



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université AMO de Bouira

Faculté des Sciences et des Sciences Appliquées

Département d'Informatique



Compte rendu de TP 1

Méthodes et Technologies d'Implémentation

Spécialité : Génie des Systèmes Informatiques

XML

Réalisé par

— FEKAR ROUMAÏSSA

— AIFAOUÏ SARA

2019/20120

0.1 TP2

0.1.1 Introduction

Le langage XML (eXtended Markup Language) est un format général de documents orienté texte. Il s'est imposé comme un standard incontournable de l'informatique. Il est aussi bien utilisé pour le stockage de documents que pour la transmission de données entre applications. Sa simplicité, sa flexibilité et ses possibilités d'extension ont permis de l'adapter à de multiples domaines allant des données géographiques au dessin vectoriel en passant par les échanges commerciaux. De nombreuses technologies se sont développées autour de XML et enrichissent ainsi son environnement.

Le langage XML dérive de SGML (Standard Generalized Markup Language) et de HTML (HyperText Markup Language). Comme ces derniers, il s'agit d'un langage orienté texte et formé de balises qui permettent d'organiser les données de manière structurée.

l'objectif de ce programme :

- Avoir une idée de XML.
- Savoir faire de simples grammaires DTD.
- Syntaxe qui ressemble à celle de HTML.

0.1.2 Méthodologie du travail

Dans le TP2 nous allons utiliser des technologies qui permettent de définir une structure stricte aux documents XML

DTD : Le DTD ou Document Type Declaration ou encore Document Type Definition est l'ensemble des règles et des propriétés que doit suivre le document XML.

On peut inclure son propre DTD au code source du fichier XML. On parlera alors d'un DTD interne.

Le DTD interne suit la syntaxe suivante : `;<!DOCTYPE élément-racine [déclaration des éléments]>`

Le DTD externe suivra la syntaxe suivante : `;<!DOCTYPE élément-racine SYSTEM "nom_du_fichier.dtd"`

XSLT : XSLT est un langage central dans le monde XML et beaucoup de qualités reconnues de XML reposent en fait sur l'utilisation de XSLT : productions de versions diffusables (HTML, PDF, etc.), pérennité des documents, ouverture des formats, interopérabilité, etc.

La Syntaxe est la suivante : `<?xml-stylesheet type="text/xsl" href="nom_fichier.xsl" ?>`

CSS : Pour afficher les balises XML, on peut faire appel aux bonnes vieilles feuilles de style (CSS), maintenant classiques dans le paysage Html. A chaque balise "inventée" dans le fichier XML, on va définir un élément de style que le navigateur pourra alors afficher.

La Syntaxe est la suivante : `<?xml-stylesheet type="text/css" href="nom_fichier.css" ?>`

0.1.3 Outils utilisés

- XML Copy Editor



XML Copy Editor est un éditeur XML léger et rapide. Mise à part la vérification de fermeture de tags, ce programme peut tout supporter (relax et strict), transformations XSL, Xpath, encodage, etc.).

0.1.4 Tests et résultats attendus

Après l'exécution le programme s'affiche les capteur suivant :

R1 : oui, on peut modifier le fichier CSS pour afficher que le nom, le prénom, et la ville en ajoutant l'instruction `display : none` ; (Voir Figure 1)



FIGURE 1 – R1

R2 : Non, on ne peut pas modifier le fichier CSS pour afficher que le nom, le prénom, et

la ville.

R3 : Oui, on peut modifier le fichier XSL pour afficher que le nom, le prénom, et la ville.(Voir figure 2)



AchourHamza	Alger
AOmarAhmed	Bouira

FIGURE 2 – R3

R4 :Oui, on peut modifier le fichier XSL pour afficher le prénom avant le nom.
(Voir figure 3)



HamzaAchour	Alger
AhmedAOmar	Bouira

FIGURE 3 – R4

R5 : ● Modifier le code pour afficher chaque personne comme suite :

(Voir figure 4)

Nom:Achour
Prenom:Hamza
Rue:25 avenue de 05 juillet 1962
Code-postale:16000
Ville:Alger
Telephone:071002500

FIGURE 4 – R5

R6 : ● Modifier le code pour afficher une table qui contient la liste des personnes.(Voir figure 5)

Annuaire

la liste des personnes

nom	Prénom	Adresse
Achour	Hamza	25 avenue de 05 juillet 1962 16000 Alger 071002500
AOmar	Ahmed	18 avenue du Ibn Badis 10000 Bouira 026942525
Alouache	Ayoub	18 avenue du Didouche Mourad 42000 Tipaza 051252525
Khaladi	Razika	18 Ahmed Ben Aissa 10000 Bouira 026666666
Benkadour	Ahmed	18 avenue du Ibn Badis 10000 Bouira 061252525
Ribiai	Hamada	18 avenue du Ibn Badis 10000 Bouira 063052525

FIGURE 5 – R6

R7 : ● Afficher que les personnes habitants à Bouira.(Voir figure 6)

Annuaire

la liste des personnes

nom	Prénom	Adresse
AOmar	Ahmed	18 avenue du Ibn Badis 10000 Bouira 026942525
Khaladi	Razika	18 Ahmed Ben Aissa 10000 Bouira 026666666
Benkadour	Ahmed	18 avenue du Ibn Badis 10000 Bouira 061252525
Ribiai	Hamada	18 avenue du Ibn Badis 10000 Bouira 063052525

FIGURE 6 – R7

0.2 Travail à domicile

0.2.1 Tests et résultats attendus

1ère partie :(Voir Code1)

Liste des etudiants

Matricule	Nom	Prenom	Moy S1	Moy S2	Moyenne Annuelle	Credit S1	Credit S2	Unite	Resultat
Q14M121	ZIANE	BILAL	12.78	12.87	12.83	30	30	Oui	Admis
Q14M120	ZITOUNI	SOUMIA	12.01	13.45	12.73	30	30	Oui	Admis
Q14M125	ZEGHDANE	HOUSSAM	13.63	11.52	12.58	30	30	Oui	Admis
Q14M122	ZIANE	MEZIANE	12.72	11.93	12.33	30	30	Oui	Admis
Q14M152	TALEB	ASMA	14.73	9.90	12.32	30	21	Oui	Admis avec Dettes
Q14M141	TOUATI	MOHAMED	11.99	10.37	11.18	30	30	Oui	Admis
Q14M126	ZEGADI	AMEL	11.04	11.25	11.15	30	30	Oui	Admis
Q14M123	ZIANE	AMIRA	10.15	11.91	11.03	30	30	Oui	Admis
Q14M135	YAHIAOUI	MOHAMED	11.32	10.56	10.94	30	30	Oui	Admis
Q14M124	ZERARI	OUASSILA	10.20	11.65	10.93	30	30	Oui	Admis
Q14M168	SENANI	HAYET	12.70	9.12	10.91	30	3	Oui	Ajourné
Q14M143	TIGRINE	AYOUB	11.30	10.30	10.8	30	30	Oui	Admis
Q14M158	SOBAIHI	FAHEM	11.97	9.49	10.73	30	23	Non	Ajourné
Q14M128	ZAIR	KAHINA	10.57	10.84	10.71	30	30	Oui	Admis
Q14M132	YOUCEFI	SORAYA	10.54	10.71	10.63	30	30	Oui	Admis
Q14M142	TOUAMA	YAZID	10.69	10.35	10.52	30	30	Oui	Admis
Q14M127	ZAKNOUN	ANOUAR	10.12	10.87	10.5	30	30	Oui	Admis
Q14M136	TOUMIAT	SARAH	10.51	10.47	10.49	30	30	Oui	Admis
Q14M130	ZAHER	SOFIA	10.11	10.80	10.46	30	30	Oui	Admis
Q14M139	TOUATI	ALI	10.35	10.40	10.38	30	30	Oui	Admis
Q14M129	ZAIDI	IBTISSAM	9.94	10.82	10.38	24	30	Non	Ajourné
Q14M134	TOUMINE	AMEL	10.18	10.56	10.37	30	30	Oui	Admis

FIGURE 7 – 1ère partie 1

Le nombre total des etudiants est :87

Le nombre des etudiants Admis est : 20

Le nombre des etudiants Admis avec Dettes est :30

Le nombre des etudiants Ajournés est :37

La moyenne de la classe est :0

FIGURE 8 – 1ère partie 2

2ème partie :(Voir Code2)

جزء عما

اسم السورة	رقم السورة	آياتها	مكية/مدنية
النبا	78	40	مكية
النازعات	79	46	مكية
عبس	80	42	مكية
التكوير	81	29	مكية
الإنفطار	82	19	مكية
المطففين	83	36	مكية
الإنشاق	84	25	مكية
البروج	85	22	مكية
المطارق	86	17	مكية
الأعلى	87	19	مكية
الغاشية	88	26	مكية
الفجر	89	30	مكية
البلد	90	20	مكية
الشمس	91	15	مكية
الليل	92	21	مكية
الضحى	93	11	مكية
الشرح	94	8	مكية
التين	95	8	مكية
العلق	96	19	مكية
الفجر	97	5	مكية
البينة	98	8	مدنية
الزلزلة	99	8	مدنية

FIGURE 9 – 2ème partie 1



FIGURE 10 – 2ème partie 2

Annexe A

Code en annexe

TP2 : R2 :

rue

display : none ;

color : #f6f6f6 ;

font-size : 16px ;

line-height : 24px ;

code-postale

display : none ;

padding : 15px ;

font-size : 20px ;

telephone

color : #e8d4a7 ;

display : none ;

R3 :

```
<?xml:version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" encoding="utf-8" doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN" doctype-system="annuaire.dtd"/>
  <xsl:template match="/">
    <html xmlns="http://www.w3.org/1999/xhtml">
      <body>
        <h1 align="center">Annuaire</h1>
        <h2 align="center">la liste des personnes</h2>
        <xsl:for-each select="annuaire/personne">
          <h3>
            <xsl:value-of select="identite/nom"/>
            <xsl:value-of select="identite/prenom"/>
          </h3>
          <p>
            <xsl:value-of select="identite/adresse/ville"/>
          </p>
          <hr/>
        </xsl:for-each>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

R4 :

```
1 <?xml:version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 <xsl:output method="html" encoding="utf-8" doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN" doctype-system="annuaire.dtd"/>
4 <xsl:template match="/">
5 <html xmlns="http://www.w3.org/1999/xhtml">
6 <body>
7 <h1 align="center">Annuaire</h1>
8 <h2 align="center">la liste des personnes</h2>
9 <xsl:for-each select="annuaire/personne">
10 <h3>
11 <xsl:value-of select="identite/prenom"/>
12 <xsl:value-of select="identite/nom"/>
13
14 </h3>
15 <p>
16 <xsl:value-of select="identite/adresse/ville"/>
17 </p>
18 <hr/>
19 </xsl:for-each>
20 </body>
21 </html>
22 </xsl:template>
23 </xsl:stylesheet>
```

R5 :

```

1  <?xml:version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3  | <xsl:output method="html" encoding="utf-8" doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN" doctype-system="annuaire.dtd"/>
4  <xsl:template match="/">
5  | <html xmlns="http://www.w3.org/1999/xhtml">
6  | <body>
7  | <h1 align="center">Annuaire</h1>
8  | <h2 align="center">la liste des personnes</h2>
9  <xsl:for-each select="annuaire/personne">
10 | <p>
11 | <b>Nom:</b><xsl:value-of select="identite/nom"/><br/>
12 | <b>Prenom:</b><xsl:value-of select="identite/prenom"/><br/>
13 | <b>Rue:</b><xsl:value-of select="identite/adresse/rue"/><br/>
14 | <b>Code postale:</b><xsl:value-of select="identite/adresse/code-postale"/><br/>
15 | <b>Ville:</b><xsl:value-of select="identite/adresse/ville"/><br/>
16 | <b>Telephone:</b><xsl:value-of select="identite/adresse/telephone"/>
17 | </p>
18 | <hr/>
19 | </xsl:for-each>
20 | </body>
21 | </html>
22 | </xsl:template>
23 | </xsl:stylesheet>

```

R6 :

```

1  <?xml:version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3  | <xsl:output method="html" encoding="utf-8" doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN" doctype-system="annuaire.dtd"/>
4  <xsl:template match="/">
5  | <html xmlns="http://www.w3.org/1999/xhtml">
6  | <body>
7  | <h1 align="center">Annuaire</h1>
8  | <h2 align="center">la liste des personnes</h2>
9  | <table border="2" align="center">
10 | <tr>
11 | <th>nom</th>
12 | <th>Prénom</th>
13 | <th>Adresse</th>
14 | </tr>
15 | <xsl:for-each select="annuaire/personne">
16 | <tr>
17 | <td><xsl:value-of select="identite/nom"/></td>
18 | <td><xsl:value-of select="identite/prenom"/></td>
19 | <td><xsl:value-of select="identite/adresse"/></td>
20 | </tr>
21 | </xsl:for-each>
22 | </table>
23 | </body>
24 | </html>
25 | </xsl:template>
26 | </xsl:stylesheet>

```

R7 :

```
1 <?xml:version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3 | <xsl:output method="html" encoding="utf-8" doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN" doctype-system="annuaire.dtd"/>
4 <xsl:template match="/">
5 | <html xmlns="http://www.w3.org/1999/xhtml">
6 | <body>
7 | <h1 align="center">Annuaire</h1>
8 | <h2 align="center">la liste des personnes</h2>
9 | <table border="2" align="center">
10 | <tr>
11 | <th>nom</th>
12 | <th>Prénom</th>
13 | <th>Adresse</th>
14 | </tr>
15 | <xsl:for-each select="annuaire/personne[identite/adresse/ville='Bouira']">
16 | <tr>
17 | <td><xsl:value-of select="identite/nom"/></td>
18 | <td><xsl:value-of select="identite/prenom"/></td>
19 | <td><xsl:value-of select="identite/adresse"/></td>
20 | </tr>
21 | </xsl:for-each>
22 | </table>
23 | </body>
24 | </html>
25 </xsl:template>
26 </xsl:stylesheet>
```

Travail à domicile :

Code1 : En utilisant le fichier csv des étudiants de TP n 1 et écrire un programme de conversion en XML en générant les champs calculés (moyennes, mentions, etc...) avec langage python

```
PyScripter - C:\Users\sara\Desktop\tpza\etud.py
File Edit Search View Project Run Tools Help

dataInputFile = open("data.txt")
dataOutputFile = open("etud.xml", "w")

dataInputFile.__next__()

dataOutputFile.write("<?xml version='1.0' encoding='UTF-8' ?>\n"
                    "<?xml-stylesheet type='text/xsl' href='etud.xsl' ?>\n"
                    "<!DOCTYPE Group_gsi SYSTEM 'etud.dtd'>\n"
                    "<Group_gsi>\n")

for line in dataInputFile:
    splitedLine = line.split("\t")
    dataOutputFile.write("<etudiant>\n"
                        "<Identite>\n"
                        "<Matricule>" + str(splitedLine[0]) + "</Matricule>\n"
                        "<Nom>" + str(splitedLine[1]) + "</Nom>\n"
                        "<Prenom>" + str(splitedLine[2]) + "</Prenom>\n"
                        "</Identite>\n"
                        "<Moy>\n"
                        "<MoyS1>" + str(splitedLine[3]) + "</MoyS1>\n"
                        "<MoyS2>" + str(splitedLine[4]) + "</MoyS2>\n"
                        "<MoyAnnuel>" + str(splitedLine[5]) + "</MoyAnnuel>\n"
                        "<CreditS1>" + str(splitedLine[6]) + "</CreditS1>\n"
                        "<CreditS2>" + str(splitedLine[7]) + "</CreditS2>\n"
                        "<Unite>" + str(splitedLine[8].split("\n")[0]) + "</Unite>\n"
                        "</Moy>\n"
                        "</etudiant>\n")

dataOutputFile.write("</Group_gsi>")
```

Définir une DTD pour définir les types des données.

```
<!ELEMENT Group\_gsi(etudiant+)>
<!ELEMENT etudiant (Identite,Moy)>
<!ELEMENT Identite (Matricule,Nom,Prenom)>
<!ELEMENT Moy (MoyS1,MoyS2,MoyAnnuel,CreditS1,CreditS2,Unite)>
<!ELEMENT Matricule (\#PCDATA) >
<!ELEMENT Nom (\#PCDATA) >
<!ELEMENT Prenom (\#PCDATA) >
<!ELEMENT MoyS1 (\#PCDATA) >
<!ELEMENT MoyS2 (\#PCDATA) >
<!ELEMENT MoyAnnuel (\#PCDATA) >
<!ELEMENT CreditS1 (\#PCDATA) >
<!ELEMENT CreditS2 (\#PCDATA) >
<!ELEMENT Unite (\#PCDATA) >
```

Créer une feuille de style en CSS

```
body{
    background-color: \#fce4ec ;
    color: black;
}
table{
    position: relative;
    margin: auto;
    border-style:solid;
    width: 1000px;
    height: 150px;
    background-color: white;
}
th{
    text-align: center;
    background-color: \#f8bbd0;
}
```

```

\#admis{
    background-color : \#c8e6c9 ;
}
\#admis\_avec\_dette{
    background-color : \#00695c ;
}
\#ajournee{
    background-color : \#e53935 ;
}
\#sa{
    position: relative ;
    margin: auto ;
    border-style:solid ;
    width: 500px ;
    height: 250px ;
    background-color: white ;
}

```

Code2 :

```

D finir une DTD
<!ELEMENT mushaf (sura*)>
<!ELEMENT sura (aya*)>
<!ATTLIST sura nb_aya CDATA #REQUIRED>
<!ATTLIST sura name CDATA #REQUIRED>
<!ATTLIST sura id CDATA #REQUIRED>
<!ATTLIST sura type CDATA #REQUIRED>
<!ELEMENT aya (#PCDATA)>
<!ATTLIST aya id CDATA #REQUIRED>

```

- Créer une feuille de style en CSS

```
body{
background-color: red

}
table{
color:blue;
position: relative;
margin: auto;
border-style:solid;
width: 1000px;
height: 150px;
background-color: white;
}
#sura{ color: #fff;
font-family: 'Righteous', cursive;
font-size: 40px;
line-height: 60px;
text-transform: uppercase;
text-shadow: 2px 2px 0 #000;
margin: 10px 0 24px;
}
#aya {font-size: 30px;
color: #f8bbd0;
text-shadow: 1px 1px 0 #000;
margin: 10px 0 24px;
line-height: 50px;
left: 53px;
}
```


- Créer une feuille de style en Xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mushaf SYSTEM "mushaf.dtd">
<?xml-stylesheet type="text/xsl" href="mushaf.css"?>
<mushaf>
<sura nb_aya="40" id="78" name="                " type="                ">
<aya id="1">

</aya>
<aya id="2">

</aya>
<aya id="3">

</aya>
<aya id="4">

</aya>
<aya id="5">

</aya>
<aya id="6">

</aya>
</sura>
</mushaf>
```

- Créer une feuille de style en Xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/
Transform" version="1.0">
<xsl:template match="/">
<html xmlns="http://www.w3.org/1999/xhtml">
```

```

<head>
<link type="text/css" rel="stylesheet" href="mushaf.css"/>
</head>
<body>
<table border="3" cellspacing="2" align="center" >
<h1 align="center">                </h1>
<tr align="center">
<th>                </th>
<th>                </th>
<th>                </th>
<th>                /                </th>
</tr>
<xsl:for-each select="mushaf/sura">
<tr align="center">
<td><xsl:value-of select="@name"/></td>
<td><xsl:value-of select="@id"/></td>
<td><xsl:value-of select="@nb_aya"/></td>
<td><xsl:value-of select="@type"/></td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```