

## PimenTech-scripts : Ajax2.js

**Author :** Guillaume Savary  
**Contact :** [guillaume@\\_nospam\\_pimentech.net](mailto:guillaume@_nospam_pimentech.net)  
**Revision :** 1.19  
**Date :** 2007-04-26  
**Copyright :** 2006 PimenTech SARL  
**Tags :** ajax javascript api pimentech

Cette bibliothèque fournit notamment l'objet loadHTML, basé sur la libYahoo Ajax.  
Téléchargement : <http://ftp.pimentech.net/src/pimentech-scripts/>

### loadHTML

loadHTML est un objet qui lie un conteneur du DOM à une URL. Il peut charger l'URL en GET ou en POST avec les données d'un formulaire et l'upload d'images.

### Initialisation

Il faut d'abord charger les fichiers javascripts suivant :

- yahoo/js/yahoo-min.js
- yahoo/js/connection-min.js
- yahoo/js/event-min.js
- pimentech/js/PIMENTECH.js
- pimentech/js/ajax2.js

Une fois la page HTML chargée, vous pouvez créer les objets qui chargeront vos blocs ajax. Le meilleur moyen est de les initialiser dans une fonction onload de <body>.

```
<body onload="onLoadFunction()">
```

```
...
```

```
function onLoadFunction () {  
    my_object = new PIMENTECH.util.LoadHTML('my_object', 'container', 'myurl'  
}
```

- 'my\_object' : reprend le nom de mon objet pour la mécanique interne.
- 'container' : id d'un conteneur dans le DOM, celui-ci contiendra les données XHTML de la réponse
- 'myurl' : url appelée par défaut, absolue ou relative, elle peut être vide ou contenir des variables get : <http://myurl.com?var=hello&var2=world>. Cette url doit renvoyer XHTML STRICT !

### Appel GET

Nous voulons qu'un lien ou un bouton charge un bloc HTML dans un Id du Dom :

```
<a onclick="my_object.load()">lien</a>
```

Ceci va appeler la méthode load() de notre objet 'my\_object' qui se contente de charger le résultat de notre url par défaut dans notre conteneur. On peut aussi charger une autre url :

```
<a onclick="my_object.load('/other_url?var=hello')">lien</a>
```

## Appel POST

Si vous voulez poster un formulaire en ajax, il faut utiliser la méthode `post()` de notre objet. De la même façon, le conteneur affichera la réponse de l'url appelée pour le post.

```
<form name="form_1" onsubmit="my_object.post('form_1', 'my_url_for_post')">
```

Les arguments de la méthode sont :

- 'form\_1' : le nom du formulaire que l'on veut poster
- 'my\_url\_for\_post' : une url qui va traiter nos données et renvoyer le XHTML STRICT qui sera affiché à la place du formulaire (message de confirmation par exemple).

## Fonctions avancées

Il est possible de déclarer une méthode `afterLoad()` et `afterPost()` qui vont, comme leur nom l'indique être lancées après le `load()` et après le `post()` :

```
function things_todo_after_load () {  
    ... some stuff ...  
}  
my_object.afterLoad = things_todo_after_load;
```

Pour ne pas voir le chargement de l'url appelée par `load()` ou `post()`, il faut utiliser la fonction `setSilence()` juste après l'initialisation :

```
my_object = new PIMENTECH.util.LoadHTML('my_object', 'container', 'myurl');  
my_object.setSilence();
```