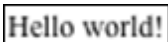
Le paramètre optionnel `rotation` permet de préciser la rotation à appliquer au texte.

Le paramètre optionnel `interligne` permet de préciser la valeur de l'interlignage si le texte comporte plusieurs lignes. Valeur par défaut = 1.

Le paramètre optionnel `étirement` permet d'appliquer horizontalement un facteur d'étirement (valeur >1) ou de condensation (valeur <1) au texte.

Exemples

1. Texte simple utilisant les propriétés de texte par défaut :



```
$SVG:=SVG_New  
$textID:=SVG_New_text($SVG;"Hello world!")
```

2. Texte en bleu, italique, souligné et aligné à droite :



```
$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;Italique +Souligné ;
Aligné à droite ;"blue")
```

3. Texte vertical :



```
$SVG:=SVG_New
$textID:=SVG_New_text ($SVG;$text;-250;0;"";48;-1;-1;"red";-90)
```

4. Texte condensé ou étiré :



```
$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)
```

Référence

SVG_New_textArea, SVG_New_tspan, SVG_New_vertical_text.

SVG_New_textArea (objetSVGParent; texte{; x{; y{; largeur{; hauteur{; police{; taille{; style{; alignement}}}}}})) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG	→ Référence de l'élément parent
texte	Texte	→ Texte à insérer
x	Numérique	→ Coordonnée sur l'axe x
y	Numérique	→ Coordonnée sur l'axe y
largeur	Numérique	→ Largeur de la zone de texte
hauteur	Numérique	→ Hauteur de la zone de texte
police	Chaîne	→ Nom de la police
taille	Entier	→ Taille des caractères en points
style	Entier	→ Style des caractères
alignement	Entier	→ Alignement
Résultat	Ref_SVG	← Référence de l'objet texte SVG

Description

La commande SVG_New_textArea insère une zone de texte dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

L'élément "textArea" est recommandé par la norme SVG tiny 1.2 et implémentée dans 4D v11 SQL à partir de la version 11.3 (cf. <http://www.w3.org/TR/SVGMobile12/text.html#TextAreaElement>). Cet élément implémente une zone de texte qui, au contraire de l'élément "text", gère automatiquement les retours à la ligne lorsque le texte dépasse la largeur demandée.

Les paramètres optionnels x et y permettent de préciser le positionnement sur l'axe x et sur l'axe y du coin supérieur gauche de la zone.

Les paramètres optionnels largeur et hauteur définissent dans l'espace de coordonnées utilisateur la taille de la zone. Si l'un ou l'autre de ces paramètres n'est pas fourni, la zone de texte s'adaptera automatiquement à son contenu.

Le paramètres optionnels police et taille permettent de spécifier la police et la taille, en points, à utiliser. Si ces paramètres sont omis, le texte sera écrit en Times New Roman 12 pts.

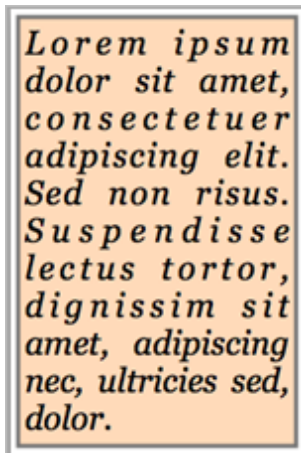
Le paramètre optionnel style précise le style de caractères à utiliser. Vous devez passer dans le paramètre style l'une des valeurs suivantes ou une combinaison de ces valeurs (vous pouvez également utiliser les constantes 4D correspondantes dans le thème "Styles de caractères") :

- 0 = Normal
- 1 = Gras
- 2 = Italique
- 4 = Souligné
- 8 = Barré

Le paramètre optionnel alignement permet de spécifier le type d'alignement appliqué au texte dessiné. Vous devez passer une des valeurs suivantes :

- 1 = Alignement par défaut (gauche)
- 2 = Aligné à gauche
- 3 = Centré
- 4 = Aligné à droite
- 5 = Justifié

Exemple



`$svg:=SVG_New`

``Positionner un rectangle de bordure`

`$rec:=SVG_New_rect ($svg;5;5;210;320;0;0;"#777";"peachpuff";3)`

``Le texte`

`$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;$txt;10;10;200;310;"Georgia";25;Italique ;5)  
`Sauvegarder le document  
SVG_SAVE_AS_TEXT ($svg;"test.svg")
```

Référence

SVG_New_text, SVG_New_vertical_text.

SVG_New_tspan (objetSVGParent; texte{; x{; y{; police{; taille{; style{; alignement{; couleur}}}})) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de l'élément parent
texte	Texte →	Texte à insérer
x	Numérique →	Coordonnée sur l'axe x
y	Numérique →	Coordonnée sur l'axe y
police	Chaîne →	Nom de la police
taille	Entier →	Taille des caractères en points
style	Entier →	Style des caractères
alignement	Entier →	Alignement
couleur	Chaîne →	Couleur du texte
Résultat	Ref_SVG ←	Référence de l'objet texte SVG

Description

La commande SVG_New_tspan crée un nouvel élément dans l'élément 'text' ou 'tspan' désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas une référence d'un élément 'text' ou 'tspan', une erreur est générée.

Les différents paramètres optionnels sont décrits avec la commande SVG_New_text. Si certains paramètres optionnels sont omis, leurs valeurs sont héritées du ou des élément(s) parent(s).

Exemples

1. Dans un texte il est possible de créer des paragraphes qui héritent des propriétés du parent.



```
$SVG:=SVG_New
  `Créer un nouveau texte en Arial, bleu, Aligné à gauche
$textID:=SVG_New_text ($SVG;"";0;0;"arial";-1;-1;Aligné à gauche;"blue")
  `Paragraphes imbriqués avec indentation et changement de taille et de style
$textID:=SVG_New_tspan ($textID;"TITRE 1"; 10; 10;""; 24; Gras+Souligné )
$textID:=SVG_New_tspan ($textID;"Titre 2"; 20; 42;""; 12; Gras)
$textID:=SVG_New_tspan ($textID;"Titre 3"; 30; 60;""; 10; Gras+Italique)
$textID:=SVG_New_tspan ($textID;"Titre 4"; 40; 78;""; 8; Italique)
```

2. Changer une propriété tout en restant dans un élément "text", ici la taille du texte :

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)
```

Référence

SVG_New_text.

SVG_New_vertical_text (objetSVGParent; texte{; x{; y{; police{; taille{; style{; alignement{; couleur{; rotation}}}}}})) → Ref_SVG

Paramètre	Type	Description
objetSVGParent	Ref_SVG →	Référence de l'élément parent
texte	Texte →	Texte à insérer
x	Numérique →	Coordonnée sur l'axe x
y	Numérique →	Coordonnée sur l'axe y
police	Chaîne →	Nom de la police
taille	Entier →	Taille des caractères en points
style	Entier →	Style des caractères
alignement	Entier →	Alignement
couleur	Chaîne →	Couleur du texte
rotation	Numérique →	Angle de rotation du texte
Résultat	Ref_SVG ←	Référence de l'objet texte VG

Description

La commande SVG_New_vertical_text insère le texte verticalement dans le conteneur SVG désigné par objetSVGParent et retourne sa référence. Si objetSVGParent n'est pas un document SVG, une erreur est générée.

Les paramètres optionnels x et y permettent de préciser le positionnement sur l'axe x et sur l'axe y du coin inférieur du premier caractère du texte.

Les paramètres optionnels police et taille permettent de spécifier la police et la taille, en points, à utiliser. Lorsque ces paramètres sont omis, le texte est écrit en *Times New Roman 12 pts*.

Le paramètre optionnel style précise le style de caractères à utiliser. Vous devez passer dans le paramètre style l'une des valeurs suivantes ou une combinaison de ces valeurs (vous pouvez également utiliser les constantes 4D correspondantes dans le thème "Styles de caractères") :

- 0 = Normal
- 1 = Gras
- 2 = Italique
- 4 = Souligné
- 8 = Barré

Le paramètre optionnel alignement permet de spécifier le type d'alignement appliqué au texte dessiné. Vous devez passer une des valeurs suivantes (vous pouvez également utiliser les constantes 4D correspondantes dans le thème "Alignement objet") :

2 = Aligné à gauche

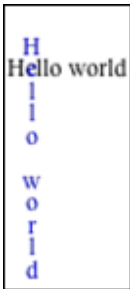
3 = Centré

4 = Aligné à droite

Le paramètre optionnel couleur contient le nom de la couleur de la police. Pour plus d'informations sur les couleurs, reportez-vous au chapitre "Couleurs et dégradés".

Le paramètre optionnel rotation permet de préciser la rotation à appliquer au texte.

Exemple



```
$SVG:=SVG_New
```

```
$textID:=SVG_New_text ($SVG;"Hello world";10;12)
```

```
$textID:=SVG_New_vertical_text ($SVG;"Hello world";22;3;"";-1;-1;Centré;"blue")
```

Référence

SVG_New_text, SVG_New_tspan.

SVG_SET_FONT_COLOR (objetSVG; couleur)

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
couleur	Chaîne	→ Couleur du texte

Description

La commande SVG_SET_FONT_COLOR permet de spécifier la couleur de la police pour l'objet SVG de référence objetSVG. Si objetSVG ne référence pas un élément valide, une erreur est générée.

Le paramètre couleur contient le nom de la couleur à utiliser. Pour plus d'informations sur les couleurs, reportez-vous à la section "Couleurs SVG".

Référence

Couleurs SVG.

SVG_SET_FONT_FAMILY (objetSVG; police)

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
police	Chaîne	→ Nom de police

Description

La commande SVG_SET_FONT_FAMILY permet de spécifier la police pour l'objet SVG de référence objetSVG.

Si objetSVG ne référence pas un élément valide, une erreur est générée.

Le paramètre police contient le nom de la police à utiliser.

SVG_SET_FONT_SIZE (objetSVG; taille)

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
taille	Entier	→ Taille des caractères en points

Description

La commande SVG_SET_FONT_SIZE permet de spécifier la taille de la police pour l'objet SVG de référence objetSVG. Si objetSVG ne référence pas un élément valide, une erreur est générée.

Le paramètre taille contient la taille de la police exprimée en points.

Référence

SVG_SET_FONT_STYLE.

SVG_SET_FONT_STYLE (objetSVG; style)

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
style	Entier	→ Style de caractères

Description

La commande SVG_SET_FONT_STYLE permet de spécifier le style du texte pour l'objet SVG de référence objetSVG. Si objetSVG ne référence pas un élément valide, une erreur est générée.

Vous devez passer dans le paramètre style l'une des valeurs suivantes ou la somme de plusieurs de ces valeurs :

- 0 = Normal
- 1 = Gras
- 2 = Italique
- 4 = Souligné
- 8 = Barré

Référence

SVG_SET_FONT_SIZE.

SVG_SET_TEXT_ANCHOR (objetSVG; alignement)

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence d'un élément SVG
alignement	Entier	→ Alignement

Description

La commande SVG_SET_TEXT_ANCHOR permet de modifier l'alignement de l'objet SVG de référence objetSVG. Si objetSVG ne référence pas un élément valide, une erreur est générée.

Vous devez passer dans le paramètre alignement l'une des valeurs suivantes :

- 1 = Alignement par défaut (gauche)
- 2 = Aligné à gauche
- 3 = Centré
- 4 = Aligné à droite
- 5 = Justifié (pour un objet textArea seulement)

Référence

SVG_New_textArea.

9

Utilitaires

SVG_Count_elements (objetSVG) → Numérique

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence SVG
Résultat	Numérique ←	Nombre d'objets

Description

La commande `SVG_Count_elements` retourne le nombre d'objets graphiques contenus dans l'objetSVG passé en paramètre. Les groupes comptent pour un objet. Pour connaître le nombre d'objets graphiques d'un groupe, passez sa référence à la commande. Si la référence SVG n'est pas valide, une erreur est générée.

Référence

`SVG_ELEMENTS_TO_ARRAYS`, `SVG_New_group`.

SVG_ELEMENTS_TO_ARRAYS (objetSVG; pointTabRefs{; pointTabTypes{; pointTabNoms{}})

Paramètre	Type	Description
objetSVG	Ref_SVG	→ Référence SVG
pointTabRefs	Pointeur	→ Tableau chaîne des références d'objets
pointTabTypes	Pointeur	→ Tableau chaîne des types d'objets
pointTabNoms	Pointeur	→ Tableau chaîne des id d'objets

Description

La commande SVG_ELEMENTS_TO_ARRAYS remplit le tableau pointé par pointTabRefs avec les références des objets graphiques de premier niveau pour la référence SVG passée dans objetSVG.

Si le pointeur optionnel pointTabTypes est passé, le tableau sera renseigné avec le type des objets.

Si le pointeur optionnel pointTabNoms est passé, le tableau sera renseigné avec l'id des objets.

Les groupes comptent pour un objet. Pour retourner ces informations pour les objets graphiques d'un groupe, passez sa référence à la commande.

Si objetSVG n'est pas valide ou si cet attribut n'existe pas, une erreur est générée.

Référence

SVG_Count_elements, SVG_New_group.

SVG_Estimate_weight (objetSVG) → Numérique

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'un document SVG
Résultat	Numérique ←	Poids en octets du document SVG

Description

La commande SVG_Estimate_weight retourne la taille en octets de l'arbre SVG dont la référence est passée dans le paramètre objetSVG. Si objetSVG n'est pas une référence valide, une erreur est générée.

Référence

SVG_Count_elements.

SVG_Find_ID (nom) → Ref_SVG

Paramètre	Type		Description
nom	Chaîne	→	id de l'élément SVG
Résultat	Ref_SVG	←	Référence de l'élément

Description

La commande SVG_Find_ID retourne la référence de l'élément dont l'ID est passé dans le paramètre nom. Si l'élément n'est pas trouvé, une erreur est générée.

Référence

SVG_Get_ID, SVG_SET_ID.

SVG_Get_options → Entier long

Paramètre	Type	Description
-----------	------	-------------

Cette commande ne requiert pas de paramètre

Résultat	Entier long ← Options
----------	-----------------------

Description

La commande SVG_Get_options retourne un entier long représentant un tableau de 32 bits, chaque bit pouvant décrire une option du composant. Vous pouvez utiliser les Opérateurs sur les bits de 4D pour tester (??) l'état d'une option, l'activer (?+) ou la désactiver (?-).

Les options actuellement disponibles sont :

Bit	Option	Défaut
1	Attribuer un ID automatiquement lors de la création d'un élément	0 (inactivé)
2	Fermer automatiquement les objets qui peuvent l'être	0 (inactivé)
3	Créer les objets avec un fond	1 (activé)
4	Coordonnées absolues pour les tracés	1 (activé)
5	Créer un code plus lisible	0 (inactivé)
6	Émettre un bip lorsqu'une erreur survient	1 (activé)
7	Ne pas afficher les erreurs 4D	0 (inactivé)
8	Images transparentes	1 (activé)

- *Attribuer un ID automatiquement lors de la création d'un élément*

Si cette option est activée, lorsque le composant crée un nouvel élément, il ajoute et renseigne systématiquement un attribut 'id' pour l'objet créé s'il n'est pas précisé.

- *Fermer automatiquement les objets*

Si cette option est activée, les objets créés avec les commandes SVG_New_arc et SVG_New_polyline_by_arrays seront fermés automatiquement.

- *Créer les objets avec un fond*

Si cette option est activée, les objets fermés seront créés avec une couleur de fond, sinon le fond est transparent.

- *Coordonnées absolues pour les tracés*

Lors du dessin de tracés avec les commandes `SVG_PATH_MOVE_TO`, `SVG_PATH_LINE_TO`, `SVG_PATH_CURVE` et `SVG_PATH_ARC`, les coordonnées passées seront interprétées comme absolues si cette option est activée ou comme relatives sinon.

- *Créer un code plus lisible*

Cette option permet de créer un code indenté et aéré mais par conséquent plus lourd, son activation est surtout intéressante durant la phase de débogage.

- *Émettre un beep lorsqu'une erreur survient*

Lorsqu'une erreur se produit et qu'aucune méthode d'erreur de la base hôte n'a été installée avec la commande `SVG_Set_error_handler`, un bip est émis si cette option est activée.

- *Ne pas afficher les erreurs 4D*

Cette option activée par défaut bloque l'affichage des erreurs 4D en installant une méthode de gestion d'erreur propre au composant 4D SVG. Vous pouvez préférer ne pas utiliser cette gestion interne et laisser 4D afficher ces messages. Cela peut être utile en cours de débogage par exemple.

- *Images transparentes*

Par défaut, les images SVG créées avec la commande `SVG_New` sont transparentes. Si vous désactivez cette option, les images seront sur fond blanc.

Exemple

Reportez-vous à l'exemple de la commande `SVG_SET_OPTIONS`.

Référence

`SVG_SET_OPTIONS`.

SVG_Get_version → Chaîne

Paramètre	Type	Description
-----------	------	-------------

Cette commande ne requiert pas de paramètre

Résultat	Chaîne	← Numéro de version
----------	--------	---------------------

Description

La commande SVG_Get_version retourne sous forme alphanumérique le numéro de version du composant. La chaîne retournée comporte toujours le numéro de version et de sous-version (par exemple "11.0" pour la version 11 ou "11.3" pour la 3e mise à jour de la version 11). Lorsqu'il s'agit d'une version beta, le numéro de distribution est précisé, préfixé de la lettre "B" (par exemple "11.3B1" pour la beta 1 de la version 11.3).

SVG_Is_reference_valid (objetSVG) → Booléen

Paramètre	Type		Description
objetSVG	Ref_SVG	→	Référence d'un élément SVG
Résultat	Booléen	←	Vrai si la référence appartient à un élément SVG

Description

La commande `SVG_Is_reference_valid` retourne Vrai si la référence passée dans le paramètre `objetSVG` est celle d'un élément d'un arbre SVG. Si l'élément n'appartient pas à un arbre SVG, la commande retourne Faux. Si `objetSVG` n'est pas une référence valide, une erreur est générée.

Référence

`SVG_References_array`.

SVG_Read_element_type (objetSVG) → Type

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence d'élément SVG
Résultat	Type ←	Type de l'élément

Description

La commande SVG_Read_element_type retourne le type de l'élément dont la référence est passée dans le paramètre objetSVG.

Si objetSVG n'est pas une référence valide ou si cet attribut n'existe pas, une erreur est générée.

SVG_Read_last_error → Entier long

Paramètre	Type	Description
-----------	------	-------------

Cette commande ne requiert pas de paramètre

Résultat	Entier long ←	Numéro de la dernière erreur
----------	---------------	------------------------------

Description

La commande SVG_Read_last_error retourne le numéro de la dernière erreur survenue lors de l'exécution d'une commande du composant 4D SVG et réinitialise cette erreur.

Le numéro d'erreur retourné peut être spécifique à une commande du composant ou une erreur générée par 4D. Les erreurs générées par le composant sont :

8850	Nombre de paramètres insuffisant
8851	Type de paramètre non valide
8852	Référence non valide
8853	Valeur incorrecte pour un attribut
8854	L'élément n'accepte pas cette commande
8855	Nom d'objet (symbole, marqueur, filtre...) invalide (ID non trouvé dans le document)
8856	Le fichier DTD n'a pas été trouvé.
8857	Valeur incorrecte pour un paramètre
8858	Erreur inconnue

Exemples

1. Soit la méthode "gest_SVG_error" décrite dans l'exemple de la commande SVG_Set_error_handler :

```

`Installation de la méthode de gestion d'erreur
$ Error_Method_Txt:=SVG_Set_error_handler ("gest_SVG_error")
`Désormais c'est la méthode gest_SVG_error qui sera exécutée en cas d'erreur

`Création d'un nouveau document SVG
$SVG:=SVG_New (1200; 900; "Test Composant SVG Component"; ""; Vrai)
SVG_SET_VIEWBOX ($SVG; 0; 0; 1500; 1000)

```


Si(SVG_Read_last_error=0)

...
Sinon

 `La méthode `gest_SVG_error` a été appelée et a reçu le numéro d'erreur
Fin de si

 `Désinstallation de la méthode de gestion d'erreurs
SVG_Set_error_handler

2. Soit la méthode `gest_SVG_error` suivante:

C_ENTIER LONG (\$1)

C_TEXTE (\$2)

 `Garder l'erreur et le contexte
NumErreur:=\$1
NomCommande:=\$2

 `Mettre la variable système OK à 0
OK := 0

 ` Installation de la méthode de gestion d'erreur
\$ Error_Method_Txt:=**SVG_Set_error_handler** ("SVG_error_mgmt")

 ` Création d'un nouveau document SVG
\$SVG:=**SVG_New** (1200; 900; "Test composant SVG"; ""; **Vrai**)
SVG_SET_VIEWBOX (\$SVG; 0; 0; 1500; 1000)
Si(OK = 1)

...
Sinon

ALERTE("Erreur N°." + **Chaîne**(NumErreur) + " pendant l'exécution de la commande \" +
 + NomCommande+ \"\")
Fin de si

 ` Désinstallation de la méthode de gestion d'erreurs
SVG_Set_error_handler

Référence

SVG_Set_error_handler.

SVG_References_array (pointeurTabRef)

Paramètre	Type	Description
pointeurTabRef	Pointeur →	Tableau chaîne des références de documents

Description

La commande SVG_References_array retourne dans le tableau pointé par pointeurTabRef la liste des références de documents SVG courantes. Cette commande est utile en cours de débogage.

A chaque fois qu'un document SVG est créé avec les commandes SVG_New, SVG_Copy ou SVG_Open_file, le composant ajoute la référence retournée par la commande dans un tableau interne. Quand le document SVG est libéré avec la commande SVG_CLEAR, le composant supprime la référence du tableau.

Référence

SVG_Is_reference_valid.



SVG_Set_error_handler {(méthode)} → Chaîne

Paramètre	Type	Description
méthode	Chaîne →	Nom de la méthode de la base hôte à installer
Résultat	Chaîne ←	Nom de la méthode précédemment installée

Description

La commande SVG_Set_error_handler permet d'installer la méthode de la base hôte en tant que méthode appelée en cas d'erreur et retourne le nom de la précédente méthode d'appel sur erreur.

Les commandes du composant 4D SVG effectuent un ensemble de vérifications élémentaires lorsqu'elles sont appelées : nombre de paramètres minimum, validité des références, de l'élément sur lequel une commande est appliquée... Le composant gère donc les erreurs de façon structurée et permet à la base hôte de récupérer les éventuelles erreurs.

En l'absence de modification du fonctionnement par défaut, si une erreur survient, un bip est émis et la commande est interrompue.

La base hôte peut récupérer dans une de ses méthodes le numéro de l'erreur et le nom de la commande à l'origine de l'interruption. Il suffit pour cela de l'installer via la commande SVG_Set_error_handler. Cette méthode recevra en premier paramètre le numéro de l'erreur et en second paramètre le nom de la commande. Elle sera appelée lorsqu'une erreur se produira et dans ce cas aucun bip ne sera généré par le composant.

Si méthode est omis ou si vous passez une chaîne vide dans ce paramètre, la méthode est désinstallée et le comportement par défaut est réactivé.

Note : La méthode de la base hôte qui sera appelée par le composant 4D SVG doit avoir la propriété "Partagée entre composants et base hôte".

Exemple

Installation de la méthode *gest_SVG_error* (méthode de la base hôte) comme méthode d'appel sur erreur :

```
$erreur;=SVG_Set_error_handler("gest_SVG_error")
```

Code de la méthode :

```
` Méthode gest_SVG_error  
ALERTE ("Erreur No." + Chaîne($1) +" pendant l'exécution de la commande \""+$2+"\"")
```

Référence

SVG_Read_last_error.

SVG_SET_OPTIONS {(options)}

Paramètre	Type	Description
options	Entier long →	Options du composant 4D SVG

Description

La commande SVG_SET_OPTIONS permet de fixer les options du composant 4D SVG avec l'entier long options. Pour plus d'informations sur le contenu de options, reportez-vous à la description de la commande SVG_Get_options.

Comme toutes les options sont fixées simultanément, cette commande doit être précédée d'un appel à la commande SVG_Get_options puis de l'utilisation des Opérateurs sur les bits de 4D.

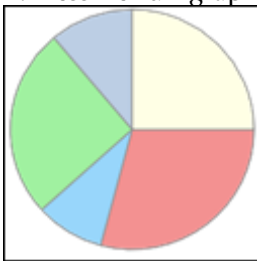
Si le paramètre option est omis, toutes les options sont réinitialisées à leur valeur par défaut (cf. commande SVG_Get_options).

Exemples

- Créer un code plus lisible :


```
$Options := SVG_Get_options
$Options := $Options ?+5 `activer l'option
SVG_SET_OPTIONS ($Options)
```

- Dessiner un graphique en camembert :



`$svg:=SVG_New`

`Activer la fermeture automatique des objets
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)
```

Référence

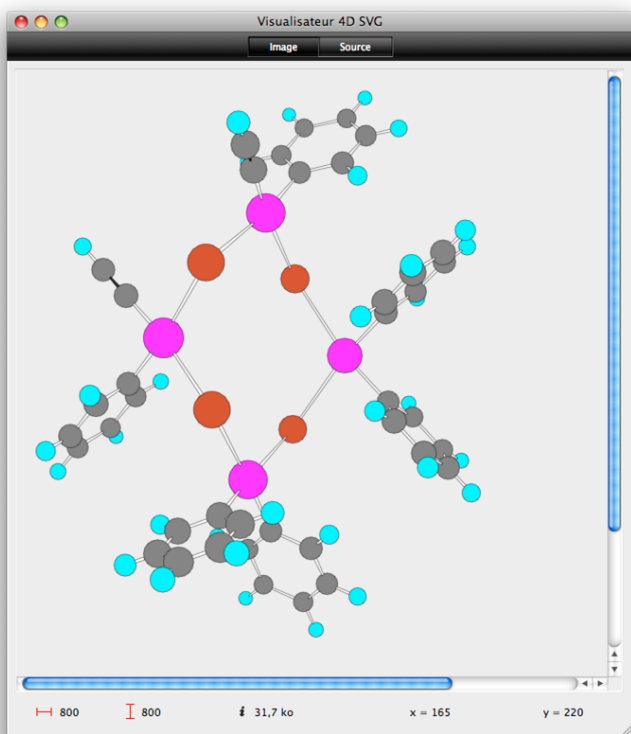
SVG_Get_options.

SVGTool_SHOW_IN_VIEWER (objetSVG)

Paramètre	Type	Description
objetSVG	Ref_SVG →	Référence de l'image à afficher

Description

La commande SVGTool_SHOW_IN_VIEWER affiche l'image SVG définie par objetSVG dans une fenêtre du Visualisateur SVG. Cet outil est fourni avec le composant SVG :



Pour plus d'informations sur le Visualisateur SVG, reportez-vous à la section Outils de développement.

Référence






































Outils de développement.

10

Annexes

Voici la liste des couleurs reconnues par le SVG. Pour plus d'informations, reportez-vous à la section Couleurs SVG.

	aliceblue	rgb(240, 248, 255)		darkslategrey	rgb(47, 79, 79)
	antiquewhite	rgb(250, 235, 215)		darkturquoise	rgb(0, 206, 209)
	aqua	rgb(0, 255, 255)		darkviolet	rgb(148, 0, 211)
	aquamarine	rgb(127, 255, 212)		deeppink	rgb(255, 20, 147)
	azure	rgb(240, 255, 255)		deepskyblue	rgb(0, 191, 255)
	beige	rgb(245, 245, 220)		dimgray	rgb(105, 105, 105)
	bisque	rgb(255, 228, 196)		dimgrey	rgb(105, 105, 105)
	black	rgb(0, 0, 0)		dodgerblue	rgb(30, 144, 255)
	blanchedalmond	rgb(255, 235, 205)		firebrick	rgb(178, 34, 34)
	blue	rgb(0, 0, 255)		floralwhite	rgb(255, 250, 240)
	blueviolet	rgb(138, 43, 226)		forestgreen	rgb(34, 139, 34)
	brown	rgb(165, 42, 42)		fuchsia	rgb(255, 0, 255)
	burlywood	rgb(222, 184, 135)		gainsboro	rgb(220, 220, 220)
	cadetblue	rgb(95, 158, 160)		ghostwhite	rgb(248, 248, 255)
	chartreuse	rgb(127, 255, 0)		gold	rgb(255, 215, 0)
	chocolate	rgb(170, 105, 30)		goldenrod	rgb(218, 165, 32)
	coral	rgb(255, 127, 80)		gray	rgb(128, 128, 128)
	cornflowerblue	rgb(100, 149, 237)		grey	rgb(128, 128, 128)
	cornsilk	rgb(255, 248, 220)		green	rgb(0, 128, 0)
	crimson	rgb(220, 20, 60)		greenyellow	rgb(173, 255, 47)
	cyan	rgb(0, 255, 255)		honeydew	rgb(240, 255, 240)
	darkblue	rgb(0, 0, 139)		hotpink	rgb(255, 105, 180)
	darkcyan	rgb(0, 139, 139)		indianred	rgb(205, 92, 92)
	darkgoldenrod	rgb(184, 134, 11)		indigo	rgb(75, 0, 130)
	darkgray	rgb(169, 169, 169)		ivory	rgb(255, 255, 240)
	darkgreen	rgb(0, 100, 0)		khaki	rgb(240, 230, 140)
	darkgrey	rgb(169, 169, 169)		lavender	rgb(230, 230, 250)
	darkkhaki	rgb(189, 183, 107)		lavenderblush	rgb(255, 240, 245)
	darkmagenta	rgb(139, 0, 139)		lawngreen	rgb(124, 252, 0)
	darkolivegreen	rgb(85, 107, 47)		lemonchiffon	rgb(255, 250, 205)
	darkorange	rgb(255, 140, 0)		lightblue	rgb(173, 216, 230)
	darkorchid	rgb(153, 50, 204)		lightcoral	rgb(240, 128, 128)
	darkred	rgb(139, 0, 0)		lightcyan	rgb(224, 255, 255)
	darksalmon	rgb(233, 150, 122)		lightgoldenrodyellow	rgb(250, 250, 210)
	darkseagreen	rgb(143, 188, 143)		lightgray	rgb(211, 211, 211)
	darkslateblue	rgb(72, 61, 139)		lightgreen	rgb(144, 238, 144)
	darkslategrey	rgb(47, 79, 79)		lightgrey	rgb(211, 211, 211)

	lightpink	rgb (255, 182, 193)
	lightsalmon	rgb (255, 160, 122)
	lightseagreen	rgb (32, 178, 170)
	lightskyblue	rgb (135, 206, 250)
	lightslategray	rgb (119, 136, 153)
	lightslategrey	rgb (119, 136, 153)
	lightsteelblue	rgb (176, 196, 222)
	lightyellow	rgb (255, 255, 224)
	lime	rgb (0, 255, 0)
	limegreen	rgb (50, 205, 50)
	linen	rgb (250, 240, 230)
	magenta	rgb (255, 0, 255)
	maroon	rgb (128, 0, 0)
	mediumaquamarine	rgb (102, 205, 170)
	mediumblue	rgb (0, 0, 205)
	mediumporchid	rgb (186, 85, 211)
	mediumpurple	rgb (147, 112, 219)
	mediumseagreen	rgb (60, 179, 113)
	mediumslateblue	rgb (113, 104, 238)
	mediumspringgreen	rgb (0, 250, 154)
	mediumturquoise	rgb (72, 209, 204)
	mediumvioletred	rgb (199, 21, 133)
	midnightblue	rgb (25, 25, 112)
	mintcream	rgb (245, 255, 250)
	mistyrose	rgb (255, 228, 225)
	moccasin	rgb (255, 228, 181)
	navajowhite	rgb (255, 222, 173)
	navy	rgb (0, 0, 128)
	oldlace	rgb (253, 245, 230)
	olive	rgb (128, 128, 0)
	olivedrab	rgb (107, 142, 35)
	orange	rgb (255, 165, 0)
	orangered	rgb (255, 69, 0)
	orchid	rgb (218, 112, 214)
	palegoldenrod	rgb (238, 232, 170)
	palegreen	rgb (152, 251, 152)
	paleturquoise	rgb (175, 238, 238)

	palevioletred	rgb (219, 112, 147)
	papayawhip	rgb (255, 239, 213)
	peachpuff	rgb (255, 218, 185)
	peru	rgb (205, 133, 63)
	pink	rgb (255, 192, 203)
	plum	rgb (224, 160, 221)
	powderblue	rgb (176, 224, 230)
	purple	rgb (128, 0, 128)
	red	rgb (255, 0, 0)
	rosybrown	rgb (188, 143, 143)
	royalblue	rgb (65, 105, 225)
	saddlebrown	rgb (139, 69, 19)
	salmon	rgb (250, 128, 114)
	sandybrown	rgb (244, 164, 96)
	seagreen	rgb (46, 139, 87)
	seashell	rgb (255, 245, 238)
	sienna	rgb (160, 82, 45)
	silver	rgb (192, 192, 192)
	skyblue	rgb (135, 206, 235)
	slateblue	rgb (106, 90, 205)
	slategray	rgb (112, 128, 144)
	slategrey	rgb (112, 128, 144)
	snow	rgb (255, 250, 250)
	springgreen	rgb (0, 255, 127)
	steelblue	rgb (70, 130, 180)
	tan	rgb (210, 180, 140)
	teal	rgb (0, 128, 128)
	thistle	rgb (216, 191, 216)
	tomato	rgb (255, 99, 71)
	turquoise	rgb (64, 224, 208)
	violet	rgb (238, 130, 238)
	wheat	rgb (245, 222, 179)
	white	rgb (255, 255, 255)
	whitesmoke	rgb (245, 245, 245)
	yellow	rgb (255, 255, 0)
	yellowgreen	rgb (154, 205, 50)



Voici une liste de liens utiles relatifs à la norme SVG :

Présentation

<http://www.w3.org/Graphics/SVG/>

Spécification

<http://www.w3.org/TR/SVG12/> (Status : Working Draft *13 April 2005*)

<http://www.yoyodesign.org/doc/w3c/svg1/> (Traduction française (Version 1.0))

Tutoriaux

<http://www.w3.org/Consortium/Offices/Presentations/SVG/1.svg>

Communauté

<http://svg.org/>

<http://svgfr.org>

<http://www.openclipart.org/>

<http://www.gosvg.net/>

Index des commandes

A

SVG_ADD_POINT..... 59

C

SVG_CLEAR.....101
SVG_Color_grey.....52
SVG_Color_RGB_from_long.....53
SVG_Copy..... 102
SVG_Count_elements..... 159

D

SVG_Define_filter.....121
SVG_Define_linear_gradient.....123
SVG_Define_marker.....126
SVG_Define_radial_gradient.....128
SVG_Define_shadow.....130
SVG_Define_solidColor.....132
SVG_Define_symbol.....133

E

SVG_ELEMENTS_TO_ARRAYS.....160
SVG_Estimate_weight.....161
SVG_Export_to_picture.....103
SVG_Export_to_XML.....104

F

SVG_Filter_Blend..... 116
SVG_Filter_Blur..... 117

SVG_Filter_Offset.....	118
SVG_Find_ID.....	162

G

SVG_GET_ATTRIBUTES.....	21
SVG_GET_DEFAULT_BRUSHES.....	54
SVG_Get_ID.....	22
SVG_Get_options.....	163
SVG_Get_version.....	165

I

SVG_Is_reference_valid.....	166
-----------------------------	-----

N

SVG_New.....	105
SVG_New_arc.....	60
SVG_New_circle.....	62
SVG_New_ellipse.....	64
SVG_New_ellipse_bounded.....	66
SVG_New_embedded_image.....	68
SVG_New_group.....	134
SVG_New_image.....	70
SVG_New_line.....	72
SVG_New_path.....	73
SVG_New_polygon.....	77
SVG_New_polygon_by_arrays.....	78
SVG_New_polyline.....	80
SVG_New_polyline_by_arrays.....	82
SVG_New_rect.....	85
SVG_New_regular_polygon.....	87
SVG_New_text.....	141
SVG_New_textArea.....	144

SVG_New_tspan.....	147
SVG_New_vertical_text.....	149

O

SVGTool_SHOW_IN_VIEWER.....	175
SVG_Open_file.....	107
SVG_Open_picture.....	108

P

SVG_PATH_ARC.....	89
SVG_PATH_CLOSE.....	91
SVG_PATH_CURVE.....	92
SVG_PATH_LINE_TO.....	93
SVG_PATH_MOVE_TO.....	94
SVG_PATH_QCURVE.....	95

R

SVG_Read_element_type.....	167
SVG_Read_last_error.....	168
SVG_References_array.....	170

S

SVG_SAVE_AS_PICTURE.....	109
SVG_SAVE_AS_TEXT.....	110
SVG_SET_ATTRIBUTES.....	23
SVG_SET_ATTRIBUTES_BY_ARRAYS.....	24
SVG_SET_DEFAULT_BRUSHES.....	55
SVG_Set_description.....	136
SVG_SET_DIMENSIONS.....	25
SVG_Set_error_handler.....	171

SVG_SET_FILL_BRUSH.....	26
SVG_SET_FILTER.....	27
SVG_SET_FONT_COLOR.....	151
SVG_SET_FONT_FAMILY.....	152
SVG_SET_FONT_SIZE.....	153
SVG_SET_FONT_STYLE.....	154
SVG_SET_ID.....	28
SVG_SET_MARKER.....	29
SVG_SET_OPACITY.....	32
SVG_SET_OPTIONS.....	173
SVG_SET_ROUNDING_RECT.....	33
SVG_SET_STROKE_BRUSH.....	34
SVG_SET_STROKE_LINECAP.....	35
SVG_SET_STROKE_LINEJOIN.....	36
SVG_SET_STROKE_WIDTH.....	37
SVG_SET_TEXT_ANCHOR.....	155
SVG_Set_title.....	137
SVG_SET_TRANSFORM_FLIP.....	38
SVG_SET_TRANSFORM_MATRIX.....	40
SVG_SET_TRANSFORM_ROTATE.....	41
SVG_SET_TRANSFORM_SCALE.....	42
SVG_SET_TRANSFORM_SKEW.....	43
SVG_SET_TRANSFORM_TRANSLATE.....	44
SVG_SET_VIEWBOX.....	45
SVG_SET_VIEWPORT_FILL.....	46
SVG_SET_VISIBILITY.....	47
SVG_SET_XY.....	48

U

SVG_Use.....	96
--------------	----

V

SVG_Validate_file.....	111
------------------------	-----