

Ex 1.

Afficher le contenu d'un fichier texte distant dans une page web en utilisant l'objet XMLHttpRequest.

1. A partir d'un éditeur de texte, créer la page web AjaxTxt.html qui permettra l'affichage du contenu d'un fichier texte en utilisant la technologie Ajax.

```
<!DOCTYPE html>
<html>
<head>
<script src="Codes.js"></script>
</head>
<body>
<h2>AJAX - XMLHttpRequest</h2>
<p id="demo">AJAX change ce texte.</p>
<button type="button" onclick="requete()">Changer texte</button>
</body>
</html>
```

2. Créer le fichier ajax_info.txt qui définit le texte à afficher.

```
<h1>Bonjour MMI ...</h1>
```

3. Créer le fichier Codes.js qui définit la fonction requete().

```
function requete() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("demo").innerHTML = this.responseText;
        }
    };
    xhttp.open("GET", "ajax_info.txt", true);
    xhttp.send();
}
```

4. Tester (attention l'accès à la page web se fait à partir d'une URL vers le serveur : http://localhost/....).

Ex 2.

Afficher l'heure locale (du client) et l'heure distante (du serveur) en utilisant l'objet XMLHttpRequest.

1. A partir d'un éditeur de texte, créer la page web AjaxHeure.html qui permettra l'affichage des heures en utilisant la technologie Ajax.

```
<!DOCTYPE html>
<html>
<head>
<script src="Codes.js"></script>
</head>
<body>
<h1> Ajax - XMLHttpRequest - PHP</h1>
<div id="htm">
<button type="button" onclick="requete()">Demande heure</button>
</div>
<div id="client"><!-- heure locale --></div>
<div id="serveur"><!-- heure distante --></div>
</body>
</html>
```

2. Créer le fichier heure.php qui définit le texte de l'heure serveur à afficher.

```
<?php
$Lheure=date("H:i:s");
echo "Heure(serveur):".$Lheure."\n";
// renvoie la réponse
?>
```

3. Créer le fichier Codes.js qui définit la fonction requete().

```
function requete() {
// Affichage de l'heure locale (le client)
var d = new Date();
var h = d.getHours();
var m = d.getMinutes();
var s = d.getSeconds();
document.getElementById('client').innerHTML = ("Heure (client) : " + h + ':' + m + ':' + s);
// Affichage de l'heure distante (le serveur )
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
document.getElementById("serveur").innerHTML = xhttp.responseText;
}
};
/* open(mode, url, boolean);
* - mode: type de requête, GET ou POST
* - url: l'endroit ou trouver les données, un fichier avec son chemin sur le disque.
* - boolean: true (asynchrone) / false (synchrone). */
xhttp.open("GET", "heure.php", true);
xhttp.send(null);
}
```

4. Tester.

Ex 3. Utilisation d'une requête GET ou POST avec l'objet XMLHttpRequest.

1. A partir d'un éditeur de texte, créer la page web AjaxGETPOST.html qui permettra l'affichage du résultat de la requête.

```
<!DOCTYPE html>
<html>
<head>
<script src = "Codes.js"></script>
</head>
<body>
<h2>The XMLHttpRequest Object</h2>
<button type="button" onclick="requeteGet()">Request data GET</button>
<button type="button" onclick="requetePost()">Request data
POST</button></br>
<p>Get : <span id="demoGet"></span></p>
<p>Post : <span id="demoPost"></span></p>
</body>
</html>
```

2. Créer le fichier demo_get.php qui définit le texte à afficher.

```
<?php
$Prenom = $_GET['fname'];
$Nom = $_GET['lname'];
echo "Bonjour ".$Nom." ".$Prenom;
?>
```

3. Créer le fichier demo_post.php qui définit le texte à afficher.

```
<?php
$Prenom = $_POST['fname'];
$Nom = $_POST['lname'];
echo "Bonjour ".$Nom." ".$Prenom;
?>
```

4. Créer le fichier Codes.js qui définit les fonctions requeteGET() et requetePOST().

```
function requeteGet() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("demoGet").innerHTML = this.responseText;
        }
    };
    xhttp.open("GET", "demo_get.php?fname=Henry&lname=Ford", true);
    xhttp.send();
}

function requetePost() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("demoPost").innerHTML = this.responseText;
        }
    };
    xhttp.open("POST", "demo_post.php", true);
    xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    xhttp.send("fname=Henry&lname=Ford");
}
```

5. Tester.

Ex 4. Afficher le contenu d'un fichier XML en utilisant l'objet XMLHttpRequest.

1. A partir d'un éditeur de texte, créer la page web AjaxXML.html qui permettra l'affichage des données du fichier XML.

```
<!DOCTYPE html>
<html>
<head>
<script src = "Codes.js"></script>
</head>
<body>
<h2>XMLHttpRequest - Lecture fichier XML</h2>
<p>AJAX XML</p>
<button type="button" onclick="requeteXML()">Lecture XML</button>
<div id="demo"></div>
</body>
</html>
```

2. Créer le fichier file.xml qui définit les données à afficher.

```
<?xml version="1.0" ?>
<data>
<text>Hello World !</text>
<text>Hello Toulon !</text>
<text>Hello John !</text>
</data>
```

3. Créer le fichier Codes.js qui définit la fonction requeteXML().

```
function requeteXML()
{
var xhr = new XMLHttpRequest();
    xhr.onreadystatechange = function()
    {
        // readyState 4 = requête terminée et status 200 = page requêtée trouvée
        if (xhr.readyState == 4 && xhr.status == 200)
        {
            ajax_parseXml(xhr.responseXML); // Traitement XML
        }
    }
    // Préparation de la requête
    var url = 'file.xml';
    xhr.open('GET', url, true);
    xhr.send(null);
}
// Fonction de traitement des données XML retournées
// Récupère le noeud XML /data/text
function ajax_parseXml(pXml)
{
    var dataNode = pXml.getElementsByTagName('data').item(0);

    var textNode = dataNode.getElementsByTagName('text');
    for(i=0;i<textNode.length;i++)
        ajax_setText(textNode.item(i).firstChild.data);
}
// Fonction de mise à jour du contenu de la div #demo
// Ajout d'un element <p> contenant le message, dans le div #demo
function ajax_setText(pText)
{
    var elt = document.getElementById('demo');
    var p = elt.appendChild(document.createElement('p'));
    p.appendChild(document.createTextNode(pText));
}
```

4. Tester.

Ex 5. Afficher le contenu d'un fichier JSON en utilisant l'objet XMLHttpRequest.

1. A partir d'un éditeur de texte, créer la page web AjaxJSON.html qui permettra l'affichage des données du fichier JSON.

```
<!DOCTYPE html>
<html>
<head>
<script src = "Codes.js"></script>
</head>
<body>
<h2>XMLHttpRequest - Lecture fichier JSON</h2>
<p>AJAX JSON</p>
<button type="button" onclick="requeteJSON()">Lecture JSON</button>
<div id="demo"></div>
</body>
</html>
```

2. Créer le fichier liens.json qui définit les données à afficher.

```
[ {
  "display": "google",
  "url": "https://www.google.fr"
},{
  "display": "qwant",
  "url": "https://www.qwant.com"
},{
  "display": "univ-tln",
  "url": "http://www.univ-tln.fr/"
} ]
```

3. Créer le fichier Codes.js qui définit la fonction requeteJSON().

```
function requeteJSON(){
var xmlhttp = new XMLHttpRequest();
var url = "liens.json";
xmlhttp.onreadystatechange = function() {
  if (this.readyState == 4 && this.status == 200) {
    var myArr = JSON.parse(this.responseText);
    myFunction(myArr);
  }
};
xmlhttp.open("GET", url, true);
xmlhttp.send();
}

function myFunction(arr) {
  var i;
  for(i = 0; i < arr.length; i++) {
    ajax_Lien(arr[i].display, arr[i].url);
  }
}

function ajax_Lien(pText, plien)
{
var elt = document.getElementById('demo');
var NewA = document.createElement('a');
NewA.setAttribute('href', plien);
var AText = document.createTextNode(pText);
NewA.appendChild(AText);
elt.appendChild(NewA);
elt.appendChild(document.createElement('br'));
}
```

4. Tester.

Ex 5. Afficher le contenu d'un fichier JSON en utilisant l'objet XMLHttpRequest.

1. A partir d'un éditeur de texte, créer la page web AjaxJSON.html qui permettra l'affichage des données du fichier JSON.

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" media="screen" type="text/css" href="styles.css">
<script src = "Codes.js"></script>
</head>
<body>
<h2>XMLHttpRequest - Lecture BDD JSON</h2>
<p>AJAX PDO JSON</p>
<button type="button" onclick="requeteJSON()">Lecture BDD</button>
<div id="demo"></div>
</body>
```

2. Créer le fichier styles.css

```
.Clnom { font-size:18px; color: blue; }
.Clprenom { font-size:14px; color: black; }
.Clage { font-size:16px; color: green; }
```

3. Créer le fichier Donnees_JSON.php qui interroge la base de données.

```
<?php
try
{
    // Connexion à la base de données
    $bdd = connexion();
    // Requete SQL
    $result = $bdd->query('SELECT Nom, Prenom, Age FROM `clients` ');
    // Fichier JSON
    $data = $result->fetchAll(PDO::FETCH_ASSOC);
    echo json_encode($data);
}
catch(Exception $e)
{
    die('Erreur : '.$e->getMessage());
    echo "Erreur connexion !";
}

//-----

function connexion()
{
    $pdo_options[PDO::ATTR_ERRMODE] = PDO::ERRMODE_EXCEPTION;
    // Informations de connection a la BDD
    $host = 'localhost';
    $dbname = 'monsite';
    $user = 'root';
    $password = '';
    // Connection
    $db = new PDO('mysql:host='.$host.';dbname='.$dbname.', $user, $password, $pdo_options);
    return $db;
}

?>
```

4. Créer le fichier Codes.js qui définit la fonction requeteJSON().

```
function requeteJSON(){
var xmlhttp = new XMLHttpRequest();
var url = "Donnees_JSON.php";
xmlhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        var myArr = JSON.parse(this.responseText);
        myFunction(myArr);
    }
};
xmlhttp.open("GET", url, true);
xmlhttp.send();
}

function myFunction(arr) {
    var i;
    for(i = 0; i < arr.length; i++) {
        ajax_aff(arr[i].Nom, arr[i].Prenom, arr[i].Age);
    }
}

function ajax_aff(nom, prenom, age)
{
var elt = document.getElementById('demo');
// Traitement nom
var NewP = document.createElement('p');
NewP.setAttribute('class', "Clnom");
var PText = document.createTextNode("Nom : "+nom);
NewP.appendChild(PText);
elt.appendChild(NewP);
// Traitement prenom
var NewP = document.createElement('p');
NewP.setAttribute('class', "Clprenom");
var PText = document.createTextNode("Prenom : "+prenom);
NewP.appendChild(PText);
elt.appendChild(NewP);
// Traitement age
var NewP = document.createElement('p');
NewP.setAttribute('class', "Clage");
var PText = document.createTextNode("Age : "+age);
NewP.appendChild(PText);
elt.appendChild(NewP);
elt.appendChild(document.createElement('hr'));
}
```

5. Tester.