

# Visualisation de données

## 02 - Scalable Vector Graphics (SVG)

22 février 2024

Noemi Romano  
noemi.romano@heig-vd.ch

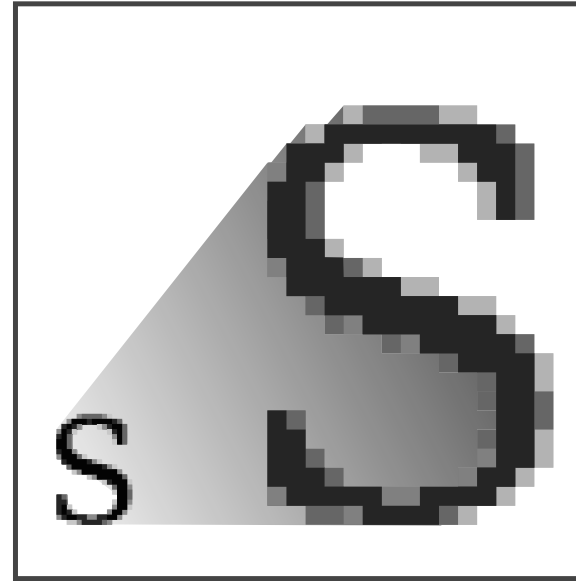


# D3 & SVG

*D3.js is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, **SVG** and CSS*

# SVG

Scalable Vector Graphics  
(*graphique vectoriel  
adaptable*)



Matriciel  
.jpeg .gif .png

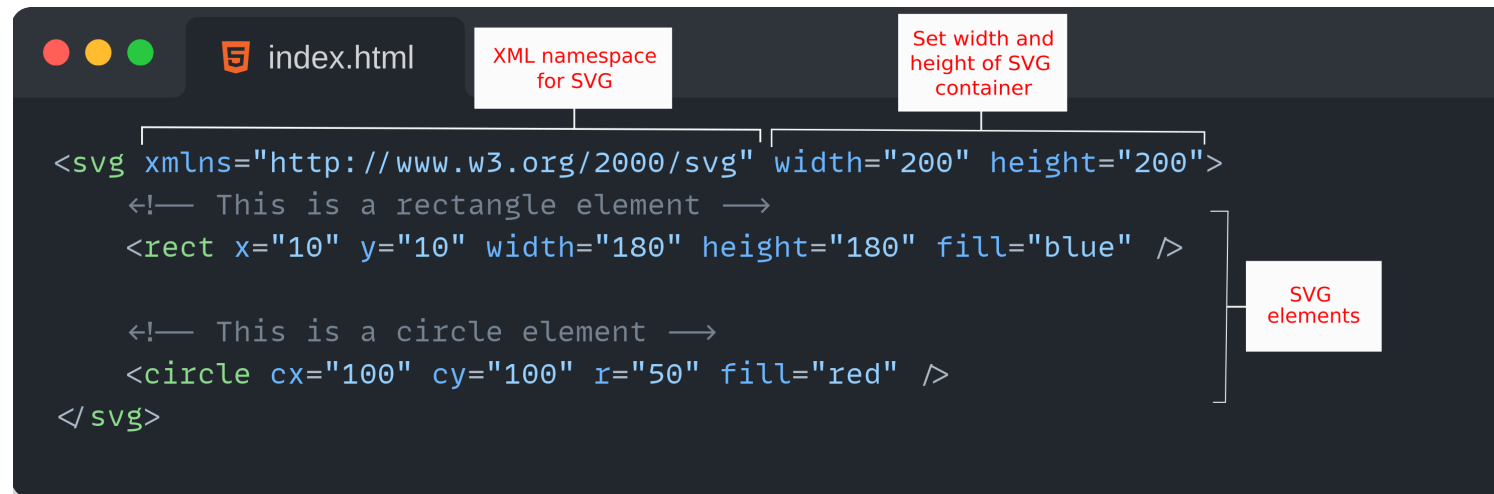


Vectoriel  
.svg

↳ [MDN: Scalable Vector Graphics \(SVG\)](#)

# Syntaxe SVG

- Langage de balisage XML (eXtensible Markup Language)
- Le document SVG est délimité par la balise `<svg>`
- **Attributs** : `width` et `height` définissent la largeur et la hauteur du conteneur SVG. `viewBox` spécifie la boîte englobante utilisée pour définir la vue sur le contenu SVG



The screenshot shows a code editor window titled 'index.html'. The code is as follows:

```
<svg xmlns="http://www.w3.org/2000/svg" width="200" height="200">
  <!-- This is a rectangle element -->
  <rect x="10" y="10" width="180" height="180" fill="blue" />

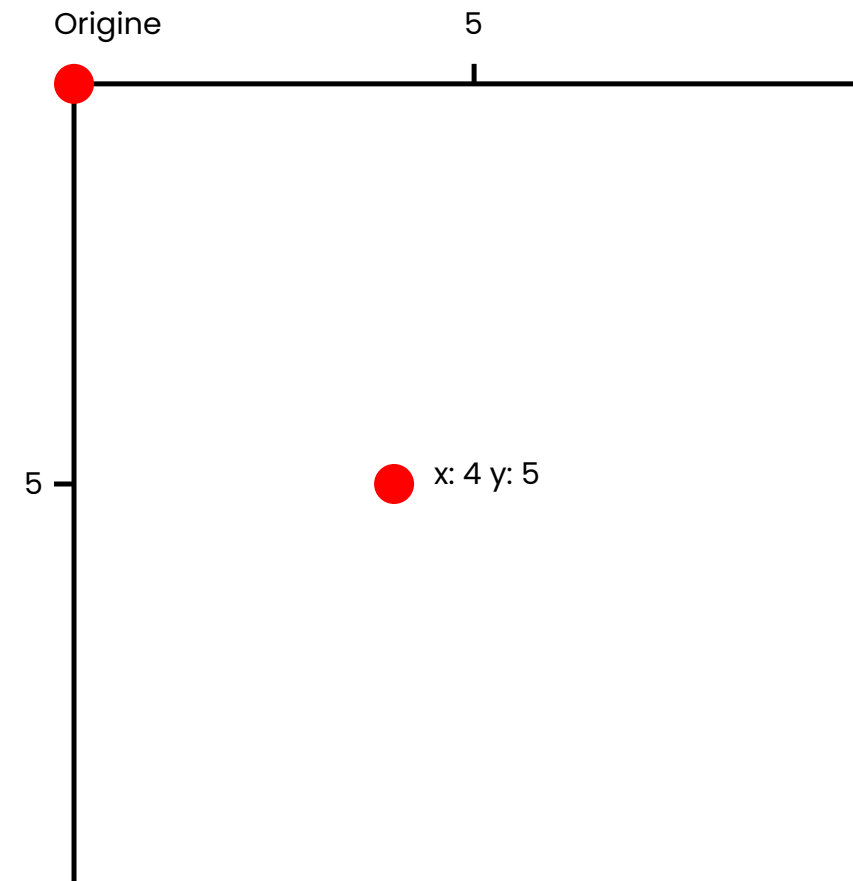
  <!-- This is a circle element -->
  <circle cx="100" cy="100" r="50" fill="red" />
</svg>
```

Annotations in the image:

- A bracket above the `xmlns` attribute points to a box labeled "XML namespace for SVG".
- A bracket above the `width` and `height` attributes points to a box labeled "Set width and height of SVG container".
- A bracket on the right side groups the `<rect>` and `<circle>` elements, pointing to a box labeled "SVG elements".

# Systeme de coordonnées

- L'**origine** du système de coordonnées SVG est située dans le **coin supérieur gauche** du conteneur SVG. Les coordonnées (0,0) représentent ce point.
- L'**axe horizontal** est appelé l'axe X. Les valeurs augmentent **de gauche à droite**
- L'**axe vertical** est appelé l'axe Y. Les valeurs augmentent **de haut en bas**
- Par défaut, les unités de mesure sont les **pixels**, mais vous pouvez spécifier d'autres unités comme em, cm, mm, etc.

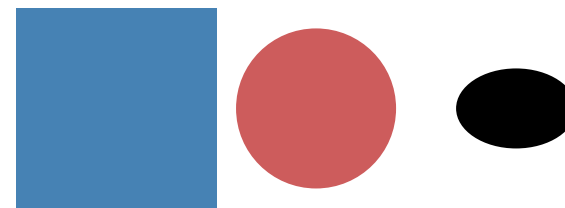


# Formes et textes

- **Formes**: Les éléments graphiques tels que `<rect>`, `<circle>`, `<line>`, etc., sont utilisés pour créer des formes.
- **Texte** : Utilisez l'élément `<text>` pour afficher du texte.
- **Couleurs et Styles** : l'attribut `fill` définit la couleur de remplissage d'une forme. L'attribut `stroke` définit la couleur de la bordure d'une forme.

```
index.html

<svg width="1000" height="500">
  <rect x="0" y="0" width="100" height="100" fill="steelblue" />
  <circle cx="150" cy="50" r="40" fill="indianred" />
  <ellipse cx="250" cy="50" rx="30" ry="20" />
  <line x1="300" y1="0" x2="500" y2="100" stroke="red" />
  <text x="500" y="80">Et bien d'autres !</text>
</svg>
```



Et bien d'au

# Path

- L'élément `<path>` est utilisé pour définir des trajectoires ou des contours de formes complexes.
- L'attribut principal utilisé est `d` (pour "data"), qui contient une séquence de commandes et de paramètres qui décrivent la forme du chemin.

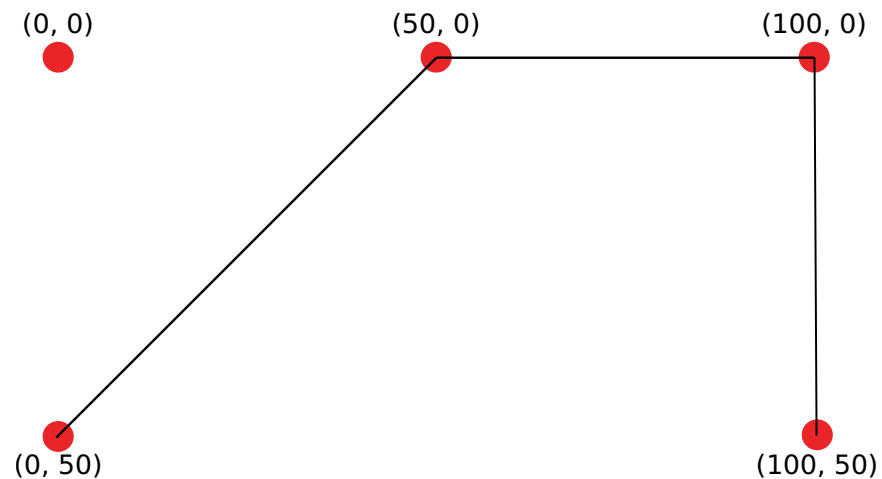
**M**ove to (M x y)

**L**ine to (L x y)

**H**orizontal line (H x)

**V**ertical line (V y)

```
index.html
<svg width=500 height=200>
  <path d="M 0 50 L 50 0 H 100 V 50" fill="none" stroke="black" stroke-width="5" />
</svg>
```



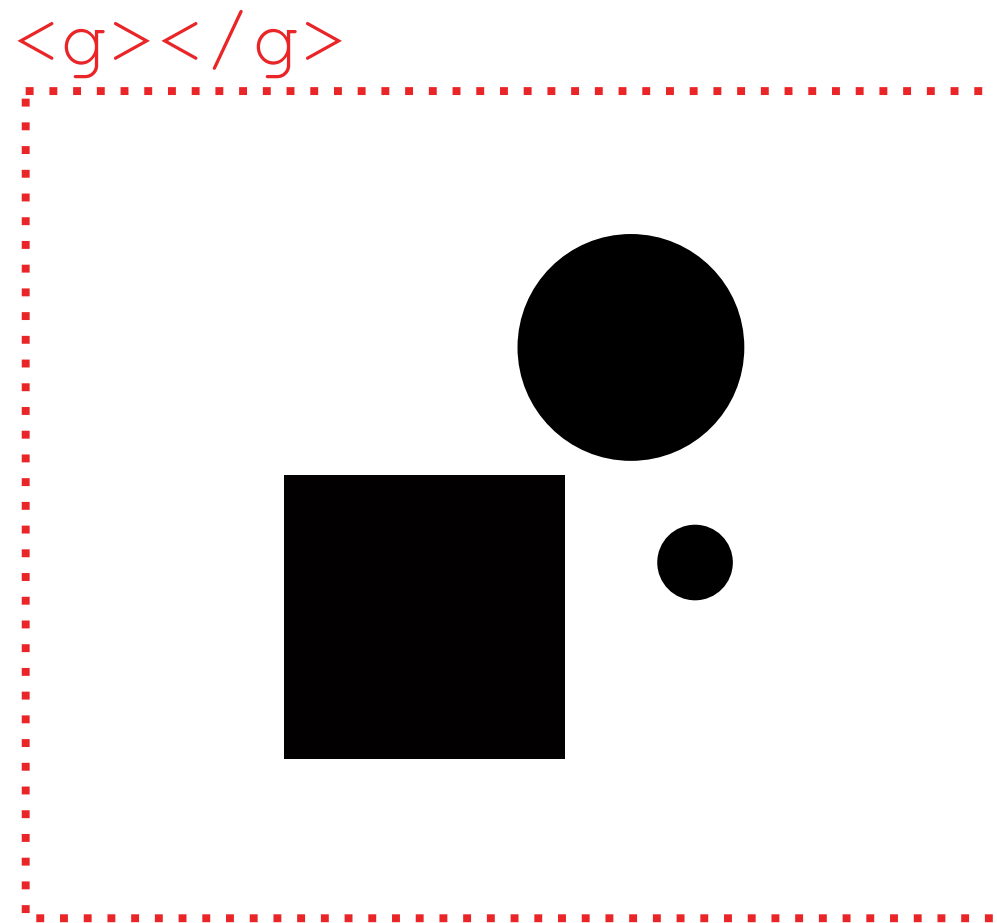


# Groupe

- L'élément `<g>` est utilisé pour regrouper plusieurs éléments SVG.
- Facilite la manipulation et la transformation d'un ensemble d'éléments en tant qu'unité.

```
index.html

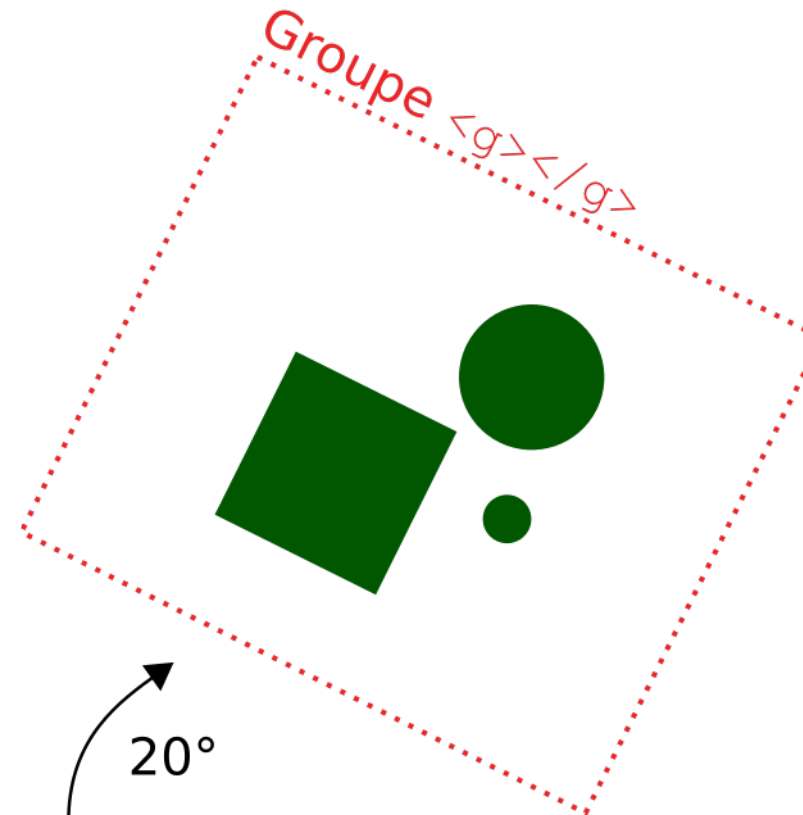
<g>
  <circle cx="150" cy="50" r="30" />
  <rect x="100" y="60" width="50" height="50" />
  <circle cx="160" cy="55" r="10" />
</g>
```



# Transformations

- `translate(x,y)` : déplace un élément le long de l'axe X et Y
- `rotate(angle, cx, cy)` : fait pivoter un élément autour du point (cx, cy)
- `scale(facteur_echelle)` : redimensionne un élément

```
index.html
<g transform="rotate(20, 100, 50)" fill="green">
  <circle cx="150" cy="50" r="30" />
  <rect x="100" y="60" width="50" height="50" />
  <circle cx="160" cy="55" r="10" />
</g>
```

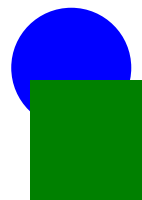


# Animations

- Prend en charge les animations avec des éléments tels que `<animate>` et `<animateTransform>`.
- Permet la création d'effets visuels dynamiques.
- Les navigateurs préfèrent les animations avec CSS 🤪

```
index.html
<svg xmlns="http://www.w3.org/2000/svg" width="300" height="200">
  <!-- Cercle avec animation de translation horizontale -->
  <circle cx="50" cy="100" r="30" fill="blue">
    <animate attributeName="cx" from="50" to="250" dur="3s" repeatCount="indefinite" />
  </circle>

  <!-- Rectangle avec animation de translation verticale -->
  <rect x="100" y="50" width="60" height="60" fill="green">
    <animateTransform attributeName="transform" type="translate" values="0 0; 0 80; 0 0" dur="3s" repeatCount="indefinite" />
  </rect>
</svg>
```



# Exercices

Rendez-vous sur **GitHub**