

Programmation sur Internet

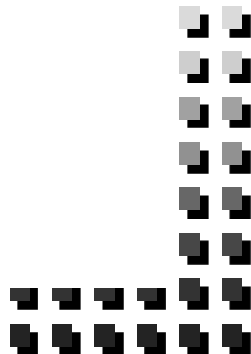
cours 3+ : bibliothèque jquery-ajax

Jean-Michel Ilié

IUT Paris 5

—

département informatique



Processus ajax

Rôle de Ajax

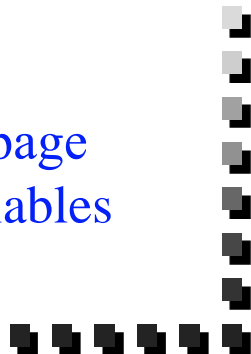
- ☞ faire une requête HTTP dynamiquement pour invoquer un script serveur PHP
- ☞ La requête HTTP peut être paramétré (cf transmission POST ou GET)
- ☞ Récupérer ce qui est transmis dynamiquement

Principe de transmission

- ☞ synchrone : les requêtes et l'interface sont gérées dans une même socket.
- ☞ asynchrone : une socket séparée s'occupe de traiter la requête
 - Avantage : l'interface utilisateur n'est pas bloqué

Que transmettre

- ☞ **contenu** de pages web : html, ... à placer dynamiquement dans la page
- ☞ **données** et collections de données : affectation dynamiques de variables javascript.



Intérêts des bibliothèques javascript pour ajax

Encapsulation

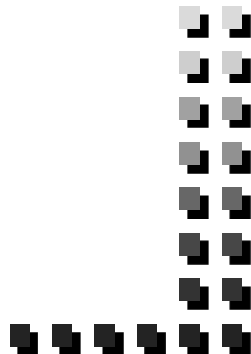
☞ gère en interne les disparités entre types de navigateur

Paramètres

☞ indicateurs de contrôle des procédure (timeOut, ...)

☞ référence sur les fonctions à exécuter (retour, error, ...)

Propose des fonctions raccourcis



La méthode \$.ajax

La méthode
\$.ajax
est spécifiée
génériquement
pour tous
les objets de la collection
référéncée par \$

Exple

```
$.ajax({  
  type: "POST",  
  url: "test.php",  
  success: function(retour){  
    alert("Données retournées : " + retour );  
  }  
});
```

Paramètre :
un objet au format JSON
(equiv. tableau associatif)

Les sorties du script PHP invoquées,
test.php, sont accessibles
dans le paramètre *retour*

👉 **type**

GET ou POST

👉 **url**

fichier désiré sur le serveur WEB (html, php, ...)

👉 **success**

(référence) fonction invoquée quand la requête a
abouti positivement.

Le paramètre est le texte généré par l'URL.

Cas d'une requête sur un lien

Ici, AJAX est lancé
au clic sur le lien

*En laissant href="..." dans le lien,
cela permet un fct même si JS est
bloqué par le navigateur.*

Exple avec un lien :

```
<a href="script.php"
  class="test">
</a>
<div id= "rpse" >
</div>
```

Exple

```
$("#a.test").click(function() {
  $.ajax({
    type: "POST",
    url: $(this).attr("href"),
    success: function(retour){
      $("#rpse").empty().append(retour);
    }
  });
  return false;
});
```

La valeur de l'attribut *href*
Récupérée est celle
du lien courant, cliqué.

élimine toute propagation
aux restes des gérants
événementiels du lien

Requête en méthode GET sur un formulaire

Ici, AJAX est lancé à la
soumission
d'un formulaire .

Exple avec un formulaire:

```
<form action="script.php"  
      class="test">  
</form>  
  
<div id="rpse"> </div>
```

```
$("#form.test").submit(function() {  
    var s = $(this).serialize();  
    $.ajax({  
        type: "GET",  
        data: s,  
        url: $(this).attr("action"),  
        success: function(retour){  
            $("#rpse").empty().append(retour);  
        }  
    });  
    return false;  
});
```

Les paramètres du formulaire
sont sérialisés
et sont placés après l'URL
(param:val¶m:val&..)

Le bloc de réponse
sera vidé
pour recevoir
le nouveau contenu

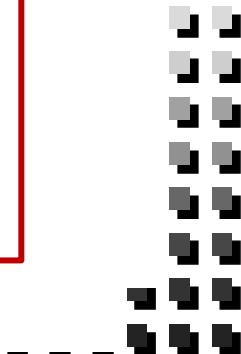
Paramètre de \$.ajax précisant les formats d'échange

👉 **contentType** Type de contenu envoyé
(par défaut : application/x-www-form-urlencoded).

👉 **dataType** Type de contenu reçu (par défaut : html)

- xml** : texte xml
- html** : texte sans mise en forme mais les scripts sont interprétés.
- script**: texte JS sans mise en forme.
- json**: texte JS au format JSON, qui sera directement interprété -> retour d'un objet Javascript.

Le format JSON permet de récupérer des données directement interprétables en JS (tableaux, objet complexe, ...)!
Il existe des extensions php pour créer directement une donnée à partir d'une chaîne reçue au format JSON



Autres paramètres de la méthode \$.ajax

👉 **beforeSend**

Fonction : interprétée avant la requête

👉 **error**

Fonction : interprétée en cas d'error

👉 **complete**

Fonction : interprétée en fin de requête
(erreur ou non)

👉 **processData**

false pour transmettre vers le serveur web
des documents-DOM (html, xml, ...)
(true par défaut)

👉 **async**

false pour communication AJAX synchrone
(true-asynchrone par défaut)

👉 **timeout**

redéfinition de la borne maximale
pour la durée de la requête

👉 **ifModified**

Réponse prise en compte uniquement si apporte
une modification (modification de contenu)

