

Examen Web/XML

*Durée : 2 heures, Documents autorisés***QCM (2,5 points)**

Une seule réponse par question. 0,5 point par bonne réponse. -0,25 par mauvaise réponse.
Vous pouvez répondre directement sur la feuille.

A - Que fait un parseur lorsqu'il rencontre une déclaration d'espace de nommage ?

- Il en parle à son parfrère
- Il vérifie que l'URI existe bien
- Rien
- Il charge la DTD trouvée à l'adresse spécifiée par l'URI
- Il vérifie la syntaxe de l'URI

B - Quelle techno est la plus efficace pour transformer en HTML une toute petite partie des données d'un document XML très volumineux ?

- DOM
- SAX
- XSLT
- Regexp avec Perl

C - Que contient le paramètre test `<xsl:variable name="test" select="/..">` ?

- Un ensemble de nœuds vide
- Tous les nœuds du document XML en entrée
- L'élément racine du document XML en entrée
- La racine du document XML en entrée

D - Dans le DOM, combien de nœuds **Text** consécutifs peut on trouver ?

- Aucun car ils sont tous transformés en nœuds **CDATASection**
- Un seul puisque les informations textuelles adjacentes sont fusionnées
- Plusieurs s'il y a des appels d'entités ou des sections CDATA ou des appels de caractères dans le texte
- Plusieurs car on peut dissocier les nœuds **Text** par programmation

E - Laquelle de ces grammaire XML ne se préoccupe pas de la mise en forme ?

- SVG
- XSLFO
- Docbook
- XHTML

Opérations sur les ensembles avec XPath (1,5 points)

Il n'y a pas d'opérateurs ensemblistes dans XPath, en revanche, on peut facilement les obtenir dans XSLT.
A quelle opération ensembliste correspond chacune des expressions suivantes ?

- `<xsl:copy-of select="$node-set1[count(. | $node-set2) = count($node-set2)]"/>`
- `<xsl:copy-of select="$node-set1 | $node-set2"/>`
- `<xsl:copy-of
select="$node-set1[count(. | $node-set2) != count($node-set2)] |
$node-set2[count(. | $node-set1) != count($node-set1)]"/>`

Examen Web/XML

*Durée : 2 heures, Documents autorisés***Problème XSLT (2 points)**

La fonction XSLT **document** () permet de charger un document XML dans une feuille de style XSLT. Elle retourne un **node-set** de ce document.

Lorsque le premier argument est une chaîne vide : **document** ('') le document XML chargé est la feuille de style elle-même. C'est très pratique pour définir des données structurées constantes, et pour y accéder : puisqu'un document XSLT est aussi un document XML, on peut utiliser XPath pour y accéder.

A partir du code suivant, écrire les expressions qui permettent de transformer un attribut **date** avec le nom du mois en clair (**24 février 2003** au lieu de **24-2-2003**).

On utilisera les fonctions XPath **substring-before** () et **substring-after** () pour extraire le jour le mois et l'année de la date.

On pourra utiliser des variables intermédiaires pour obtenir la valeur de la variable **nom-mois**.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:date="*** Pour obtenir le nom des mois ***">

  <xsl:output method="html" />

  <date:noms-mois>
    <date:mois court="jan">janvier</date:mois>
    <date:mois court="fév">février</date:mois>
    <!-- etc -->
    <date:mois court="déc">décembre</date:mois>
  </date:noms-mois>

  <xsl:template match="@date">
    <!--date au format j-m-aaaa -->
    <xsl:variable name="jour" select="mettre ici l'expression appropriée" />
    <xsl:variable name="mois" select="ici aussi" />
    <xsl:variable name="année" select="ici aussi" />
    <xsl:variable name="nom-mois" select="utiliser la fonction document() ici" />

    <xsl:text/>
    le <xsl:value-of select="$jour" />
    <xsl:text>&#160;</xsl:text>
    <xsl:value-of select="$nom-mois" />
    <xsl:text>&#160;</xsl:text>
    <xsl:value-of select="$année" />
  </xsl:template>

  <!-- etc -->
</xsl:stylesheet>
```

Exemple de donnée du document XML en entrée (on considère que la règle XSLT ci-dessus est appelée) :

```
<truc date="24-2-2003" />
```

Exemple de résultat produit :

```
Le 24 février 2003
```

Examen Web/XML

Durée : 2 heures, Documents autorisés

Syntaxe XML (4 points)

L'auteur du document XML suivant a cru pouvoir rédiger son document en se passant d'un éditeur XML. Aidez le à corriger les **16** erreurs qu'il contient, pour qu'il puisse être parsé.

Répondez sur la feuille : entourez chaque erreur et annotez la.

```
<?xml version="1.0" encoding="UTF-8" standalone="standalone"?>
<?xml-stylesheet type="text/xsl" href="fg.xsl"?>
<!-- à relire -->
<?robots index="yes" follow="no"?>
<document>
  <titre style="bold" style="big">Mes mémoires</titre>
  <auteur>
    <nom>Gump</nom>
    <prénom>Forrest</prénom>
  <auteur>
  <description xmlns="http://www.w3.org/1999/xhtml"
    style="bold" Style="big" xml:space="preserve">
    <!-- à partir d'ici, on peut utiliser
    des éléments HTML <!-- et d'autres aussi -->
    c'est pratique pour du contenu
    documentaire -->
    <p xml:space="default" align=center>Ma maman disait
      toujours : &#xA; "<i>la vie c'est comme une
      bo&icirc;te de <b>chocolat</i>, on ne sait jamais
      sur quoi on va tomber</b>".</p>
    <hr width/>
    <a xlink:href="bubbagump.avi"
      xmlns:xlink="http://www.w3.org/1999/xlink"
      href="bubbagump.avi"
      xlink:role="mon film">
      <object xmlns:xlink=""
        xlink:href="bubbagump.avi"/>
    </a>
    <script language="JavaScript">
      <![CDATA[
        function check() {
          for (int i=10; i>0; i++) {
            if ( a[b[i]]>5 ) break;
          }
        }
      ]]>
      <!-- vérifier si la boucle doit s'écrire
      avec i-- et pas i++ -->
    </script>
    <p>La suite, je ne m'en souviens plus...
  </description>
  <xml:parse processor="JAXP"/>
</document>
<remarques>
  Mon document ne parse pas
</remarques>
<!-- il y a quelques erreurs -->
fin du document
```