

# Ajax

SAjax, script.aculo.us et autres Framework en PHP

Jean David Olekhnovitch

[jd@ecomis.fr](mailto:jd@ecomis.fr)

Version du 9/3/2007

# Ajax

- Casse le schéma  
“une interaction = un changement de page”
- Passe par des requêtes RPC (Remote Procedure Call)
- S'appuie sur les progrès effectués en Javascript



# Ajax côté client

- Certains éléments (formulaires...) provoquent un appel RPC sur le serveur
  - Chaque page intègre donc un mini-client RPC
- Les résultats sont récupérés et transformés en HTML
- Le HTML est intégré dynamiquement à la page
  - Via du Javascript

# Exemple avec Simple Ajax (SAjax)

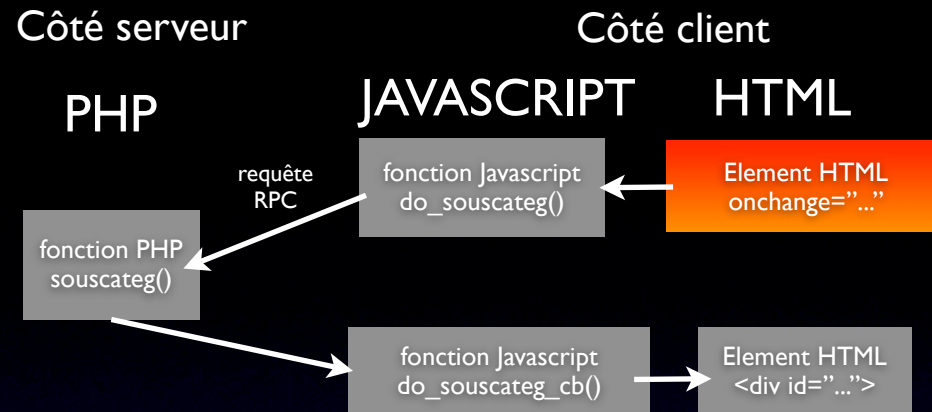
- Ajax n'est en effet qu'un ensemble de recommandation
- Diverses implémentations en différents langages
- Certaines implémentations sont de véritables Framework



# Exemple

- Problème classique : catégorie/sous catégorie
  - Adapter la liste des sous catég en fonction du choix de la catégorie
- Un premier combo (liste) de sélection provoque un appel sur le serveur
- Le serveur retourne un second combo qui sera affiché dynamiquement

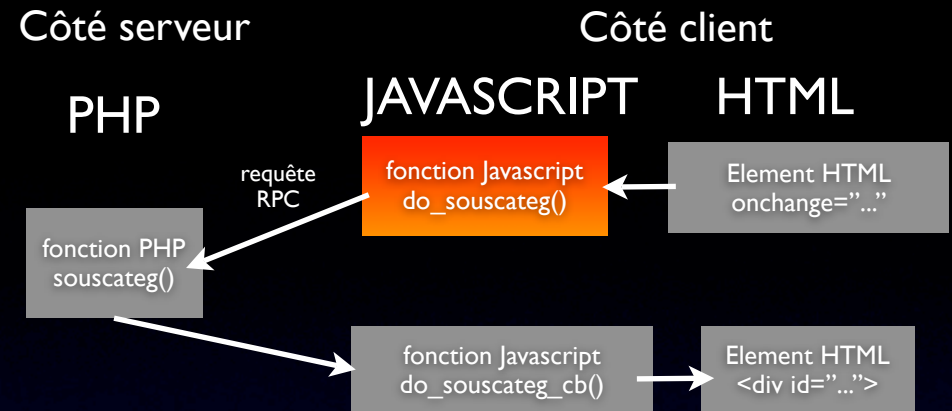
# HTML appelant



```
Catégorie : <select name="categ" id="categ"
onchange="do_souscateg(); return false;">
  <option>Veuillez sélectionner...</option>
  <option>XX</option>
  <option>XXXXX</option>
</select>
<div id="souscateg"></div>
```



# Partie Javascript : appel

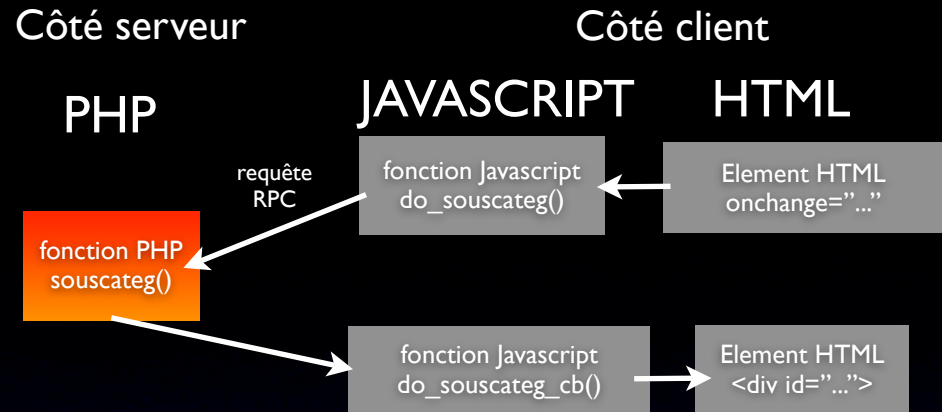


```
function do_souscateg() {  
    var categ = document.getElementById  
    ("categ").value;  
    x_souscateg(categ, do_souscateg_cb);  
}
```

Callback

# Partie PHP

<?

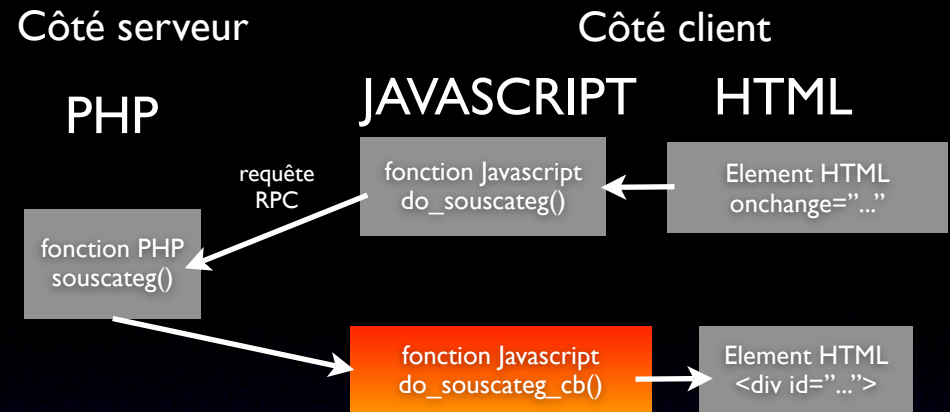


```
require("Sajax.php"); // import du Framework Sajax
```

```
function souscateg($categ) { // génération sous catég
    // requête permettant de récup liste sous catég
    // (appel objets métiers)
    // génération liste
    $ret='<select name="souscateg">';
    $ret=$ret.'<option>XXXX</option>';
    $ret=$ret.'</select>';
    return $ret;
}
```



# Partie Javascript : Callback



```
<script>
```

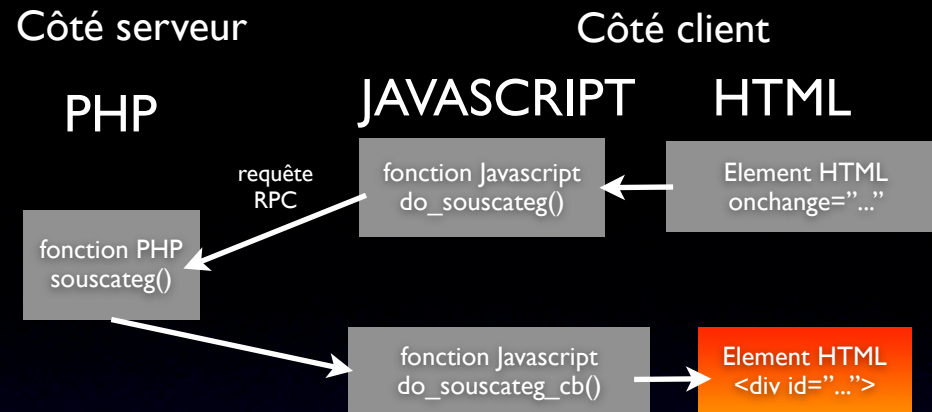
```
<? // insertion du client RPC en Javascript
```

```
sajax_show_javascript();
```

```
?>
```

```
function do_souscateg_cb(valretour) {  
    document.getElementById("souscateg").  
        innerHTML = valretour;  
}
```

# HTML généré



Catégorie : <select name="categ" id="categ"  
onchange="do\_souscateg(); return false;">  
 <option>Veuillez sélectionner...</option>  
 <option>XX</option>  
 <option>XXXXX</option>  
</select>  
<div id="souscateg"></div>



# Structure du fichier

```
<?
require("Sajax.php");// import du Framework Sajax

// écriture des méthodes PHP appelables à distance

sajax_init();
// $sajax_debug_mode = 1;
sajax_export("nomMethodePHP");
sajax_handle_client_request();
?>
```

```
<script>
<? // insertion du client RPC en Javascript
sajax_show_javascript();
?>
function appelMethode_cb(valretour) // callback
{ ... }
function appelMethode()
{ ... }
</script>
```

```
<html>
<input ..... onchange="appelMethode(); return false;">
...
<div id="emplacement"></div>
</html>
```

# Schéma récapitulatif

Côté serveur

PHP

fonction PHP  
souscateg()

Côté client

JAVASCRIPT

fonction  
Javascript  
do\_souscateg()

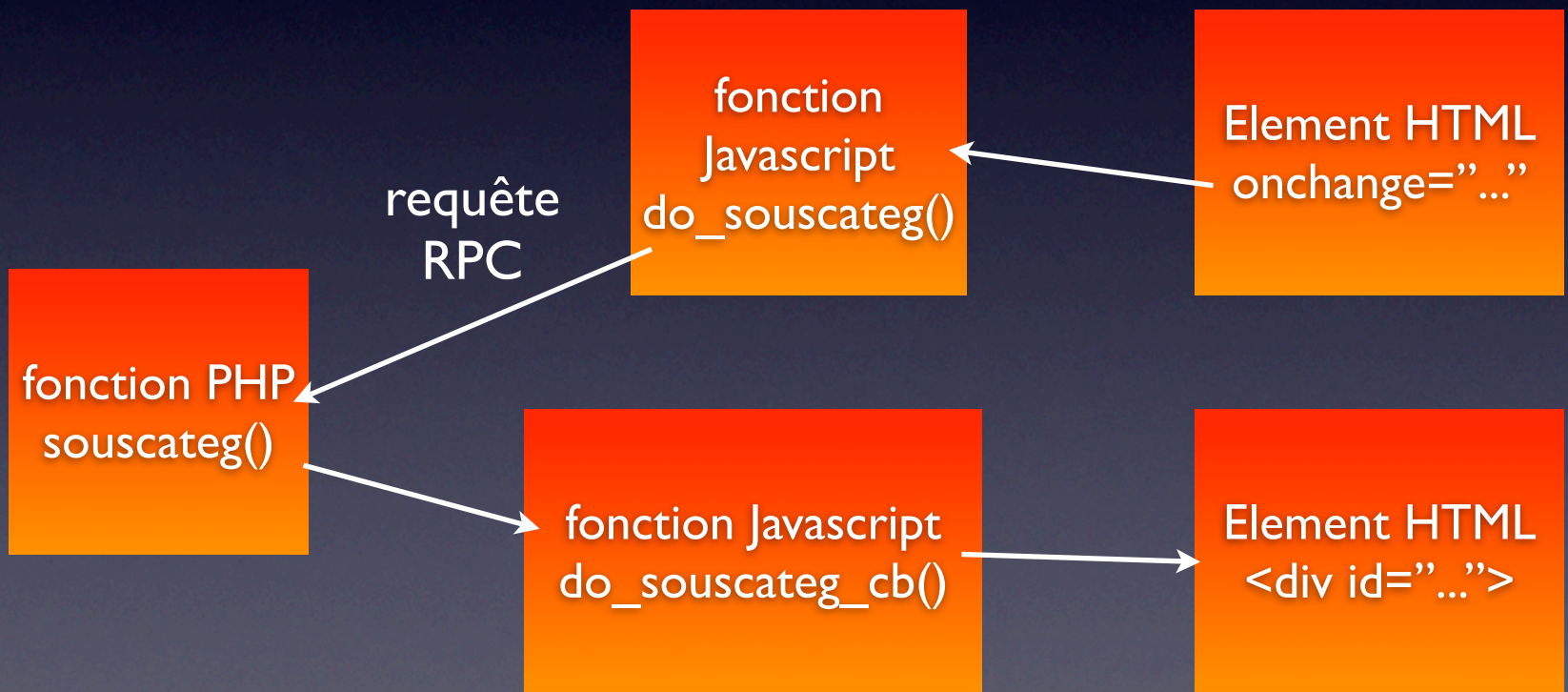
HTML

Element HTML  
onchange="..."

requête  
RPC

fonction Javascript  
do\_souscateg\_cb()

Element HTML  
<div id="...">





# Frameworks Ajax

- Ajax s'avère rapidement assez lourd à l'usage
- Il existe de nombreux framework pour exploiter ses capacités
  - Appels RPC simplifiés
  - Amélioration de l'IHM

# Ajax et l'interface utilisateur

- “Dérive” du concept initial
  - Puisque le Javascript est utilisé abondamment..
  - ... et que l'interactivité est améliorée...
  - On en profite pour créer divers Widgets enrichissant l'IHM



# script.aculo.us

- Librairie Javascript
- <http://script.aculo.us>
- Des objets Javascript réutilisables
  - Extension du Framework Prototype
- Partie serveur non gérée
  - Mais simple à mettre en place (appel de pages isolées)
  - Un projet existe pour Ruby on Rails

# Initialisation

- Deux fichiers .js a stocker avec votre site
- Entête de page :

```
<html>
  <head>
    <title>Page de tests</title>
    <script type="text/javascript" src="js/prototype.js"></script>
    <script type="text/javascript" src="js/scriptaculous.js"></script>
  </head>
  ...
```



# Exemple simple : effets visuels sur du texte

```
<p id="desproges" style="display:none">
```

Qui, parmi vous qui êtes ici ce soir et qui ne semblez globalement pas plus hébétés que le commun des électeurs, qui accepterait de se lever publiquement, comme j'ose le faire, pour exiger l'extermination des mercières, le rétablissement de la peine de mort pour les chanteurs illettrés, l'émasculatation des architectes paysagistes et l'interdiction de stationner devant chez moi.</p>

...

```
<h1 id="title">
```

```
<a href="#" onclick="new Effect.Appear('desproges', 1)">Cliquez  
ici, merci !</a>
```

```
</h1>
```

Appel à un objet  
script.aculo.us

# D'autres effets de texte

```
<a href="#" onclick="new Effect.BlindDown('montexte', 1)">BlindDown</a>  
<a href="#" onclick="new Effect.BlindUp('montexte', 1)">BlindUp</a> |  
<a href="#" onclick="new Effect.Highlight('montexte', 1)">Highlight</a> |  
<a href="#" onclick="new Effect.Grow('montexte', 1)">Grow</a> |  
<a href="#" onclick="new Effect.Shrink('montexte', 1)">Shrink</a> |  
<a href="#" onclick="new Effect.Squish('montexte', 1)">Squish</a> |  
<a href="#" onclick="new Effect.SwitchOff('montexte', 1)">SwitchOff</a> |  
<a href="#" onclick="new Effect.Shake('montexte', 1)">Shake</a> |  
<a href="#" onclick="new Effect.Pulsate('montexte', 1)">Pulsate</a> |  
<a href="#" onclick="new Effect.Fold('montexte', 1)">Fold</a> |  
<a href="#" onclick="new Effect.DropOut('montexte', 1)">DropOut</a>
```

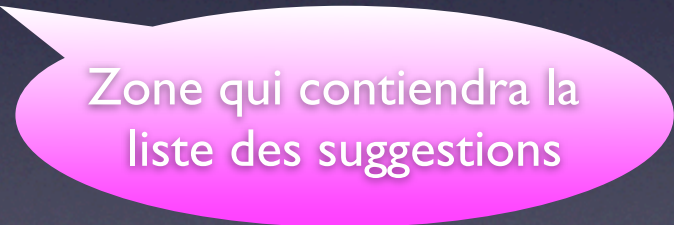


# Ex d'interaction serveur : l'autocomplétion

- Permet d'anticiper la frappe d'un texte
  - Suggestions en provenance du serveur
  - Nécessite une interaction forte avec le serveur, via un appel RPC
- Avec `script.aculo.us`, tout est géré directement :
  - L'appel du serveur
  - L'affichage des résultats dynamiquement

# I - Le champ de saisie

```
<form action="" method="post">  
  <label for="prenom">Quel est votre pr&eacute;nom ?</label>  
  <input type="text" size="30" name="prenom" id="prenom" />  
  <div id="suggestions"></div>  
</form>
```



Zone qui contiendra la  
liste des suggestions



# II - L'objet script.aculo.us

```
<script type="text/javascript">  
new Ajax.Autocompleter (  
  'prenom',    // champ de saisie  
  'suggestions', // zone d'affichage des résultats  
  'prenoms.php', // PHP contenant le code serveur  
  {method: 'post', paramName: 'prenom'}  
);  
</script>
```



Paramétrage de l'appel  
serveur

# III - Le RPC serveur ("Server Side")

prenoms.php :

```
<?php
$tab = array("John", "Johnny", "Jennifer", "Bob", "Patrick", "Patricia", "Pascal");
$q = $_POST['prenom']; $i = 0;
if ($q != "") {
    echo '<ul>';
    foreach($tab as $prenom) {
        if (substr(strtolower($prenom),0,strlen($q)) == strtolower(stripslashes($q))) {
            echo '<li><a href="#" onclick="return false">'.$prenom.'</a></li>';
            if (++$i >= 10) die('<li>...</li></ul>'); // on limite l'affichage à 10 entrées
        }
    }
    echo '</ul>';
}
?>
```

Paramètre  
envoyé par le client

Tous les  
affichages seront reroutés  
sur le client



# Drag'n'Drop avec scriptaculous

- On va utiliser 3 éléments principaux du framework :
  - Les éléments Draggable
  - Les éléments Droppable
  - La fonction Ajax.Updater

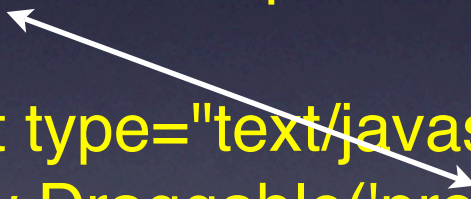
# Élément que l'on peut “tirer” : draggable

- Il s'agit de textes ou d'images

`<div>`

``

`<script type="text/javascript">  
 new Draggable('product_1', {revert:true})  
</script>`



`</div>`



# Emplacement où poser l'élément : Droppable

```
<div id="panier">  
  Glissez vos produits ici  
</div>  
<script type="text/javascript">  
  Droppables.add('panier',  
    {accept:'products',  
      onDrop:function(element)  
      {  
        // ajax.updater  
      }  
    })  
</script>
```

l'emplacement qui  
recevra les éléments

Fonction  
appelée lors du  
“drop”

# Appel Ajax : Updater

Emplacement  
où sera appelé le  
résultat

Fonction  
appelée lors du  
“drop”

```
new Ajax.Updater('panier', 'ajout.php',  
  {  
    parameters:'id=' + encodeURIComponent(element.id)  
  })
```

Paramètres  
envoyés à  
ajout.php



# Le serverside

- Et on termine par le script PHP qui va insérer le produit dans le panier, et générer le visu du panier

```
<?php
$id=$_POST['id'];
// insertion du produit 'id' dans le panier
...
// génération du panier
echo "<h3>Votre panier : </h3>";
echo "...";
?>
```