

AJAX

Formateur : NSEKE Charles

OBJECTIFS

Définition du principe AJAX

Parcourir un objet XML Javascript.

Accéder aux données via le DOM et XMLHttpRequest

Identifier les forces et faiblesses d'AJAX

PRINCIPE AJAX

DÉFINITION

AJAX = Asynchronous JavaScript + XML

approche s'appuyant les technologies : HTML ou XHTML, les feuilles de styles CSS, JavaScript, le modèle objet de document (DOM)... et surtout l'objet XMLHttpRequest

Il est surtout utilisé pour l'échange de données en JS.

ACCÉDER AUX
DONNÉES VIA
XMLHTTPREQUEST

ACCÉDER À DES DONNÉES EN AJAX

Pour accéder à des données via XMLHttpRequest il faut:

```
const request = new XMLHttpRequest();
```

```
const aSynchrone = false;
```

```
request.open('GET', 'https://www.google.com', aSynchrone);
```

```
request.send(null);
```

```
// pour accéder au contenu de la réponse
```

```
request.responseText ou request.responseXml
```

TP 1 : LIRE DU TEXT

LECTURE D'UN TEXTE EN AJAX

- 1- créer un fichier `exemple.txt` et écrivez-y `hello world!`
- 2- fait un requête ajax pour récupérer le contenu de ce fichier.
- 3- afficher la réponse de cette requête dans une page `html`

TP 2: LIRE DU XML

A FAIRE LUNDI

récupérer le fichier emails.xml

récupérer son contenu en ajax inspirez vous de ce qu'on a fait avec le fichier personnes.json.

afficher les emails dans une page html.

TP 3: LIRE DU JSON

charger les données du fichier personnes.json que vous trouverez sur mon github

afficher les infos contenues dans ce json dans une page html

INTRODUCTION À JQUERY

JQUERY C'EST QUOI ?

jQuery est une bibliothèque JavaScript open source qui simplifie l'écriture de script JS.

Avec jQuery on peut:

- manipuler le DOM
- Gérer les évènements
- Faire des requêtes Ajax
- Ajouter des effets visuels (animations)
- ...



AJOUT DE JQUERY DANS SA PAGE HTML

depuis un repo distant (recommandé)

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```

OU

télécharger le script (<https://jquery.com/download/>) et mettez le dans votre page html

```
<script src="jquery-3.3.1.min.js"></script>
```

JQUERY ET LE DOM - SÉLECTION D'ÉLÉMENT

Pour sélectionner un élément html avec jQuery.

- une balise

`$('nom_de_la_balise')`

exemple: pour récupérer tous les boutons de la page html

on fait `$('button')`

en JS natif on faisait `document.getElementsByTagName('button')`

JQUERY ET LE DOM- SÉLECTION D'ÉLÉMENT

Pour sélectionner un élément html avec jQuery.

- un élément connaissant son id

`$('#idElement')`

exemple: pour récupérer l'élément d'id écran de la page html

on fait `$('#ecran')`

en JS natif on faisait `document.getElementById('ecran')`

JQUERY ET LE DOM- SÉLECTION D'ÉLÉMENT

Pour sélectionner un élément html avec jQuery.

- un ou plusieurs éléments connaissant le nom de la classe
classe

`$('.nomClasse')`

exemple: pour récupérer les éléments de classe nom de la page html

on fait `$('.nom')`

en JS natif on faisait `document.getElementsByClassName('nom')`

JQUERY ET LE DOM- MODIFICATION D'ÉLÉMENTS

- **text()** retourne la valeur du texte contenu dans la balise et **text('new text')** met à jour le texte contenu dans la balise
- **val()** pour les éléments de type input **.val()** retourne la valeur et **val(newVal)** met à jour la valeur.
- **html()** retourne la valeur de l'html contenu dans la balise et **html('new html code')** met à jour le html contenu dans la balise

JQUERY ET LE DOM - GESTION D'ÉVÈNEMENTS

Pour attacher un événement à un élément avec jQuery on fait:

```
$(“button”).bind(‘click’, function() {});
```

il existe une fonction pour chaque type d'évènement pour attacher un click à un bouton on peut par exemple faire

```
$(“button”).click(function() {});
```

JQUERY ET LE DOM - GESTION D'ÉVÈNEMENTS

click(function à exécuter) pour gérer l'évènement click

dblclick(function à exécuter) pour gérer l'évènement double click

hover(function à exécuter) pour gérer l'évènement survol de la souris

keypress(function à exécuter) pour gérer l'évènement appui d'une touche du clavier

JQUERY ET LE DOM GESTION D'ÉVÈNEMENTS

live(événement, fonction à exécuter) live permet d'attacher l'évènement passé en paramètre à tous les éléments concernés et aussi au prochain qui seront ajouté.

exemple on a un bouton dans la page html

on fait `$('#button').live('click', function(){});`

si dans notre code on ajoute un nouveau bouton, ce bouton aura automatiquement l'évènement click attaché.

live est à remplacer par on pour des version jquery >= 1.7

Plus d'évènements: <https://api.jquery.com/category/events/>

JQUERY ET LE DOM

```
$.ajax({  
  dataType: "json",  
  url: url,  
  data: data, // données transmises dans la requête  
  success: success --la fonction qui s'exécute quand on reçoit une réponse sans erreur  
});
```

plus de détails ici: <https://api.jquery.com/jquery.getjson/>

D'AUTRES LIBRAIRIES JS

ReactJS, Angular, Vue.js.

Node.js

EN RÉSUMÉ

QUESTION POUR LUNDI ET MARDI

pourquoi utiliser Ajax?

revenir sur les états que peut avoir une requête ajax

revenir sur les type de réponse xml et txt

JQuery

Jquery et Ajax

inconvénients d'utiliser Ajax

TP FINAL: INTERFACE DE GESTION D'UNE CLASSE

Je vous propose de mettre en place des interfaces de gestion d'une classe.

En vous appuyant sur HTML/CSS et JavaScript (Jquery ou natif), réaliser le TP que vous retrouverez à cette URL.

récupérer le fichier serveur-classe.zip :

<https://github.com/nsekecharles/formation-js>

AVANT

```
java -jar build/libs/gestionnaire-classe-1.0-SNAPSHOT.jar  
--dataBaseDir=/Users/nsekecharles/Desktop/data
```

sous windows :

```
java -jar gestionnaire-classe.jar  
--dataBaseDir=C:\Users\IPI\Downloads\serveur-classe\serveur-  
classe\data --server.port=9000
```

LES WEB SERVICES À UTILISER

<http://localhost:8081/students> en POST + {"nom": "Karine", "dateNaissance": "16/04/1991"} en param pour créer un élève

<http://localhost:8081/students> en GET récupération des élèves

<http://localhost:8081/subjects> ajout d'une matière en POST + {"name": "Maths"} en param

<http://localhost:8081/subjects> en GET récupération de toutes les matières

<http://localhost:8081/notes> ajout d'une note en POST + {"studentId": 1, "subjectId": 2, "note": 19} en param

<http://localhost:8081/notes?eleveId=1> en GET récupération des notes d'un élève

LES PAGES À CRÉER

Création d'une page pour créer un élève

Création d'une page pour afficher tous les élèves

Création d'une page pour ajouter une matière

Création d'une page pour ajouter une note à un élève

Création d'une page pour afficher toutes les notes d'un élève

POUR FAIRE UNE REQUETE AJAX POST

```
$.ajax({url:"http://localhost:8081/students",  
data:JSON.stringify({nom:'charles',  
dateNaissance:'12/12/2018'}),  
contentType:"application/json", dataType:"json",  
method:'POST'});
```

POUR FAIRE UNE REQUETE AJAX GET

```
$.ajax(  
  
  {url:"http://localhost:8081/students",  
  contentType:"application/json",  
  dataType:"json",method:'GET'})  
  
  .done(function(data) {  
  
    // ici vous gérez la réponse  
  
  })
```