

---

# TD-3 : SVG et MathML

Benoît Valiron

## 1. Écriture d'une DTD

(reprise du TD-2)

On souhaite décrire le contenu d'une bibliothèque et des livres empruntés avec un document XML. La racine sera `bibliothèque`, et ses fils seront d'abord `livre` puis `utilisateur`. Chaque `livre` possèdera un attribut `id`, un attribut `lang`, un fils `titre`, un fils `éditeur` et un fils `emplacement`. L'attribut `id` sera de type ID (obligatoire). `lang` pourra être `fr`, en ou `de`. Par défaut, ce sera `fr`. L'élément `utilisateur` aura comme fils `nom` (obligatoire), `prénom` (optionel), `adresse` (optionel), et comme attribut `emprunts` (optionel, valeur par défaut " "), de type IDREFS.

Écrivez la DTD correspondante, puis produisez un fichier contenant 3 livres et deux utilisateurs Alice et Bob. Alice aura emprunté les deux premiers livres et Bob le dernier.

### Note

Cet exercice est l'archétype de ce que vous pourriez avoir à faire en examen.

## 2. Dessin en SVG

On rappelle que la racine d'un document SVG est l'élément `svg` et que les sous éléments sont sous la formes:

```
<circle cx="70" cy="100" r="50" />
<rect x="150" y="50" width="135" height="100" />
<line x1="325" y1="150" x2="375" y2="50" />
<polyline points="50,250 75,350 100,250 125 350" />
<polygon points=" 250,250 297,284 279,340" />
<ellipse cx="400" cy="300" rx="72" ry="50" />
<texte x="100" y="100">Hello</texte>
```

Par ailleurs, ils peuvent prendre les attributs `stroke` (couleur de trait), `stroke-width` (épaisseur de trait) et `fill` (couleur de l'intérieur de la forme). Par défaut, l'intérieur d'une forme est blanche, de même que la couleur de son trait.

Les lignes suivantes pourraient apparaître dans une DTD pour SVG. Finissez-les.

```
<!ELEMENT svg ...
<!ELEMENT rect ...
<!ATTLIST rect ...
```

Attention au couleurs par défaut.

Écrivez un fichier SVG qui contient:

- Un rectangle de largeur 100, de hauteur 200, dont le coin supérieur gauche est placé aux coordonnées (50,10), est rempli de rouge et dont le contour orange est de largeur 10.
- Un cercle de rayon 50, dont le centre est au point de coordonnées (80, 120), de couleur bleu. Le contour est noir et de largeur 2 pixels.
- Une ellipse de rayon en x 20, de rayon en y 40, de centre (120, 120), de même remplissage et contour que le cercle précédant.

- Un polygone qui entoure le tout, de couleur intérieur jaune et de contour noir de largeur 1. (Attention, il est opaque).
- Du texte en noir sous le polygone.

### 3. Un peu de MathML

On rappelle que les éléments d'une formule MathML sont les suivants : `mi` pour introduire une variable, `mo` pour un opérateur, `mn` pour un nombre. La concaténation de formules se fait à l'aide de l'élément `mrow`. Les formules complexes sont `msqrt` pour la racine carré, `mroot` pour la racine générale, `menced` pour les parenthèses, `mfrac` pour les fractions, `msup` pour les exposants. Deux opérateurs utiles mais non-accessibles sur le clavier sont les suivants. "·" s'obtient avec `&#183;` et "±" avec `&#177;`.

Reproduisez la formule 1 en MathML.

**Figure 1. Une formule...**

$$\frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2}$$

Une invocation de la DTD pour MathML est la suivante :

```
<!DOCTYPE math
  PUBLIC "-//W3C//DTD MathML 2.0//EN"
  "http://www.w3.org/Math/DTD/mathml2/mathml2.dtd">
```

Définissez des ENTITY pour les deux symboles afin de leur donner un nom plus lisible. Comment les ajouteriez-vous à la déclaration ?

### 4. Espaces de noms, XHTML, SVG et MathML

Une DTD simplifiée pour XHTML qui incluerait du SVG et du MathML pourrait commencer comme suit :

```
<!ELEMENT html (head,body)>
<!ELEMENT head title>
<!ELEMENT body (h1|h2|p)*>
<!ELEMENT h1 (#PCDATA)>
<!ELEMENT h2 (#PCDATA)>
<!ATTLIST h1 align (center | left | right)>
<!ATTLIST h2 align (center | left | right)>
<!ELEMENT p (#PCDATA | b | i | s:svg | m:math)*>
<!ELEMENT b (#PCDATA | i)*>
<!ELEMENT i (#PCDATA | b)*>
<!-- Après ça, la définition de s:svg et de m:math et de leurs
  sous-éléments -->
...

```

Créez un document XHTML avec un titre (de première classe), deux paragraphe avec leur propre titre (deuxième classe), dans le premier, un dessin SVG et dans le deuxième, une formule MathML. Attention aux espaces de noms.

### 5. Encore des entités

Considérez le fichier XML suivant.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE ville SYSTEM "ville.dtd">
<ville xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#">
  <nom>&grenoble;</nom>
  <geo:lat>45.196349</geo:lat>
  <geo:long>5.73226</geo:long>
</ville>
```

Si le fichier valide, que contient la DTD ?

Même question, avec une entité paramètre pour `geo`. Comment changeriez-vous l'appel à la DTD dans le fichier XML pour pouvoir changer de préfixe ?