TP 5 : JQUERY ET AJAX

« A » pour Asynchrone. « JA » pour Javascript et « X » pour … JSON (ou XML historiquement). Généralement, la mise en place de requêtes AJAX nécessite un serveur. Néanmoins pour faciliter la compréhension, les fichiers à interroger dans ce TP seront accessibles localement.

1.1 CHARGER UN FICHIER

load(url, données, complete_function) La méthode load( ) permet de charger d'une façon simple un fichier selon le procédé mis en place par AJAX.

- url : une chaîne de caractères contenant l'URL du fichier à charger
- données (optionnel) : liste de paires de la forme clé/valeur qui seront envoyées en tant que données au serveur.
- fonction 'complete' (optionnel) : la fonction exécutée en cas de réussite de la requête. Par défaut, les données sont chargées dans l'élément sélectionné.

```
/* Appel d'un fichier html */
$("#resultat").load(
    "http://example.com/resource.html #selecteurJQuery",
    function(){   alert("C'est chargé!"); }
);
```

Exercice (1.1.1) : Charger avec le bouton 1, le contenu entier de 'option1.html' dans la 'div', et parallèlement avec le bouton 2 et 'option2.html'

Exercice (1.1.2) : Charger uniquement le contenu de l'élément avec l'identifiant « box » de 'option2.html' avec le bouton.

1.2 CHARGER SELON LA MÉTHODE GET OU POST

GET et POST sont des types de requêtes HTTP. GET est davantage utilisé pour recevoir des données et POST pour en envoyer. D'autres différences concernent :

- Dans le cas d'un retour arrière, GET ne fait rien, POST renvoie à nouveau les données.
- GET affiche les données dans son URL, POST cache les données.

*Par exemple : http://annuaire.com/retrouver.php**?**nom=Dupont**&**prenom=Luc**&**ville=Tarbes*

- GET est limité à 2048 caractères dans sa requête. POST n'a pas de restriction.
- GET gère uniquement des chaînes de caractère, POST gère le binaire.

Il existe d'autres types de requêtes HTTP (DELETE, PUT, CONNECT, etc.)

$.get(url, données, fonction, type) envoie une requête HTTP GET.

- url : une chaîne de caractères contenant l'URL du fichier visé.
- données (optionnel) : liste de paires de la forme clé/valeur qui seront envoyées en tant que données au serveur.
- success : la fonction qui doit être exécutée en cas de réussite de la requête.

- type (optionnel) : chaîne de caractères qui spécifie le type de données transmises à la fonction : “xml”, “html”, “script”, “json”, “jsonp”, ou “text”.

```
/* Exemple de requête GET */
$.get("meteo_demain.php",
     function(data){
         console.log("Demain il fera : " + data );
     },
     "text"
);
```

$.post(url, données, fonction,type) envoie une requête POST.

- url : une chaîne de caractères contenant l’URL du fichier à charger
- données : liste de paires de la forme clé/valeur qui seront envoyées en tant que données au serveur.
- success : la fonction qui doit être exécutée en cas de réussite de la requête.
- type (optionnel) : chaîne de caractères qui spécifie le type de données transmises à la fonction : “xml”, “html”, “script”, “json”, “jsonp”, ou “text”.

```
/* Exemple de requête POST */
$.post("formulaire.php",
     {"sujet" : $("#sujet").val(),
      "commentaire" : $("#commentaire").val()
     },
     function(data){
         console.log(data.attributjson) ;
     },
     "json"
);
```

Exercice (1.2) : Au clic du bouton, afficher dans la d’identifiant « droit » le contenu de l’article ‘lorem.txt’. Le contenu textuel chargé doit être ‘justifié’ (typographiquement). La méthode $.get() doit être utilisée.

1.3 CHARGER UN SCRIPT

$.getScript(url, fonction) charge un script JavaScript du serveur en utilisant la méthode http GET et exécute celui-ci.

- url : une chaîne de caractères qui indique l’adresse du script à charger .
- fonction (optionnel) : la fonction à exécuter si la requête est réussie.

```
$.getScript("test.js", function(){
     alert("Script chargé!");
});
```

Exercice (1.3) : Déclencher au clic du bouton un message d’alerte JavaScript chargé à partir du serveur (cf. ‘alert.js’).

2 LA REQUETE AJAX COMPLETE

Cette méthode permet d'effectuer une requête AJAX en maîtrisant, grâce aux nombreuses options disponibles, les différents paramètres et étapes de celle-ci.

ajax(options) réalise une requête HTPP.

- url (obligatoire) : une chaîne de caractères contenant l'adresse de la requête
- type (optionnel) : chaîne de caractères qui spécifie le type de requête HTTP à utiliser (GET ou POST). La valeur par défaut est GET. D'autres méthodes d'envoi HTTP peuvent être utilisés, comme PUT ou DELETE, mais celles-ci ne sont pas supportées par tous les navigateurs
- dataType (optionnel) : une chaîne de caractères qui spécifie le format des données qui seront renvoyées par le serveur (xml, html, json ou script) :
    - "xml" : retourne un document XML qui pourra être traité par jQuery
    - "html" : retourne du code HTML au format texte
    - "script" : évalue une réponse en JavaScript et retourne cette dernière au format texte
    - "json" : évalue une réponse en JSON et retourne un objet JavaScript.

- Trois fonctions :
    - success : exécuté si l'appel réussi. C'est cette fonction qui va être appelée pour, par exemple, mettre à jour le site web.
    - error : exécuté si une erreur survient.
    - complete : exécuté une fois l'appel terminé.

```
/* Exemple de requête AJAX de type POST pour un chat online */
$.ajax({
     url : 'tchat_online.php',
     type : 'POST',
     data : 'user=' + user_name + '&content='+ content_data,
     dataType : 'json',
     success : function(code_json, statut){
                $("#retour-user").text("Message envoyé");

     },
     error : function(resultat, statut, erreur){
                console.log("Erreur de type : " + erreur);
     },
     complete : function(resultat, statut){
                console.log("Appel AJAX terminé");
     }
}) ;
```

Exercice (2) : Au clic d'un lien, faire apparaître un contenu html depuis la page 'liste.html' en utilisant la méthode ajax( ). Etant donné qu'il s'agit uniquement de récupérer des informations, une requête GET sera adaptée. Attention au lien : il est

nécessaire de prévenir l'action par défaut. Egalement, faites apparaître dans la console les différents arguments donnés aux fonctions 'success', 'error' and 'complete'.

3 SERIALISER LES DONNEES

serialize() Transforme les données des champs de formulaire en une chaîne de caractères de type 'valeur1=' + valeur1 + '&valeur2=' + valeur2 + ' … '

Ce procédé est fort utile pour envoyer des données au serveur par une requête AJAX sous un format compatible avec les langages de programmation côté serveur.

```
var mydata = $("#mon-formulaire").serialize() ;
```

Exercice (3) : Soit un formulaire de départ. Au clic du bouton, les données renseignées sont sérialisées et affichée dans l'encart 'résultat'.

4 APPLICATIONS

Pour les applications, il est attendu que vous utilisiez à la fois le contenu du cours, et que vous vous **DOCUMENTIEZ**. La *jQuery API Documentation* est une documentation de qualité : http://api.jquery.com/

4.1 UNE ICÔNE DE CHARGEMENT

Les requêtes AJAX peuvent parfois entraîner une relative lenteur au chargement du fichier externe en fonction de leur taille, qualité de la connexion, etc. Pour éviter que l'utilisateur ne soit dérangé par ces moments de latence, le concepteur peut prévoir une icône indiquant le chargement en cours. Ce peut être un gif animé qui disparaît lorsque la requête a abouti avec succès.

Des icônes sont téléchargeables ici : http://ajaxload.info/

Exercice (4.1) : Soit le fichier de départ qui comporte simplement un bouton et une division destinée à recevoir le fichier (lorembis.html). Lorsque la requête AJAX est lancée par le bouton, il faut que l'icône ('ajax-loader.gif') apparaisse dans la pendant le chargement. Une fois le contenu chargé, il faut charger le fichier dans la de contenu et faire disparaître l'icône de chargement.

4.2 UN LEXIQUE EN AJAX

Elaborons un petit lexique informatique dont les définitions sont chargées par des requêtes AJAX dans la page principale.

Exercice (4.2) : Au clic sur la lettre A ou B, une requête load( ) est exécutée afin de charger les définitions correspondantes (lettre_a.html ou lettre_b.html) dans la division lexique.

4.3 AFFICHAGE DYNAMIQUE DE PHOTOS FLICKR

$.getJSON(options) Charge des données JSON à partir d'une page appelée avec la méthode GET.

- url (String): URL de la page à charger contenant des données JSON,
- params (Map): (optionnel) paires de clé/valeur qui seront envoyées au serveur
- callback : fonction à exécuter quand les données seront complètement chargées.

```
/* Soit l'objet JSON de structure suivante : */
({
   "title" : "Mon objet JSON sur un service web",
   "modified" : "2015-05-12T21:34:34Z"
   "items" : [
         { "description" : "Premier élément de l'objet",
           "tags" : "mot-clef1, mot-clef2, mot-clef3"
          },
        { "description" : "Deuxième élément de l'objet",
          "tags" : "mot-clef2, mot-clef4, mot-clef7"
         }
    ]
})
/* fonction getJSON pour récupérer les informations */
$.getJSON("http://exemple.com/fichier.json", function(data_json){
          console.log("Contenu de l'attribut title : "+   data_json.title);
          console.log("Contenu de l'attribut title : "+   data_json. modified);

          /* boucle sur les objets items */
         $.each(data.items, function(index, element) {
                console.log("Itération dans la boucle numéro : " + index) ;
                console.log("Description de l'objet : " + element.description) ;
               console.log("Tags de l'objet : " + element.tags) ;

       }) ;
});
```

Exercice (4.3.1) :

Le site flickr http://www.flickr.com/services/feeds/docs/photos_public/ met à disposition des flux de données sous différents formats dont JSON.

Ecrire un script jQuery qui permet l'affichage de photos publiques contenues dans un flux et taggées avec le mot-clé « Tarbes » sur flickr.

URL direct ver les photos publiques taggées Tarbes :

http://api.flickr.com/services/feeds/photos_public.gne?tags=tarbes&tagmode=any&format=json&jsoncallback=?

Un conseil : pour récupérer les données, vous avez besoin d'aller voir la structure des données JSON sur ce site.

Amélioration importante : le visiteur est redirigé sur la page flickr correspondante lorsqu'il clique sur la photo.