

SÉRIALISATION

Laboratoire n°3
Christophe Peretti
Samuel Darcey

Table des matières

1 Introduction.....	2
2 Modification du fichier xml	2
3. Implémentation	2
3.1 Génération des pages.....	2
3.2 Templates de tri	3
3.3 Affichage du tableau	4
3.4 Fichier de film	4
3.5 Css.....	6
4. Captures	7
.....	10
5 Conclusion	10

1 Introduction

Le laboratoire SER – Plex consistant à simuler les activités d'un complexe cinématographique, est constitué de 4 laboratoires constituant les différentes parties du projet. Le présent document porte sur le troisième laboratoire : Construction de fichier html avec xsl, et utilisation de css.

Dans ce laboratoire, il s'agissait concevoir un fichier xsl, qui prendrait le fichier xml généré dans le deuxième laboratoire, et qui en générerait un ou plusieurs fichiers html, afin d'avoir des pages web représentant le cinéma.

2 Modification du fichier xml

Dans le précédent laboratoire, nous avons décidé d'épurer le xml afin de ne garder que les quelques champs les plus utiles. Au final, il nous manquait plusieurs informations essentielles qui devraient figurer sur un site web de cinéma, comme les notes et les critiques. Dans ce laboratoire, nous avons commencé par modifier notre fichier java de génération du xml, afin que celui-ci comporte toutes les balises demandées.

3. Implémentation

3.1 Génération des pages

Pour ce laboratoire, nous avons décidé de générer une seule page principale ayant les projections de film triées par date, titre et note moyenne. Pour les descriptions de film, nous générons une page par film.

Sur la capture suivante, ce trouve en premier la génération du fichier index.html, qui appelle les trois templates permettant de sélectionner les projections par date, titre ou note (ces templates se trouvent plus bas dans le code). Nous avons décidé de mettre ces trois formes de tri sur une seule page, afin de ne pas multiplier le nombre de page html générée. Une autre possibilité aurait été, grâce à du javascript, de générer du contenu dynamique en fonction du choix de tri de l'utilisateur.

```
<!-- Generation du fichier index.html -->
<xsl:result-document method="html" href="index.html" >
  <html>
    <head>
      <xsl:call-template name="entete_html"/>
    </head>
    <body>
      <a name="home"/>
      <h1>Liste des projections</h1>
      <xsl:apply-templates select="cinema" mode="organisee_par_date"/>
      <xsl:apply-templates select="cinema" mode="organisee_par_titre"/>
      <xsl:apply-templates select="cinema" mode="organisee_par_note"/>
      <footer>
      </footer>
    </body>
  </html>
</xsl:result-document>

<!-- Generation des fichiers des films -->
<xsl:for-each select="//film">
  <xsl:variable name="titre" select="titre" />
  <xsl:result-document method="html" href="film_{$titre}.html" >
    <html>
      <head>
        <xsl:call-template name="entete_html"/>
      </head>
      <body>
        <xsl:call-template name="menu" />
        <xsl:apply-templates select="." mode="detail"/>
        <footer>
        </footer>
      </body>
    </html>
  </xsl:result-document>
</xsl:for-each>
</xsl:template>
```

Dans la deuxième partie, nous pouvons voir la génération des pages html des films, ayant le titre de celui-ci dans le nom du fichier. Ces fichiers comprennent l'ensemble des informations d'un film.

3.2 Templates de tri

Voici les trois templates de tri des projections. Nous pouvons voir qu'ils sont presque identiques, les seules différences sont le titre et la méthode de tri. Dans les trois cas nous appliquons le template qui s'applique sur film en mode tableau.

```
<xsl:template match="cinema" mode="organisee_par_date">
  <h2>Liste organisée par date</h2>
  <table class="index" cellspacing="10">
    <xsl:call-template name="entete_tableau"/>
    <tbody>
      <xsl:for-each select="//projection">
        <xsl:sort order="ascending" select="date"/>
        <xsl:sort order="ascending" select="film/titre"/>
        <tr>
          <xsl:apply-templates select="film" mode="tableau"/>
        </tr>
      </xsl:for-each>
    </tbody>
  </table>
</xsl:template>

<xsl:template match="cinema" mode="organisee_par_titre">
  <h2>Liste organisée par titre</h2>
  <table class="index" cellspacing="10">
    <xsl:call-template name="entete_tableau"/>
    <tbody>
      <xsl:for-each select="//projection">
        <xsl:sort order="ascending" select="film/titre"/>
        <tr>
          <xsl:apply-templates select="film" mode="tableau"/>
        </tr>
      </xsl:for-each>
    </tbody>
  </table>
</xsl:template>

<xsl:template match="cinema" mode="organisee_par_note">
  <h2>Liste organisée par note moyenne</h2>
  <table class="index" cellspacing="10">
    <xsl:call-template name="entete_tableau"/>
    <tbody>
      <xsl:for-each select="//projection">
        <xsl:sort order="descending" select="film/noteMoyenne"/>
        <tr>
          <xsl:apply-templates select="film" mode="tableau"/>
        </tr>
      </xsl:for-each>
    </tbody>
  </table>
</xsl:template>
```

3.3 Affichage du tableau

Dans l’affichage des tableaux de projections, nous allons simplement chercher dans le xml les dates de projection, le numéro de la salle du cinéma, le titre du film correspondant ainsi que sa durée et la note moyenne des critiques.

```
<xsl:template match="film" mode="tableau">
  <td><xsl:value-of select="../date"/></td>
  <td><xsl:value-of select="../numeroSalle"/></td>
  <xsl:variable name="titre_film" select="titre" />
  <td><a href="film_{$titre_film}.html"><xsl:value-of select="titre"/></a></td>
  <td><xsl:value-of select="duree"/></td>
  <td><xsl:value-of select="noteMoyenne"/></td>
  <br/><br/>
</xsl:template>
```

L’entête des tableaux était appelé avant le parcours des projections, dans les templates précédents. Nous pouvons voir que le titre du film est encadré d’une balise <a>, qui permet de créer un lien hypertexte. La référence de ce lien est le fichier html du film en question. Il est donc possible d’accéder à ce fichier en cliquant directement sur le film présent dans une projection.

3.4 Fichier de film

Les fichiers des films contiennent presque toutes les infos contenues dans le xml. En haut de la page, un lien pour retourner sur la page d’accueil est disponible. L’image de chaque film est redimensionnée grâce au css. Pour la liste des acteurs, nous affichons à chaque fois l’acteur et son rôle correspondant dans le film. Pour voir les infos supplémentaires, il suffit de survoler le nom d’un acteur. Dans notre fichier xml, pour chaque film nous avons la liste des rôles, avec l’id de l’acteur. La liste des acteurs se trouve dans la projection, et non dans le film. Il a donc fallu grâce au xsl, retrouver l’acteur correspondant à l’id présent dans le rôle.

```
<xsl:template match="film" mode="detail">
  <h1><xsl:value-of select="titre"/></h1>
  <xsl:variable name="url_img" select="image" />
  
  <h2>Durée</h2>
  <xsl:value-of select="duree" />min
  <h2>Synopsis</h2>
  <xsl:value-of select="synopsis" />
  <h2>Critiques</h2>
  <table>
    <thead>
      <th>Note</th>
      <th>Avis</th>
    </thead>
    <tbody>
      <xsl:for-each select="critiques/critique">
        <xsl:sort data-type="number" order="descending" select="note"/>
        <tr>
          <td><xsl:value-of select="note" /></td>
          <td><xsl:value-of select="texte" /></td>
        </tr>
      </xsl:for-each>
    </tbody>
  </table>
  <h2>Genres</h2>
  <xsl:value-of select="genres" />
  <h2>Mots-clés</h2>
  <xsl:value-of select="motCle" />
  <h2>Langues</h2>
  <xsl:value-of select="langages" />
```

```

<h2>Acteurs</h2>
<table>
  <thead>
    <th>Nom</th>
    <th>Rôle</th>
  </thead>
  <tbody>
    <xsl:for-each select="roles/role">
      <xsl:sort data-type="number" order="ascending" select="@place"/>
      <xsl:variable name="id_acteur" select="idActeur" />
      <tr>
        <td class="infoActeur"><xsl:value-of select="../../../../acteurs/acteur[@id=$id_acteur]/nom" />
          <span>
            <table>
              <tr>
                <td>Genre :</td>
                <td><xsl:value-of select="../../../../acteurs/acteur[@id=$id_acteur]/sexe" /></td>
              </tr>
              <tr>
                <td>Date Naissance :</td>
                <td><xsl:value-of select="../../../../acteurs/acteur[@id=$id_acteur]/dateNaissance" /></td>
              </tr>
              <tr>
                <td>Date décès :</td>
                <td><xsl:value-of select="../../../../acteurs/acteur[@id=$id_acteur]/dateDeces" /></td>
              </tr>
            </table>
            Biographie :<br/>
            <xsl:value-of select="../../../../acteurs/acteur[@id=$id_acteur]/biographie" />
          </span>
        </td>
        <td><xsl:value-of select="personnage" /></td>
      </tr>
    </xsl:for-each>
  </tbody>
</table>
</xsl:template>

```

3.5 Css

Nous avons repris le fichier de css proposé comme exemple, et nous avons ajouté la gestion du survol des acteurs avec la souris, afin de créer une petite fenêtre avec les infos des acteurs.

```
body {
    background-image: linear-gradient(to bottom, #F0F0F0 0%, #CCFF99);
    padding: 0;
    margin-top: 10px;
    margin-bottom: 10px;
    margin-right: 10px;
    margin-left: 30px;
}
h1 {
    margin-top: 50px;
    margin-left: 0px;
}
h2 {
    margin-top: 50px;
    margin-left: 0px;
}
td {
    text-align: left;
    vertical-align: top;
    padding-top: 5px;
    padding-bottom: 5px;
}
th {
    border: 1px gray solid;
}
.infoAuteur span {
    display:none;
    padding:15px;
    max-width:70%;
}
.infoAuteur:hover span {
    display:block;
    position:absolute;
    background-color:white;
    border:solid 1px black;
    pointer-events: none;
    font-weight:normal;
}
.infoAuteur:hover {
    font-weight:bold;
}
table.index {
    margin-top:-100px;
}
footer {
    height:500px;
}
```

4. Captures

Liste des projections

Liste organisée par date

Date	Salle	Titre	Durée	Note moyenne
Fri Jun 03 20:40:36 CEST 2016	Flon 1	Casino Royale	144	4.3
Sat Jun 04 22:30:05 CEST 2016	Flon 1	GoldenEye	130	2.5
Sun Jun 05 22:30:04 CEST 2016	Flon 2	Django Unchained	165	4.5

Liste organisée par titre

Date	Salle	Titre	Durée	Note moyenne
Fri Jun 03 20:40:36 CEST 2016	Flon 1	Casino Royale	144	4.3
Sun Jun 05 22:30:04 CEST 2016	Flon 2	Django Unchained	165	4.5
Sat Jun 04 22:30:05 CEST 2016	Flon 1	GoldenEye	130	2.5

Liste organisée par note moyenne

Date	Salle	Titre	Durée	Note moyenne
Sun Jun 05 22:30:04 CEST 2016	Flon 2	Django Unchained	165	4.5
Fri Jun 03 20:40:36 CEST 2016	Flon 1	Casino Royale	144	4.3
Sat Jun 04 22:30:05 CEST 2016	Flon 1	GoldenEye	130	2.5

[Projections](#)

Casino Royale



Durée

144min

Synopsis

Casino Royale introduces James Bond before he holds his license to kill. But Bond is no less dangerous, and with two professional assassinations in quick succession, he is elevated to '00' status. Bond's first 007 mission takes him to Uganda where he is to spy on a terrorist, Mollaka. Not everything goes to plan and Bond decides to investigate, independently of MI6, in order to track down the rest of the terrorist cell. Following a lead to the Bahamas, he encounters Dimitrios and his girlfriend, Solange. He learns that Dimitrios is involved with Le Chiffre, banker to the world's terrorist organizations. Secret Service intelligence reveals that Le Chiffre is planning to raise money in a high-stakes poker game in Montenegro at Le Casino Royale. MI6 assigns 007 to play against him, knowing that if Le Chiffre loses, it will destroy his organization. 'M' places Bond under the watchful eye of the beguiling Vesper Lynd. At first skeptical of what value Vesper can provide, Bond's interest in her deepens as they brave danger together and even torture at the hands of Le Chiffre. In Montenegro, Bond allies himself with Mathis MI6's local field agent, and Felix Leiter who is representing the interests of the CIA. The marathon game proceeds with dirty tricks and violence, raising the

Critiques

Note	Avis
5	Super film!
5	Excellent!!
3	Mouais...

Genres

Action, Thriller, Adventure

Mots-clés

gadget-car, motorboat, shot-in-the-eye, cult-film, wager, car-accident, spa, crane, shower, martini, killed-in-an-elevator, explosion, jungle, dead-woman-with-eyes-open, casino, eye-patch, corpse, male-nudity, slow-motion-scene, foot-chase, terrorist-base, martial-arts, held-at-gunpoint, parkour, africa, village, chase, beating, fatal-attraction, exploding-bus, heart-attack, drugged-drink, free-running, murder-by-gunshot, banker, violence, shot-in-the-back, airplane, shot-in-the-forehead, poisoned, video-camera, villa, montenegro, adultery, femme-fatale, silenced, hotel-receptionist, shot-in-the-head, shootout, helicopter, woman-drowned, gambling-syndicate, beach-resort, hotel, flashback, lifting-male-in-air, lifting-an-adult-into-the-air, madagascar, poison, bodyworks, attempted-poisoning, laptop-computer, flood, reboot-of-series, assassin, machine-gun, corporal-punishment, prisoner, shot-in-the-chest, super-villain, machete, assassination, bahamas, double-cross, gadget, henchman, restaurant, taser, trapped-in-an-elevator, gunfight, animated-credits, sequel, fire-sprinkler, shot-in-the-foot, balisong, blood, based-on-novel, electrocution, security-camera, shot-in-the-leg, car-chase, mass-murder, swimming, stabbed-in-the-chest, dead-girl, walther-p99, blockbuster, surveillance, mongoose, pistol, altered-version-of-studio-logo, blood-on-shirt, espionage, fight, computer, bare-chested-male-bondage, origin-story, cult-figure, secret-device, tanker, returning-character-with-different-actor, title-spoken-by-character, pinball-machine, sailing-boat, double-agent, construction-site, money, gambling, stabbing, police-officer, building-collapse, murder, south-africa, neck-breaking, hammock, italy, nassau-bahamas, tied-to-a-chair, miami-florida, money-laundering, ak-47, playing-cards, horse, airport, embassy, rogue-agent, lost-love, bond-girl, hit-in-the-crotch, arch-villain, suspense, terrorism, dead-woman, prequel-to-sequel, soldier, bomb, sequel-mentioned-during-end-credits, crushed-to-death, snake, cocktail, train, strangulation, siren, exploding-body, secret-service, horse-and-carriage, blood-spatter, shotgun, death, official-james-bond-series, uganda, beach, hospital, horse-riding, tough-guy, secret-agent, cell-phone, lifting-someone-into-the-air, cia-agent, action-hero, lagoon, aston-martin, drowning, body-bag, yacht, 2000s, psychopath, darkroom, foot-pursuit, torture, stairwell, poker-the-card-game, black-and-white-prologue, prague-czech-republic, flipping-car, infidelity, army, exhibit, london-england, hostage, venice-italy, police-car, plot-twist, newspaper-headline, off-screen-murder, damsel-in-distress, police-arrest, british-secret-service, terrorist, passport, penumbra, surprise-ending

Langues

French, English

Acteurs

Nom	Rôle
Craig, Daniel	James Bond
Green, Eva	Vesper Lynd
Mikkelsen, Mads	Le Chiffre
Dench, Judi	M

Genre : FEMININ

Date Naissance : Sun Dec 09 00:00:00 CET 1934

Date décès : -

Biographie :

Attended Mount School in York, and studied at the Central School of Speech and Drama. She has performed with Royal Shakespeare Company, the National Theatre, and at Old Vic Theatre. She is a ten-time BAFTA winner including Best Actress in a Comedy Series for "A Fine Romance" (1981) (qv) in which she appeared with her husband, 'Michael Williams (I)' (qv), and Best Supporting Actress in "A Handful of Dust" (1988) (qv) and "A Room with a View" (1985) (qv). She received an ACE award for her performance in the television series "Mr. and Mrs. Edgell" (1985) (TV) (qv). She was made an Officer of the Order of the British Empire in 1970, and was created Dame of Order of the British Empire in 1988.

Sinclair, Malcolm	Dryden
Sammel, Richard	Adolph Gettler
Pistor, Ludger	Mendel
Millson, Joseph	Carter
Shaw, Darwin	Fisher
Schick, Clemens	Kratt
Avena, Emmanuel	Leo
Chadbon, Tom	Stockbroker
Ade	Infante
Barberini, Urbano	Tomelli
Chin, Tsai	Madame Wu
Levi Leroy, Charlie	Gallardo
Ristovski, Lazar	Kaminofsky
So, Tom	Fukutu
von Lehnendorff, Veruschka Gräfin von Wallenstein	

5 Conclusion

Grâce à ce laboratoire, nous avons corrigé les éléments manquant du précédent, et nous avons pu générer avec succès des fichiers html, se basant sur le fichier xml des projections, et sur la css pour le style. L'appel des templates xsl est très puissante, et permet de générer beaucoup de contenu, de façon « dynamique » en fonction des différents champs du fichier xml.