

TP1 : document xml valide ,bien formé et DTD

```

EXO1 : EXO2
-----TP1_EXO1_EXO2_DTD_EXTERNE_INTERNE.xml-----
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
//declaration DTD externe
<!DOCTYPE catalogue SYSTEM "DTD_TP1.dtd">
<!DOCTYPE catalogue [
<!ELEMENT catalogue (cours+,personne+)>
<!ELEMENT cours (titre,description,professeur,assistant?)>
<!ELEMENT personne (nom,prenom,email,adresse?)>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT description (#PCDATA)> <!ELEMENT professeur
EMPTY>
<!ELEMENT assistant EMPTY>
<!ELEMENT nom (#PCDATA)><!ELEMENT prenom (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT adresse (ville,zone,rue,num)>
<!ELEMENT ville (#PCDATA)><!ELEMENT zone (#PCDATA)>
<!ELEMENT rue (#PCDATA)><!ELEMENT num (#PCDATA)>
<!ATTLIST cours id ID #REQUIRED type CDATA #REQUIRED>
<!ATTLIST professeur nsom ID #REQUIRED cin IDREF #REQUIRED
grade CDATA #IMPLIED >
<!ATTLIST assistant cin IDREF #REQUIRED ><!ATTLIST personne cin
ID #REQUIRED >
<!ENTITY info "informatique"> ]>
<catalogue>
<cours id="INFO-X-80" type="&info;">
<titre>Technologie XML</titre>
<description>cours sur XML &amp; technologies
adjacentes</description>
<professeur nsom="A118845" cin="AD465" grade="professeur
abilité"/>
<assistant cin="AG466" />
&chap1;
-----Pour TP2-----
<prix>1000</prix>
<sessions>
<session>
<numero>1</numero><date_debut>12/03/2013</date_debut>
<nombre_min>10</nombre_min>
<nombre_max>30</nombre_max></session></sessions>

```

```

</cours>
<personne cin="AD465" >
<nom>ayoubi</nom>
<prenom>Mohamed</prenom>
<email>ayoubi@gmail.com</email>
</personne>
<personne cin="AG466">
<nom>Nabil</nom>
<prenom>Zahir</prenom>
<email>zahir@gmail.com</email>
<adresse>
<ville>Salé</ville>
<zone>Sala al Jadida</zone>
<rue>Avenue Moulay Rachid</rue>
<num>Num : 38</num>
</adresse> </personne>
</catalogue>
-----TP1_ENTITE_CHAP1.xml-----
<?xml version="1.0" encoding="UTF-8"?>
<chapitres>
<chapitre1><intitule>Analyse</intitule>
<duree>6 heures</duree></chapitre1></chapitres>
----- DTD_TP1.dtd -----
<!ELEMENT catalogue (cours+,personne+)>
<!ELEMENT cours
(titre,description,professeur,assistant?,chapitres)>
<!ELEMENT personne (nom,prenom,email,adresse?)>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT professeur EMPTY>
<!ELEMENT assistant EMPTY>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT adresse (ville,zone,rue,num)>
<!ELEMENT ville (#PCDATA)>
<!ELEMENT zone (#PCDATA)>
<!ELEMENT rue (#PCDATA)>
<!ELEMENT num (#PCDATA)>
<!ELEMENT chapitres (chapitre1)>
<!ELEMENT chapitre1 (intitule,duree)>
<!ELEMENT intitule (#PCDATA)>
<!ELEMENT duree (#PCDATA)>
<!ATTLIST cours id ID #REQUIRED type CDATA
#REQUIRED>
<!ATTLIST professeur nsom ID #REQUIRED cin IDREF
#REQUIRED grade CDATA #IMPLIED >
<!ATTLIST assistant cin IDREF #REQUIRED >
<!ATTLIST personne cin ID #REQUIRED >
<!ENTITY info "informatique">
<!ENTITY chap1 SYSTEM "TP1_ENTITE_CHAP1.xml">

```

Questions TP1

Verifier la validiter du doc XML et établir sa DTD.(EXO1 ET EXO2)

Questions TP2

Verifier la validiter du doc XML et établir son schéma xml.(de mm)

Questions TP3

Exo1 :

2-Améliorer la feuille de style afin d’obtenir l’affichage de la figure 1

Titre,auteur,et annee sont en *gras*(<b>)

Les élément d’un livre sont séparé par des sauts de ligne (<br>)

Les livres sont séparés par des marques de *paragraphes*(<p>)

3-modifier la feuille de style afin d’avoir la forme de la figure 2.

4-afficher le tableau de resultat suivant.

Exo2 :

1-afficher le tableau suivant

2-les cours triée par prix

3-les cours ayant plus d’une session(triés)

4-les cours ayant le prix le plus petit.

```

-----TP1_EXO3_CV_DTD_EXTERNE.xml-----
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE cv SYSTEM "DTD_TP1_EXO3.dtd">
<cv cin="F563671">
<personne sexe="masculin">
<nom>BOUJIMI</nom> <prenom>Mohammed</prenom>
<age>21 ans</age> <situation>célibataire</situation>
<adresse>
<village>Hay Essalam Aouinat Essarak</village>
<num>A14,N 26 BIS</num><ville>OUJDA</ville>
</adresse>
<mobile type="portable">0639-666616</mobile>
<email>mohammedboujmi@gmail.com</email>
<statut>Elève-Ingénieur</statut>
<niveau>2 ème année</niveau>
<ecole specialite="GL">ENSIAS</ecole>
<permis_conduite>B</permis_conduite></personne>
<formations><formation>
<annee>2015-2016</annee>
<description>Génie logiciel à l'École Nationale Supérieure
d'Informatique et d'Analyse des Systèmes (ENSIAS, Rabat).</description>
</formation>
<formation>
<annee>2013-2015</annee>
<description>Classes préparatoires aux grandes écoles d'ingénieurs
(Filière: TSI, Ville : OUJDA)</description>
</formation>
<formation>
<annee>2013</annee>
<description>Baccalauréat Sciences et Technologies (option Sciences
et Technologies Electriques) avec mention Très Bien ( Lycee Mehdi Ben
Barka , OUJDA)</description>
</formation>
</formations>
<competences>
<competence>Langage de Programmation: C, JAVA , SQL , PL/SQL ,
Assembleur.</competence>
<competence>Bibliothèques & Framework:
Swing</competence>
<competence>Technologies web : PHP,HTML,CSS.</competence>
<competence>SGBD: Oracle, MySQL</competence>
<competence>Systèmes d'exploitations: Windows,
Linux.</competence>
</competences>
<langues>
<langue>Arabe : Langue maternelle.</langue>
<langue>Français : Bonne maîtrise.</langue>
<langue>Anglais : Assez bonne maîtrise.</langue></langues>
<divers>
<loisirs>Loisirs: Documentaires, Jeux vidéo, football.</loisirs>
<parascolaire>Membre d'organisation des olympiades ENSIAS
V2.0.</parascolaire>
</divers>
</cv>
-----DTD_TP1_EXO3.dtd-----
<!ELEMENT cv (personne,formations,competences,langues,divers)>
<!ELEMENT personne
(nom,prenom,age,situation,adresse,mobile,email,statut,niveau,ecole,pe
rmis_conduite)>
<!ELEMENT formations (formation+)>
<!ELEMENT competences (competence+)>
<!ELEMENT langues (langue+)>
<!ELEMENT divers (loisirs+,parascolaire+)>
<!ELEMENT adresse (village,num,ville)>
<!ELEMENT formation (annee,description)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
<!ELEMENT age (#PCDATA)>
<!ELEMENT situation (#PCDATA)>
<!ELEMENT village (#PCDATA)>
<!ELEMENT num (#PCDATA)>
<!ELEMENT ville (#PCDATA)>
<!ELEMENT mobile (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT statut (#PCDATA)>
<!ELEMENT niveau (#PCDATA)>
<!ELEMENT ecole (#PCDATA)>
<!ELEMENT permis_conduite (#PCDATA)>
<!ELEMENT annee (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT competence (#PCDATA)>
<!ELEMENT langue (#PCDATA)>
<!ELEMENT loisirs (#PCDATA)>
<!ELEMENT parascolaire (#PCDATA)>
<!ATTLIST cv cin ID #REQUIRED>
<!ATTLIST personne sexe (masculin|feminin) #REQUIRED>
<!ATTLIST mobile type (fixe|portable) "portable">
<!ATTLIST ecole specialite (GL|BI|IWIM|SSI|ISEM|IEL) "GL">

```

Questions TP5

Utilisez les fichiers XML films.xml et artistes.xml

1. nom et prénom des acteurs du fil « Blade Runner ». L’élément résultat contiendra un élément acteur pour chacune de ces informations présentées sous forme d’éléments ;

2. Liste des titres de film et leurs années de sortie (en attribut) ;

3. titres de films triés selon leurs dates et de sorties par ordre décroissant et par ordre alphabétique selon le titre pour ceux sortis la même année ;

4. titres de films d’action, nom du premier rôle et nom du metteur en scène ;

5. nombre de films sortis avant 19990 et sortis après 1990

6. tous les films et les informations associés mais sans les rusées et sans les rôles ;

7. titre des films où ont joué ensemble Harrison Ford et Rutger Hauer

8. titres des films triés par genre (TypeFilm comme attribut de l’élément TITRE si le genre existe sinon ne pas ajouter l’attribut) ;

9. les rôles joués par John Travolta dans ses films dont vous préciserez le titre (comme élément) et l’année (comme attribut du titre) ;

10. la liste des acteurs (triés par id), avec la liste des films (triés par ordre alphabétique) dans lesquels ils ont joué ;

TP5 : XQuery

```

-----Artistes.XML-----
<?xml version="1.0" encoding="iso-8859-1"?>
<ARTISTES>
<ARTISTE id="6">
<ACTNOM>Cameron</ACTNOM><ACTPNOM>James</ACTPNOM>
<ANNEENAISS>1954</ANNEENAISS> </ARTISTE>
<ARTISTE id="3"><ACTNOM>Hitchcock</ACTNOM>
<ACTPNOM>Alfred</ACTPNOM><ANNEENAISS>1899</ANNEENAISS>
</ARTISTE>
<ARTISTE id="4"> <ACTNOM>Scott</ACTNOM>
<ACTPNOM>Ridley</ACTPNOM> <ANNEENAISS>1937</ANNEENAISS>
</ARTISTE>
<ARTISTE id="5">
<ACTNOM>Weaver</ACTNOM> <ACTPNOM>Sigourney</ACTPNOM>
<ANNEENAISS></></ARTISTE>
<ARTISTES>
-----Film.XML-----
<?xml version="1.0" encoding="iso-8859-1"?>
<FILMS>
<FILM annee="1958"><TITRE>Vertigo</TITRE>
<GENRE>Drame</GENRE><PAYS>USA</PAYS> <MES idref="3"/>
<ROLES>
<ROLE><PRENOM>James</PRENOM> <NOM>Stewart</NOM>
<INTITULE>John Ferguson</INTITULE> </ROLE>
<ROLE><PRENOM>Kim</PRENOM><NOM>Novak</NOM>
<INTITULE>Madeleine Elster</INTITULE> </ROLE>
</ROLES>
<RESUME>Scottie Ferguson, ancien inspecteur de police, est sujet
au vertige depuis qu'il a vu mourir son
colleque. Elster, son ami, le charge de surveiller sa femme, Madeleine,
ayant des tendances
suicidaires. Amoureux de la jeune femme Scottie ne remarque pas le piege
qui se trame autour
de lui et dont il va etre la victime... </RESUME>
</FILM></FILMS>
-----Requette XQuery-----
1)
<acteurs>
{for $x in doc("films.xml")//FILMS/FILM,
$y in $x/ROLES/ROLE where $x/TITRE="Blade Runner"
return <acteur> {$y/NOM} {$y/PRENOM} } </acteurs>
2)+3
<films>
{for $x in doc("films.xml")//FILMS/FILM order by $x/@annee descending,
$x/TITRE descending
return <film annee_de_sortie="{ $x/@annee}"> { $x/TITRE} </film>
}</films>
4)
<films>
{for $x in doc("films.xml")//FILMS/FILM,
$y in doc("artistes.xml")//ARTISTES/ARTISTE
where $y/@id=$x/MES/@idref and $x/GENRE="Action"
order by $x/@annee descending, $x/TITRE descending
return <film> { $x/TITRE} { $x/ROLES/ROLE[1]/NOM} {$y/ACTNOM} } </films>
5)
let $y:=doc("films.xml")//FILMS/FILM
return <result>
<number_inf_1990> {count($y[@annee<1990])} </number_inf_1990>
<number_sup_1990> {count($y[@annee>=1990])} </number_sup_1990>
</result>
6-
for $x in doc("films.xml")//FILM
return element movie
{ $x/TITRE, $x/GENRE, $x/PAYS, $x/MES, $x/RESUME }
7)
<films>
{for $x in doc("films.xml")//FILMS/FILM
where (some $r in $x/ROLES satisfies( $r/ROLE/NOM="Ford" and
$r/ROLE/PRENOM="Harrison"))
and (some $r in $x/ROLES satisfies( $r/ROLE/NOM="Hauer" and
$r/ROLE/PRENOM="Rutger"))
return <film annee_de_sortie="{ $x/@annee}"> { $x/TITRE} { $x/ROLES}
} </films>
8-
for $x in doc("films.xml")//FILM
return if (exists($x/GENRE))
then element movie{
element TITRE{
attribute typeFilm { $x/GENRE},
data{ $x/TITRE} } }
else element movie{
element TITRE{
data{ $x/TITRE} } }
}
9)
<films>
{for $x in doc("films.xml")//FILMS/FILM,
$y in $x/ROLES/ROLE where $y/NOM="Travolta" and $y/PRENOM="John"
return element film
{ attribute annee{ $x/@annee}, element title { $x/TITRE/text() },
element role { $y/INTITULE/text() } }
}</films>
10)
for $x in doc("artistes.xml")//ARTISTE
order by $x/@id
return element acteur{ $x/ACTNOM, $x/ACTPNOM,
for $films in doc("films.xml")//FILM
where exists($films/ROLES/ROLE[PRENOM = $x/ACTPNOM and NOM =
$x/ACTNOM])
return element movie{ $films/TITRE } }

```

Questions TP4

EXO1 :

2-Améliorez l'exemple pour afficher la liste des ouvrages en français

EXO2:utilisant catalogue.xml

1-Afficher les cours disponibles sur le catalogue.

2-Afficher les cours ayant plus d'une session.

3-modifier le prix (7000 au lieu de 3000) du cours ayant id= « MATH-0-1 »

En créant un autre fichier XML contenant la nouvelle valeur

4-créer un fichier html du doc XML.

## TP4 : DOM &amp; Sax

## bibliographie.XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<?xml-stylesheet href="TP3-Q3-Q4.xsl" type="text/xsl"?>
<bibliographie>
<livre key="Michard01" lang="fr">
<titre>XML langage et applications</titre>
<auteur>Alain Michar </auteur> <annee>2001</annee>
<editeur>Eyrolles </editeur> <identification isbn="2-212-09206-7"/>
<editions> 2006-08-13 2006-08-14 2006-08-15 </editions>
<url>http://www.editions-eyrolles/livres/michard/</url>
</livre>
<livre key="Samuel1" lang="fr">
<titre>Réseaux et systèmes informatiques mobiles</titre>
<auteur>Samuel Pierre </auteur> <annee>2003</annee>
<editeur> Presses Polytechnique de Montréal </editeur>
<identification isbn="978-2553010385"/>
</livre>
<livre key="Zeldman03" lang="en">
<titre>Designing with web standards</titre>
<auteur>Jeffrey Zeldman</auteur> <annee>2003</annee>
<editeur> New Riders </editeur>
<identification isbn="0-7357-1201-8"/></livre></bibliographie>
```

## catalogue.XML

```
<catalogue>
<ours id="INFO-H-509" type="Informatique">
<titre>Technologies XML</titre>
<prix unite="MAD">4000</prix>
<session num="1"> <date>12-03-2015</date></session>
<session num="2"> <date>28-12-2015</date></session>
<session num="3"> <date>14-1-2016</date></session>
</ours>
<ours id="MATH-0-1" type="Mathematique">
<titre>Recherche Opérationnelle</titre>
<prix unite="Euro">3000</prix>
<session num="1"> <date>12-03-2013</date></session>
<session num="2"> <date>28-12-2015</date></session>
</ours></catalogue>
```

## EX01

```
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.Document;import org.w3c.dom.Element;
import org.w3c.dom.Node;import org.w3c.dom.NodeList;
public class DomParser {
public static void main(String[] args) {
try{
/* Cr  ation du parseur en appelant la m  thode statique factory */
DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
DocumentBuilder builder = factory.newDocumentBuilder();
/* Chargement du fichier XML */
Document document = builder.parse("src/Bibliographie.xml");
/* R  cup  rer le contenu de l'  l  ment racine */
Element racine = document.getDocumentElement();
/* R  cup  ration de tous les livres inclus dans bibliographie */
NodeList livres = racine.getElementsByTagName("livre");
System.out.println("Les livres disponibles en Fran  ais sont : " );
for (int i = 0 ; i < livres.getLength() ; i++) {
System.out.println("-----");
/* R  cup  rer les noeuds */
Element livre = (Element)livres.item(i);
String lang = livre.getAttribute("lang"); // R  cup  rer l'attribut lang
Node Titre = livre.getElementsByTagName("titre").item(0);
Node Auteur = livre.getElementsByTagName("auteur").item(0);
Node Annee = livre.getElementsByTagName("annee").item(0);
if (lang.equals("fr")){ /* Afficher les noeuds */
System.out.println(Titre.getFirstChild().getNodeValue());
System.out.println(Auteur.getFirstChild().getNodeValue());
System.out.println(Annee.getFirstChild().getNodeValue()); }
} catch (Exception e) { e.printStackTrace(System.out);}}
}
TP2 : Sh  ma XML    EX01
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
<xsd:element name="catalogue">
<xsd:complexType> <xsd:sequence>
<xsd:element name="cours" maxOccurs="unbounded"
minOccurs="0">
<xsd:complexType><xsd:sequence>
<xsd:element name="titre" type="xsd:string"/>
<xsd:element name="description" type="xsd:string"/>
<xsd:element name="professeur"><xsd:complexType>
<xsd:attribute name="insom" type="xsd:ID"
use="required"></xsd:attribute>
<xsd:attribute name="cin" type="xsd:IDREF"
use="required"></xsd:attribute>
<xsd:attribute name="grade"
type="xsd:string"></xsd:attribute>
</xsd:complexType>
</xsd:element>
<xsd:element name="assistant">
<xsd:complexType>
<xsd:attribute name="cin" type="xsd:IDREF"
use="required"></xsd:attribute>
</xsd:complexType>
</xsd:element>
<xsd:element name="prix" type="xsd:float"/>
<xsd:element name="sessions">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="session" maxOccurs="unbounded">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="numero" type="xsd:int"/>
<xsd:element name="date_debut" type="dateType"/>
<xsd:element name="nombre_min" type="max"/>
<xsd:element name="nombre_max" type="max"/>
</xsd:sequence></xsd:complexType></xsd:element>
</xsd:sequence></xsd:complexType> </xsd:element>
</xsd:sequence>
<xsd:attribute name="id" type="xsd:string" use="required" />
<xsd:attribute name="type" type="xsd:string" use="required" />
</xsd:complexType></xsd:element>
```

## EX02

```
import java.io.File;import java.io.FileWriter;import javax.xml.parsers.*;
import javax.xml.transform.*;import javax.xml.transform.stream.*;
import javax.xml.transform.stream.*;import org.w3c.dom.*;
public class Boujmi {
public static void main(String[] args) {
try{
/* Cr  ation du parseur en appelant la m  thode statique factory */
DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
DocumentBuilder builder = factory.newDocumentBuilder();
/* Chargement du fichier XML */
Document document = builder.parse("src/categorie.xml");
/* R  cup  rer le contenu de l'  l  ment racine */
Element racine = document.getDocumentElement();
/* R  cup  ration de tous les cours inclus dans catalogue */
NodeList courses = racine.getElementsByTagName("cours");
/* --- QUESTION 1 --- */
System.out.println("Les cours disponibles sont : " );
for (int i = 0 ; i < courses.getLength() ; i++) {
/* R  cup  rer les noeuds */
Element cours = (Element)courses.item(i);
Node Titre = cours.getElementsByTagName("titre").item(0);
Node Titre = cours.getElementsByTagName("titre").item(0);
/* Afficher les noeuds */
System.out.println(Titre.getFirstChild().getNodeValue());}
/* --- QUESTION 2 --- */
System.out.println("\n\nLes cours ayant plus d'une session sont : " );
for (int i = 0 ; i < courses.getLength() ; i++) {
Element cours = (Element)courses.item(i);
Node Titre = cours.getElementsByTagName("titre").item(0);
int nbr = cours.getElementsByTagName("session").getLength();
if (nbr > 1){
System.out.println(Titre.getFirstChild().getNodeValue());}
}
/* --- QUESTION 3 --- */
System.out.println("\n\nModification du prix du cours ayant l'id   gale   
MATH-0-1 " );
for (int i = 0 ; i < courses.getLength() ; i++) {
Element cours = (Element)courses.item(i);
String id = cours.getAttribute("id");
if (id.equals("MATH-0-1")){
Node Prix = cours.getElementsByTagName("prix").item(0);
System.out.println("Ancienne valeur : " +
Prix.getFirstChild().getNodeValue());
Prix.setTextContent("7000");
//Prix.setNodeValue("7000"); (ne marche pas)
System.out.println("Nouvelle valeur : " +
Prix.getFirstChild().getNodeValue());
TransformerFactory tfact = TransformerFactory.newInstance();
Transformer transformer = tfact.newTransformer();
DOMSource source = new DOMSource(document);
FileWriter fw = new FileWriter("src/catalogue_V2.xml");
StreamResult result = new StreamResult(fw);
transformer.transform(source, result);}
}
/* --- QUESTION 4 --- */
Source srcXML = new DOMSource(document);
/* Cr  ation du fichier HTML de sortie */
File fHtml = new File("result.html");
Result resultHTML = new DOMResult(fHtml);
//ce fichier deja fait dans le TP3
StreamSource srcXML = new StreamSource("src/categorie.xml");
Transformer leTransf =
(TransformerFactory.newInstance()).newTransformer(srcXML);
leTransf.transform(srcXML,resultHTML);
} catch (Exception e) { e.printStackTrace(System.out); } }
TP2 : Sh  ma XML (suite)
```

```
<xsd:element name="personne" maxOccurs="unbounded"
minOccurs="0">
<xsd:complexType><xsd:sequence>
<xsd:element name="nom" type="xsd:string" />
<xsd:element name="prenom" type="xsd:string" />
<xsd:element name="email" type="email_Type" />
<xsd:element name="adresse" minOccurs="0">
<xsd:complexType><xsd:sequence>
<xsd:element name="ville" type="xsd:string" />
<xsd:element name="zone" type="xsd:string" />
<xsd:element name="rue" type="xsd:string" />
<xsd:element name="num" type="xsd:string" />
</xsd:sequence></xsd:complexType></xsd:element>
</xsd:sequence>
<xsd:attribute name="cin" type="xsd:ID" use="required" />
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:simpleType name="dateType">
<xsd:restriction base="xsd:string">
<xsd:pattern value="[0-9]{2}/[0-9]{2}/[0-9]{4}" />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="email_Type">
<xsd:restriction base="xsd:string">
<xsd:pattern value="[a-z]@[a-z]+\.[a-z]{2,3}" />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="max">
<xsd:restriction base="xsd:string">
<xsd:pattern value="[0-9]{1,2}" />
</xsd:restriction>
</xsd:simpleType>
</xsd:schema>
```

## TP5 : XSLT

## EX01

```
/* --- QUESTION 2 --- */
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns="http://www.w3.org/1999/xhtml" >
<xsl:output method="html"/>
<xsl:template match="/bibliographie">
<html> <head> <title>Liste des ouvrages</title> </head> <body>
<h1>Liste des ouvrages</h1><br/>
<xsl:apply-templates select="livre" /> </body>
</html> </xsl:template>
<xsl:template match="livre">
<p>
<b>Titre : </b> <xsl:value-of select="titre" /><br/>
<b>Auteur : </b> <xsl:value-of select="auteur" /><br/>
<b>Annee : </b> <xsl:value-of select="annee" /><br/><br/></p>
</xsl:template></xsl:stylesheet>
/* --- QUESTION 3-4 --- */
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema" exclude-result-
prefixes="xs" version="2.0">
<xsl:template match="/">
<html>
<head><title> Ouvrages </title></head>
<body>
<h1>Liste des ouvrages</h1>
<b>p style="font-size:20;">Liste des ouvrages Fran  ais</p><b><br/>
<xsl:for-each select="bibliographie/livre[@lang='fr']">
<p>
<b>Titre : </b> <xsl:value-of select="titre" /><br/>
<b>Auteur : </b> <xsl:value-of select="auteur" /><br/>
<b>Annee : </b> <xsl:value-of select="annee" /><br/> </p>
</xsl:for-each>
<b>p style="font-size:20;">Liste des ouvrages Anglais</p><b>
<xsl:for-each select="bibliographie/livre[@lang='en']">
<p>
<b>Titre : </b> <xsl:value-of select="titre" /><br/>
<b>Auteur : </b> <xsl:value-of select="auteur" /><br/>
<b>Annee : </b> <xsl:value-of select="annee" /><br/>
</p> </xsl:for-each>
<table style="text-align:center" border="solid black 2px">
<tr>
<th></th>
<th style="background:black;color:white">Francais</th>
<th style="background:black;color:white">Angalais</th>
<th style="background:black;color:white">Totale</th></tr>
<tr>
<td style="background:grey;color:white">Nombre d'ouvrage</td>
<td><xsl:variable name="count"
select="count(bibliographie/livre[@lang='fr'])"/>
<xsl:value-of select="$count"/></td>
<td><xsl:variable name="count"
select="count(bibliographie/livre[@lang='en'])"/>
<xsl:value-of select="$count"/></td>
<td><xsl:variable name="count" select="count(bibliographie/livre)"/>
<xsl:value-of select="$count"/></td>
</tr>
</table> </body></html> </xsl:template> </xsl:stylesheet>
EX02
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:math="http://exslt.org/math"
extension-element-prefixes="math">
<xsl:template match="/">
<html> <head> <title> Catalogue </title> </head>
<body>
<table style="text-align:center" border="solid black 2px">
<tr> <th> </th>
<th> </th>
<th style="background:black;color:white">Type</th>
<th style="background:black;color:white">Nombre de session</th>
<th style="background:black;color:white">Prix</th></tr>
<xsl:for-each select="catalogue/cours">
<tr>
<td style="background:grey;color:white">
<xsl:value-of select="titre" /></td>
<td style="background:grey;color:white">
<xsl:value-of select="@type" /></td>
<td><xsl:variable name="count" select="count(session)"/>
<xsl:value-of select="$count"/></td>
<td style="background:grey;color:white">
<xsl:value-of select="prix" /> MAD</td></tr>
</xsl:for-each>
</table><p style="font-size:20px">Liste des cours trie   par prix</p>
<xsl:for-each select="catalogue/cours">
<xsl:sort select="prix" order="descending" />
<xsl:value-of select="titre" />
( <xsl:value-of select="prix" /> MAD)<br/><br/>
</xsl:for-each>
<p style="font-size:20px">Liste des cours ayant plus d'une session</p>
<xsl:for-each select="catalogue/cours">
<xsl:sort select="count(session)" order="descending" />
<xsl:if test="count(session) > 1">
<xsl:value-of select="titre" />
( <xsl:value-of select="count(session)"/> sessions)<br/><br/></xsl:if>
</xsl:for-each>
<p style="font-size:20px">Le cours le moins ch  re</p>
<xsl:value-of select="/catalogue/cours [prix =
math:min(/catalogue/cours/prix)]/titre" />
( <xsl:value-of select="/catalogue/cours [prix =
math:min(/catalogue/cours/prix)]/prix" /> MAD)
</body></html></xsl:template></xsl:stylesheet>
```