

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

# AJAX

Emeline DOREY

11 février 2014

emeline.dorey@gmail.com

# Sommaire

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

- 1 Présentation
  - Historique
  - Signification
- 2 Avant Ajax
- 3 Avec Ajax
- 4 Principe
- 5 L'objet XMLHttpRequest
  - Création
  - Utilisation
  - Exemple complet
- 6 La notation JSON

# Présentation

## Historique

### AJAX

Emeline DOREY

Présentation

**Historique**

Signification

Avant Ajax

Avec Ajax

Principe

L'objet

XMLHttpRequest

Création

Utilisation

Exemple complet

La notation JSON

- Le terme AJAX a été introduit :
  - le 18 février 2005,
  - par l'informaticien Américain Jesse James Garrett,
  - dans un article sur le site Web "Adaptive Path".
- Ensuite, il a rapidement gagné en popularité.

# Présentation

## Signification

### AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

- Ajax n'est pas une technologie en elle-même.
- Ce n'est pas non plus un langage.
- Ajax est un acronyme :
  - A ⇒ Asynchronous
  - J ⇒ JavaScript
  - A ⇒ And
  - X ⇒ XML
- Littéralement : "XML et JavaScript asynchrones".

# Présentation

## Signification

### AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

- Ajax est le mariage d'un ensemble de technologies libres qui existent sur le Web :
  - XHTML : pour le balisage du document ;
  - CSS : pour le style du document ;
  - DOM et JavaScript : pour afficher et interagir dynamiquement avec les éléments du document ;
  - XMLHttpRequest (XHR) : l'objet permettant d'échanger et de manipuler les données de manière asynchrone avec le serveur Web ;
  - XML ou JSON : pour les échanges de données entre le client et le serveur.

# Avant Ajax

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

- Les applications Web traditionnelles permettent aux utilisateurs d'effectuer des choix :
  - suivre un lien,
  - remplir un formulaire,
  - valider un formulaire...
- A chaque fois, une requête est envoyée au serveur HTTP.
- Ce serveur agit en fonction de l'action et des données reçues, puis renvoie une nouvelle page.

Ces requêtes sont dites "synchrones".

# Avant Ajax

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

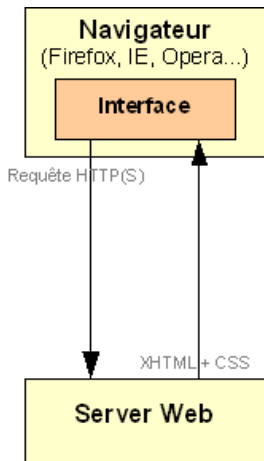
Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON



Classique

# Avant Ajax

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

- Inconvénients :

- Une grande partie du code XHTML est commune aux différentes pages envoyées  $\Rightarrow$  consommation inutile d'une partie de la bande passante.
- A chaque interaction avec l'application, une requête au serveur HTTP doit être réalisée  $\Rightarrow$  le temps de réponse de l'application dépend en grande partie du temps de réponse du serveur HTTP.
- Cela conduit à des interfaces utilisateurs plus lentes.

Mais, aujourd'hui, les navigateurs mettent les éléments communs en cache  $\Rightarrow$  le chargement de nouvelles pages n'oblige pas le serveur à redonner à chaque fois les mêmes informations.



# Avec Ajax

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

- Les applications utilisant Ajax peuvent envoyer des requêtes au serveur HTTP pour récupérer uniquement les données nécessaires, grâce à la requête HTTP XMLHttpRequest.

Ces requêtes sont dites "asynchrones".

# Avec Ajax

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

- Avantages :
  - Ajax apporte beaucoup plus d'interactivité aux applications Web.
  - La quantité de données échangées entre le navigateur et le serveur HTTP est fortement réduite  $\Rightarrow$  les applications sont plus réactives.
  - Une partie du traitement est réalisé sur la machine d'où provient la requête  $\Rightarrow$  le temps de traitement de la requête côté serveur est réduite.

# Avec Ajax

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

### Attention

Le serveur est déchargé d'une partie du travail, c'est au client de manipuler l'information retournée, grâce à du code JavaScript  $\Rightarrow$  il est nécessaire de disposer d'un navigateur Web supportant le JavaScript. Sinon, il sera impossible de naviguer correctement.

# Avec Ajax

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

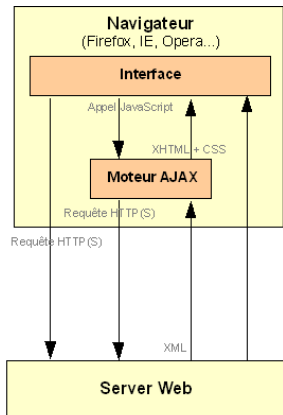
Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON



AJAX

# Principe

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

- Le principe de base d'Ajax est :
  - d'intercepter les événements survenant sur la page ⇒ utilisation de JavaScript,
  - insérer dynamiquement dans la page un contenu provenant d'un serveur Web, véhiculé par un document XML (toujours en utilisant JavaScript).
- L'objet XMLHttpRequest est la pierre angulaire de cette méthode : il permet à JavaScript d'effectuer une requête vers le serveur, sans que l'utilisateur ne le voie, de façon asynchrone.
- Pendant ce temps, l'utilisateur garde la main.

# L'objet XMLHttpRequest

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

- L'objet XMLHttpRequest permet de récupérer des données sur le serveur et de les afficher, grâce à JavaScript, sans rechargement de la page.
- Cet objet n'est pas standardisé  $\Rightarrow$  son implémentation varie selon les navigateurs.
  - Microsoft dispose des fonctions :
    - **ActiveXObject("Msxml2.XMLHTTP")**
    - **ActiveXObject("Microsoft.XMLHTTP")**
  - Les autres navigateurs utilisent la fonction **XMLHttpRequest()**.

# L'objet XMLHttpRequest

## Création

### AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

```
function getXhr(){
  var xhr=null;
  if (window.XMLHttpRequest){ // Firefox et autres
    xhr = new XMLHttpRequest();
  }
  else if (window.ActiveXObject){ // IE
    try {
      xhr = new ActiveXObject(" Msxml2.XMLHTTP");
    }
    catch (e) {
      xhr = new ActiveXObject(" Microsoft.XMLHTTP");
    }
  }
  else {
    alert ("Votre navigateur ne supporte pas les objets _
XMLHttpRequest...");
    xhr = false;
  }
  return xhr;
}
```

# L'objet XMLHttpRequest

## Utilisation - Les propriétés

### AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

- Une fois créé, cet objet peut être manipulé grâce à ses propriétés et méthodes.
- Les propriétés de l'objet XMLHttpRequest sont :
  - **onreadystatechange** : Gestionnaire d'événements pour les changements d'état. Il faut assigner une fonction à cette propriété pour effectuer des traitements sur les données renvoyées après la requête.
  - **readyState** : statut de l'objet.
  - **responseText** : Réponse sous forme de chaîne de caractères.
  - **responseXML** : Réponse sous forme d'objet DOM.
  - **status** : code numérique de réponse du serveur HTTP (200 ⇒ OK, 404 ⇒ Page non trouvée...).
  - **statusText** : message accompagnant le code de réponse.



# L'objet XMLHttpRequest

## Utilisation - Les propriétés

### AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

- **readyState** est un entier indiquant l'état de l'objet, qui est utilisé pour connaître l'état de la requête.
- Les codes possibles sont :
  - 0 = non initialisé ;
  - 1 = ouverture. La méthode **open()** a été appelée avec succès ;
  - 2 = envoyé. La méthode **send()** a été appelée avec succès ;
  - 3 = en train de recevoir. Des données sont en train d'être transférées, mais le transfert n'est pas terminé ;
  - 4 = terminé. Les données sont chargées.

# L'objet XMLHttpRequest

## Utilisation - Les méthodes

### AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

- Les propriétés de l'objet XMLHttpRequest sont :
  - **open("méthode", "url", flag)** : Ouvre la connexion avec le serveur (en option, on peut ajouter un login et un mot de passe).
    - **méthode** ⇒ "GET" ou "POST".
    - **url** ⇒ l'url à laquelle on va envoyer notre requête.
    - Si la méthode est **GET**, on met les paramètres dans l'url.
    - **flag** ⇒ **true** si l'on veut un dialogue asynchrone, sinon, **false**.
  - **setRequestHeader("champ", "valeur")** : Assigne une valeur à un champ d'entête HTTP qui sera envoyé lors de la requête. Par exemple, pour un POST :
    - **champ** ⇒ "Content-Type"
    - **valeur** ⇒ "application/x-www-form-urlencoded"

# L'objet XMLHttpRequest

## Utilisation - Les méthodes

### AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

- **abort()** : Abandonne la requête.
- **getAllResponseHeaders()** : Renvoie l'ensemble de l'entête de la réponse sous forme de chaîne de caractères.
- **getResponseHeader("champEntete")** : Renvoie la valeur d'un champ d'entête HTTP.
- **send(contenu)** : Effectue la requête, éventuellement en envoyant les données.
  - Si la méthode est **GET**, on met **null** en paramètre.
  - Si la méthode est **POST**, on met les paramètres à envoyer, sous la forme :

**"nomparam1=valeurparam1&nomparam2=valeurparam2"**

# L'objet XMLHttpRequest

## En résumé

### AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

## Les étapes d'une requête AJAX

- 1 Création d'un objet XMLHttpRequest
- 2 Définition des actions à réaliser lors des différentes réponses du serveur ⇒ **onreadystatechange()**
- 3 Récupération des éventuels paramètres à envoyer
- 4 Ouverture d'une connexion avec le serveur ⇒ **open()**
- 5 Envoi de la requête au serveur ⇒ **send()**

# L'objet XMLHttpRequest

Exemple1 complet - ajaxExemple1.html

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html;_charset=UTF
    -8">
  <script type="text/javascript" src="ajaxExemple1.js"></script>
</head>
<body>
  <form name="ajax" method="POST" action="">
    <p>
      <input type="button" value="Submit" onclick="submitForm();
        ">
    </p>
    <div id="zoneModif">
      Texte de depart
    </div>
  </form>
</body>
</html>
```

# L'objet XMLHttpRequest

Exemple1 complet - ajaxExemple1.js

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

```
function getXhr() {...}

function submitForm()
{
    var xhr = getXhr();
    xhr.onreadystatechange = function()
    {
        //On ne fait quelque chose que si on a tout reçu et que le serveur
        //est OK
        if(xhr.readyState == 4)
        {
            //La requête s'est bien déroulée => on applique les traitements
            //vus
            if(xhr.status == 200)
            {
                document.getElementById("zoneModif").innerHTML = xhr.
                    responseText;
            }
        }
        ...
    }
}
```

# L'objet XMLHttpRequest

Exemple1 complet - ajaxExemple1.js

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

```
function submitForm()
{
    ...
    //La requete ne s'est pas bien deroulee
    else
    {
        document.getElementById("zoneModif").innerHTML ="Erreur:␣
        code_d'etat␣" + xhr.status + "␣" + xhr.statusText;
    }
}
};
xhr.open("GET", "tab.html", true);
xhr.send(null);
}
```

# L'objet XMLHttpRequest

Exemple1 complet - tab.html

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation

Exemple complet

La notation JSON

```
<h1>Test Ajax</h1>
<table border="1">
  <tr colspan="2">Tableau</tr>
  <tr>
    <td>colonne 1</td>
    <td>colonne 2</td>
  </tr>
</table>
```



# L'objet XMLHttpRequest

## Exemple1 complet - Ecran initial

### AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation

**Exemple complet**

La notation JSON

Submit

Texte de départ

# L'objet XMLHttpRequest

## Exemple1 complet - Ecran final

### AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation

**Exemple complet**

La notation JSON

Submit

## Test Ajax

Tableau

colonne 1	colonne 2
-----------	-----------

# L'objet XMLHttpRequest

## Exemple2 complet - Autocomplétion

### AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation

**Exemple complet**

La notation JSON

### Exemple d'autocomplétion

- 1 Côté serveur  $\Rightarrow$  **autocompletion.php**
- 2 Côté client  $\Rightarrow$  **completionPays.html** et **completionPays.js**

# La notation JSON

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

- JSON : JavaScript Object Notation.
- La notation JSON est un format permettant le passage des données à un programme JavaScript.
- Ce format est un des formats utilisés dans le cadre d'Ajax.
- Il présente l'avantage d'être plus léger que XML et d'être très facilement récupérable par le script JavaScript.

# La notation JSON

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

- Il utilise la notation des objets de JavaScript, sous la forme :

```
var objet = {"propriete1" : "valeur1" , " propriete2" : "valeur2" }
```

- **objet** a ainsi les attributs **propriete1** et **propriete2**, ayant respectivement pour valeurs **valeur1** et **valeur2**.

# La notation JSON

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

- Un tableau d'objets est entouré de "[]"
- Chaque objet est représenté entre "{}"
- Plusieurs objets sont séparés par des ","
- Un attribut et sa valeur sont séparés par " :"
- Les attributs sont séparés par des ","

```
[  
  { "fruit" : "orange",  
    "couleur" : "orange" },  
  { "fruit" : " fraise ",  
    "couleur" : "rouge" }  
]
```

- **fruit** et **couleur** sont des attributs.
- **orange** et **rouge** sont des valeurs d'attributs.

# La notation JSON

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

- Cette notation peut également être utilisée pour les méthodes des objets :

```
var objetAvecMethodesJSON = {"disBonjour" : function() {alert('bonjour')  
;}} , "proprete2" : "valeur2" };
```

- L'objet peut être récupéré d'une requête Ajax au format habituel grâce à la méthode eval :

```
var objet = eval(xhr.responseText);
```

# Comparaison XML / JSON

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

## Avantages XML :

- Largement utilisé et reconnu par tous les langages de programmation.
- Facile à lire.
- Langage de balises permettant de bien structurer les données ⇒ Pratique pour exploiter des données "riches" et/ou "complexes" :
  - Il est facile d'effectuer des recherches dans les données.
  - Les technologies XPath ou XSLT peuvent être utilisés sur un arbre XML pour réaliser des recherches complexes ou des transformations puissantes.



# Comparaison XML / JSON

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

## Inconvénients XML :

- Génère des échanges de données conséquents.
- Ces données doivent être analysées ("parsées") dès réception.

# Comparaison XML / JSON

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

## Avantages JSON :

- Vitesse de traitement.
- Simple à mettre en oeuvre.
  - organisation des données de façon comparable  $\Rightarrow$  bien plus léger qu'XML.
  - JSON est reconnu nativement par JS (il utilise la syntaxe des objets JS) :
    - pas besoin de parser le fichier pour extraire les informations serveurs ;
    - objets directement utilisables.
- Normalisé  $\Rightarrow$  toute application supportant ce format peut accéder à son contenu sans rien ajouter.

# Comparaison XML / JSON

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

## Inconvénients JSON :

- Donnée typée  $\Rightarrow$  ne pourra pas être envoyé en tant que tel.
- Envoi sous forme de texte  $\Rightarrow$  devra donc être évalué par JS avant de pouvoir être utilisée.

# Comparaison XML / JSON - Exemple : Définition

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

### Exemple simple : Définition d'un menu déroulant

Nous souhaitons définir les éléments qui constituent un menu déroulant.

- Soit le menu déroulant "Fichier".
- Ce menu contient trois entrées, pour lesquelles on souhaite connaître le nom et l'action associée.

# Comparaison XML / JSON - Exemple : Fichier au format XML

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

```
<?xml version="1.0" encoding="UTF-8"?>
<racine>
  <menu>Fichier</menu>
  <commandes>
    <commande>
      <titre>Nouveau</titre>
      <action>CreerDocument</action>
    </commande>
    <commande>
      <titre>Sauvegarder</titre>
      <action>SauverDocument</action>
    </commande>
    <commande>
      <titre>Exporter</titre>
      <action>ExporterDocument</action>
    </commande>
  </commandes>
</racine>
```

# Comparaison XML / JSON - Exemple : Fichier au format JSON

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

```
{
  "menu": "Fichier",
  "commandes": [
    {
      "titre": "Nouveau",
      "action": "CréerDocument"
    },
    {
      "titre": "Sauvegarder",
      "action": "SauverDocument"
    },
    {
      "titre": "Exporter",
      "action": "ExporterDocument"
    }
  ]
}
```

# Utilisation du format JSON

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

Un fichier au format JSON permet :

- De charger de l'information à partir du serveur.
- De transmettre de l'information au serveur.
  - Exemple : envoyer le contenu d'un formulaire.

# Utilisation du format JSON

## AJAX

Emeline DOREY

Présentation

Historique  
Signification

Avant Ajax

Avec Ajax

Principe

L'objet  
XMLHttpRequest

Création  
Utilisation  
Exemple complet

La notation JSON

Trois aspects à prendre en compte :

- Côté client  $\Rightarrow$  le traitement par le navigateur.
  - JSON fait partie de la norme JS  $\Rightarrow$  les données au format JSON sont assignées à une variable, qui devient un objet du programme.
- Côté serveur  $\Rightarrow$  le traitement par le serveur.
  - Utilisation du format JSON dans différents langages (PHP, Java...)  $\Rightarrow$  par des parseurs qui permettent d'accéder au contenu.
- La transmission des données entre le client et le serveur.
  - Utilisation de l'objet XMLHttpRequest.