

jQuery

Motivation

- La manipulation du DOM en JavaScript est lourde et vite pénible.
- Elle pose aussi souvent des problèmes de compatibilité entre les différents navigateurs.
- jQuery permet de simplifier et normaliser cela.
- <http://code.jquery.com/>
 - (pour le cours, prendre une version normale, la version minimifed est pour une mise en production).

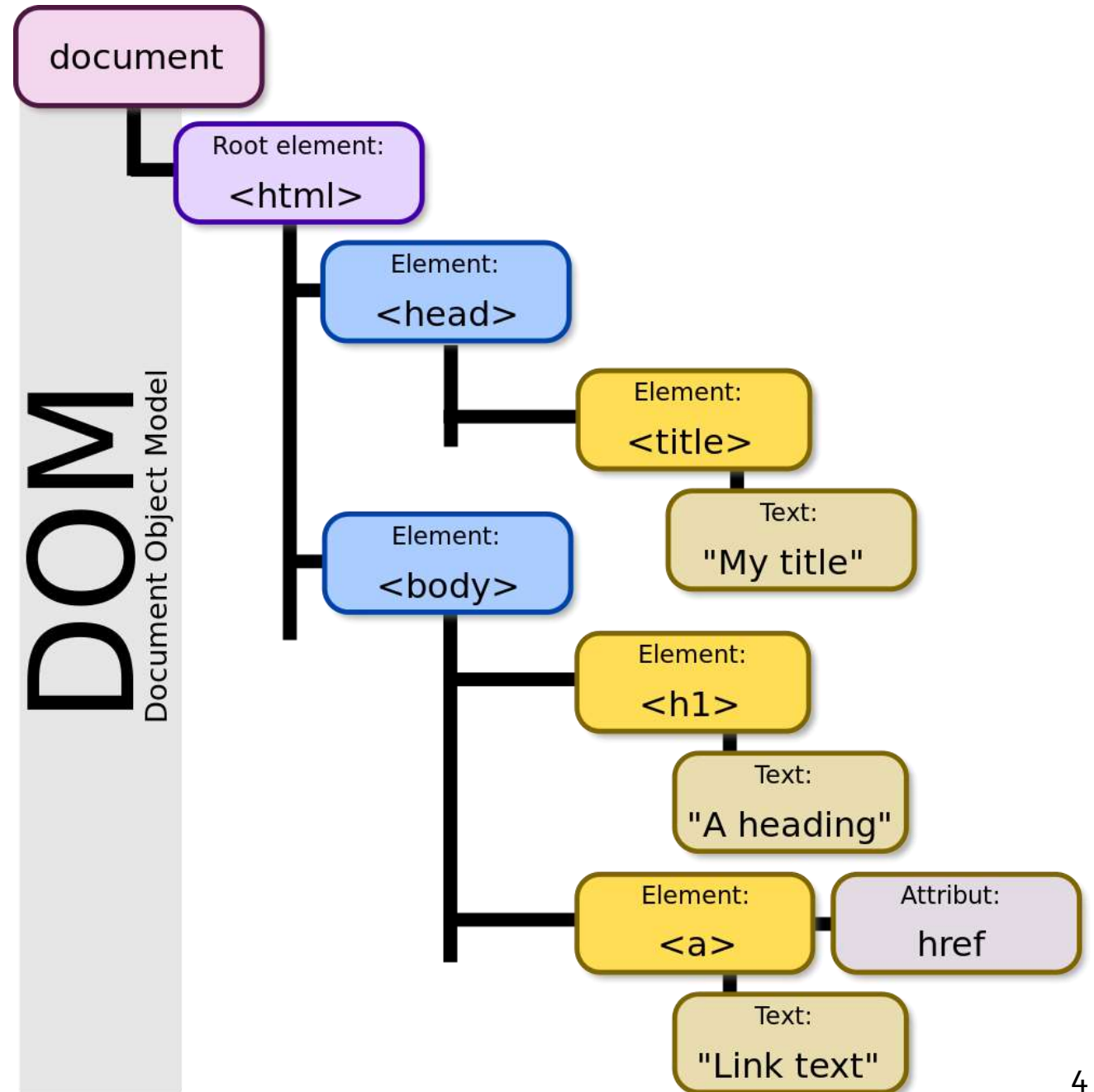


Write less, do more

- Bibliothèque JavaScript
- Assure une compatibilité multi-navigateurs
- Simplifie l'écriture de script
- Utilisé par de nombreux acteurs de premier plan du Web
- Peut demander à un serveur de mettre à jour une partie d'une page en utilisant du code AJAX

Le DOM (rappel)

- Document Object Model
- Structure de données qui représente un document HTML comme une arborescence
- Les balises HTML sont les nœuds



Attendre l'obtention complète du DOM

- `jQuery(document).ready(function() {
 // Ici, le DOM est entièrement défini
});`
- Utilisons toujours l'alias de la fonction jQuery noté \$
- `$(document).ready` peut être simplifié :
 `$(function() {
 // Ici, le DOM est entièrement défini
 });`

Hello World!

- Présentation de l'exemple 1 pour afficher Hello World!
- Voici le contenu de myscript.js

```
$(function() {  
    $('#texteJQ').html('Hello World! est affiché via jQuery.');
```
- });
- \$('#texteJQ') renvoie un objet jQuery
- .html est une fonction jQuery applicable sur un objet jQuery

Sélection d'éléments

- Sélectionner les nœuds du DOM qui nous intéressent
`$('Sélecteur')` où Sélecteur représente un sélecteur CSS ou un sélecteur jQuery

Sélecteur	<code>\$('Sélecteur')</code> renvoie sous forme d'objet(s) jQuery
<code>#texteJQ</code>	la balise d'identifiant texteJQ
<code>.bleu</code>	toutes les balises de classe bleu
<code>h1</code>	toutes les balises <code><h1></code>
<code>h1,h2</code>	toutes les balises <code><h1></code> et <code><h2></code>

Sélection d'éléments (suite)

Sélecteur jQuery	qui renvoie sous forme d'objet(s) jQuery toutes les balises
<code>\$('ul')</code>	<code></code>
<code>\$('ul.bleu')</code>	<code></code> de classe bleu
<code>\$('div ul')</code>	<code></code> contenues dans une <code><div></code>
<code>\$('li[class]')</code>	<code></code> qui ont un attribut class
<code>\$('li[class="impair"]')</code>	<code></code> qui ont un attribut class de valeur impair
<code>\$('div img[width="40"]')</code>	<code></code> qui ont un attribut width de valeur 40 contenues dans une <code><div></code>

Que renvoie la fonction jQuery \$('Sélecteur') ?

- Un objet jQuery qui représente sous forme d'objet(s) jQuery un (ou plusieurs) élément(s) du DOM
- C'est une sorte de tableau !
 - Qui a une propriété .length
 - \$('a').length donne le nombre de liens hypertextes du document
 - \$('a')[0] donne le premier lien hypertexte de la page sous forme d'élément du DOM et non sous forme d'objet jQuery !
 - \$('a:first') donne le premier lien hypertexte de la page sous forme d'objet jQuery

Appliquer une méthode à une sélection

- `$('Sélecteur').méthode(paramètres)`
- `text()` retourne ou modifie la valeur textuelle stockée dans l'élément
 - Traite la chaîne de caractères en paramètre comme du texte simple
- `html()` retourne ou modifie le code HTML stocké dans l'élément
 - Traite la chaîne de caractères en paramètre comme de l'HTML
- <http://stackoverflow.com/questions/1910794/what-is-the-difference-between-jquery-text-and-html>

Getteurs et setteurs ?

- `$('#texteJQ').html();`
 - Getteur !
- Pour mémoriser une valeur dans une variable écrire
`var mavariable = ...`
- `$('#texteJQ').html('Hello World! est affiché via jQuery.');`
 - Setteur !

Outils de développement

- En ligne
 - JSBIN <http://jsbin.com/?html,js,output>
 - Développement partagé
 - Débogage
 - JSFIDDLE <http://jsfiddle.net/>
- Hors ligne
 - Console Chrome
 - Console Firefox (+ extension Firebug)

Sélecteurs : une liste complète...

- Consultez l'excellente documentation en ligne <http://api.jquery.com/category/selectors/>
- Sélecteurs hiérarchiques
 - Expression parent > enfant
 - Recherche tous les éléments définis par le sélecteur "enfant", fils direct de l'élément parent spécifié par le sélecteur "parent"
 - Expression :last
 - Retourne le dernier élément de la sélection
 - Par exemple `$('div > ul > li:last').css("color","red");`

Modifier le CSS

- `$('Sélecteur').css('attribut')`
 - Récupère la valeur d'un attribut CSS
- `$('Sélecteur').css('attribut','valeur')`
 - Définit un attribut CSS
- `$('Sélecteur').css({"attribut1":"valeur1",
"attribut2":"valeur2"})`
 - Définit plusieurs attributs CSS

Exercices

Instruction jQuery	Effet
<code>\$('input').css('background','yellow');</code>	Tous les éléments de type input reçoivent un arrière-plan de couleur jaune.
<code>\$('#password').css('background','purple');</code>	Le champ de type password reçoit un arrière-plan de couleur violette.
<code>\$('#img').css('width','200px');</code>	Les champs images (tag img) sont redimensionnés pour avoir une largeur de 200 px.
<code>\$('#input:first').focus();</code> <code>\$('#:focus').css('background','green');</code>	La première instruction donne le focus au premier champ input et la deuxième colore son arrière-plan en vert.

Parcourir les éléments sélectionnés

- `$('Sélecteur').each(function(index) {
 // $(this) donne l'élément courant
});`
- index est une variable JavaScript qui représente la position de l'élément dans la sélection (0, 1, 2, etc. jusqu'à ce que tous les éléments soient passés en revue)
- <http://api.jquery.com/each/>
- <http://api.jquery.com/eq/>

La méthode val()

- Connaître ou modifier la valeur de zones de texte, radio boutons, cases à cocher, listes déroulantes dans un formulaire
- <http://api.jquery.com/val/>

Instruction jQuery	Effet
<code>\$('#nom').val()</code>	Obtient la valeur entrée dans la zone de texte <code><input type="text" id="nom"></code>
<code>\$('#nom').val('Yop')</code>	Ecrit Yop dans la zone de texte <code><input type="text" id="nom"></code>

Exercices

Instruction jQuery	Effet
<code>\$('input[name="sexe"]:checked').val();</code>	Renvoie la valeur H ou F du radio button sélectionné Homme <code><input type="radio" name="sexe" value="H"></code> Femme <code><input type="radio" name="sexe" value="F"></code>
<code>\$('input[type=radio]:checked').val();</code>	
<code>\$('input[type=radio]').val(['H']);</code>	Sélectionne le radio button dont la valeur est H
<code>\$('#fonction').val();</code>	Renvoie l'option sélectionnée dans <code><select id="fonction"></code> <code><option VALUE="étudiant"> Etudiant </option></code> <code><option</code> <code>VALUE="enseignant">Enseignant</option></code> <code></select></code>

Les événements définis directement dans le DOM

- Javascript en « direct »

```
<input type="button" value="exempledebutton"  
onclick="alert('événement onclick déclenché');" />
```

- Via une fonction séparée

```
<input type="button" value="exempledebutton"  
onclick="maFonction()" />
```

```
<script>function maFonction(){ ... }</script>
```

- Liste des événements :

http://www.w3schools.com/jsref/dom_obj_event.asp

Ajout d'un *listener* pour gérer un événement

- Voir [jQuery API](#) ou [W3Schools](#)
- On attache un *listener* sur un *event* suite à un sélecteur
 - `$('.a').click(function(){ alert("Click sur une balise de class a!"); });`
- Il existe une manière de lier générique : le *on*
 - `$('.c').on("mouseenter", function(){
 $('.c').text("Mouse au-dessus");
});`
- Voir un exemple sur [fiddle](#)

Notions concernant les événements

- Bubbling
 - Voir l'exemple sur [fiddle](#)
- Unbind
 - <http://api.jquery.com/unbind/>
 - Passer en paramètre l'*event* à ne plus gérer (déliier)

Insertion d'éléments dans le DOM

- `.append()`

Avant `<div class='a'>`
 `<div class='b'>b</div>`
 `</div>`
`$('.a').append("<div class='c'>c</div>");`
Après `<div class='a'>`
 `<div class='b'>b</div>`
 `<div class='c'>c</div>`
 `</div>`

- `.prepend()`

- `.after()` : ajoute après l'élément sélectionné, voir le [fiddle](#)

- `.before()`

- Voir [stackoverflow](#), et [W3Schools](#)