

TP AJAX

La technologie AJAX

Objectif : vous introduire aux principes des appels de serveurs asynchrone

Créé par : Neila Ben Lakhal

TP AJAX

La technologie AJAX

Exercice 1. Afficher l'heure sur le client et sur le serveur

Compléter les pages web **heure.html** et **heure.php** pour permettre l'affichage de l'heure locale (du client) et de l'heure distante (du serveur) lorsqu'on clique sur le lien « Demande heure » de la **heure.html** en utilisant la technologie Ajax :

Heure.html :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
  <head>
    <title>Exercice n°1 : Ajax (XMLHttpRequest)</title>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <meta http-equiv="Pragma" content="no-cache">
    <style type="text/css">
      #htm { position:absolute; left:1px; top:100px; width:250px; height:150px; font-size:16px; }
      #client { position:absolute; left:250px; top:100px; width:250px; height:150px; font-size:16px; color: red; }
      #serveur { position:absolute; left:500px; top:100px; width:250px; height:150px; font-size:16px; color: blue; }
    </style>
    <script type="text/javascript">
      function requete(){//à compléter

      //cette fonction :
      //1-fourni l'heure client (utiliser l'objet Date (JS)) et l'affiche à l'endroit adéquat de //heure.html
      //2-envoie une requête ajax demandant la page heure.php pour donner l'heure coté serveur et //l'afficher à l'endroit adéquat de heure.html
      }
    </script>
  </head>
  <body>
    <h1>Exemple n°1 : Ajax (XMLHttpRequest)</h1>
    <div id="htm">
      &nbsp;&nbsp;&nbsp;<span style="cursor: pointer; text-decoration: underline"
      onclick="requete()">Demande heure</span>
    </div>
    <div id="client"><!-- heure client s'affichera ici --></div>
    <div id="serveur"><!-- heure serveur s'affichera ici --></div>
  </body>
</html>
```

Heure.php :

```
<?php
// à compléter : utiliser la fonction date() qui renvoi la date serveur
?>
```

Exercice 2. Lecture d'un document XML se trouvant dans le serveur avec AJAX

- Créer le fichier **dvd.xml** qui fournit la liste d'items suivants :

```
<?xml version="1.0" encoding="utf-8"?>
<dvd>
  <item>
    <title>Princesse Mononoké</title>
    <director>Hayao Miyazaki</director>
    <price>14,99</price>
  </item>
  <item>
    <title>Mon voisin Totoro</title>
    <director>Hayao Miyazaki</director>
    <price>19,99</price>
  </item>
  <item>
    <title>Blood, The Last Vampire</title>
    <director>Hiroyuki Kitakubo</director>
    <price>11,99</price>
  </item>
</dvd>
```

- Compléter page web **affichage-dvd.html** qui permettra l'affichage d'une liste de DVD **dvd.xml** en utilisant la technologie Ajax (voir capture d'écran de la page suivante) :

affichage-dvd.html :

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
  <head>
    <title>Ajax (XML)</title>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <style type="text/CSS">
      #header { }
      #content { }
      .item { border-left: 2px solid red; padding-left: 5px; }
      .title { font-size:18px; color: blue; }
      .author { font-size:14px; color: black; }
      .price { font-size:16px; color: green; }
    </style>
    <script type="text/javascript">
function getXMLHttpRequest(){ // à completer
}

var xhr = getXMLHttpRequest();

function requete()

  xhr.onreadystatechange = handleHttpResponse;

{ // à completer
}

  function handleHttpResponse()
{ // à completer
}

    </script>
  </head>
  <body>
    <h1>Ajax (XML)</h1>
    <div id="header">
      &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<span style="cursor: pointer; text-decoration: underline"
onclick="requete()">Requête liste DVD</span>
    </div>
    <div id="content"> affichage ici</div>
  </body>
</html>

```

Ajax (XML)

Requête liste DVD

Princesse Mononoké

Réalisateur : Hayao Miyazaki

Prix : 14,99 €

Mon voisin Totoro

Réalisateur : Hayao Miyazaki

Prix : 19,99 €

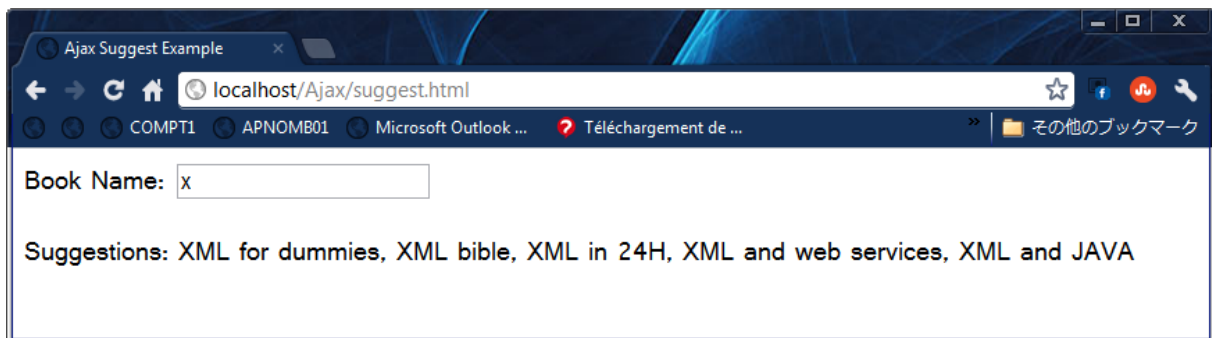
Blood, The Last Vampire

Réalisateur : Hiroyuki Kitakubo

Prix : 11,99 €

Exercice 3.

- À l'instar de GOOGLE SUGGEST, le but de cet exercice est de développer la page web **suggest.html** (voir capture d'écran) :



- Explication : Lorsque l'utilisateur insère « x » par exemple, l'événement JS **onkeyup** se produit, une requête Ajax est envoyée au serveur à une page PHP **gethint.php** qui se chargera de chercher tous les livres qui commencent par la chaîne insérée et de les retourner. À fur et à mesure que la saisie avance, la liste de nom de livres suggérés change.

- Pour vous aider, le fichier xmlFn.js est fourni pour vous avec les fonctions utiles pour gérer des requêtes AJAX :

```

/* Helper functions for using XMLHttpRequest (XHR) and
 * accessing XML documents
 */

/*
 * sendXHR - takes a URL
 *           - sets up the event handler to call
 *             a function named stateChange()
 *           - NOTE: you must write a function named doStuff()
 *           - sends a request for the given URL
 *           - xmlhttp must be a global variable
 */
function sendXHR(url)
{
    if (xmlhttp == null) {
        xmlhttp = new XMLHttpRequest();
    }
    if (xmlhttp == null) {
        alert ("Your browser does not support XMLHttpRequest.");
        return;
    }
    xmlhttp.onreadystatechange = stateChange;
    xmlhttp.open ("GET", url, true);
    xmlhttp.send (null);
}

/* stateChange - event handler called from sendXHR
 *               - if the response has loaded and
 *                 the status code is 200 (OK), call a
 *                 function named doStuff()
 */
function stateChange()
{
    if (xmlhttp.readyState == 4) {
        if (xmlhttp.status == 200) {
            // status is 200 OK
            doStuff(); // you must write this function
        } else {
            alert ("Problem retrieving data: " + xmlhttp.statusText);
        }
    }
}

/*
 * openXML - returns an xmlDoc object that holds
 *           the contents of an XML file
 */
function openXML (xmlFilename)
{
    var xmlDoc = null;

    // send the request for the XML file
    var xmlhttp = new window.XMLHttpRequest();
    xmlhttp.open ("GET", xmlFilename, false);
    xmlhttp.send (null);

    // process the response
    if (xmlhttp.responseXML != null) {
        xmlDoc = xmlhttp.responseXML.documentElement;
    } else {
        alert ("Cannot load the XML file " + xmlFilename);
    }
    return xmlDoc;
}

```

- Travail demandé : Compléter les fichiers : suggest.html et suggest.php

- **Suggest.html**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Ajax Book Suggest </title>
<script type="text/javascript" src="xmlFn.js"></script>

<script type="text/javascript">

var xmlhttp = null;

function showHint(str)
{
    // check to see if textbox is blank
    //à compléter
}

    // build request to run PHP script on server
var url = "gethint.php";
url = url + "?q=" + str;

    // send request to PHP script
sendXHR(url);
}

function doStuff()
{
    //à compléter
}

</script>
</head>

<body>

<form action="">
Book Name:
<input type="text" id="txt1" onkeyup="showHint(this.value)" />
</form>

<p>Suggestions: <span id="txtHint">suggestion ici</span></p>

</body>
</html>
```

- **Gethint.php :**

```
<?php
// Fill up array with book names
$names = array ("Web programming", "XML for dummies", "XML bible", "J2EE for dummies", "J2EE
complete guide", "SOA guide", "XML in 24H", "teach yourself Java", "Java bible", "XML and web
services", "XML and JAVA", ".NET web services");

//à completer

// get the q parameter from the URL
$q = $_GET["q"];

// lookup all hints from array if length of q > 0

// Set output to "no suggestion" if no hint was found
// or to correct value

// output the response
echo $response;
?>
```