

AJAX

Magali Contensin



24 mai 2012

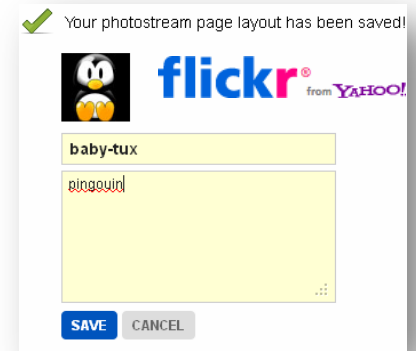
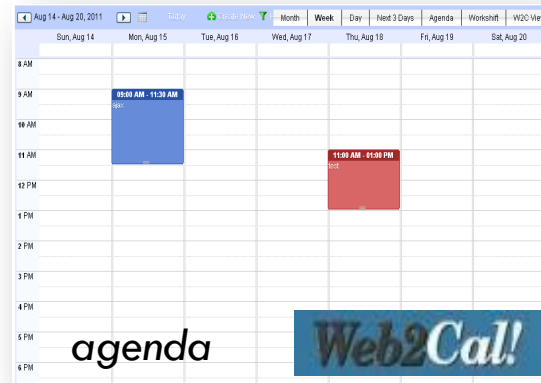
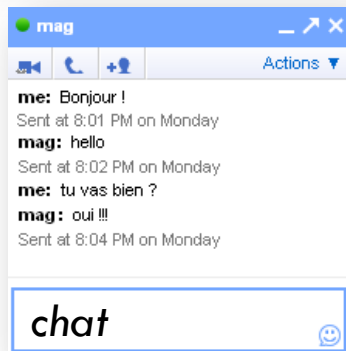
ANF Mathrice – Angers

Plan



- Présentation d'AJAX
- Principe
- Etapes
- Inconvénients
- Avantages

Présentation d'AJAX



Présentation d' **AJAX**

Asynchronous

Javascript

And

XML

Combinaison de technologies :

(X)HTML & CSS – présentation et contenu

DOM – manipulation du contenu

Objet XMLHttpRequest – communication avec le serveur



XML – échange de données (pas obligatoire – texte, JSON)

JavaScript

Choisir une agence :

Nom

Pays




synchrone

Choisir une agence :

Nom

Pays



asynchrone

Jesse James Garrett - 2005

Ajax: A New Approach to Web Applications

Principe

AJAX

Choisir une agence :

Nom

Pays Ville

En

- Allemagne
- Angleterre**
- Belgique
- Espagne
- France
- Italie

- Marseille
- Marseille**
- Paris
- Toulouse
- Nice

Choisir une agence :

Nom

Pays Ville

- Londres
- Londres**
- Manchester

Actualiser une partie d'une page web

Principe

AJAX

Choisir une agence :

Nom

Pays



serveur

1

Principe **AJAX**

Choisir une agence :

Nom

Pays France

En

- Allemagne
- Angleterre**
- Belgique
- Espagne
- France
- Italie

2



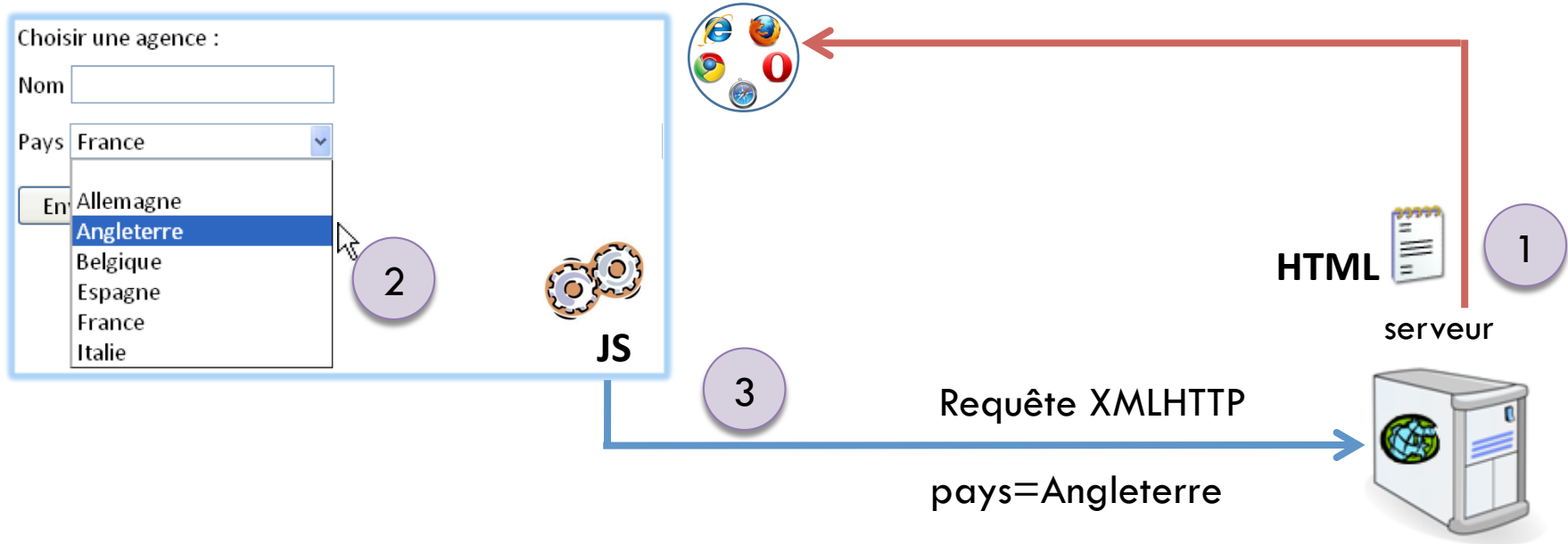
HTML  1

serveur

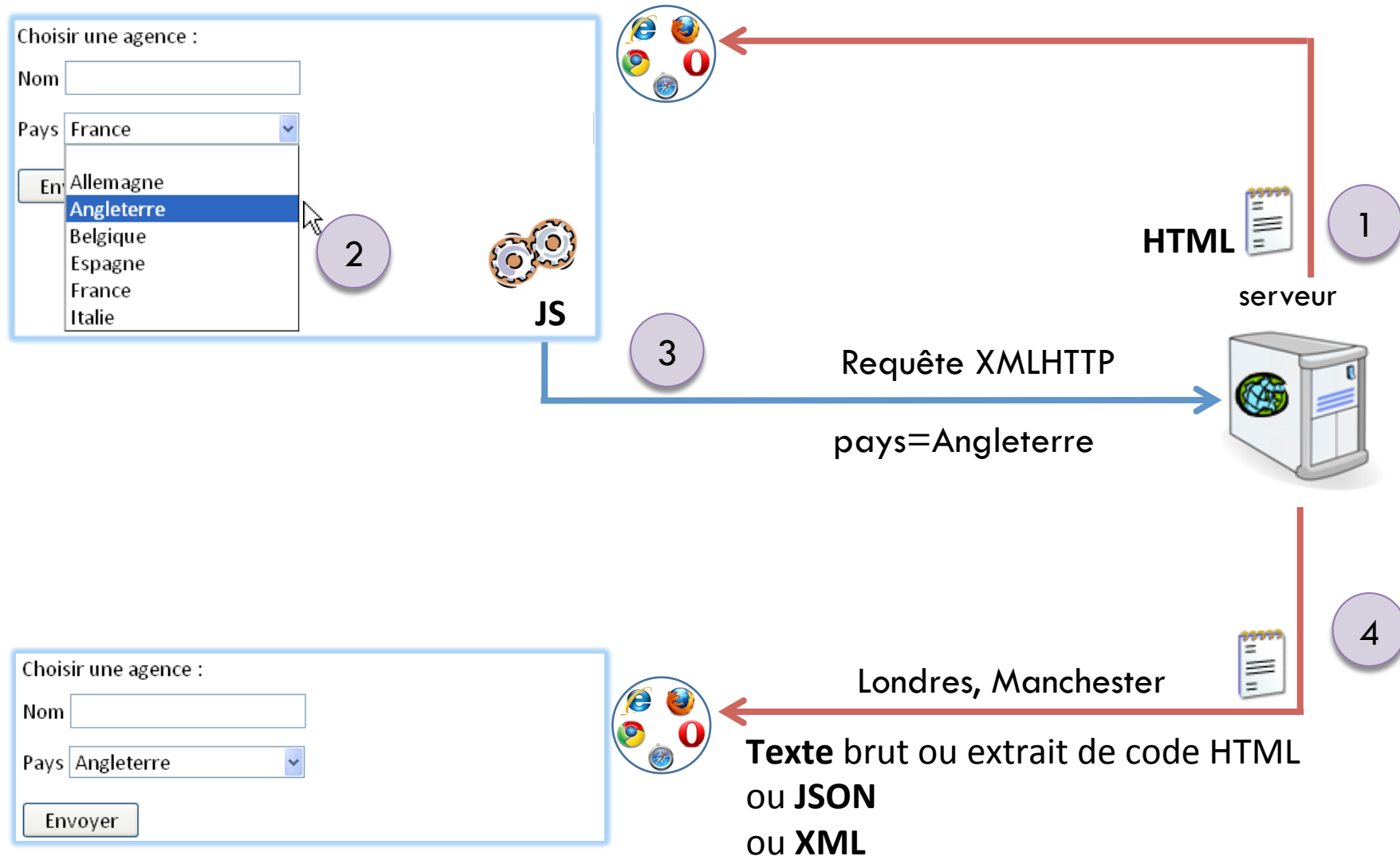


Principe

AJAX

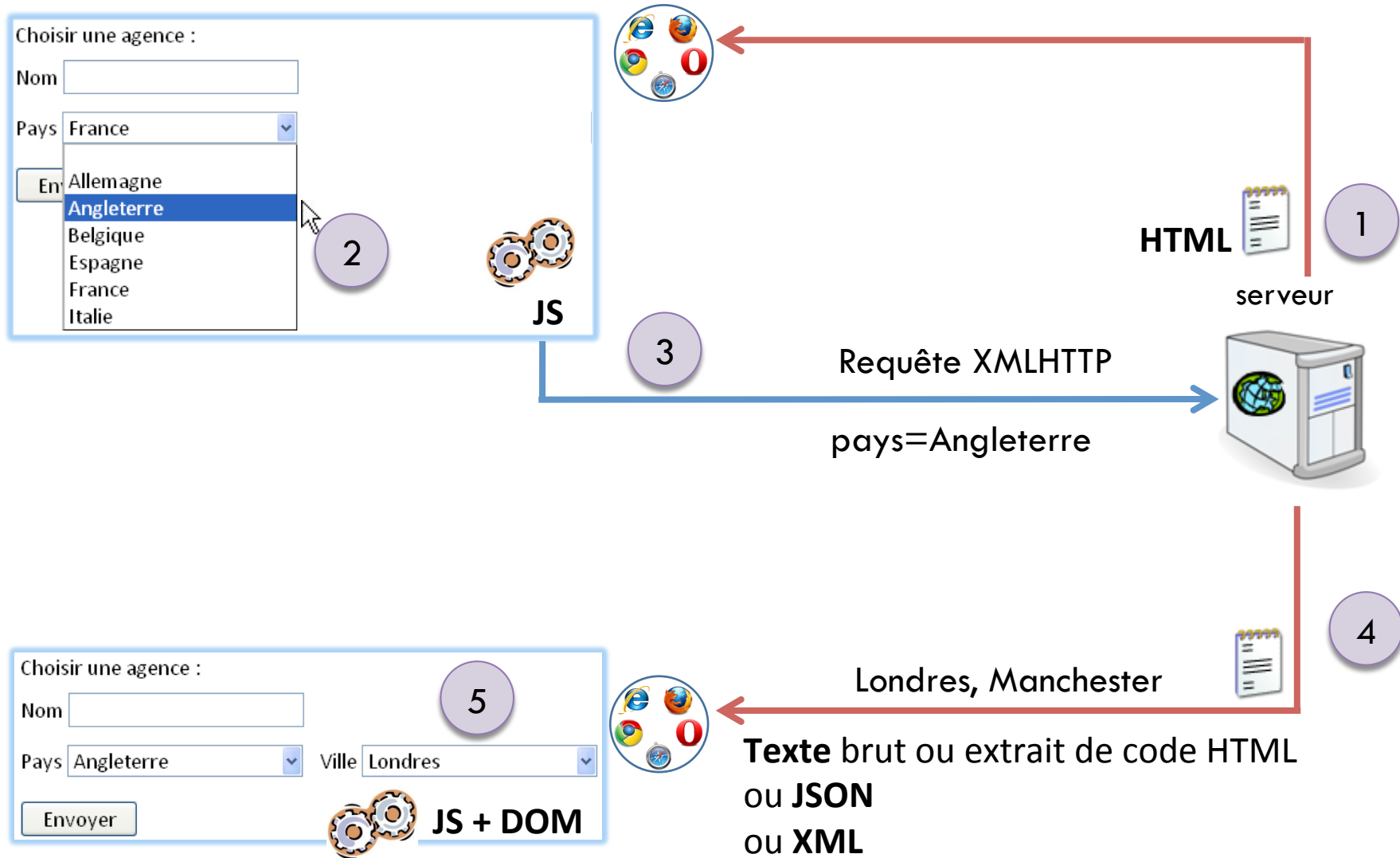


Principe **AJAX**



Principe

AJAX



Etape 1 – page web HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>test</title>
<script type="text/javascript" src="ajax.js"></script>
</head>
<body>
  <form action="traite_form.php" method="post">
    <div> Choisir une agence : <br />
      Nom <input type="text" name="nom" /> <br />
      Pays <select name="pays" onchange="showCities(this.value, 'cities')">
        <option value="" selected="selected"></option>
        <option value="Allemagne">Allemagne</option>
        <option value="Angleterre">Angleterre</option>
        <option value="Belgique">Belgique</option>
        <option value="Espagne">Espagne</option>
        <option value="France">France</option>
        <option value="Italie">Italie</option>
      </select>
      <span id="cities"></span> <br />
      <input type="submit" value="Envoyer" class="button" />
    </div>
  </form>
</body>
</html>
```



Choisir une agence :

Nom

Pays
En Allemagne
Angleterre
Belgique
Espagne
France
Italie

1. Page web HTML

- Inclure un script JavaScript
- Définir l'action
- Identifier la zone à modifier

Etape 2 – script côté serveur (E/S)


Choisir une agence :

Nom

Pays France ▼

En

- Allemagne
- Angleterre**
- Belgique
- Espagne
- France
- Italie



JS



`http://.../liste_villes.php?pays=Angleterre`

Requête XMLHTTP

serveur



Réponse XMLHTTP

Londres;Manchester







liste_villes.php

2. Script côté serveur

- **Entrées** : données de la requête
- **Sortie** : JSON/texte/XML

Etape 3 – script JavaScript


```
function showCities(pays, id_elt) {  
  var node = null;  
  var xml_http = null;  
  try {  
    node = document.getElementById(id_elt);  
    if (pays.length == 0) { // si aucun pays selectionne  
      node.innerHTML = "";  
      return;  
    }  
    xml_http = new XMLHttpRequest();  
  }  
}
```

    **cities**
Angleterre

3. Script JS

a) Créer la fonction appelée dans l'événement de la page HTML


```
} // fin fonction showCities
```

 ``

Etape 3 – script JavaScript

```
function showCities(pays, id_elt) {  
  var node = null;  
  var xml_http = null;  
  try {  
    node = document.getElementById(id_elt);  
    if (pays.length == 0) { // si aucun pays selectionne  
      node.innerHTML = "";  
      return;  
    }  
    xml_http = new XMLHttpRequest();  
  }  
}
```

↑ cities
↑ Angleterre

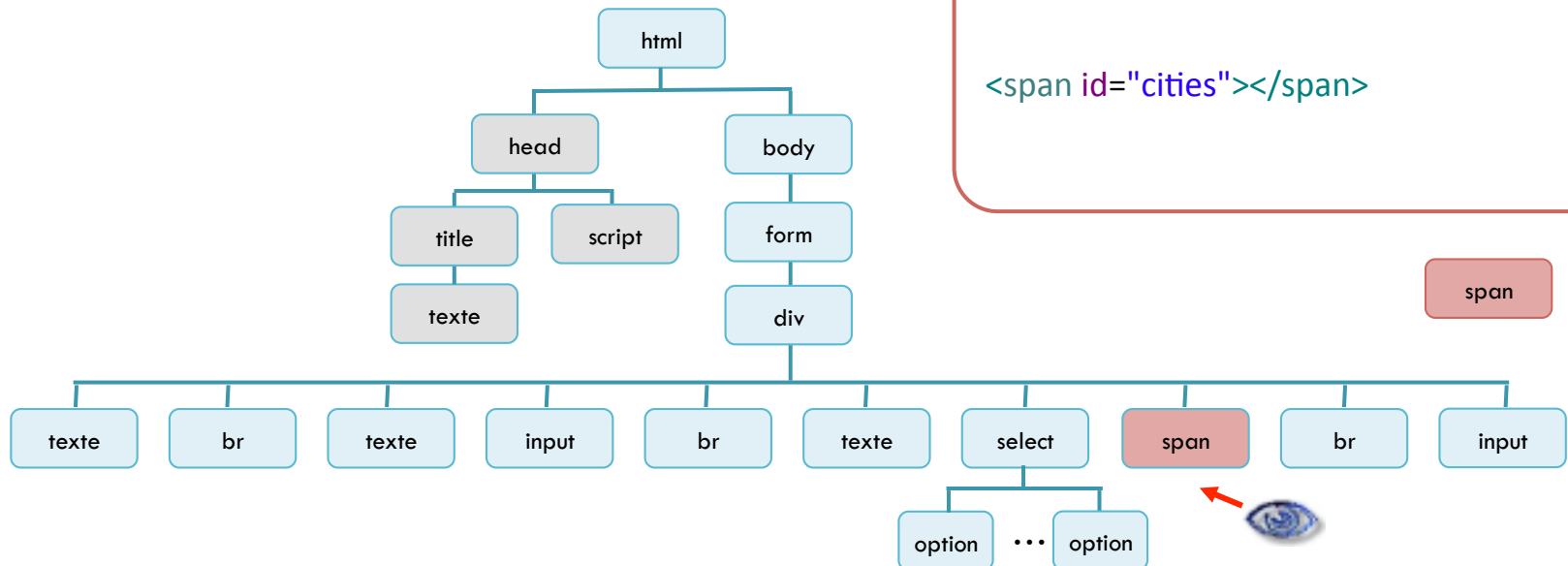


3. Script JS

a) Créer la fonction appelée dans l'événement de la page HTML

```
} // fin fonction showCities
```

```
<span id="cities"></span>
```



Etape 3 – script JavaScript

```
function showCities(pays, id_elt) {  
  var node = null;  
  var xml_http = null;  
  try {  
    node = document.getElementById(id_elt);  
    if (pays.length == 0) { // si aucun pays selectionne  
      node.innerHTML = "";  
      return;  
    }  
    xml_http = new XMLHttpRequest();  
  }  
}
```

Diagramme illustrant les annotations dans le code :

- Deux flèches bleues pointent de la variable `node` vers le mot `cities` et le mot `Angleterre`.
- Une flèche orange pointe de la variable `xml_http` vers un pictogramme d'œil.

3. Script JS

b) Créer l'objet XMLHttpRequest

```
} // fin fonction showCities
```

```
<span id="cities"></span>
```

span

node

Etape 3 – script JavaScript

```
function showCities(pays, id_elt) {  
  var node = null;  
  var xml_http = null;  
  try {  
    node = document.getElementById(id_elt);  
    if (pays.length == 0) { // si aucun pays selectionne  
      node.innerHTML = "";  
      return;  
    }  
    xml_http = new XMLHttpRequest();  
    xml_http.onreadystatechange = function(){
```

 cities

Angleterre



```
} // fin fonction anonyme
```

3. Script JS

c) Définir le gestionnaire d'événements
(appelé quand l'état de l'objet change)

```
} // fin fonction showCities
```

```
<span id="cities"></span>
```

Etats de l'objet

- 0 : UNSENT – objet construit
- 1 : OPENED – objet initialisé (méthode open)
- 2 : HEADERS_RECEIVED – requête envoyée au serveur (méthode send)
- 3 : LOADING – réponse en cours de chargement
- 4 : DONE – terminé

Etape 3 – script JavaScript

```
function showCities(pays, id_elt) {  
  var node = null;  
  var xml_http = null;  
  try {  
    node = document.getElementById(id_elt);  
    if (pays.length == 0) { // si aucun pays selectionne  
      node.innerHTML = "";  
      return;  
    }  
    xml_http = new XMLHttpRequest();  
    xml_http.onreadystatechange = function(){  
      var tab_villes = new Array();  
      var code_html = "";  
      if ((this.readyState == 4) && (this.status == 200)) {  
        // ...  
      }  
    } // fin etat 4 DONE et reponse HTTP 200  
  } // fin fonction anonyme
```

Diagramme illustrant les annotations dans le code :

- Un pointeur bleu pointe de "cities" vers "id_elt" dans la signature de la fonction.
- Un pointeur bleu pointe de "Angleterre" vers "pays" dans la signature de la fonction.
- Un pointeur orange pointe de "4" dans "this.readyState == 4" vers un globe terrestre.

3. Script JS

c) Définir le gestionnaire d'événement
- Attendre que la réponse HTTP soit **complète**

```
} // fin fonction showCities
```

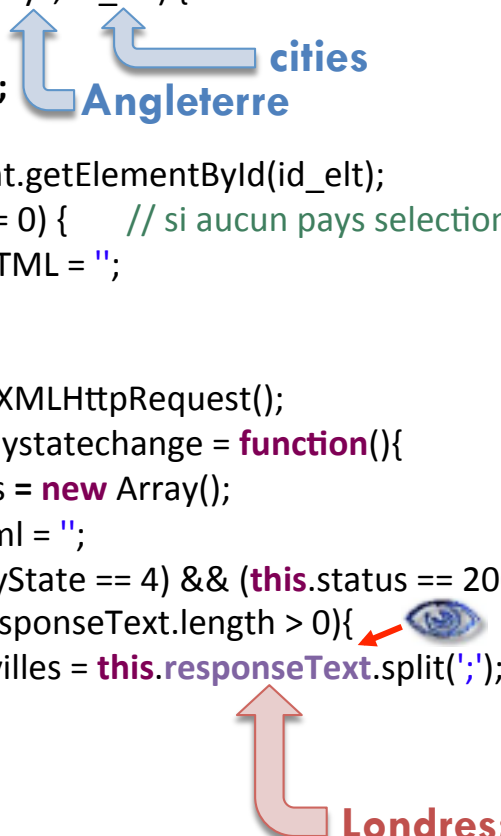
```
<span id="cities"></span>
```

Etats de l'objet

0 : UNSENT – objet construit
1 : OPENED – objet initialisé (méthode open)
2 : HEADERS_RECEIVED – requête envoyée au serveur (méthode send)
3 : LOADING – réponse en cours de chargement
4 : **DONE – terminé**

Etape 3 – script JavaScript

```
function showCities(pays, id_elt) {  
  var node = null;  
  var xml_http = null;  
  try {  
    node = document.getElementById(id_elt);  
    if (pays.length == 0) { // si aucun pays selectionne  
      node.innerHTML = "";  
      return;  
    }  
    xml_http = new XMLHttpRequest();  
    xml_http.onreadystatechange = function(){  
      var tab_villes = new Array();  
      var code_html = "";  
      if ((this.readyState == 4) && (this.status == 200)) {  
        if (this.responseText.length > 0){  
          tab_villes = this.responseText.split(';');  
        }  
      }  
    } // fin etat 4 DONE et reponse HTTP 200  
  } // fin fonction anonyme
```



3. Script JS

c) Définir le gestionnaire d'événement

- Attendre que la réponse HTTP soit complète
- Récupérer la réponse envoyée par le serveur :
format texte/JSON : propriété **responseText**
format XML : propriété **responseXML**

```
} // fin fonction showCities
```

```
<span id="cities"></span>
```

span

node

Etape 3 – script JavaScript

```
function showCities(pays, id_elt) {  
  var node = null;  
  var xml_http = null;  
  try {  
    node = document.getElementById(id_elt);  
    if (pays.length == 0) { // si aucun pays selectionne  
      node.innerHTML = "";  
      return;  
    }  
    xml_http = new XMLHttpRequest();  
    xml_http.onreadystatechange = function(){  
      var tab_villes = new Array();  
      var code_html = "";  
      if ((this.readyState == 4) && (this.status == 200)) {  
        if (this.responseText.length > 0){  
          tab_villes = this.responseText.split(';');  
          // Londres;Manchester  
        }  
      }  
    }  
  } // fin etat 4 DONE et reponse HTTP 200  
} // fin fonction anonyme
```

↑ cities
↑ Angleterre

↑ Londres;Manchester

3. Script JS

- c) Définir le gestionnaire d'événement
- Attendre que la réponse HTTP soit complète
 - Récupérer la réponse envoyée par le serveur
 - Modifier le contenu de la page (manipulation de l'arbre du document)

```
} // fin fonction showCities
```

```
<span id="cities"></span>
```

span

node

tab_villes

Londres

Manchester

0

1

Etape 3 – script JavaScript

```
function showCities(pays, id_elt) {  
  var node = null;  
  var xml_http = null;  
  try {  
    node = document.getElementById(id_elt);  
    if (pays.length == 0) { // si aucun pays selectionne  
      node.innerHTML = "";  
      return;  
    }  
    xml_http = new XMLHttpRequest();  
    xml_http.onreadystatechange = function(){  
      var tab_villes = new Array();  
      var code_html = "";  
      if ((this.readyState == 4) && (this.status == 200)) {  
        if (this.responseText.length > 0){  
          tab_villes = this.responseText.split(';');  
          for (var i = 0; i < tab_villes.length; i++) {  
            code_html += '<option value="' + i + "'>' + tab_villes[i] + '</option>';  
          }  
          node.innerHTML = 'Ville <select name="city">' + code_html + '</select>';  
        } else {  
          node.innerHTML = 'echec';  
        }  
      }  
    } // fin etat 4 DONE et reponse HTTP 200  
  } // fin fonction anonyme
```

 cities
Angleterre



3. Script JS

- c) Définir le gestionnaire d'événement
- Attendre que la réponse HTTP soit complète
 - Récupérer la réponse envoyée par le serveur
 - Modifier le contenu de la page (manipulation de l'arbre du document)

} // fin fonction showCities

```
<span id="cities"></span>
```

code_html

```
<option value="0">Londres</option>  
<option value="1">Manchester</option>
```

span

node

tab_villes

Londres


Manchester


0

1

Etape 3 – script JavaScript

```
function showCities(pays, id_elt) {  
  var node = null;  
  var xml_http = null;  
  try {  
    node = document.getElementById(id_elt);  
    if (pays.length == 0) { // si aucun pays selectionne  
      node.innerHTML = "";  
      return;  
    }  
    xml_http = new XMLHttpRequest();  
    xml_http.onreadystatechange = function(){  
      var tab_villes = new Array();  
      var code_html = "";  
      if ((this.readyState == 4) && (this.status == 200)) {  
        if (this.responseText.length > 0){  
          tab_villes = this.responseText.split(';');  
          for (var i = 0; i < tab_villes.length; i++) {  
            code_html += '<option value="' + i + "'>' + tab_villes[i] + '</option>';  
          }  
          node.innerHTML = 'Ville <select name="city">' + code_html + '</select>';  
        } else {  
          node.innerHTML = 'echec';  
        }  
      }  
    } // fin etat 4 DONE et reponse HTTP 200  
  } // fin fonction anonyme
```





3. Script JS

- c) Définir le gestionnaire d'événement
- Attendre que la réponse HTTP soit complète
 - Récupérer la réponse envoyée par le serveur
 - Modifier le contenu de la page (manipulation de l'arbre du document)

```
} // fin fonction showCities
```

```
<span id="cities"></span>
```

```
Ville <select name="city">  
  <option value="0">Londres</option>  
  <option value="1">Manchester</option>  
</select>
```

span

node

Etape 3 – script JavaScript

```
function showCities(pays, id_elt) {  
  var node = null;  
  var xml_http = null;  
  try {  
    node = document.getElementById(id_elt);  
    if (pays.length == 0) { // si aucun pays selectionne  
      node.innerHTML = "";  
      return;  
    }  
    xml_http = new XMLHttpRequest();  
    xml_http.onreadystatechange = function(){  
      var tab_villes = new Array();  
      var code_html = "";  
      if ((this.readyState == 4) && (this.status == 200)) {  
        if (this.responseText.length > 0){  
          tab_villes = this.responseText.split(';');  
          for (var i = 0; i < tab_villes.length; i++) {  
            code_html += '<option value="' + i + "'>' + tab_villes[i] + '</option>';  
          }  
          node.innerHTML = 'Ville <select name="city">' + code_html + '</select>';  
        } else {  
          node.innerHTML = 'echec';  
        }  
      }  
    } // fin etat 4 DONE et reponse HTTP 200  
  } // fin fonction anonyme
```

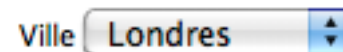
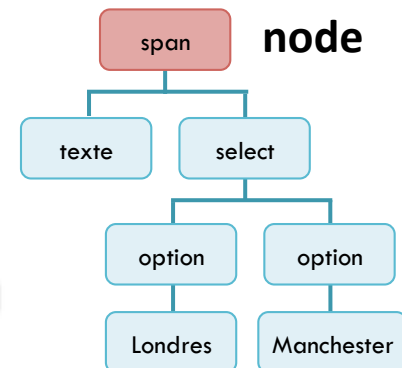


3. Script JS


- c) Définir le gestionnaire d'événement
- Attendre que la réponse HTTP soit complète
 - Récupérer la réponse envoyée par le serveur
 - Modifier le contenu de la page (manipulation de l'arbre du document)


```
} // fin fonction showCities
```

```
<span id="cities">  
  Ville <select name="city">  
    <option value="0">Londres</option>  
    <option value="1">Manchester</option>  
  </select>  
</span>
```



Etape 3 – script JavaScript

```
function showCities(pays, id_elt) {  
    var node = null;  
    var xml_http = null;  Angleterre  
    try {  
        node = document.getElementById(id_elt);  
        if (pays.length == 0) { // si aucun pays selectionne  
            node.innerHTML = "";  
            return;  
        }  
        xml_http = new XMLHttpRequest();  
        xml_http.onreadystatechange = function(){  
            var tab_villes = new Array();  
            var code_html = "";  
            if ((this.readyState == 4) && (this.status == 200)) {  
                if (this.responseText.length > 0){  
                    tab_villes = this.responseText.split(';');  
                    for (var i = 0; i < tab_villes.length; i++) {  
                        code_html += '<option value="' + i + "'>' + tab_villes[i] + '</option>';  
                    }  
                    node.innerHTML = 'Ville <select name="city">' + code_html + '</select>';  
                } else {  
                    node.innerHTML = 'echec';  
                }  
            }  
        } // fin etat 4 DONE et reponse HTTP 200  
    } // fin fonction anonyme
```

```
// envoyer la requete   
xml_http.open("GET",  
              "liste_villes.php?pays=" + pays,  
              true); // asynchrone  
xml_http.send(null); // GET -> pas de donnees  
} catch (e) {  
    return;  
}  
} // fin fonction showCities
```

3. Script JS

d) Envoyer la requête au serveur web

- Récupérer les données à envoyer au serveur
- Initialiser (méthode open)
- Envoyer (méthode send)

Etape 3 – script JavaScript

Prendre en compte la latence

⇒ Informer l'utilisateur qu'une opération est en cours en tâche de fond



Choisir une agence :

Nom

Pays



Envoyer

```
xml_http.onreadystatechange = function() {  
    var tab_villes = new Array();  
    var code_html = "  
    if (this.readyState < 4) {   
        node.innerHTML = "<img src='loader.png' alt='loading...'>";   
    } else {  
        // si etat 4 DONE et que code reponse HTTP 200 manipuler le DOM pour afficher le select  
        ...  
    }  
} // fin fonction anonyme
```


Etape 3 – script JavaScript

Support par les navigateurs

```
function getXmlHttpRequest(){  
  
    var xml_http_obj = null;  
    try {  
        xml_http_obj = new XMLHttpRequest(); // Firefox, IE >= 7, Safari, Opera, Chrome  
    } catch(e){  
        try{  
            xml_http_obj = new ActiveXObject("Msxml2.XMLHTTP"); // IE 6  
        } catch(e){  
            alert ("Browser does not support Ajax");  
        }  
    }  
    return xml_http_obj;  
}
```

Inconvénients

- navigateur
 - ▣ JavaScript activé
 - ▣ support de l'objet XMLHttpRequest
 - => prévoir une version qui fonctionne sans Ajax
- navigation
 - ▣ historique de navigation (boutons back/forward)
 - => fonction de *callback* qui restaure la page
 - ▣ signets
- sécurité
 - ▣ JavaScript activé
- pas d'indexation du contenu dynamique
- latence

Avantages

- Amélioration de l'expérience utilisateur
 - ▣ mise à jour locale de la page
 - ⇒ supprime l'effet de rafraîchissement de la page entière
 - ⇒ diminution des temps d'attente (réponse plus rapide aux actions de l'utilisateur)
 - ▣ contrôle en temps réel du contenu des formulaires
 - ▣ asynchrone => permet de continuer à remplir un formulaire
- Réduction de la bande passante
 - seule l'information nécessaire est transmise
- Réduction de la charge du serveur
 - ▣ une partie des traitements est réalisée par le client (mise en page du résultat)
 - ▣ le serveur n'a pas à générer la totalité de la page à chaque demande
- Aucune installation nécessaire (pas de plugin)

AJAX, utilisé à bon escient, fournit une expérience utilisateur similaire à celle des applications installées sur l'ordinateur.

En savoir plus

- ❑ *W3C XMLHttpRequest Level 2 – Working Draft 2012*
<http://www.w3.org/TR/XMLHttpRequest>
- ❑ *Ajax: A New Approach to Web Applications – J.J. Garrett 2005*
<http://adaptivepath.com/ideas/ajax-new-approach-web-applications>
- ❑ *Ajax Tutorial*
<http://www.xul.fr/en-xml-ajax.html>
- ❑ *Ajax loader icon*
<http://www.ajaxload.info>
- ❑ *Liste Frameworks Ajax*
http://en.wikipedia.org/wiki/List_of_Ajax_frameworks
<http://www.xul.fr/ajax-librairies.html>
- ❑ *Firebug (Onglet Net – XHR)*
<http://getfirebug.com/network>
- ❑ *OWASP – Ajax and Other « Rich » Interface Technologies*
https://www.owasp.org/index.php/Ajax_and_Other_%22Rich%22_Interface_Technologies
- ❑ *OWASP Ajax Security Guidelines*
https://www.owasp.org/index.php/OWASP_AJAX_Security_Guidelines

Questions ?