



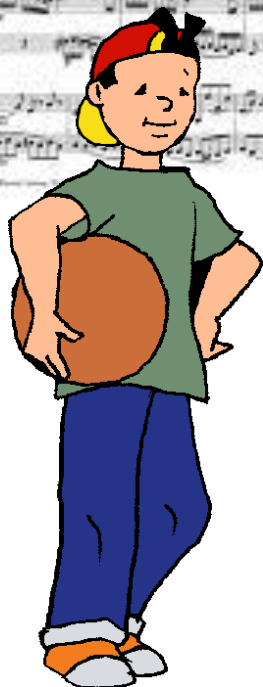
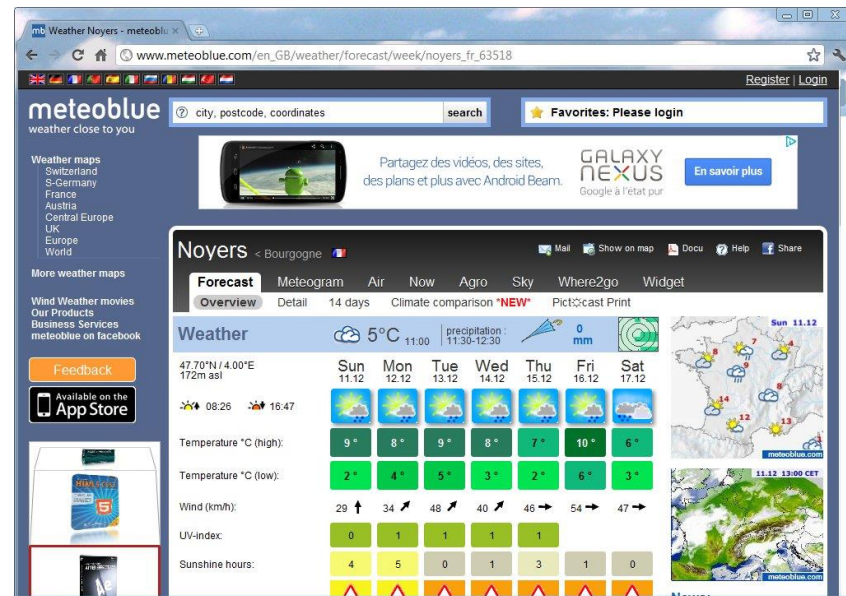
XML et multimédia

Jean-Claude MOISSINAC
Novembre 2014





Qu'ont-ils en commun?



$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

éléments de la France suivante avec MathML



Qu'ont-ils en commun?

MusicML

SVG

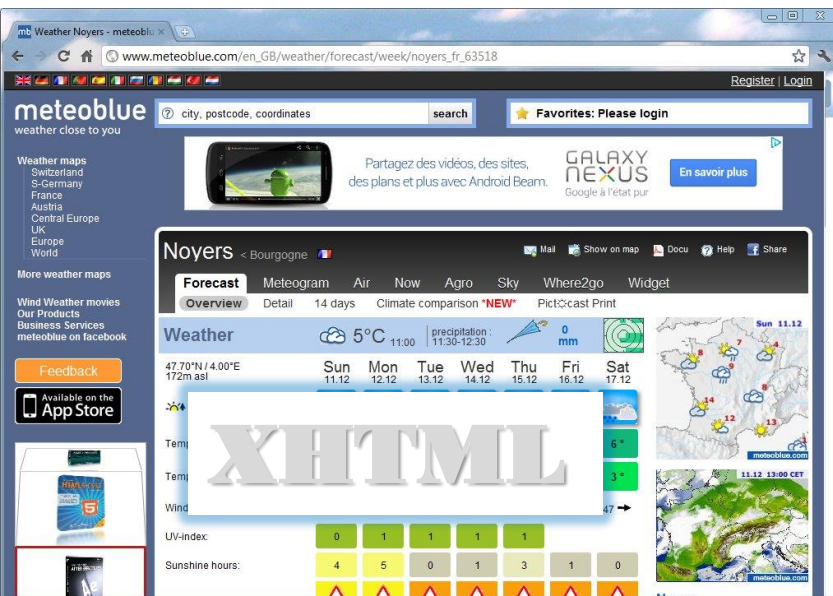
MMS/SMIL



MathML

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

XHTML



```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<title>Java 1&egrave;re ann&eacute;e</title>
```

```
<meta name="generator" content="S5" />
```

```
<meta name="version" content="S5 1.1" />
```

```
...
```

```
<!-- style sheet links -->
```

```
<link rel="stylesheet"
href="ui/default/slides.css" type="text/css"
media="projection" id="slideProj" />
```

```
...
```

```
<!-- S5 JS -->
```

```
<script src="ui/default/slides.js"
type="text/javascript"></script>
```

```
</head>
```

```
<body>
```

```
<div class="layout">
```

```
<div id="controls"></div>
```

```
<div id="currentSlide"></div>
```

```
<div id="header"></div>
```

```
<div id="footer">
```

```
<h1>T&eacute;lecom ParisTech</h1>
```

```
<h2>Programmation O.O. en Java</h2>
```

```
</div>
```




```
<score-partwise version="3.0">
```

```
<part-list>
```

```
<score-part id="P1">
```

```
<part-name>Music</part-name>
```

```
</score-part>
```

```
</part-list>
```

```
<part id="P1">
```

```
<measure number="1">
```

```
<attributes>
```

```
<divisions>1</divisions>
```

```
<key>
```

```
<fifths>0</fifths>
```

```
</key>
```

```
<time>
```

```
<beats>4</beats>
```

```
<beat-type>4</beat-type>
```

```
</time>
```

```
<clef>
```

```
<sign>G</sign>
```

```
<line>2</line>
```

MMS/SMIL



MMS/SMIL

<smil>

<head>

<meta name="title" content="vacation photos" />

<meta name="author" content="Danny Wyatt" />

<layout>

<root-layout width="160" height="120"/>

<region id="Image" width="100%"
height="80" left="0" top="0" />

<region id="Text" width="100%"
height="40" left="0" top="80" />

</layout>

</head>

<body>

<par dur="8s">

<text src="FirstText.txt" region="Text" />

<audio src="FirstSound.amr"/>

</par>

<par dur="7s">

<text src="SecondText.txt" region="Text" />

<audio src="SecondSound.amr" />

</par>

</body>

</smil>

MathML

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

MathML

<math>

<mrow>

<mi>x</mi>

<mo>=</mo>

<mfrac>

<mrow>

<mrow>

<mo>-</mo>

<mi>b</mi>

```
<math>
  <mrow>
    <mi>x</mi>
    <mo>=</mo>
    <mfrac>
      <mrow>
        <mrow>
          <mo>-</mo>
          <mi>b</mi>
        </mrow>
        <mo>&PlusMinus;</mo>
        <msqrt>
          <mrow>
            <msup>
              <mi>b</mi>
              <mn>2</mn>
            </msup>
            <mo>-</mo>
            <mrow>
              <mn>4</mn>
              <mo>&InvisibleTimes;</mo>
              <mi>a</mi>
              <mo>&InvisibleTimes;</mo>
              <mi>c</mi>
            </mrow>
          </mrow>
        </msqrt>
      </mrow>
      <mrow>
        <mn>2</mn>
        <mo>&InvisibleTimes;</mo>
        <mi>a</mi>
      </mrow>
    </mfrac>
  </mrow>
</math>
```



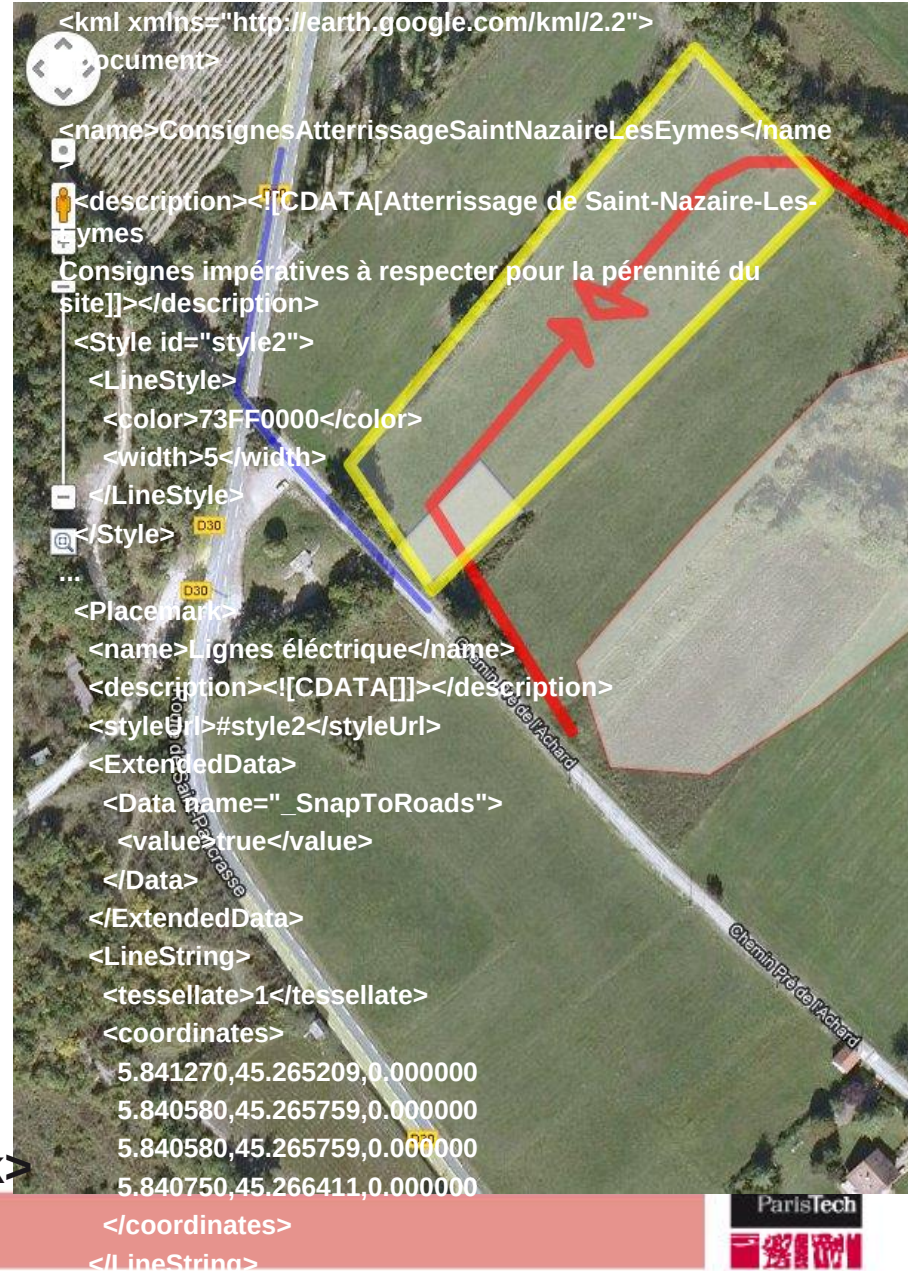
```
<?xml version='1.0' encoding='iso-8859-1'?>
<svg width='100%' height='100%' xmlns='http://www.w3.org/2000/svg'
xmlns:xlink='http://www.w3.org/1999/xlink' viewBox='0 0 1600 1200'>
<g id="s7"> <g fill='black' display='inline' fill-rule='evenodd' stroke='none'
<path id='C1582' fill='#000000' opacity='1.000000' display='inline' d='M424
<path id='C1572' fill='#Dadbe0' opacity='1.000000' d='M382 1057l1 0l1 0l1
<path id='C1570' fill='#Ffbd94' opacity='1.000000' d='M394 675l1 0l0 -1l0 -
<path id='C1514' fill='#708f70' opacity='1.000000' d='M389 623l1 0l0 -1l1 0
...
</g>
</svg>
```


RSS



```
<rss version="2.0">
  <channel>
    <title>Mon site</title>
    <description>Exemple de flux RSS
2.0</description>
    <lastBuildDate>Sat, 07 Sep 2002
00:00:01 GMT</lastBuildDate>
    <link>http://www.example.org</link>
    <item>
      <title>Actualité N°1</title>
      <description>Ceci est ma première
actualité</description> <pubDate>Sat, 07
Sep 2002 00:00:01 GMT</pubDate>
      <link>http://www.example.org/actu1</link>
    </item> </channel> </rss>
```

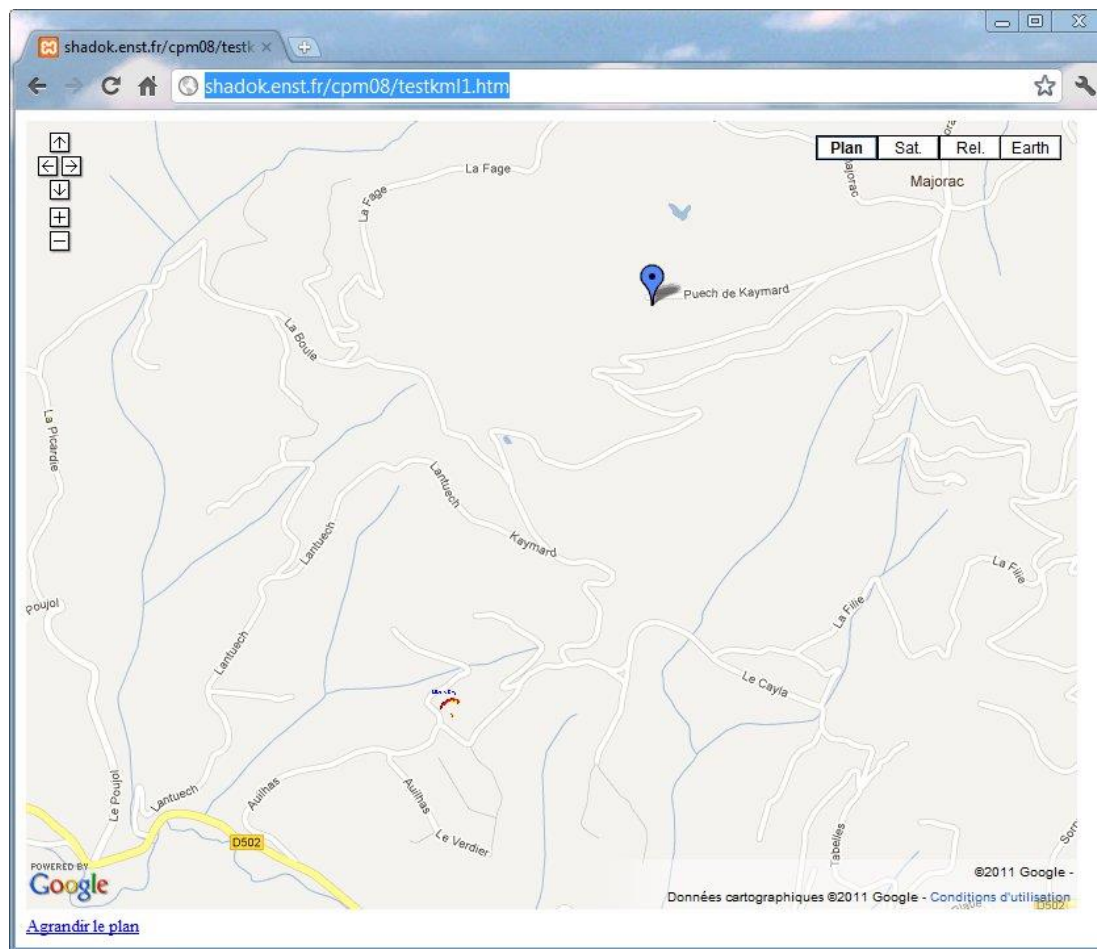
KML



```
<kml xmlns="http://earth.google.com/kml/2.2">
  <document>
    <name>ConsignesAtterrissageSaintNazaireLesEymes</name>
    <description><![CDATA[Atterrissage de Saint-Nazaire-Les-
Eymes
Consignes impératives à respecter pour la pérennité du
site]]></description>
    <Style id="style2">
      <LineStyle>
        <color>73FF0000</color>
        <width>5</width>
      </LineStyle>
    </Style>
    <Placemark>
      <name>Lignes électriques</name>
      <description><![CDATA[]]></description>
      <styleUrl>#style2</styleUrl>
      <ExtendedData>
        <Data name="_SnapToRoads">
          <value>true</value>
        </Data>
      </ExtendedData>
      <LineString>
        <tessellate>1</tessellate>
        <coordinates>
          5.841270,45.265209,0.000000
          5.840580,45.265759,0.000000
          5.840580,45.265759,0.000000
          5.840750,45.266411,0.000000
        </coordinates>
      </LineString>
```



Cas d'usage: une carte dans un site (1)





Cas d'usage: une carte dans un site (2)

```
<kml xmlns="http://earth.google.com/kml/2.2">
```

```
  <Document>
```

<http://shadok.enst.fr/cpm08/Test.kml>

```
    <name>Le Kaymard</name>
```

```
    <description><![CDATA[ma belle description]]></description>
```

```
    <Style id="style1"> <IconStyle> <Icon>
```

```
      <href>http://paraveyron.free.fr/site/images/akeymard.png</href>
```

```
    </Icon> </IconStyle> </Style>
```

```
    <Placemark>
```

```
      <name>Deco du Kaymard</name>
```

```
      <description><![CDATA[]]></description>
```

```
      <styleUrl>#style2</styleUrl>
```

```
      <Point>
```

```
        <coordinates>2.478168,44.546211,0.000000</coordinates>
```

```
      </Point>
```

```
    </Placemark>
```

```
    <Placemark>
```

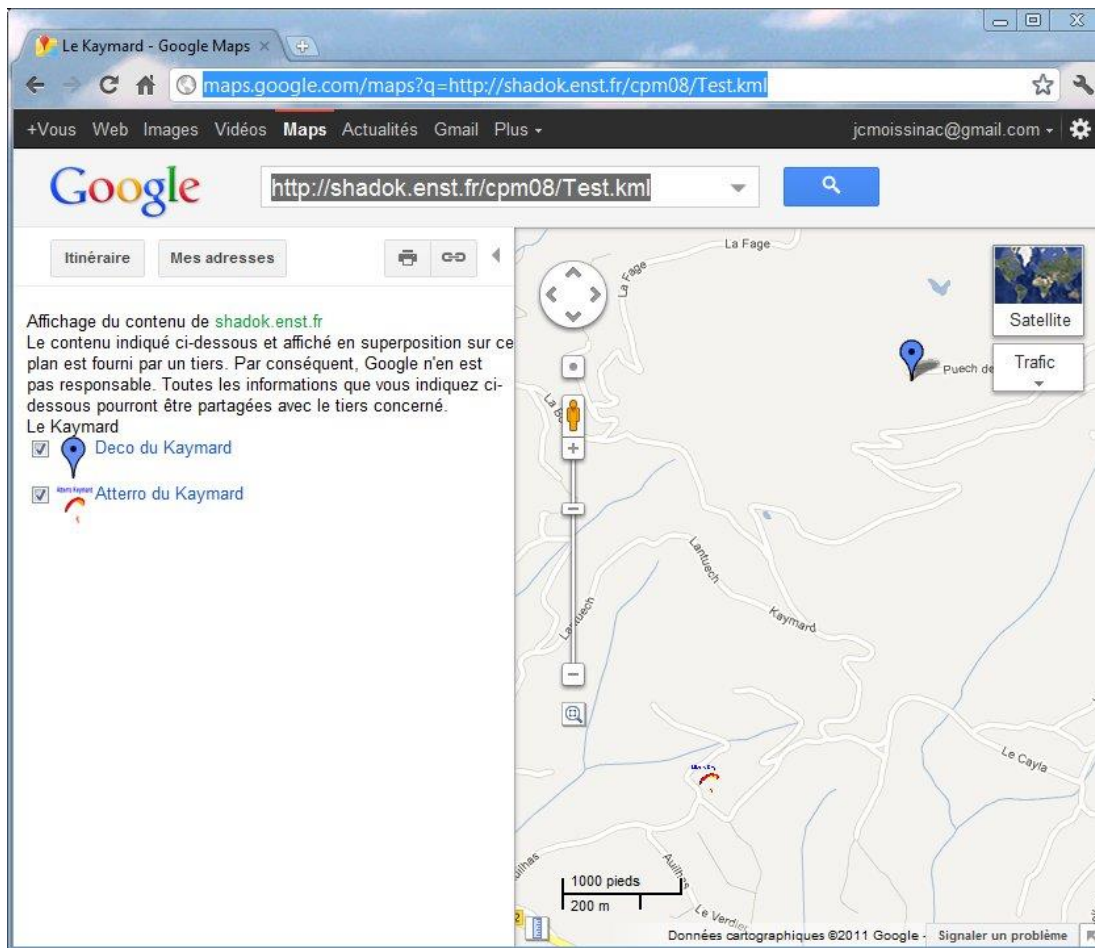
```
      <name>Atterro du Kaymard</name>
```

```
    ...
```




Cas d'usage: une carte dans un site (3)

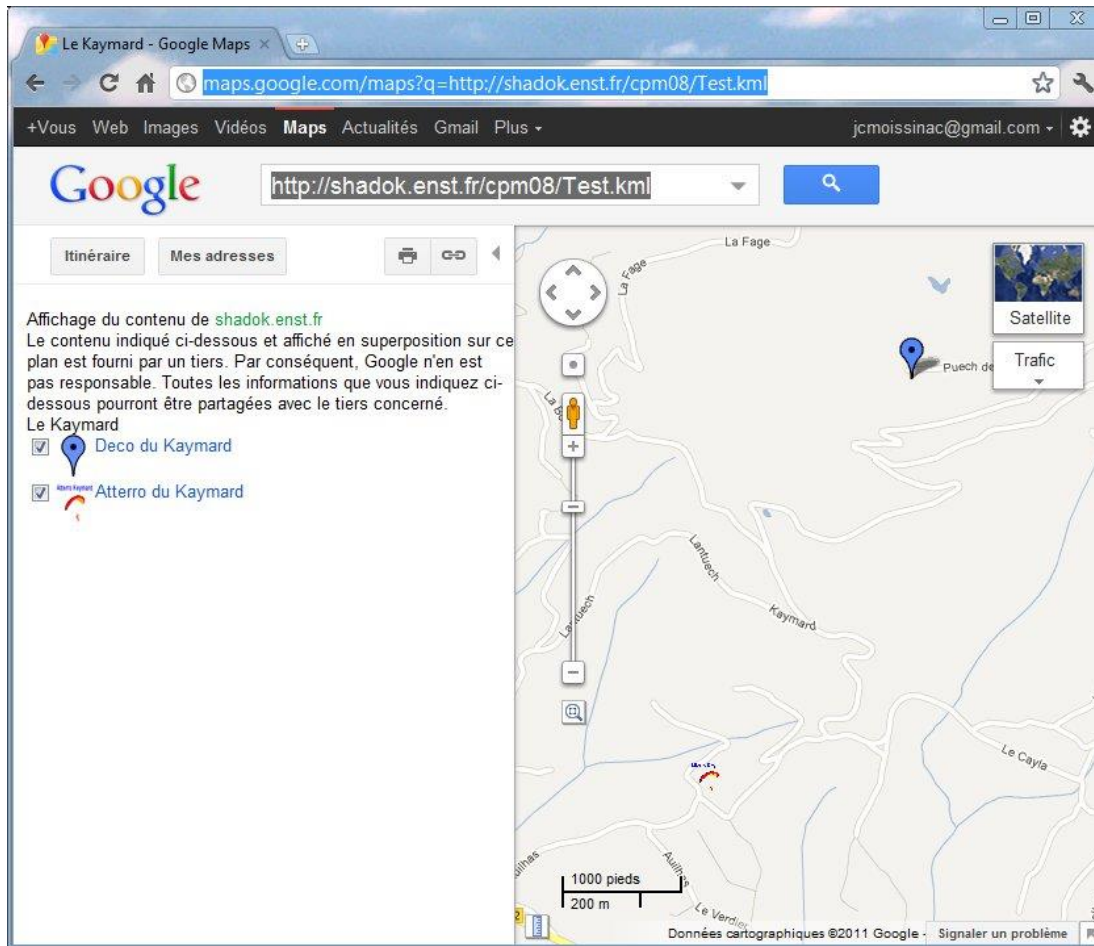
<http://maps.google.com/maps?q=http://shadok.enst.fr/cpm08/Test.kml>





Cas d'usage: une carte dans un site (3)

<http://maps.google.com/maps?q=http://shadok.enst.fr/cpm08/Test.kml>





Cas d'usage: une carte dans un site (3)

```
<html>
```

```
<body>
```

```
<h1>Ceci est ma super carte:</h1>
```

```
<iframe width="800" height="600" frameborder="0" scrolling="no"
```

```
  marginheight="0" marginwidth="0"
```

```
src="http://maps.google.com/maps?q=http:%2F%2Fshadok.enst.fr%2Fcpm08%2FTes  
.kml&output=embed">
```

```
</iframe>
```

```
<br />
```

```
<small>
```

```
<a
```

```
href="http://maps.google.com/maps?q=http:%2F%2Fshadok.enst.fr%2Fcpm08%2FTes  
t.kml">Agrandir le plan</a>
```

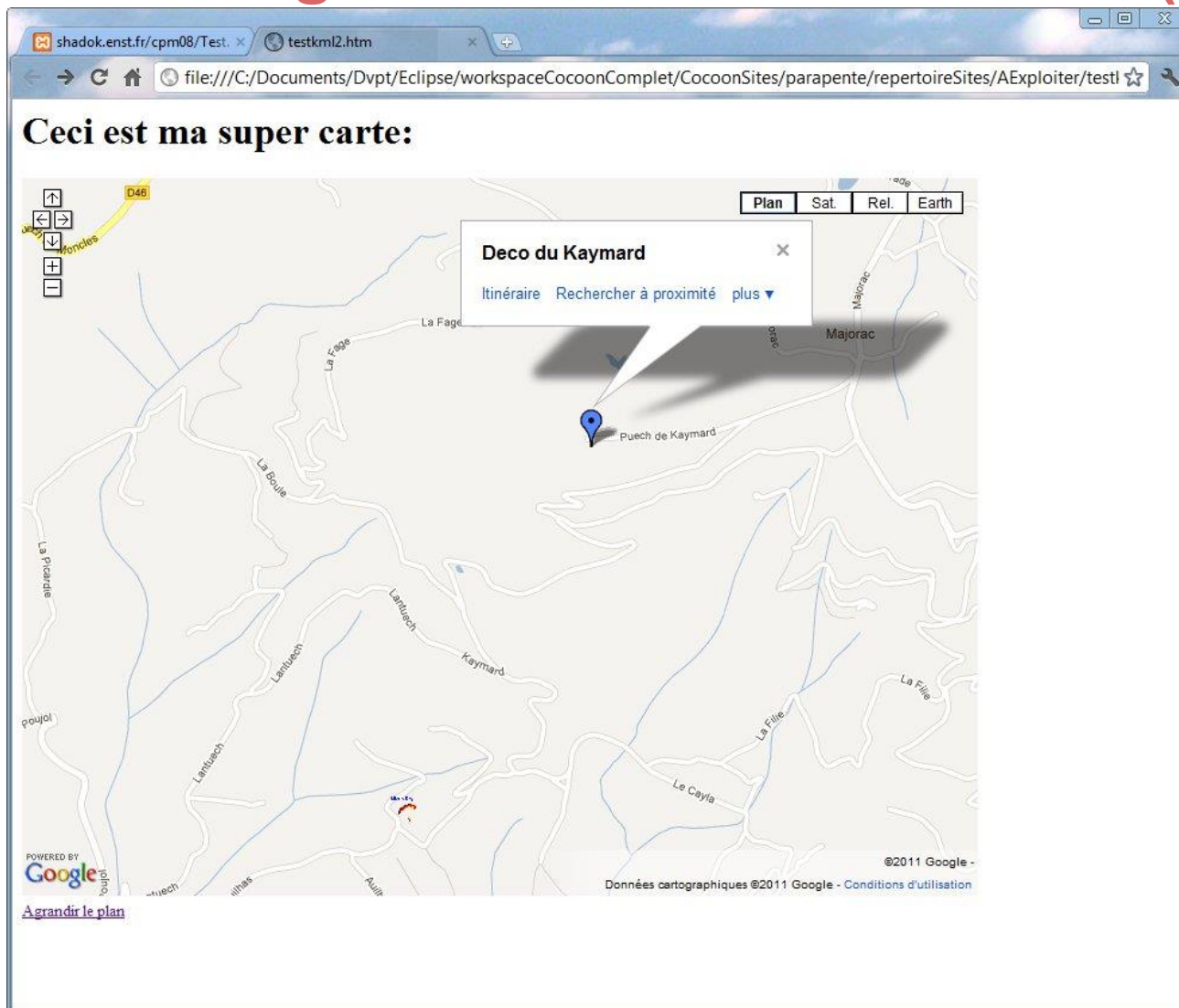
```
</small>
```

```
</body>
```

```
</html>
```



Cas d'usage: une carte dans un site (3)





Qu'est-ce que XML?

- Une syntaxe unique...
- ... pour tout représenter!



Exemple

- **<book>**
 - **<chapter>**
 - **<title>Introduction</title>**
 - **</chapter>**
 - **<chapter>**
 - **<title>Récit</title>**
 - **<subChapter>**
 - **<title>Partie 1</title>**
 - **</subChapter>**
 - **<subChapter>**
 - **<title>Partie 2</title>**
 - **</subChapter>**
 - **</chapter>**
 - **<chapter>**
 - **<title>Index</title>**
 - **</chapter>**
- **</book>**



XML: eXtensible Markup Language

- XML est un système générique de représentation semi-structurée de données
- XML propose une syntaxe pour cette structuration
- XML est basé sur un principe de balisage des données
- XML a été défini par le W3C pour disposer d'un modèle de partage de données sur le Web
- XML propose un cadre commun à tous les types de documents
- XML se décline en grammaires spécialisées
 - XHTML, MathML, ...
- XML autorise les documents composés à l'aide de plusieurs grammaires
- XML et son environnement de programmation et d'outils est adapté aux transformations entre grammaires XML ou vers d'autres formats





Représentation semi-structurée de données

<dvds>

<dvd vu="oui">

<titre>Jules et Jim</titre>

<realisateur>F.Trufaut</realisateur>

</dvd>

<dvd vu="oui">

<titre>Plein Soleil</titre>

<acteur>Alain Delon</acteur>

</dvd>

<dvd vu="oui">

<titre>La vie de Brian</titre>

<realisateur>Terry Gilliam</realisateur>

</dvd>

...

</dvds>

<dvd vu="oui">

<titre>Jules et Jim</titre>

<realisateur>F.Trufaut</realisateur>

<comment>

<p>Un de mes films préférés

</comment>

</dvd>



Balisage des données

Jules et Jim

F.Truffaut

Vu

-

Plein Soleil

Vu

Alain Delon

-

La vie de Brian

...

<dvds>

<dvd vu="oui">

<titre>Jules et Jim</titre>

<realisateur>F.Truffaut</realisateur>

</dvd>

<dvd vu="oui">

<titre>Plein Soleil</titre>

<acteur>Alain Delon</acteur>

</dvd>

<dvd vu="oui">

<realisateur>Terry Gilliam</realisateur>

<titre>La vie de Brian</titre>

</dvd>

...

</dvds>

- Interfaces logicielles de manipulation
- Outils d'analyse
- Outils de vérification
- Langages de transformation
- Outils de compression
- Moyen structuré de lecture de données



Exemple de document XML (1)

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0">
  <channel>
    <title>Website Feed</title>
    <description>Website Feed coded manually</description>
    <link>http://www.yourdomain.com</link>
    <item>
      <title>Sale this week only</title>
      <description>All household products are 50%...</description>
      <link>http://www.yourdomain.com/sales.htm</link>
    </item>
    <item>
      <title>A Special Event</title>
      <description>A Special Teleconference for ...</description>
      <link>http://www.yourdomain.com/events.htm</link>
    </item>
    <item>
      <title>Announcing new Products</title>
      <description>Announcing a new line of products</description>
      <link>http://www.yourdomain.com/products.htm</link>
    </item>
  </channel>
</rss>
```




Exemple de document XML (2)

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<rss version="2.0">
```

```
  <channel>
```

```
    <title>Website Feed</title>
```

```
    <description>Website Feed coded manually</description>
```

```
    <link>http://www.yourdomain.com</link>
```

```
    <item>
```

```
    ...
```

```
    </item>
```

```
    <item>
```

```
    ...
```

```
    </item>
```

```
    <item>
```

```
    ...
```

```
    </item>
```

```
  </channel>
```

```
</rss>
```



Exemple de document XML (3)

`<item>`

`<title>Sale this week only</title>`

`<description>All household products are 50%...</description>`

`<link>http://www.yourdomain.com/sales.htm</link>`

`</item>`



Document XML

- Un document XML est un arbre
- Les nœuds de l'arbre sont des **ELEMENTS** qui peuvent être qualifiés par des attributs

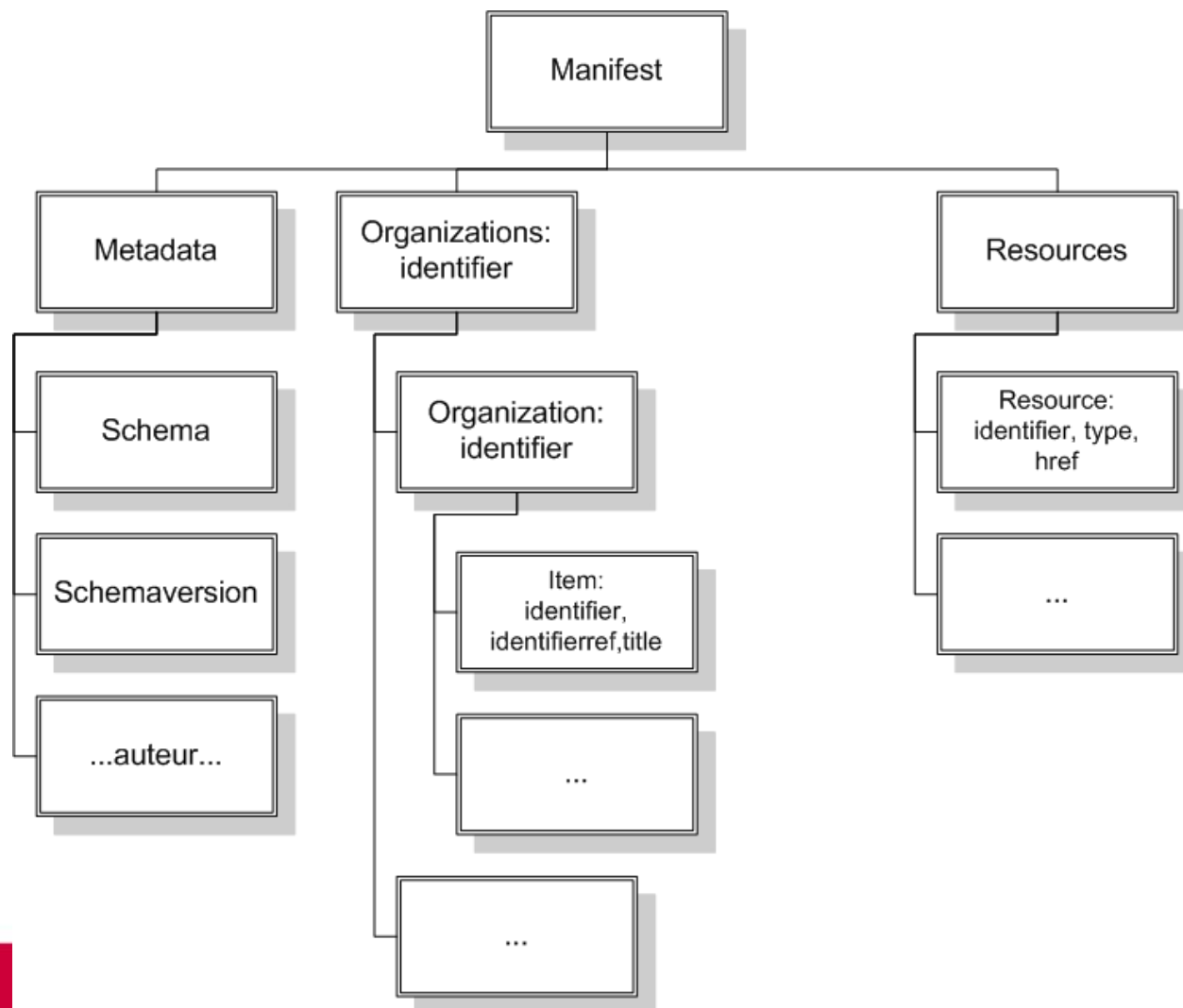


Exemple de document XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<manifest>
  <metadata>  <schema>IMS Content</schema>
               <schemaversion>1.1</schemaversion>
  ... </metadata>
  <organizations>
    <organization identifier="SESSION">
      <item identifier="SLIDE1_1" identifierref="REF_SLIDE1_1"
        title="Cryp..."><datetime>0</datetime></item>
      <item identifier="SLIDE2_1" identifierref="REF_SLIDE2_1" title="Plan de l'exposé">
        ...
      </organization>
    </organizations>
    <resources>
      <resource identifier="REF_SLIDE1_1" type="webcontent" href="jpg\Slide1_1.htm">
        <file href="jpg\Slide1_1.htm" />
        <file href="jpg\Slide1_1.jpg" />
      </resource>
      <resource identifier="REF_SLIDE2_1" type="webcontent" href="jpg\Slide2_1.htm">
        ...
      </resource>
    </resources>
  </manifest>
```



Exemple d'arbre XML





Atouts du XML

- **Une mise en œuvre légère possible**
- **Une grande souplesse d'adaptation et d'évolution**
 - Facilite la compatibilité arrière et avant
- **Naturellement multi-plateforme**
 - Format texte
- **Informations sémantiques sur le document et possibilités de recherche**
 - Le document XML est généralement composé de données encadrées par des balises qui décrivent ces données
- **Relative simplicité de passage d'une grammaire à une autre**
 - Passer des données à une présentation des données (ex: HTML)
 - Passer d'une version de fichier à une autre



Documents bien formés

Balisage correct:

- **Toute balise ouvrante est associée à exactement une balise fermante de même nom**
- **Toute balise ouverte dans un contexte doit être fermée dans ce même contexte**
 - Si on trouve la balise ouvrante `<B>` comme 'fille' de la balise `<A>` alors on doit trouver la balise fermante `</B>` comme 'fille' de la même balise `<A>`



Éléments et attributs

- Les éléments sont représentés par les balises
- Les attributs précisent un élément et apparaissent dans la balise ouvrante
 - `<item identifier="SLIDE1_1" identifierref="REF_SLIDE1_1" title="Cryptographie, certification / PKI, et services sécurisés ">`
 - Élément item
 - Attributs: identifier, identifierref, title
 - Un attribut est défini par un couple (nom, valeur) décrit par la syntaxe `nom=« valeur »`

- Tous les éléments de tous langage XML peuvent avoir un ID
- Les IDs présents dans un document doivent être uniques
- L'ID permet de désigner un élément de façon simple et non ambiguë
- getElementbyID est une fonction de l'API standard DOM pour accéder à un élément par son ID



Grammaires XML pour le multimédia

- **XHTML**
- **SVG -> Scalable Vector Graphics**
- **SMIL**
 - Synchronized Multimedia Integration Language
- **HTML+TIME -> HTML+déroulement temporel**
- **XMT -> Représentation XML de MPEG-4 BIFS**
- **InkML -> annotations de documents**
- **MPEG-7, MPEG-21**





Grammaires associées

Entre autres:

■ **IMS/Scorm**

- Représentation de ressources pédagogiques

■ **Trans-13**

- Représentation de transcriptions

■ **MathML**

- Représentation de formules mathématiques

■ **MusicML**

- Représentation de partitions

■ **XUL, XAML...**

- Représentations d'interfaces utilisateur

■
.....

■ SOAP

- Communication de messages entre applications

■ RDF

- Méta-données

■ XSLT, XSL-FO

- Transformations de documents XML



Les langages d'interface

- UIML
- XUL -> Mozilla/Firefox
- XAML -> Microsoft
- XML Layouts pour Android



XML LayoutsAndroid

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

```
<windows Width="600" Height="480" Text="Mon Programme">  
  <FlowPanel>  
    <Label Name="Montexte" FontSize="20"> Mon application </label>  
    <Button Width="80" Click="BoutonClic"> Fermer </Button>  
  </FlowPanel>  
</windows>
```

```
<Canvas ID="root" xmlns="http://schemas.microsoft.com/2003/xaml"  
  xmlns:def="Definition">  
  <TextBlock>Hello World!</TextBlock>  
</Canvas>
```

```
<window  
xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">  
  <box>  
    <description>Hello world!</description>  
  </box>  
</window>
```



getElementbyID exemple

```
<?php
```

```
$doc = new DomDocument;
```

```
// Nous devons valider notre document avant de se référer à l'ID
```

```
$doc->validateOnParse = true;
```

```
$doc->Load('book.xml');
```

```
echo "L'élément dont l'ID est 'book' est :
```

```
" . $doc->getElementById('books')->tagName . "\n";
```

```
?>
```



Création d'un 'langage'

Conseils

- Les éléments donnent une information sur la structure du document
- Les attributs donnent des précisions sur un élément
- Les éléments sont plus faciles à manipuler (parcours de l'arbre...)
 - => tendance à affiner une description par ajout d'éléments
 - => alourdissement du document





Création d'un 'langage' (2)

<personne>

<nom>Martin</nom>

<prenom> Pascal </prenom>

<ville>Paris</ville>

...

</personne>

■ Versus

**<personne nom=« Martin » prenom=« Pascal »
ville=« Paris » />**



Définition d'un 'langage' (1)

- **XML peut être qualifié de meta-langage**
 - Un langage pour définir des langages
- **XML permet de définir son propre ensemble de balises et leurs attributs**
- **Des ensembles de balises ont été standardisés**
 - Par divers organismes
 - Pour diverses applications
- **Un 'schéma' ou une 'DTD' peut servir à définir un langage et à valider les documents écrits dans ce langage**
- **Un document peut être typé en lui associant un schéma ou une DTD, mais ce n'est pas obligatoire**



Définition d'un 'langage' (2)

- **Un langage peut être défini pour représenter un ensemble semi-structuré de données**
 - On parle quelquefois de 'grammaire' au lieu de 'langage'
- **Il peut être défini**
 - Syntaxiquement par
 - un schéma exprimé en XML
 - ou une DTD (représentation non-XML)
 - Sémantiquement par un document descriptif de la sémantique des balises (éléments) et attributs utilisés



Définition d'un 'langage' (3)

Un schéma ou une DTD permettent de définir plus ou moins précisément

■ Les balises qui composent le langage

- Leur nom
- Leur nombre (0 à N ou exactement N ou ...)
- Leur relation aux autres balises (fille de..., sœur de ...)
- Le contenu autorisé de ces balises

■ Les attributs que peuvent avoir ces balises

- Leur nom
- Les valeurs possibles de ces attributs



Document XML valide

- **Un document XML est dit valide si**
 - Il est bien formé
 - Il est conforme à son type s'il en a un (cad à son schéma ou à sa DTD)





Exemples extraits de la DTD SVG (1)

- **<!ELEMENT rect**
(%descTitleMetadata;,(animate|set|animateMotion|animateColor|animateTransform
■ **%geExt;%rectExt;)*) >**
- **<!ATTLIST rect**
■ **%stdAttrs;**
■ **%testAttrs;**
■ **%langSpaceAttrs;**
■ **externalResourcesRequired %Boolean; #IMPLIED**
■ **class %ClassList; #IMPLIED**
■ **style %StyleSheet; #IMPLIED**
■ **%PresentationAttributes-Color;**
■ **%PresentationAttributes-FillStroke;**
■ **%PresentationAttributes-Graphics;**
■ **transform %TransformList; #IMPLIED**
■ **%graphicsElementEvents;**
■ **x %Coordinate; #IMPLIED**
■ **y %Coordinate; #IMPLIED**
■ **width %Length; #REQUIRED**
■ **height %Length; #REQUIRED**
■ **rx %Length; #IMPLIED**
■ **ry %Length; #IMPLIED**
■ **>**



Exemples extraits de la DTD SVG (2)

Par ailleurs, on trouve

```
<!ENTITY % Boolean "(false | true)">
```

...

```
<!ENTITY % Coordinate "CDATA">
```

...

```
■ <!ENTITY % stdAttrs "id ID #IMPLIED  
■ xml:base %URI; #IMPLIED">
```

Pour en savoir plus sur les DTD:

<http://www.mutu-xml.org/xml-base/slides/PUB-PRESSHEMA-FR.html>

Extrait d'une description sémantique d'une grammaire

■ 9.2 The 'rect' element

- The 'rect' element defines a rectangle which is axis-aligned with the current user coordinate system. Rounded rectangles can be achieved by setting appropriate values for attributes rx and ry.

■ *Attribute definitions:*

■ x = "<coordinate>"

- The x-axis coordinate of the side of the rectangle which has the smaller x-axis coordinate value in the current user coordinate system.
If the attribute is not specified, the effect is as if a value of "0" were specified.
Animatable: yes.

■ y = "<coordinate>"

- The y-axis coordinate of the side of the rectangle which has the smaller y-axis coordinate value in the current user coordinate system.
If the attribute is not specified, the effect is as if a value of "0" were specified.
Animatable: yes.

■ width = "<length>"

- The width of the rectangle.
A negative value is an error (see Error processing). A value of zero disables rendering of the element.
Animatable: yes.

■ height = "<length>"

- The height of the rectangle.
A negative value is an error (see Error processing). A value of zero disables rendering of the element.
Animatable: yes.

Principalement

- **Syntaxe spécifique (pas XML)**
- **Faiblesse du typage**
 - Pas de mécanismes précis du genre 'valeur entière entre 0 et 15'

Conseil

Comprendre un minimum de choses car certains types de documents ne sont définis que par une DTD



Introduction des schémas

Définition dans une syntaxe XML

- De la structure d'arbre d'objets typés et valués
- Avec expression de contraintes fortes
- Possibilités de définitions partielles
- Possibilités d'inclusion de définitions externes
- Prise en compte des espaces de nom
 - Pour faciliter le mariage de documents respectant plusieurs schémas

Plusieurs grammaires proposées

RelaxNG semble très utilisée

<http://www.oasis-open.org/committees/relax-ng/tutorial.html>



Extrait du schéma RNG SVG (1)

```
<element name="rect">
  <zeroOrMore>
    <ref name="SVG.Description.class"/>
  </zeroOrMore>
  ...
  <optional>
    <attribute name="x">
      <ref name="SVG.Coordinate.datatype"/>
    </attribute>
  </optional>
  <attribute name="width">
    <ref name="SVG.Length.datatype"/>
  </attribute>
  ...
</element>
```




Domaines nominaux: namespaces

- Espaces de noms
- Permet de préciser à quel type se réfère un élément
- But: éviter les collisions de noms, lorsqu'un document est composé d'éléments de plusieurs grammaires

- Exemple:

```
<monelement xmlns:xmlamoi="http://xmlamoi.org/DTDProjet/1.0">  
  <p>un paragraphe ... de l'espace de noms par défaut</p>  
  <xmlamoi:p>un paragraphe ... de l'espace de noms xmlamoi</p>  
</monelement>
```

- <http://www.w3.org/TR/REC-xml-names/>

■ Web Services

- Des services rendus via le web qui fournissent une réponse à une requête

■ Exemple: geonames

- Des services liés à la géographie et aux adresses



Exemple de service Geonames

<http://api.geonames.org/findNearByWeatherXML?lat=48.8&lng=2.33&username=demo>

```
<geonames>
<observation>
<observation>LFPO 131100Z 23018G28KT 6000 -RA SCT012 BKN016 11/10 Q1000 TEMPO 4000 RA
BKN012</observation>
<observationTime>2011-12-13 11:00:00</observationTime>
<stationName>Paris-Orly</stationName>
<ICAO>LFPO</ICAO>
<countryCode>FR</countryCode>
<elevation>96</elevation>
<lat>48.71666666666667</lat><lng>2.3833333333333333</lng>
<temperature>11</temperature>
<dewPoint>10</dewPoint>
<humidity>93</humidity>
<clouds>scattered clouds</clouds>
<weatherCondition>n/a</weatherCondition>
<hectoPascAltimeter>1000.0</hectoPascAltimeter>
<windDirection>230.0</windDirection>
<windSpeed>18</windSpeed>
</observation>
</geonames>
```



Exemple de service Geonames (2)

<http://api.geonames.org/srtm3?lat=50.01&lng=10.2&username=demo>

Donne

206

<http://api.geonames.org/findNearbyPlaceName?lat=48.8&lng=2.33&username=demo>

Donne (extrait)

<name>Arcueil</name>

<lat>48.79993</lat>

<lng>2.33256</lng>

<countryName>France</countryName>



L'exemple de Cocoon

■ Producteur de données

- Fournit un flux (XML) de données 'matière première'

■ Transformateur de données

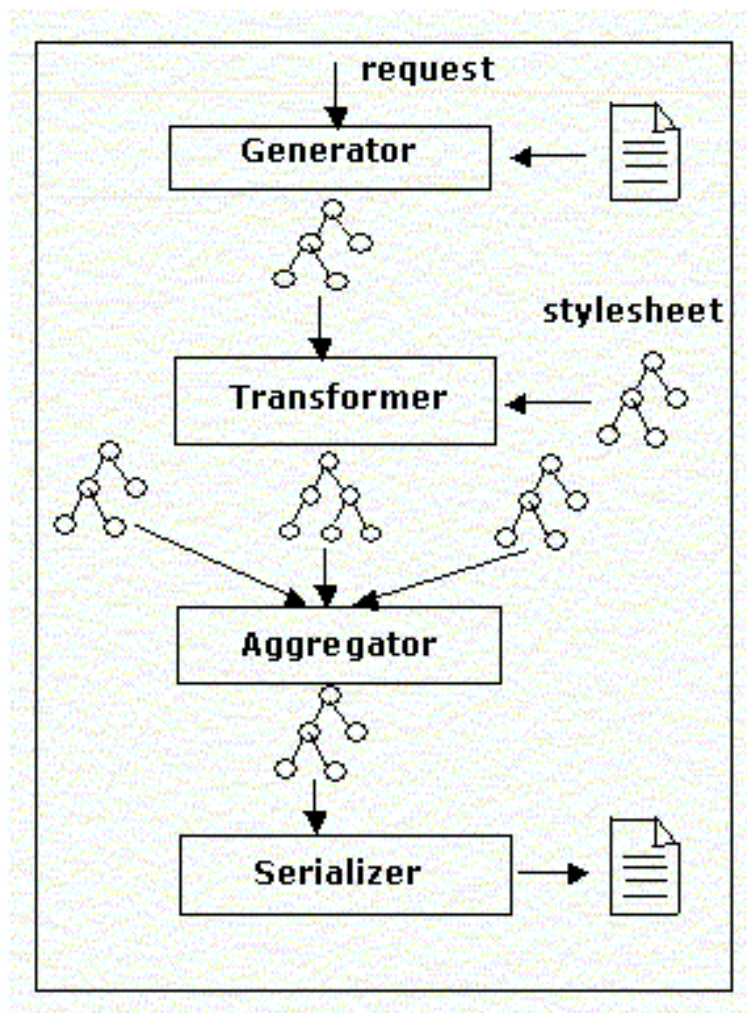
- Produit le flux (XML) de présentation

■ Sérialiseur

- Envoie le flux produit

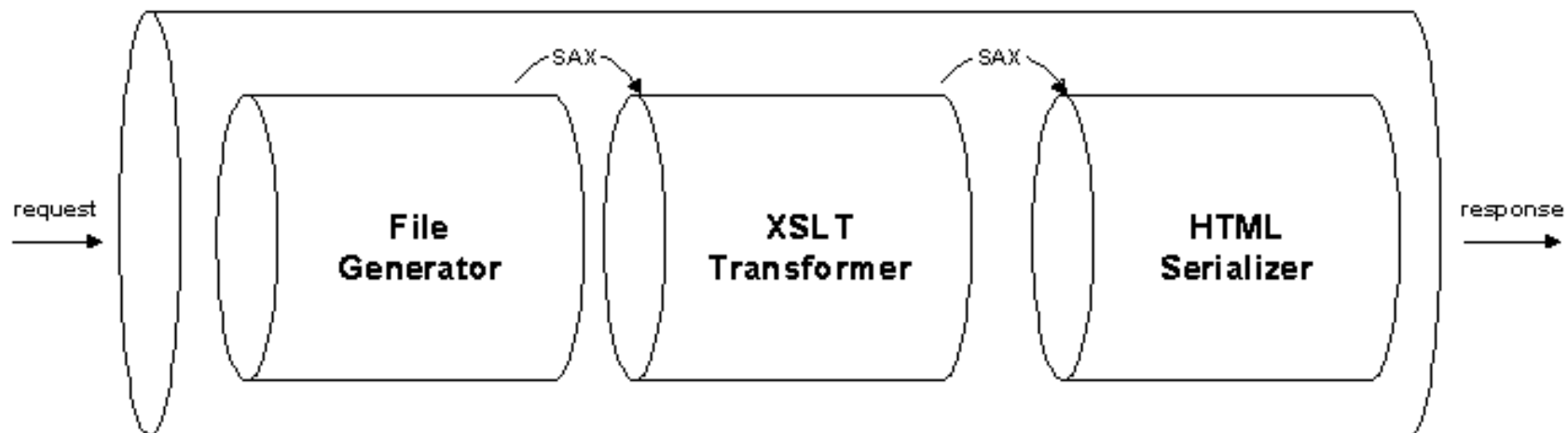


Pipeline Cocoon: principe





Pipeline Cocoon: exemple

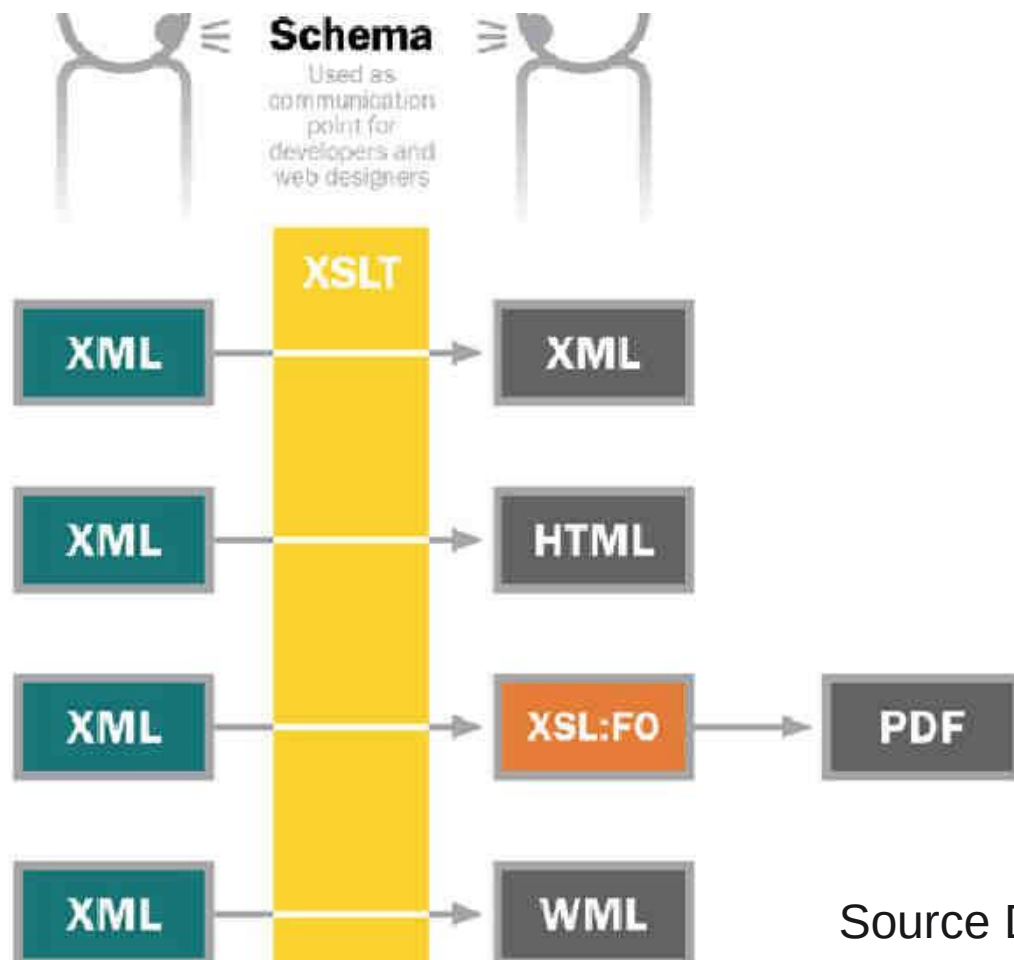




Transformations XSLT

- **Permet de transformer des données XML en un nouveau texte**
- **Par exemple:**
 - D'un fichier XML en un fichier HTML
 - D'un fichier dans une grammaire XML en un fichier dans une autre grammaire XML
 - D'un fichier XML en un fichier texte brut
- **XSLT est un langage de programmation de ces transformations**
- **Conçu pour séparer clairement les données de la présentation**

XSLT: possibilités



Source D.O.Tech

- **Langage basé sur des règles appliquées à des portions du document d'origine**
- **Templates:**
 - Définissent des transformations de portions de documents XML
- **XPATH**
 - Langage pour désigner une portion de document XML

■ XPATH

- www.w3.org/TR/xpath

■ XSLT

- www.w3.org/TR/xslt

■ Sites

- www.devguru.com/Technologies/xslt/quickref/xslt_index.html
- Site de patterns, site xml.org
- <http://wiki.cocoondev.org/downloads/XSLT.pdf>

■ Livres

- patterns

- Une transformation exprimée en langage XSLT
- Contient des règles paramétrées (templates) qui définissent comment une partie de l'arbre source est transformée en une partie de l'arbre résultat
- Utilise
 - Des 'patterns' qui désignent des portions de la source
 - Des 'templates' qui définissent le résultat

Exemples XPATH

■ « / »

- Désigne la racine du document source

■ « . »

- Désigne le contenu du nœud courant

■ « .. »

- Désigne le parent du nœud courant

■ « ../@couleur »

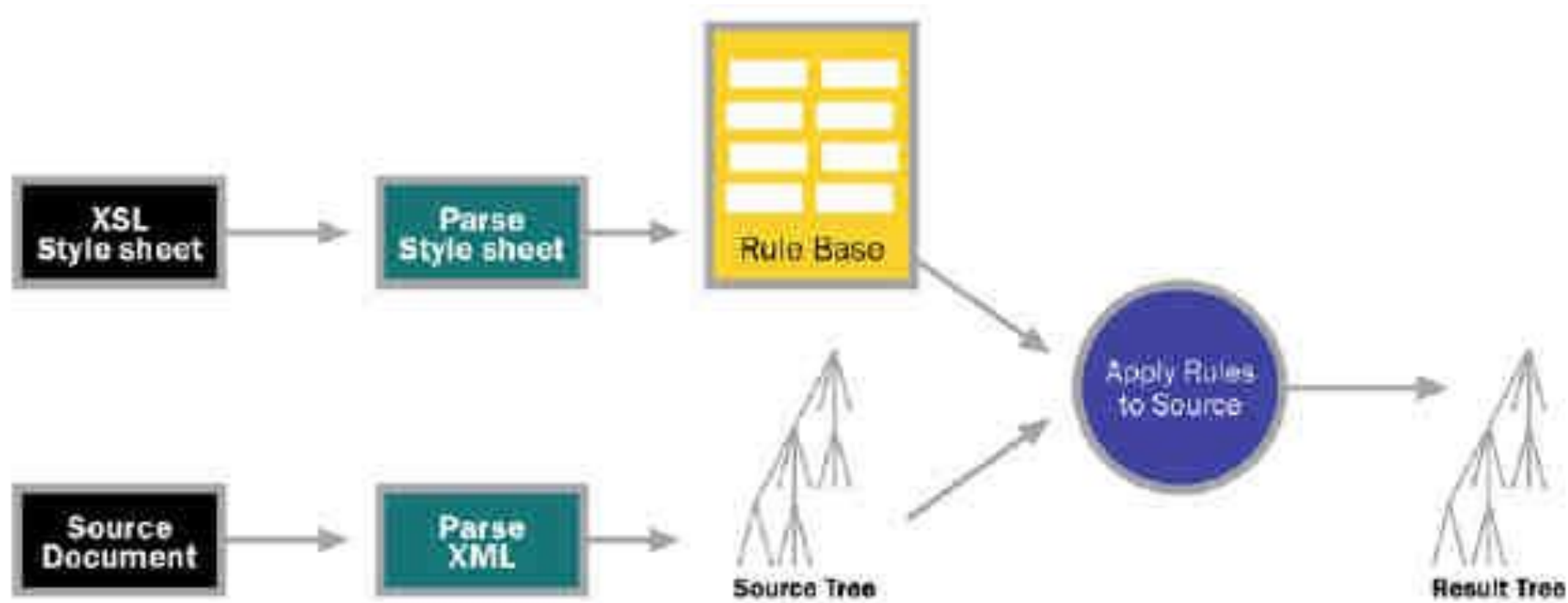
- Désigne l'attribut couleur du parent du nœud courant

■ « /svg/text »

- Désigne la liste des éléments text fils d'un nœud svg à partir de la racine



Modèle de traitement



Source D.O.Tech



Hello world (multimédia?)

- Source
- Transformation en texte
- Transformation en SVG (x2)
- Transformation en XHTML
- Transformation en HTML+TIME
- Traitement par un processeur XSLT
- Résultats



Fichier exemple/salut.xml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>  
<salut>  
    Bonjour à tous  
</salut>
```




Source vers html

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html"/>
<xsl:template match="/">
  <html>
    <body><xsl:apply-templates/></body>
  </html>
</xsl:template>
<xsl:template match="salut">
  <h1>
    <xsl:value-of select="."/>
  </h1>
</xsl:template>
</xsl:stylesheet>
```



Source vers SVG

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns="http://www.w3.org/2000/svg">
<xsl:output method="xml" indent="yes"/>
<xsl:template match="/">
  <svg width="100%" height="100%" viewBox="0 0 640
    480" » xmlns="http://www.w3.org/2000/svg">
    <xsl:apply-templates/>
  </svg>
</xsl:template>
<xsl:template match="salut">
  <text x="100" y="240">
    <xsl:value-of select="."/>
  </text>
</xsl:template>
</xsl:stylesheet>
```

Processeurs XSLT

■ Msxsl

- `Msxsl.exe salut.xml svg.xsl -o salut.svg`

■ Xalan

- `java org.apache.xalan.xslt.Process -in salut.xml -xsl
svg.xsl -out salut.svg`
- En PHP: `echo $xml -> XALAN_shell_transform(
'fichier.xsl' , 'fichier.xml');`

■ xt, xsltproc, saxon...

■ Navigateurs

- <http://shadok.enst.fr/cpm08/bouteille.xml>



Exemple xslt dans un navigateur

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="bouteille1.xsl"?>
<bouteille>
  <marque>Cristaline</marque>
  <composition>
    <ion_positif>calcium 71mg/l</ion_positif>
    <ion_negatif>nitrate 1mg/l</ion_negatif>
    <ion_positif>magnésium 5,5mg/l</ion_positif>
    <ion_negatif>chlorure 20mg/l</ion_negatif>
    <autres_materiaux>fer</autres_materiaux>
  </composition>
  <source>
    <ville>St-Cyr la Source</ville>
    <departement>Loiret</departement>
  </source>
  <code_barre>3274080005003</code_barre>
  <contenance>150cl</contenance>
  <ph>7,45</ph>
</bouteille>
```



Références XSLT

■ Livres

- XSLT Fondamental, Eyrolles, Philippe Drix
- XSLT, Wrox, Mickael Kay

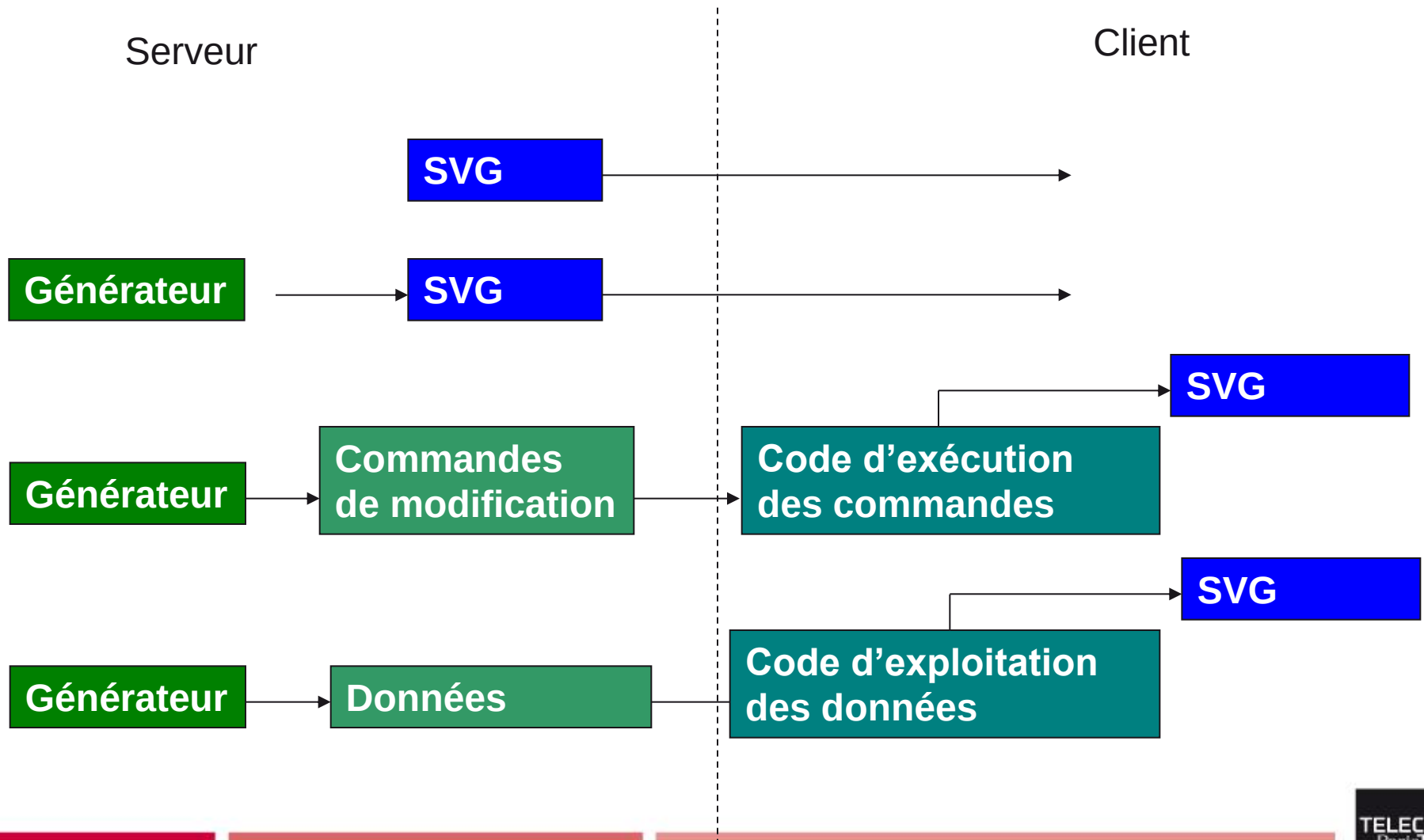
■ Sites

- Xmlfr.org

- **Oxygen XML Editor**
- **Eclipse Webtools**
- **Ainsi que dans les autres frameworks de développement**

- **SVG + script + XMLHttpRequest**
- **Modification d'une scène SVG initiale par du script**
- **Exemple d'implémentation:**
 - SVG+AJAX+REX
 - Voir <http://svgmpeg4.blogspot.com/2006/07/un-zeste-de-svg-ajax-et-rex.html>

Utilisations de SVG





Utilisations de SVG

■ Statique

■ Dynamique

- Création côté serveur en PHP, PERL, JSP...
- AJAX avec envoi de commandes
 - Voir <http://svgmpeg4.blogspot.com/2006/07/un-zeste-de-svg-ajax-et-rex.html>
- AJAX avec envoi de données

- **AJAX simple avec envoi de contenu SVG, page vide au départ**

<http://shadok.enst.fr/cpm08/minajax.htm>



Evolution permanente

Ce qui se passe dans le monde

■ Web & TV

- TV sur le Web et Web sur la TV

■ ePub3

- Livre et web, livre interactif, livre multimedia

■ HTML5

■ Web sémantique