

# Solution du travail à domicile .

## membre du groupe :

- MEBARKIA Abdenour .
- SABEG Hichem .
- TAOUS Mohamed .

## Exercice 01:

### 1).Document xml modélisant un cv :

```
<?xml version="1.0" encoding="UTF-8"?>
<cv>
  <!-- Coordonnees -->
  <coordonnees>
    <nom>MEBARKIA</nom>
    <prenom>Abdenour</prenom>
    <adresse>
      <codepostal>40000</codepostal>
      <ville> Khenchela </ville>
    </adresse>
    <dateDeNaissance>13/06/2001</dateDeNaissance>
    <tel>06.55.xx.xx.xx</tel>
    <mel>noromeb85@gmail.com</mel>
  </coordonnées>
  <diplôme>Baccalauréat </diplôme>
  <année>2019</année>
  <lieu>Khenchela</lieu>
  <mention>Bien</mention>
</cv>
```

### 2). La DTD associés:

```

<!ELEMENT cv
<!ELEMENT coordonnées (nom,prenom,adresse,dateDeNaissance,tel,mel) >
<!ELEMENT nom (#PCDATA) >
<!ELEMENT prenom (#PCDATA) >
<!ELEMENT dateDeNaissance (#PCDATA) >
<!ELEMENT tel (#PCDATA) >
<!ELEMENT mel (#PCDATA) >
<!ELEMENT adresse (codepostal,ville) >
<!ELEMENT code postal (#PCDATA) >
<!ELEMENT ville (#PCDATA) >
<!ELEMENT année (#PCDATA) >
<!ELEMENT diplôme (#PCDATA) >

```

## Exercice 02:

### 1).Exemple xml validés par la DTD :

```

<?xml version="1.0" encoding="UTF-8"?>
<person>
  <name>
    <family> ..... </family>
    <given> ..... </given>
  </name>
  <email>.....</email>
  <link>.....</link>
</person>

```

### 2).Schéma xml équivalent la DTD :

```

<?xml version="1.0"?>
<xs:schema version="1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xsd:element name="person">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:ATTLIST ref="person photo"/>
        <xsd:ATTLIST ref="person salary"/>

```

```

        <xsd:element name="name"/>
        <xsd:complextype>
            <xsd:sequence>
                <xsd:element ref="family"/>
                <xsd:element ref="given"/>
            </xsd:sequence>
        </xsd:complextype>
    </xsd:sequence>
</xsd:complextype>
</xsd:element>
</xs:schema>

```

### **Exercice 03:**

#### **1).Code xml équivalent :**

```

<?xml version="1.0" encoding="UTF-8"?>
<etudiants>
    <etudiant >
        <nom> talbi </nom>
        <prenom> houssem </prenom>
        <note> 14 </note>
    </etudiant>
    <etudiant >
        <nom> hassad</nom>
        <prenom> alli</prenom>
        <note> 12 </note>
    </etudiant>
    <etudiant >
        <nom> mah</nom>
        <prenom> raf</prenom>
        <note> 11 </note>
    </etudiant>
</etudiants>

```

### **Exercice 04:**

#### **première partie :**

## 1).Code du fichier “rafikxsl.xsl” :

```
<?xml version="1.0"?>
<xs:schema version="1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xsd:element name="personne">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="etudiant"/>
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="nom" type="xs:string"/>
            <xsd:element name="prenom" type="xs:string"/>
            <xsd:element name="niveau" type="xs:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xs:schema>
```

## 2).Le schéma xml “etude.xsd” :

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:output method="html"/>
  <xsl:template match="/">
    <html>
      <head>
        <title>newstylesheet.xsl</title>
      </head>
      <body>
        <xsl:apply-template/>
      </body>
    </html>
  </xsl:template>
```

```

<xsl:template match="personne">
  <table border="2">
    <tr>
      <th>nom</th>
      <th>prenom</th>
      <th>niveau</th>
    </tr>
    <xsl:for-each select="etudiant">
      <xsl:sort select="nom"/>
      <tr>
        <td> <xsl:value-of select="nom"/> </td>
        <td> <xsl:value-of select="prenom"/> </td>
        <td> <xsl:value-of select="niveau"/> </td>
      </tr>
    </xsl:for-each>
  </table>
</xsl:template>
</xsl:stylesheet>

```

## deuxième partie :

[#Dans un fichier xml "books.xml" situé dans le dossier : bookstore/book on cherche les titres de livres à moins de 30 \\$ et on les renvoie \(les titres\).](#)

```

for $x in doc("books.xml")/bookstore/book
where $x/price>30
order by $x/title
return $x/title

```

