

# Techniques à Objets et Internet - 2007

## Programmation Web (J.-F. Perrot)

### *Un exemple d'exploitation multimodale d'un document XML : le menu du restaurant*

- *affiché dans le navigateur via XSLT-HTML*
- *imprimé en PDF via XSLT-FO*
- *alimentant un dialogue avec AJAX-DOM*

#### I. [Le document XML en question : fichier Menu/Menu.xml](#)

#### II. [Affichage par le navigateur](#)

1. [La transformation XSLT vers HTML \(fichier Menu/Menu.xsl\)](#)
2. [La feuille de style CSS \(fichier Menu/Menu.css\)](#)
3. [Le code JavaScript qui engendre la date](#)

#### III. [Le fichier PDF engendré](#)

1. [La transformation XSLT vers FO \(\*Formatting Objects\*\)](#)
2. [Mise en œuvre effective de la transformation vers PDF](#)

#### IV. [Le dialogue "commande au restaurant" en AJAX](#)

1. [Le code-source de la page](#)
2. [L'inspecteur DOM à l'étape "choix du plat principal"](#)
3. [Le fichier montrerMenu.js qui gère l'affichage du menu](#)
4. [Le fichier menuAjax.js, qui distribue le jeu](#)
5. [Le script destinataire d'Ajax : GestMenu.php](#)
6. [Le fichier menuDOM.js, notre spécialiste DOM](#)
7. [Enfin la batterie de fonctions auxiliaires indispensables, fichier auxDOM.js](#)

#### V. [Note sur l'usage de tcpmon](#)

#### I. Le document XML en question : fichier Menu/Menu.xml

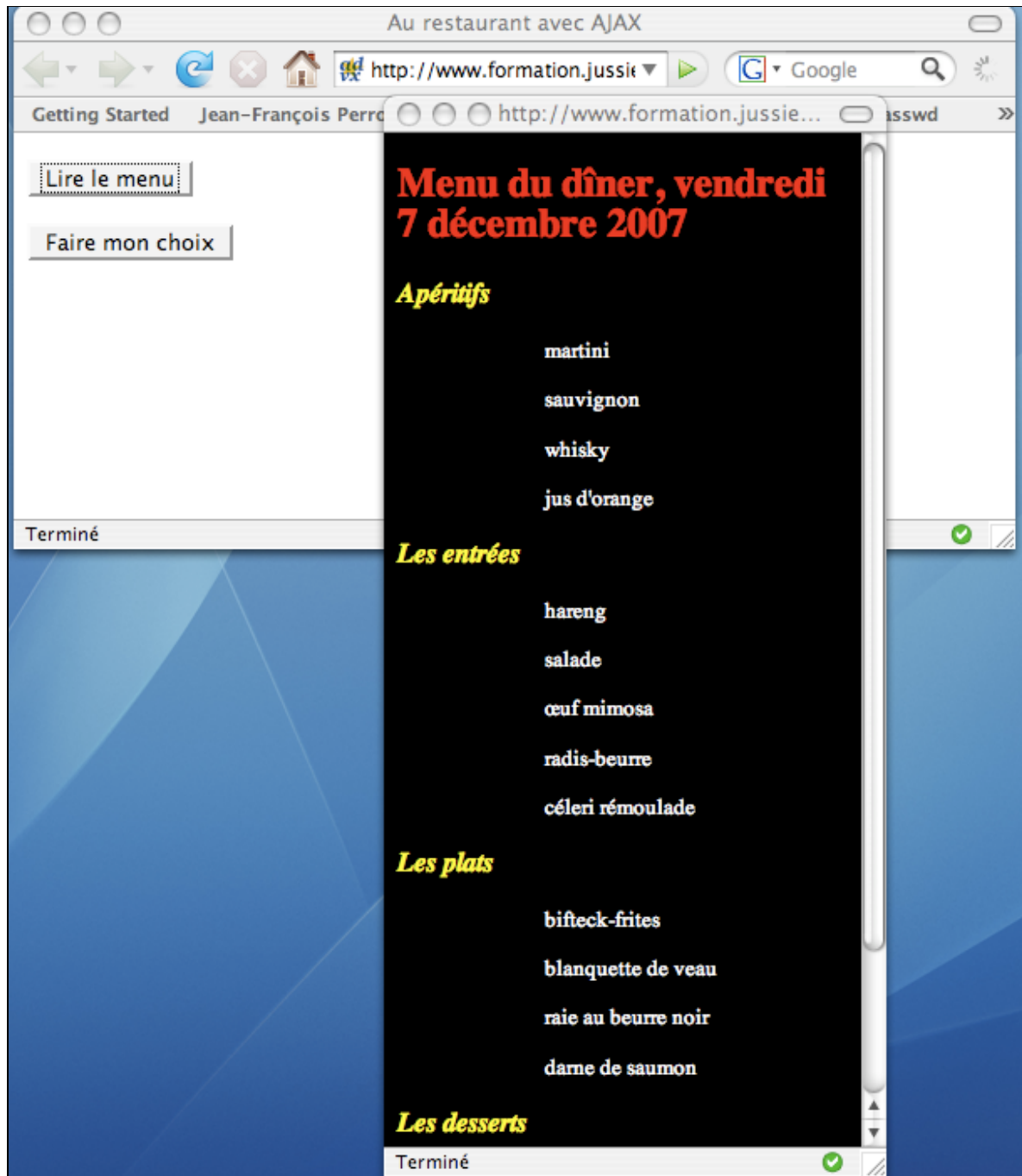
---

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="Menu.xsl"?>
```

```
<liste>
  <étape nom="apéritif">
    <annonce>Vous boirez bien quelque chose ?</annonce>
    <choix>martini</choix>
    <choix>sauvignon</choix>
    <choix>whisky</choix>
    <choix>jus d'orange</choix>
  </étape>
  <étape nom="entrée">
    <annonce>Choisissez d'abord l'entrée</annonce>
    <choix> hareng</choix>
    <choix> salade </choix>
    <choix> œuf mimosa </choix>
    <choix> radis-beurre </choix>
    <choix> céleri rémoulade </choix>
  </étape>
  <étape nom="plat">
    <annonce>Choisissez ensuite votre plat principal</annonce>
    <choix> bifteck-frites</choix>
    <choix> blanquette de veau</choix>
    <choix> raie au beurre noir </choix>
    <choix> darne de saumon </choix>
  </étape>
  <étape nom="dessert">
    <annonce>Enfin, quel dessert vous ferait plaisir ?</annonce>
    <choix> fruits de saison </choix>
    <choix> crème glacée </choix>
    <choix> glace </choix>
  </étape>
  <étape nom="fin">
    <annonce>Bon appétit !</annonce>
  </étape>
</liste>
```

---

## II. Affichage par le navigateur



## 1. La transformation XSLT vers HTML (fichier Menu/Menu.xsl)

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version='1.0'>

<!-- ===== Pre'sentation du menu ===== -->

<xsl:output method="html"/>
<xsl:template match="/"> <!-- Le "cadre HTML -->
  <html>
    <head>
      <meta content="text/html; charset=UTF-8" http-equiv="content-type" />
      <title>Restaurant : le menu</title>
      <link rel="stylesheet" type="text/css" href="Menu.css"/>

      <script type="text/javascript" src="laDate.js" />
    </head>
  </html>
</template>
</xsl:stylesheet>
```

```

</head>
<body onload="faireLaDate();">

  <h2 id="date"> Menu du jour </h2>
    <xsl:apply-templates/>

</body>
</html>
</xsl:template>

<!-- ===== Traitement au niveau "étape" ===== -->

<xsl:template match="liste/étape">
  <xsl:choose>
    <xsl:when test="@nom='apéritif'">
      <h3 id='apéritif'><i>Apéritifs</i></h3>
    </xsl:when>
    <xsl:when test="@nom='entrée'">
      <h3 id='entrée'><i>Les entrées</i></h3>
    </xsl:when>
    <xsl:when test="@nom='plat'">
      <h3 id='plat'><i>Les plats</i></h3>
    </xsl:when>
    <xsl:when test="@nom='dessert'">
      <h3 id='dessert'><i>Les desserts</i></h3>
    </xsl:when>
  </xsl:choose>
  <blockquote>
    <xsl:apply-templates select=".*"/>
  </blockquote>
</xsl:template>

<!-- ===== Traitement au niveau "intérieur à l'étape" ===== -->

<xsl:template match="choix">
  <p><xsl:value-of select="."/></p>
</xsl:template>

<xsl:template match="annonce">
  <!-- ne rien faire -->
</xsl:template>

<!-- ===== et c'est tout ===== -->

</xsl:stylesheet>

```

---

## 2. La feuille de style CSS (fichier Menu/Menu.css)

---

```

/* Un style pour les menus...*/

body { background : black;
      font-family : Arial,Helvetica,sans-serif;
}

h2 { color : red }

h3 { color : yellow;
     font-weight : bold;
     font-size : 12pt;

```

```
        line-height : 14pt;
    }
    p { color : white;
        font-size : 10pt;
        line-height : 12pt;
        text-indent : 0.5in;
    }
```

---

### 3. Le code JavaScript qui engendre la date

et l'insère dans la page via DOM (fichier Menu/laDate.js)

---

```
// Calculer la date du jour et l'afficher

function faireLaDate () {
    var tabJours=["dimanche","lundi", "mardi", "mercredi", "jeudi", "vendredi",
                  "samedi"];

    var tabMois=["janvier", "f\u00E9vrier", "mars", "avril", "mai", "juin",
                "juillet", "ao\u00FBt", "septembre", "octobre", "novembre", "d\u00E9cembre"];

    var aujourdHui = new Date();

    var leQuantieme = aujourdHui.getDate();
    var leJourDeLaSemaine = tabJours[aujourdHui.getDay()];
    var leMois = tabMois[aujourdHui.getMonth()];
    var lAnnee = aujourdHui.getFullYear();
    var lHeure = aujourdHui.getHours();

    var repas = lHeure<15?"d\u00E9jeuner":"d\u00EEner";

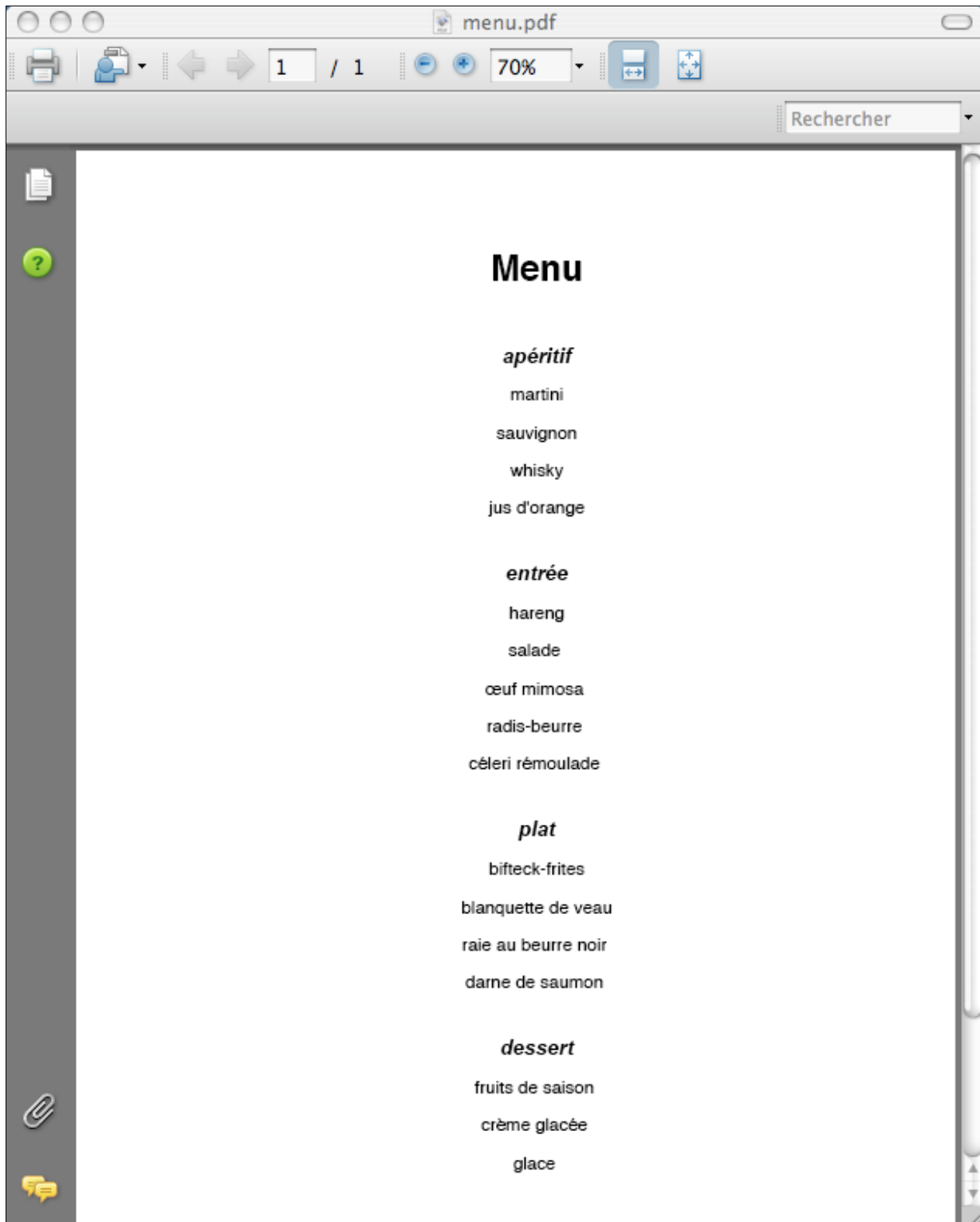
    var msg = "Menu du "+repas+", "
              + leJourDeLaSemaine+" "+ leQuantieme+" "+leMois+" "+lAnnee;

    var leTitre = document.getElementById("date");

    leTitre.replaceChild(document.createTextNode(msg), leTitre.firstChild);
}
```

---

## III. Le fichier PDF engendré



## 1. La transformation XSLT vers FO (*Formatting Objects*)

- fichier VersPDF/menu2pdf.xsl)

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fo="http://www.w3.org/1999/XSL/Format"
  version='1.0'>
```

```

<xsl:output method="xml" indent="yes" encoding="utf-8" />

<xsl:template match="/">
<fo:root>
  <fo:layout-master-set>
    <fo:simple-page-master master-name="menu"
      page-height="29.7cm"
      page-width="21cm"
      margin-top="1cm"
      margin-bottom="1cm"
      margin-left="1.5cm"
      margin-right="0.5cm">
      <fo:region-body margin-top="1cm" margin-bottom="0cm" column-count="1"/>
      <fo:region-before extent="3cm"/>
      <fo:region-after extent="1.5cm"/>
    </fo:simple-page-master>
  </fo:layout-master-set>

  <fo:page-sequence master-reference="menu">
    <fo:flow flow-name="xsl-region-body">

      <fo:block text-align="center">

        <fo:block font-size="24pt" font-weight="bold"
          space-before="12pt" space-after="12pt">
          <xsl:text>Menu</xsl:text>
        </fo:block>

        <xsl:apply-templates select="liste/étape[@nom != 'fin']"/>

      </fo:block>

    </fo:flow>
  </fo:page-sequence>
</fo:root>

</xsl:template>

<xsl:template match="étape">
  <fo:block keep-together="always">

    <fo:block font-size="14pt" font-weight="bold" font-style="italic"
      space-before="24pt" space-after="6pt">
      <xsl:value-of select="@nom"/>
    </fo:block>

    <xsl:apply-templates select="choix" />
    <!-- on saute l'annonce -->

  </fo:block>
</xsl:template>

<xsl:template match="choix">
  <fo:block space-before="6pt" space-after="6pt">
    <xsl:value-of select="."/>
  </fo:block>
</xsl:template>

</xsl:stylesheet>

```

---

## 2. Mise en œuvre effective de la transformation vers PDF

(en Java, pour changer un peu)

fichier VersPDF/Menu2Pdf.java, qui nécessite l'installation de [Apache-FOP](#).

---

```
import org.w3c.dom.*;
import org.xml.sax.*;

import javax.xml.parsers.*;
import javax.xml.transform.*;
import javax.xml.transform.dom.*;
import javax.xml.transform.sax.*;
import javax.xml.transform.stream.*;

import org.apache.fop.apps.*;

import java.io.File;
import java.io.FileOutputStream;
import java.io.OutputStream;

public class Menu2Pdf {

    public static void main(String[] args) {

        String fichIn = args[0]; // la source XML
        String fichFst = args[1]; // la "feuille de style" XSLT
        String fichOut = args[2]; // le fichier de sortie en PDF

        Driver driver = new Driver();
        driver.setRenderer(Driver.RENDER_PDF);

        try {
            OutputStream out = new FileOutputStream(fichOut);
            driver.setOutputStream(out);
            Result res = new SAXResult(driver.getContentHandler());

            DocumentBuilder parseur =
                DocumentBuilderFactory.newInstance().newDocumentBuilder();
            Document doc = parseur.parse(fichIn);
            DOMSource don = new DOMSource(doc);

            StreamSource fst = new StreamSource(new File(fichFst));
            Transformer tf =
                TransformerFactory.newInstance().newTransformer(fst);

            tf.transform(don, res);
            out.close();

        } catch (Exception e) {
            System.out.println("Erreur " + e.getMessage());
        }
    } // main
} // Menu2Pdf
```

---

## IV. Le dialogue "commande au restaurant" en AJAX

Lancement : Menu/AuRestaurant.php



## 1. Le code-source de la page

engendré par le script PHP

```
<html>
<head>
  <meta content="text/html; charset=UTF-8" http-equiv="content-type">
  <title>Au restaurant avec AJAX</title>
  <script type="text/javascript" src="montrerMenu.js"></script>
  <script type="text/javascript" src="menuAjax.js"></script>
```

```

<script type="text/javascript" src="menuDOM.js"></script>
<script type="text/javascript" src="auxDOM.js"></script>

<script type="text/javascript">
    fichMenu = 'Menu.xml';
    suivant = 'apéritif';
</script>

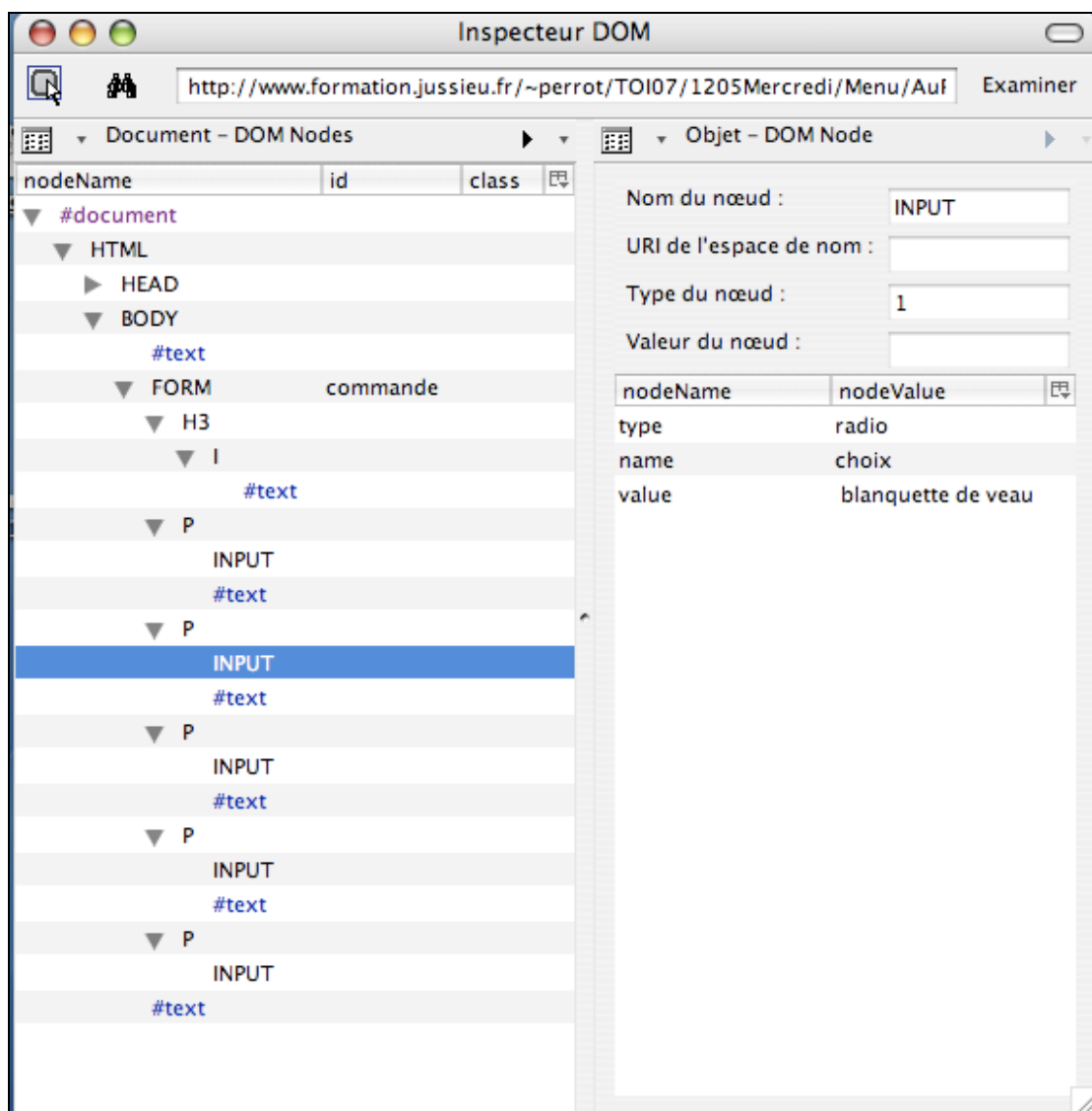
</head>
<body>
<form href="" id="commande" onsubmit="au_suivant();">
    <p><input type="button" onclick="montrerMenu();" value="Lire le menu" />
    </p>
    <p><input type="submit" value="Faire mon choix" /></p>
</form>
</body>

</html>

```

---

## 2. L'inspecteur DOM à l'étape "choix du plat principal"



### 3. Le fichier `montrerMenu.js` qui gère l'affichage du menu

---

```
// gestion de la fenêtre montrant le menu
// à partir du fichier XML (assorti d'une feuille de style XSLT et d'une CSS)
// dont le nom est contenu dans la variable fichMenu,
// laquelle est initialisée dans la partie dynamique du script, créée par PHP.

var fenMenu;
var yapaLmenu = true;

function montrerMenu() {
    if( yapaLmenu ){
        yapaLmenu = false;
        fenMenu =
            window.open(fichMenu, 'Menu', 'resizable=yes,scrollbars=yes,width=300');
    } else {
        if( fenMenu.closed ){ // même jeu
            fenMenu =
                window.open(fichMenu, 'Menu', 'resizable=yes,scrollbars=yes,width=300');
        }else{
            alert('La fen\u00Eatre est d\u00E9j\u00E0 ouverte !');
        }
    }
}
}
```

---

### 4. Le fichier `menuAjax.js`, qui distribue le jeu

---

```
/* Le menu en "AJAX-intensive"

Chaque étape comporte une requête au serveur.
L'étape suivante est donnée par la variable "suivant",
    initialisée par le script dynamique engendré par PHP initial
    (en effet, c'est le menu qui contient l'info sur le point de départ).
Les infos sur l'étape à traiter sont donc demandées au serveur par AJAX,
    via le script GestMenu.php.

*/

//===== Variables globales =====
var fichMenu;
/* le nom du fichier XML contenant le menu exploité (chaîne)
    initialisée par le script dynamique
*/
var courant = "debut";
/* le nom de l'étape en cours (chaîne)
    initialisée à une valeur spéciale testée par trouverChoix()
*/
var suivant = null;
/* le nom de l'étape à suivre (chaîne)
    doit être initialisée par le script dynamique
*/
var choisi = new Object();
/* Tableau associatif : étapes -> choix de l'utilisateur
    en vue de la récapitulation finale
*/

//===== AJAX =====

var xhr = new XMLHttpRequest();
```

```
// fonction appelée par le submit du formulaire affiché
// -----
function au_suivant(){
    var msg1 = encodeURIComponent(fichMenu);
    var msg2 = encodeURIComponent(suivant);

    trouverChoix();
    courant = suivant;
    xhr.open('GET', 'GestMenu.php?nomMenu='+msg1+'&etape='+msg2, false);
    xhr.send(null);
    etapeSuivante(xhr.responseXML);
}

function trouverChoix(){
    if( courant == "debut" ) return;
    //sinon...
    var lesBoutons = document.getElementsByName('choix');
    for ( var i=0; i<lesBoutons.length; i++ ) {
        if ( lesBoutons[i].checked ) {
            choisi[courant]= lesBoutons[i].value;
            break;
        }
    }
}

function etapeSuivante(dom){
    var oldCommande = document.getElementById('commande');
    var nouvCommande;

    if( courant !='fin' ){
        nouvCommande = suiteCommande(dom);
        suivant = nomEtapeSuivante(dom);
    }else{
        nouvCommande = recapitulation(dom);
    }
    document.body.replaceChild(nouvCommande, oldCommande);
}

```

---

## 5. Le script destinataire d'Ajax : GestMenu.php

---

```
<?php
/* Ce script extrait du fichier XML ($leMenu)
   les informations sur l'étape demandée ($etape),
   sous la forme d'un document XML ainsi conçu :
   <menu>
       <suivant>l'étape suivante</suivant>
       <annonce>l'annonce de l'étape courante</annonce>
       <etape>
           <choix>premier choix</choix>
           <choix>deuxième choix</choix>
           .....
       </etape>
   </menu>

   Le cas échéant, certains éléments pourront manquer
   (cas de l'étape finale qui engendre
   <?xml version="1.0"?>ndi/      Appli/
   <menu><suivant></suivant><annonce>Bon app&#xE9;tit !</annonce><etape/>
   </menu>)
*/

```

```

$nomMenu = $_GET['nomMenu'];
$etape = $_GET['etape'];

/*
$nomMenu = "Menu.xml";
$etape = 'plat';
*/

require_once("LireMenu.php");
$leMenu = xml_to_php($nomMenu);

$etapeSuivante = $leMenu->lesEtapes[$etape]; // chaine
$lAnnonce = $leMenu->lesAnnonces[$etape]; // chaine
$leChoix = $leMenu->lesChoix[$etape]; // tableau de chaines

$dom = new DOMDocument();
$rac = $dom->createElement("menu");
$dom->appendChild($rac);

$suiv = $dom->createElement("suivant");
$rac->appendChild($suiv);
$suiv->appendChild($dom->createTextNode($etapeSuivante));

$ann = $dom->createElement("annonce");
$rac->appendChild($ann);
$ann->appendChild($dom->createTextNode($lAnnonce));

$etp = $dom->createElement("etape");
$rac->appendChild($etp);
foreach( $leChoix as $choix ){
    $elt = $dom->createElement("choix");
    $etp->appendChild($elt);
    $elt->appendChild($dom->createTextNode($choix));
}

header("Content-Type: text/xml"); // pour responseXML
echo $dom->saveXML();

?>

```

---

Pour la lecture du fichier-source Menu.xml,  
voir [l'exemple LireMenu-2 dans le cours DOM-PHP](#).

## 6. Le fichier menuDOM.js, notre spécialiste DOM

---

```

/* on entre dans l'anatomie du dom renvoyé par GestMenu.php
à savoir
<menu>
  <suivant>l'étape suivante</suivant>
  <annonce>l'annonce de l'étape courante</annonce>
  <etape>
    <choix>premier choix</choix>
    <choix>deuxième choix</choix>
    .....
  </etape>
</menu>
*/

function nomEtapeSuivante(dom){
    return dom.getElementsByTagName('suivant')[0].firstChild.nodeValue;
}

```

```
function suiteCommande(dom){
    var lAnnonce = dom.getElementsByTagName('annonce')[0].firstChild.nodeValue;
    var elemChoix = dom.getElementsByTagName('etape')[0].childNodes;
    var lesChoix = new Array(elemChoix.length);

    for( var i=0; i< elemChoix.length; i++ ){
        lesChoix[i] = elemChoix[i].firstChild.nodeValue;
    }

    return faireForm(lAnnonce, lesChoix);
}

function recapitulation(dom){
    var lAnnonce = dom.getElementsByTagName('annonce')[0].firstChild.nodeValue;
    var table = faireTable(choisi);
    var res = document.createElement('div');

    res.appendChild (h3_it(lAnnonce));
    res.appendChild (table);
    return res;
}

// Synthèse du formulaire de l'étape
// -----
function faireForm(lAnnonce, lesChoix) {
    var leForm = document.createElement('form');
    leForm.setAttribute("onsubmit", "au_suivant();");
    leForm.setAttribute("id", "commande");

    leForm.appendChild(h3_it(lAnnonce));
    var lesBoutons = boutonsRadios('choix', lesChoix);
    for( var i=0; i<lesBoutons.length; i++ ){
        leForm.appendChild(lesBoutons[i]);
    }
    leForm.appendChild(boutonSubmit('envoyez'));

    return leForm;
}

// Construction de la table récapitulative à partir d'un tab. étapes -> choix
// -----
function faireTable(tabChoix) {
    var tb = document.createElement("table");
    var tr;
    var td;

    tb.setAttribute("border", "1");
    for( nom in tabChoix ){
        tr = document.createElement("tr");
        td = document.createElement("td");
        td.appendChild(document.createTextNode(nom));
        tr.appendChild(td);
        td = document.createElement("td");
        td.appendChild(document.createTextNode(tabChoix[nom]));
        tr.appendChild(td);
        tb.appendChild(tr);
    }
    return tb;
}
```

---

## 7. Enfin la batterie de fonctions auxiliaires indispensables, fichier `auxDOM.js` :

---

```
// Fonctions de construction des éléments de base
// -----

function h3_it(txt) { // Titre "h3" en italiques
    var par = document.createElement("h3");
    var it = document.createElement("i");
    it.appendChild(document.createTextNode(txt));
    par.appendChild(it);
    return par;
}

function boutonRadio(nom, valeur){ //empaqueté dans un <p> avec légende
    var boutRad = document.createElement("input");
    boutRad.setAttribute("type", "Radio");
    boutRad.setAttribute("name", nom);
    boutRad.setAttribute("value", valeur);

    var par = document.createElement("p");
    par.appendChild(boutRad);
    par.appendChild(document.createTextNode(valeur));
    return par;
}

function boutonsRadios(nom, lchoix) { //tableau de boutons de même nom
    var lbr = new Array(lchoix.length);
    for( var i=0; i<lchoix.length; i++){
        lbr[i] = boutonRadio(nom, lchoix[i]);
    }
    return lbr;
}

function boutonSubmit(libelle) { //empaqueté dans un <p>
    var varsub = document.createElement("input");
    varsub.setAttribute("type", "submit");
    varsub.setAttribute("value", libelle);

    var par = document.createElement("p");
    par.appendChild(varsub);
    return par;
}
//-----

// Un auxiliaire utile
// -----

function vider(elt) {
    while( elt.hasChildNodes() ){
        elt.removeChild(elt.firstChild);
    }
}

```

---

## V. Note sur l'usage de tcpmon

Dans la mise au point de ce genre de programme, on a souvent à examiner le comportement des différentes pièces du mécanisme dans le contexte normal d'exécution.

Le calcul de JavaScript peut s'observer côté client, via le navigateur,

mais texte renvoyé par le script PHP destinataire d'Ajax est plus difficile à voir.

C'est là qu'un outil visualisant les échanges TCP comme [le tcpmon d'Axis](#) révèle tout son intérêt.

Pensez-y !

TCPMonitor

Admin Port 12345

Stop Listen Port: 12345 Host: www.format Port: 80 ☐ Proxy

| State | Time                | Request Host | Target Host              | Request...                           |
|-------|---------------------|--------------|--------------------------|--------------------------------------|
| ---   | Most Recent         | ---          | ---                      | ---                                  |
| Done  | 2007-12-08 12:18:11 | localhost    | www.formation.jussieu.fr | GET /~perrot/TOI07/1205Mercredi/M... |
| Done  | 2007-12-08 12:18:36 | localhost    | www.formation.jussieu.fr | GET /~perrot/TOI07/1205Mercredi/M... |
| Done  | 2007-12-08 12:19:30 | resend       | www.formation.jussieu.fr | GET /~perrot/TOI07/1205Mercredi/M... |

Remove Selected Remove All

GET /~perrot/TOI07/1205Mercredi/Menu/GestMenu.php?nomMenu=Menu.xml&etape=ap%C3%A9ritif HTTP/1.1  
Host: www.formation.jussieu.fr:12345  
User-Agent: Mozilla/5.0 (Macintosh; U; Intel Mac OS X; fr; rv:1.8.1.2) Gecko/20070219 Firefox/2.0.0.2  
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,\*/\*;q=0.5  
Accept-Language: fr,en-us;q=0.7,en;q=0.3  
Accept-Encoding: gzip,deflate  
Accept-Charset: UTF-8,\*  
Keep-Alive: 300

HTTP/1.1 200 OK  
Date: Sat, 08 Dec 2007 11:19:37 GMT  
Server: Apache/2.2.3 (Mandriva Linux/PREFORK-1.1.20060mlcs4)  
X-Powered-By: PHP/5.1.6 with Hardening-Patch  
Content-Length: 226  
Keep-Alive: timeout=5, max=100  
Connection: Keep-Alive  
Content-Type: text/xml

```
<?xml version="1.0"?>
<menu>
  <suivant>entr&#xE9;e</suivant>
  <annonce>Vous boirez bien quelque chose ?</annonce>
  <etape>
    <choix>martini</choix>
    <choix>sauvignon</choix>
    <choix>whisky</choix>
    <choix>jus d'orange</choix>
  </etape>
```

☒ XML Format ☐ Numeric Save Resend Switch Layout Close