

FMIN201 : Galaxie XML
3 mai 2010 - 13h-15h : A.5.05 (38)
Pierre Pompidor - Tout document autorisé

Soit le fichier de données XML et le code XSL suivants :

Fichier de données XML

```
<?xml version="1.0" ?>
<?xml-stylesheet href="examen_GXML_FMIN201_2010_session_1.xsl" type="text/xsl" ?>
<A>
  <B id="1"> B1
    <C id="2"> C1
      <B id="3"> B2 </B>
    </C>
    <C id="4"> C2 </C>
  </B>
  <B id="5"> B3 </B>
  <B id="6"> B4 </B>
  <B id="7"> B5
    <C id="8"> C3 </C>
  </B>
</A>
```

Code XSL (examen_GXML_FMIN201_2010_session_1.xsl)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dyn="http://exslt.org/dynamic">

<xsl:output method="html" encoding="ISO-8859-1" />
<xsl:strip-space elements="*" />

<xsl:template match="/">
  <html>
    <body>
      <ul>
        <xsl:apply-templates select="//*" />
      </ul>
    </body>
  </html>
</xsl:template>

<xsl:template match="*[count(*) = 0]">
  <xsl:call-template name="parcours">
    <xsl:with-param name="courant" select="@id" />
    <xsl:with-param name="liste" select="name()" />
  </xsl:call-template>
</xsl:template>

<xsl:template match="*">
</xsl:template>
```

```

<xsl:template name="parcours">
  <xsl:param name="courant" /> <xsl:param name="liste" />
  <xsl:variable name="xpath" select="concat('//', $liste, '[count(*) = 0]')" />
  <xsl:variable name="comptage" select="count(dyn:evaluate($xpath))" />
  <li> <xsl:value-of select="$xpath" /> : <xsl:value-of select="$comptage" /> fois </li>
  <xsl:choose>
    <xsl:when test="//*[*/@id = $courant]/@id">
      <xsl:call-template name="parcours">
        <xsl:with-param name="courant" select="//*[*/@id = $courant]/@id" />
        <xsl:with-param name="liste" select="concat(name(//*[*/@id = $courant]), '/', $liste)" />
      </xsl:call-template>
    </xsl:when>
  </xsl:choose>
</xsl:template>

</xsl:stylesheet>

```

1 Expliquez le fonctionnement du programme XSL (12 points) :

- en spécifiant ce qui devrait s'afficher dans le navigateur (ou dans la console en utilisant xsltproc) (7 points)
- en expliquant la signification des expressions XPath suivantes :
 - //* (0,5 point)
 - *[count(*) = 0] (1,5 points)
 - //*[*/@id = \$courant]/@id (3 points)

2 Expliquez les fonctionnalités du code JQUERY suivant (4 points) :

```

<html>
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <script src="jquery-1.4.2.js" type="text/javascript"></script>
    <script type="text/javascript">
      $(document).ready(function() { var html = "<option> Faites votre choix </option>";
        $.getJSON('donnees.json', function(data) {
          $.each(data, function(entryIndex, entry) {
            html += "<option>"+entry['nom']+"</option>";
          });
          $('select:eq(0)').append(html);
        });
        $('select:eq(0)').change(function(){
          var elem = "select:eq(0) option:selected";
          $("select:eq(1)").append($("<option></option>").text($(elem).text()));
          $(elem).remove();
        });
      });
    </script>
  </head>
  <body> <select/> <select/> </body>
</html>

```

3 Expliquez les fonctionnalités du code FLEX suivant par rapport aux données XML (toujours sans paraphrase...) (4 points) :

```
<?xml version="1.0" encoding="utf-8"?>
<donnees>
  <donnee nom="d1">
    contenu donnee 1
  </donnee>
  <donnee nom="d2">
    contenu donnee 2
  </donnee>
  ...
</donnees>
```

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"
  creationComplete="initApp()" >

<mx:Script>
<![CDATA[
  import mx.controls.Alert;
  import mx.containers.Panel;
  import mx.controls.Text;
  import mx.rpc.events.ResultEvent;

  private function initApp():void { serv.send(); }

  private function serv_result(evt:ResultEvent):void {
    var xml:XML = XML(evt.result);
    var donnees:XMLList = xml..donnee;
    for each(var donnee:XML in donnees) {
      creePanel(donnee.@nom, donnee.text());
    }
  }

  private function creePanel(nom:String, contenu:String):void {
    var panel:Panel = new Panel();
    panel.label=nom;
    panel.title=nom;
    var texte:Text = new Text();
    texte.text = contenu;
    panel.addChild(texte);
    viewStack.addChild(panel);
  }
]]>
</mx:Script>

<mx:HTTPService id="serv" url="donnees.xml" resultFormat="e4x" result="serv_result(event);" />
<mx:VBox>
  <mx:LinkBar dataProvider="{viewStack}" />
  <mx:ViewStack id="viewStack" />
</mx:VBox>
```

</mx:Application>