

Préface

"Avec SVG, le Web graphique passe de l'illustration à l'information graphique. Scalable Vector Graphics est la clé pour créer des contenus visuels riches et utiles pour le Web. Enfin, les créateurs ont le format graphique ouvert dont ils ont besoin pour faire des graphiques professionnels, non seulement objets visuels sur le Web, mais aussi contenus indexables et réutilisables."

- Tim Berners-Lee, directeur du W3C et père du World Wide Web.

"Les créateurs atteignent de plus larges audiences avec une variété croissante de supports pour le Web, du téléphone à l'ordinateur. Ils ont besoin de graphiques adaptables à ces différents supports. Mais avant tout, ils veulent gérer leurs graphiques de la même manière que leurs textes et leurs données, qui aujourd'hui sont au format XML. SVG est spécifiquement fait pour le leur permettre."

- Chris Lilley, leader de l'activité graphique au W3C.

Introduction

Scalable Vector Graphics (SVG) est un nouveau format graphique révolutionnaire qui libère le potentiel de l'information graphique sur le Web. SVG est un langage de description des graphiques bi-dimensionnels en **XML**. SVG inclut un rendement de qualité, des possibilités de zoom et de panoramique, les filtres sur les objets, le remplissage des formes avec des textures et des gradients, les masques, les animations et l'interactivité et bien d'autres choses encore! SVG est conçu pour incorporer les autres spécifications du **W3C** comme **DOM**, **CSS**, **XSLT** , **SMIL**.

SVG donne aux développeurs, graphistes et éditeurs la possibilité de créer des documents de faible taille, interactifs, de qualité pour une utilisation sur le Web. SVG permet le positionnement au pixel près d'objets graphiques tels que des formes, du texte aussi bien que d'images bitmap au format PNG ou JPEG et supporte plus de 16 millions de couleurs.

SVG est révolutionnaire dans le sens où il est pleinement adaptable au support en temps réel par l'usage de styles et de scripts. Les utilisateurs peuvent adapter les couleurs, les polices le contenu et les objets graphiques du côté client.

Vous pouvez libérer le pouvoir de SVG en utilisant des scripts. SVG utilise le modèle de document du W3C **DOM** (Document Object Model) pour atteindre les objets. SVG a son propre modèle compatible qui étend le DOM, le **SVG DOM**. En utilisant ces deux modèles, les scripts rendent SVG très ouvert et puissant.

Ce livre traite également des scripts. Nous partirons d'**EcmaScript** et du DOM, puis nous montrerons comment accéder aux objets du document SVG et les manipuler par la programmation.

Le livre aborde également la création de graphiques interactifs et montre en détail comment gérer les événements. ***Apprivoiser SVG*** couvre les notions mathématiques indispensables (vecteurs, matrices) pour la manipulation des transformations applicables aux objets.

Objectifs de ce livre

L'approche de "Apprivoiser SVG"

L'objectif de ce livre est de permettre à des lecteurs novices en SVG d'atteindre une maîtrise correcte pour la création et la programmation en SVG à travers une suite d'exemples et de scénarios concrets. Nous voulons aussi présenter des notions telles qu'utiliser un style, les gradients et les filtres, l'animation, l'interactivité par le script, la publication, la création d'applications SVG.

Tout au long de ce livre la présentation de nouveaux concepts est accompagnée d'exemples, de code et d'illustrations explicatives. A la fin de ce livre le lecteur sera paré pour surfer sur la vague SVG et intégrer SVG aux technologies XML.

L'intégration des aspects graphiques et de la programmation n'est pas facile mais nous espérons avoir réussi ce challenge.

Le côté référence de ce livre sera utile aux développeurs et graphistes qui veulent des bases solides pour utiliser la puissance de SVG. Nous avons essayé de couvrir avec de nombreux exemples tous les recoins de SVG.

Le propos d' ***Apprivoiser SVG*** est d'informer et de donner envie d'utiliser SVG en couvrant les aspects graphiques et de programmation par le traitement en profondeur des exemples. Ce livre est destiné à celles et ceux qui, novices dans le domaine de SVG, veulent découvrir toute la puissance de celui-ci. Ce livre traite d'exemples concrets qui peuvent servir de référence. La communauté des programmeurs, développeurs internet, éditeurs et graphistes trouvera dans ce livre une aide appréciable dans le champ de ses compétences.

Public visé

Comme SVG est multiple, le livre s'adresse à des publics variés :

- Graphistes et développeurs Web
- Professionnels de la gestion de données
- Cartographes
- Dessinateurs techniques
- La communauté XML
- La communauté Flash

Ce livre voudrait inciter les créateurs de divers contenus à utiliser SVG. Ce livre donne de solides bases pour démarrer avec SVG. Les notions fondamentales sont couvertes de manière aussi précise et concise que possible. D'un autre côté, beaucoup ont approché ce nouveau format/langage et nous avons ajouté "SVG Concepts", challenges, et exercices à la fin de chaque chapitre pour introduire des sujets plus complexes à leur destination.

Conventions et terminologie

Nous utilisons plusieurs styles de texte pour différencier les informations données. Voici les styles utilisés et leur signification :

Alignement

L'alignement aide à la lisibilité du code. Les conventions suivantes sont appliquées :

- objets fils
- éléments occupant plusieurs lignes

La marge pour les éléments fils est de deux espaces.

Les éléments occupant plusieurs lignes sont découpés pour que les attributs soient alignés.

Voici un exemple de cet alignement:

```
<svg width="200" height="100">
  <circle cx="10" cy="86" r="15"
        fill="red"
        stroke="black" />
  <!-- code for the car goes here -->
</svg>
```

Mise en exergue de termes importants

Quand vous rencontrerez pour la première fois un terme important, il sera en gras, puis en style normal dans la suite du texte.

Nous utilisons la police Courier pour le code.

Le code associé au texte sera présenté ainsi :

```
<svg>
  <ellipse cx="50" cy="50" rx="10" ry="20" />
</svg>
```

Pour un bloc de code important :

```
<svg width="200" height="100">
  <circle cx="10" cy="86" r="15"
        fill="red"
        stroke="black" />
  <!-- code for the car goes here -->
</svg>
```

Commandes de menu

Les commandes de menu auront cette forme :

Menu > Sous-menu > Sous-menu

Script

Le langage de script utilisé dans ce livre est **EcmaScript**, que le **W3C** accepte comme standard de script. Cet élément de script est inclus de cette manière :

```
<script type="text/ecmascript">
<![CDATA[
    // script content goes here ..
]]>
</script>
```

Eléments SVG

Il est essentiel d'avoir une structure commune pour introduire un nouvel élément à travers ce livre.

Nous utiliserons cette structure :

- Texte introductif
- Syntaxe de l'élément
- Explication des attributs spécifiques de cet élément
- Exemple concernant cet élément : image et code

La syntaxe sera présentée comme sur cet exemple

Syntaxe:

```
<marker id="name"
      refX="coordinate"
      refY="coordinate"
      markerWidth="length"
      markerHeight="length"
      markerUnits="strokeWidth | userSpaceOnUse"
      viewBox="min-x min-y width height"
      orient="auto | angle"
      style-attribute="style-attribute">
  <!-- marker content here -->
</marker>
```

Pour utiliser pleinement ce livre

- Compréhension de HTML XML
- Un visualiseur SVG (Voyez l'annexe C)

Chapitre 1

"Make everything as simple as possible, but not simpler."
- Albert Einstein (1879-1955)

Objectifs du chapitre

- Histoire de SVG
- SVG et XML
- Bitmap et Vectoriel
- Concepts SVG
- Visualiser SVG
- Créer en SVG

Introduction

Apprivoiser SVG explore le monde de Scalable Vector Graphics. SVG est un format graphique qui a été développé comme un standard industriel ouvert pour les graphiques et qui est géré par le W3C. SVG décrit les graphiques en utilisant une grammaire XML.

SVG est compatible avec les standards Web tels que DOM, XML Namespace, Xpath, Xlink, XPointer et d'autres que nous verrons dans ce livre. L'important est que chacun de ces standards fonctionne parfaitement avec SVG. Ceci permet à SVG d'être stylisé, dynamique, animé, interactif et extensible; ce qui en fait un standard extraordinaire.

Dans SVG nous trouvons la synthèse de la nouvelle génération du Web. C'est pourquoi SVG est réellement un standard étonnant et robuste pour la présentation Web.

Commençons avec cette figure. Vous devez avoir un visualiseur SVG pour la voir dans votre navigateur, l'annexe C vous aidera à en choisir un.

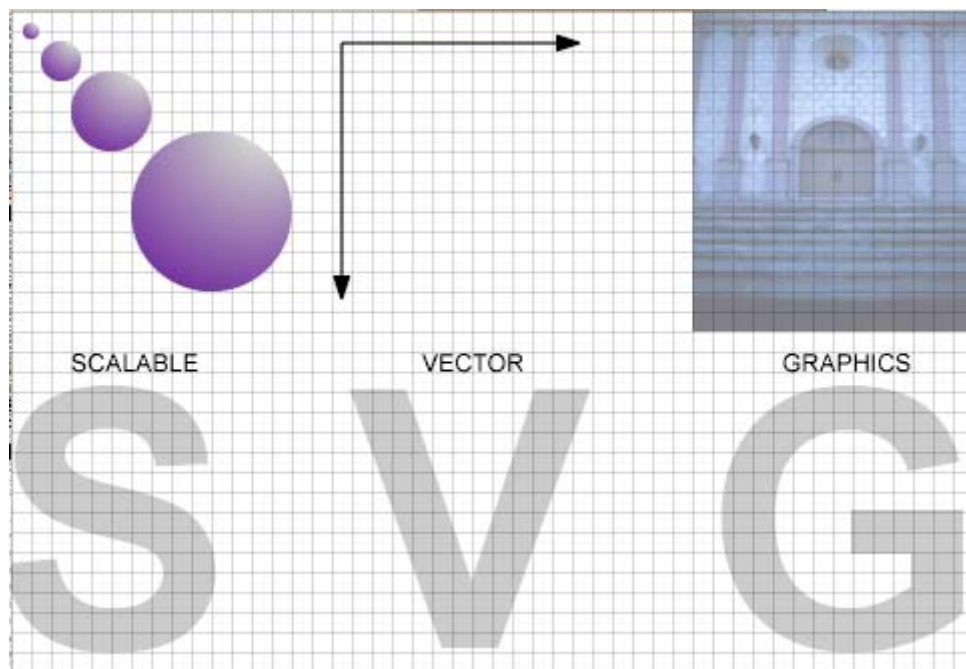


Figure 1-1: Image SVG

Et SVG apparût

Ce format graphique a suscité beaucoup d'enthousiasme quand il est apparu dans le développement du Web. SVG est la création du World Wide Web Consortium (W3C), qui est un consortium créé pour définir les standards ouverts de l'industrie internationale du Web.

SVG émergea grâce au groupe de travail du World Wide Web Consortium formé en 1998. Le **W3C SVG Working Group** continue à accroître les possibilités de SVG. Le groupe de travail du W3C est composé de représentants de groupes industriels à travers le monde. Il y a plus d'une vingtaine de membres qui représentent des groupes comme : Adobe Systems, Sun Microsystems, IBM, Corel, Macromedia, Hewlett-Packard, Microsoft, et AOL/Netscape.

Avec le développement de XML apparaît l'opportunité de créer un langage basé sur XML pour décrire les graphiques. Le W3C confia au groupe de travail sur le SVG la mission de développer un format graphique XML qui puisse produire des graphiques de grande qualité. SVG le fait et bien d'autres choses encore comme nous allons le voir.

En 1998, le W3C avait deux propositions pour un nouveau format graphique. Les deux, **Precision Graphics Markup Language (PGML)** et **Vector Markup Language (VML)** utilisaient une grammaire XML. Adobe proposait PGML basé sur son expérience avec **Postscript** et **PDF**, PGML est en fait issu de PDF. La proposition de Microsoft VML, était utilisé dans Internet Explorer, mais son développement stoppa à l'automne 1998.

Après mûre réflexion le W3C décida de combiner les meilleurs aspects des deux langages PGML et VML dans un nouveau langage baptisé SVG.

Le 2 août et à nouveau le 2 novembre 2000, le W3C accepta le passage de SVG à **Candidate Recommendation** et demanda aux développeurs de commencer à implémenter SVG. Une année et demie plus tard le premier projet de SVG est publié, Chris Lilley, le responsable du groupe de travail annonce le passage des spécifications de SVG 1.0 au statut de **W3C Recommendation**. Les spécifications de SVG 1.0 sont finalisées par le W3C le 4 septembre 2001. Ceci signifie que SVG a été testé et est un standard industriel prêt pour une complète implémentation. Le groupe de travail sur le SVG continue à étendre les possibilités de SVG avec leur travail sur SVG 1.1 ,SVG 2.0, SVG Mobile et SVG Tiny.

En janvier 2003 SVG 1.1 et SVG Mobile sont finalisés. Le travail continue sur SVG 1.2, SVG 2.0 et SVG Print qui supporteront certainement le formatage de texte qui est un souhait de nombreux utilisateurs.

Tableau de marche :

Document	WD1	WD2	LC	Ends	CR	PR	REC
SVG 1.0	-	-	-	-	-	-	5 Sep 2001
SVG 1.1	-	-	-	-	-	11 Nov 2002	14 Jan 2003
SVG Mobile Profiles	-	-	-	-	-	11 Nov 2002	14 Jan 2003
SVG 1.2	11 Nov 2002	[Jan 2003]	[May 2003]	[July 2003]	[August 2003]	[Dec 2003]	[Jan 2004]

SVG Print Requirements	[Jan 2003]	-	-	-	-	-	-
SVG Print	[Mar 2003]	-	-	-	-	-	-
Authoring Tool Guidelines	[Feb 2003]	-	-	-	-	-	-
Accessibility Techniques	[Mar 2003]						

Légende: WDn = nième projet; LC = dernier appel pour les remarques (i.e., dernier projet); Ends = Fin pour les commentaires; CR = Candidate Recommendation; PR = Proposed Recommendation; REC = W3C Recommendation.
[Feb '02] = date prévue.

SVG est XML

SVG utilise la grammaire eXtensible Markup Language (XML), le standard du 21^e siècle pour les documents Web. D'autres applications basées sur XML sont en usage ou en développement XHTML, MathML, SMIL, X3D, XFORMS, et de nombreuses autres. L'adoption rapide de ces technologies a grandement amélioré la production de documents. Comme vous le voyez, SVG est en phase avec le futur du Web.

SVG étant une application de XML bénéficie de tous les avantages qu'apporte XML. SVG est du texte et un standard ouvert.

Si vous êtes familier avec XML vous verrez qu'il n'y a rien de nouveau dans la structure du langage SVG. Comme la figure 1-2 le montre, SVG est composé de texte avec des balises faites d'éléments et d'attributs. Ce diagramme montre des composants du langage SVG utilisant la grammaire et la syntaxe XML :

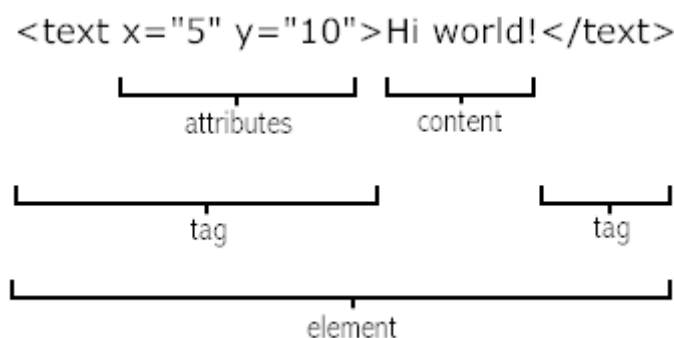


Figure 1-2: Syntaxe de l'élément SVG

SVG est efficace

SVG est un format graphique basé sur du texte. Ceci signifie qu'un fichier SVG est simplement un fichier texte. Les fichiers SVG sont très petits comparés aux autres formats graphiques.

L'image de la figure 1-1 nécessite un fichier d'environ 2.3 KB. (Notons qu'il fait appel à une image externe jpeg positionnée au-dessus du "G"). Si nous utilisons **gzip** pour **compresser** le fichier SVG, il pèse moins de 1 KB. C'est peu!

Vous trouverez ceci encore plus étonnant quand vous découvrirez que vous pouvez redimensionner votre image sans aucune perte de qualité, sauf si elle utilise une image bitmap. Comme SVG est conforme au modèle de document (DOM) et aux feuilles de style, vous avez accès à tous les éléments et attributs de la figure pour les manipuler. Si vous souhaitez déplacer un de ces éléments, il vous suffit d'ajouter un élément d'animation ou un script, ce qui n'augmentera guère la taille du fichier.

SVG est puissant

Depuis des décennies, les programmeurs cherchaient la meilleure manière d'afficher des graphiques en deux dimensions. L'utilisation d'un format vectoriel permet plus d'interactivité, autorise des fonctionnalités de zoom et de panoramique et favorise l'animation programmée. Une fois chargé sur le poste client, SVG peut être réutilisé à différentes tailles, avec différentes fonctionnalités et même différentes données.

Un ensemble de technologies peut faire de SVG le format graphique d'une application XML. Par exemple, ECMAScript, CSS, XSLT et XSL-FO peuvent être appliqués à SVG, côté client, pour produire des documents Web réellement dynamiques, interactifs et conviviaux. De plus, SVG étant conforme à XML, il est extensible, ce qui veut dire que SVG peut être visualisé avec d'autres langages ou applications basés sur XML tels que XHTML, SMIL, XFORMS et MathML.

SVG a plusieurs avantages sur les autres formats graphiques utilisés sur le Web.
En particulier:

- Redimension – Les images SVG gardent la même qualité dans un zoom ou un panoramique, ce qui est important pour des supports portables, pour l'affichage de cartes, de diagrammes et pour l'impression
- Fichier texte – Développeurs et graphistes peuvent éditer le code SVG avec des outils simples et variés.
- Taille des fichiers – Les techniques de compression et l'efficacité du vocabulaire produisent des fichiers de très petite taille, ce qui est pratique pour le Web.
- Recherche – Le contenu SVG étant du texte peut être indexé et permet une recherche.
- Infinité de couleurs et polices - 16 millions de couleurs et support pour des polices incorporées, permettant un même aspect à l'écran et en impression.
- Filtres sur les objets – Ombrages, flous, et effets de lumière peuvent être appliqués quand le SVG est rendu côté client. Les effets de filtres dynamiques sont une véritable innovation pour les graphiques du Web.
- Animation – SVG offre des possibilités d'animation avec des éléments incorporés ou grâce au script en modifiant les objets du SVG DOM.

- Interactivité – Le script permet une véritable interactivité.
- XML – La compatibilité avec XML, HTML4, XHTML et la conformité à CSS, XSL-FO, et au DOM signifient que SVG utilise les feuilles de style, les scripts, peut être étendu, interactif et s'intègre facilement aux autres langages XML.

SVG est redimensionnable (Graphique vectoriel)

Récemment, les développeurs Web ne pouvaient utiliser que des images **bitmap** dans les navigateurs. Les principaux formats bitmap utilisés sont:

- Graphics Interchange Format (GIF)
- Joint Photographic Experts Group (JPEG)
- Portable Network Graphics (PNG)

Les images bitmap sont affichées par des matrices de pixels, ce qui oblige à stocker l'information pour chaque pixel du graphique.

Les formats de graphiques vectoriels tels que SVG et **SWF** utilisent les vecteurs pour afficher l'image. Ils contiennent des objets géométriques tels que des lignes, des courbes.

Quoique les images bitmap conviennent dans de nombreuses situations, les graphiques vectoriels ouvrent de nouvelles possibilités pour le Web graphique. Par exemple comparez les deux affichages de la même image SVG.



Figure 1-3: Taille originale

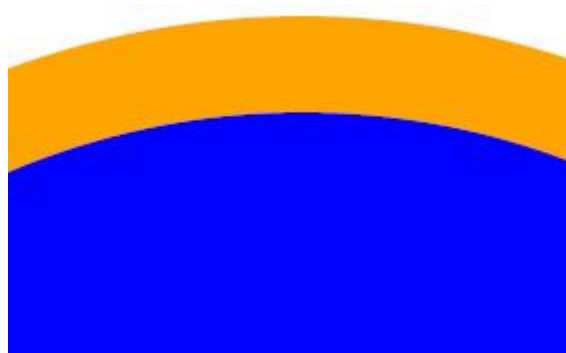


Figure 1-4: Après un zoom

Remarquez que nous n'avons rien perdu dans la qualité de l'affichage. Les visualiseurs vectoriels sont capables de recalculer comment l'image doit être rendue à partir de la description du cercle qui constitue cette image. C'est un avantage décisif de SVG et des formats vectoriels.

Les graphiques vectoriels sont plus efficaces pour l'impression et l'animation, ainsi que pour l'interactivité (y compris zoom et panoramique).

Une des limitations des images bitmap est la manière dont elles décrivent le graphique. Elles ne le décrivent pas comme formes ou objets, mais doivent décrire chaque pixel de l'image. Les images bitmap peuvent utiliser des techniques de **compression** efficaces et être chargées progressivement, cependant ce que montrent ces illustrations confirme que la possibilité de redimensionner les images vectorielles est un avantage par rapport aux images bitmap.



Figure 1-5: Image originale

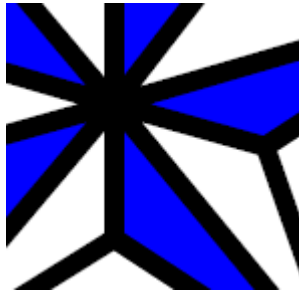


Figure 1-6: Image vectorielle à 300% Figure 1-7: Image bitmap à 300%

L'image bitmap devient très **pixelisée** quand sa taille est augmentée alors que l'image vectorielle garde toutes ses qualités! SVG est un format graphique vectoriel. Vous savez maintenant pourquoi SVG est nommé Scalable Vector Graphics.

Les blocs en SVG

Scalable Vector Graphics (SVG) utilise trois types d'objets graphiques : formes vectorielles, texte et images bitmap. Expliquons en détail.

SVG est composé d'objets graphiques

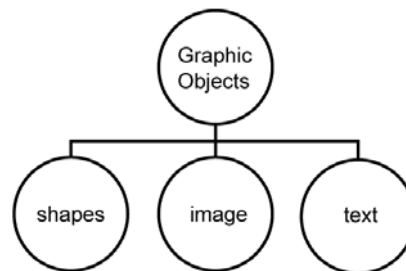


Figure 1-8: Objets graphiques en SVG

Les formes en SVG

Cette image peut être obtenue en utilisant la balise `<image>` de HTML ou en utilisant SVG avec la balise `<circle>`.

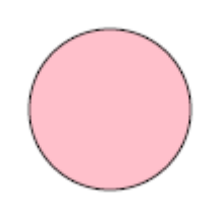


Figure 1-9: Cercle en SVG

HTML document :

```
<html>
  <head>
    <title>Pink Circle</title>
  </head>
  <body>
    
  </body>
</html>
```

SVG document :

```
<?xml version="1.0" ?>
<svg width="100" height="100">
  <title>Pink Circle</title>
  <circle cx="50" cy="50" r="40" fill="pink" stroke="black"/>
</svg>
```

Si les deux documents peuvent produire le même cercle, pourquoi tant d'intérêt pour le format SVG? Quels sont les avantages d'utiliser SVG plutôt que HTML ? C'est à cette question que nous essayons de répondre dans ce livre.

Les formes SVG comprennent les éléments suivants : `<path>`, `<line>`, `<rect>`, `<circle>`, `<ellipse>`, `<polyline>` et `<polygon>`. L'élément `<path>` peut décrire des formes très complexes en utilisant des courbes de Bézier, cubiques et quadratiques. Nous allons étudier les formes dans le chapitre 2 et l'élément `<path>` dans le chapitre 4.

Les formes ne sont pas toujours la meilleure méthode pour rendre le contenu d'une image, aussi le groupe de travail SVG du W3C a ajouté deux autres types d'objets : texte et images. Comment afficher ou insérer des images SVG sera expliqué par de nombreux exemples.

Images en SVG

En SVG, les images bitmap sont des objets graphiques. Ceci permet au développeur ou à l'utilisateur d'appliquer une animation ou un filtre à une image bitmap comme nous le voyons dans la figure 1-1. Notez que l'image bitmap dans l'angle supérieur droit est semi-opaque! Nous découvrirons comment le faire dans le prochain chapitre. Les spécifications SVG exigent qu'un visualiseur supporte au minimum les deux formats JPEG et PNG.

Texte en SVG

Après formes et images, le troisième type d'objet graphique en SVG est le texte. Comme un objet graphique, le texte peut être manipulé par un script comme les formes et les images!

Le texte SVG est composé de caractères nommés d'une manière technique de "glyphes". Nous expliquerons les différences entre caractères et glyphes, ainsi que bien d'autres possibilités pour le texte dans le chapitre 5.

SVG en pratique

SVG est déjà présent dans l'industrie informatique dans des produits majeurs, des outils et des applications Web.

Visualiser SVG

Avant de commencer à travailler avec SVG, assurez-vous que vous avez un visualiseur SVG. Référez-vous à l'annexe C pour vous aider dans votre choix.

Créer SVG

Maintenant que vous savez un peu mieux ce qu'est SVG, créons notre premier SVG. La manière la plus rapide est de taper le code de notre premier exemple. Prenez votre éditeur de texte favori et entrez dans le monde du SVG.

Cet exemple est très court et simple – un rectangle noir.

L'un des avantages de SVG est sa simplicité d'utilisation. Vous pouvez créer des images SVG sans logiciel graphique. Notre premier exemple montre la structure complète d'un document SVG.

Ligne 1: `<?xml version="1.0" ?>`
Ligne 2: `<svg>`
Ligne 3: `<rect x="100" y="50" width="100" height="100" />`
Ligne 4: `</svg>`

Ce code crée cette image :

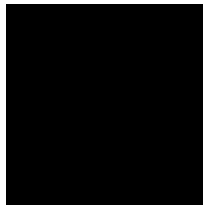


Figure 1-10: rectangle avec remplissage par défaut

Line 1: La déclaration XML avec la version XML du document.
Line 2: Ouvre le document SVG
Line 3: Crée une forme SVG en utilisant l'élément `<rect>` en précisant
 a: le coin supérieur gauche du rectangle avec les attributs 'x' et 'y'
 b: la largeur et la hauteur du rectangle avec les attributs 'width' et 'height'.
Line 4: Ferme le document SVG

Notez que la couleur de remplissage par défaut est noire. Nous traiterons du style en détail dans le chapitre 5.

Sommaire

Félicitations! Vous venez de créer une page Web utilisant SVG. Si vous trouvez cet exercice trop primaire, nous allons rapidement aborder des sujets plus délicats.

Dans la suite nous allons analyser formes, texte et images. Comme SVG est idéal pour une utilisation sur le Web, nous allons examiner tous les aspects de son utilisation sur le Web. Vous trouverez dans ce livre de nombreux exemples accompagnés de leur code qui devraient vous être d'une grande aide.

Dans ce chapitre nous avons vu ce qu'était SVG et commencé à l'apprivoiser en créant et visualisant notre premier exemple. Nous espérons que vous êtes intéressé(e) et voulez en savoir plus..