SVG

SVG signifie Scalable Vector Graphics, en francais graphique vectoriel adaptable.

Ce format est basé sur du XML et permet de définir des éléments graphiques pour le web. Il peut également être combiné avec du CSS. Son principal avantage est qu’une image peut être redimensionnée sans perte de qualité. Enfin le poids d’une image vectoriels est beaucoup léger.

Contrairement aux formats d’images matricielles qui sont en fait des tableaux de carrés colorés basés sur les pixels, le SVG va construire l’image à partir de points reliés. Ce format gère quelques formes géométriques de base comme les rectangles, les cercles mais également des chemins qui utilisent les courbes de Bézier et permettent ainsi d’obtenir presque n’importe quelle forme.

Il est possible de coder le SVG à la main grâce à un éditeur de texte. En revanche, pour réaliser des images moyennement complexes, il est préférable d'utiliser un éditeur graphique dédié, comme Inkscape. Il possède une fonctionnalité appelée Trace Bitmap qui permet de passer d'une image matricielle à une image vectorielle.

**SVG et HTML**

Il existe une balise <svg></svg>

Cette balise va créer une fenêtre qu’on va pouvoir dimensionner avec les attributs classiques width et height.

On peut modifier l’échelle grâce à l’attribut viewbox.

Exemple :

<svg width="300" height="200"></svg>

On aura donc un élément SVG dont le système de coordonnée ira de 0 à 300 en largeur et de 0 à 200 en hauteur

<svg width="300" height="200" viewBox="0 0 30 20"></svg>

En ajoutant une notion de viewbox, nous transformons ce système de coordonnées afin qu'il aille de 0 à width à l'horizontale et de 0 à height à la verticale. Ici, ce sera donc 300/30 (=10) en x ce qui équivaut à 10 unités d'élément SVG pour 1 unité

**Les formes géométriques**

**Le rectangle** :

* balise : <rect/>
* attributs : x et y (coordonnées du coin supérieur gauche) puis width et height (longueur et largeur du rectangle)

<rect x="25px" y="25px" width="150" height="100"/>

**Le cercle ou ellipse :**

* balise : <circle/>
* attributs : cx et cy (coordonnées du centre du cercle) puis r (rayon du cercle)

<circle cx="110" cy="110" r="50" />

**Les lignes :**

* balise : <line/>
* attributs : x1 et y1 (coordonnées du point de départ) puis x2 et y2 (coordonnées du point de d’arrivée)
* attributs d’apparence : couleur (stroke), de largeur (stroke-width), de pointillés (stroke-dasharray) ou de fin de ligne (stroke-linecap)

<line x1="150" y1="50" x2="250" y2="150" stroke="blue" stroke-width="5" stroke-dasharray="5,3,2" />

<line x1="50" y1="100" x2="300" y2="100" stroke="#0a989a" stroke-width="20" stroke-linecap="round" />

**Les chemins :**

* balise : <path/>
* attributs : d

<path d="M 100,180 L 140,0 L 180,180 L 220,0 L 260,180 L 300,0 L 330,180"

* attributs de remplissage : stroke (couleur du contour) fill (couleur de remplissage) stroke-opacity (opacité du contour de 0 à 1) fill-opacity (opacité du remplissage de 9 à 1)

**La balise <g></g>**

Permet de regrouper plusieurs éléments svg

**L’animation SVG**

L’élément animate : Utilisé pour animer un seul attribut ou une seule propriété au cours du temps.

Attention : les balises de forme géométriques orphelines deviennent des balises ouvrantes et fermantes

<rect width="100" height="50">

<animate attributeName="width" attributeType="XML"

fill="freeze"

from="0" to="300"

begin="0s" dur="3s"/>

</rect>

On peut enchainer les animations.

<rect x="15" y="0" width="80" height="200" rx="20" ry="20" style="fill:black;"/>

<circle cx="55" cy="40" r="25" style="stroke:black; stroke-width: 1px;fill: #F4F4F4;">

<animateColor attributeName="fill" id="rect1"

repeatDur="3"

from="green" to="yellow"

dur="3s"/>

</circle>

<circle cx="55" cy="100" r="25" style="stroke:black; stroke-width: 1px;fill: #F4F4F4;" >

<animateColor attributeName="fill" id="rect2"

repeatDur="3"

from="yellow" to="orange"

begin="rect1.end" dur="3s"/>

</circle>

<circle cx="55" cy="160" r="25" style="stroke:black; stroke-width: 1px;fill: #F4F4F4;" >

<animateColor attributeName="fill" id="rect3"

repeatDur="3"

from="orange" to="red"

begin="rect2.end" dur="3s"/>

</circle>

L'élément : set

Cet élément permet de fixer une valeur pendant une certaine durée de temps. Dans cet exemple le rectangle rose est affiché pendant 3 secondes. L'attribut CSS visibility est modifié pendant ce laps de temps, ensuite il récupère sa valeur initiale déclarée dans l'élément g.

L'élément : animateTransform

Avant de vous parler de l'animation des transformations, il faut une brève introduction sur le concept de transformations en SVG. Il y a cinq valeurs possibles de transformations pour l'attribut transform d'un élément SVG :

* La translation (translate);
* La rotation (rotate);
* Le mise à l'échelle (scale);
* L'inclinaison (skewX, skewY);

Ces transformations telles quelles ne sont pas extraordinairement utiles. Mais les transformations peuvent être animées grâce à l'élément animateTransform et là cela devient utile ! Pour la mise en œuvre, c'est similaire aux exemples précédents.

L'élément animatemotion

L'élément animateMotion permet de faire défiler un objet sur un path. Il y a deux façons de faire. La première consiste à utiliser un attribut path dans l'élément animateMotion dans lequel on renseignera les coordonées du path. La deuxieme à utiliser la balise mpath comme enfant direct d'animateMotion avec l'attribut xlink:href qui sera lié à l'id d'un path existant.

Source : https://www.alsacreations.com/tuto/lire/1421-svg-initiation-syntaxe-outils.html