

Technologies Web

jQuery

Alexandre Pauchet

INSA Rouen - Département ASI

BO.B.RC.18, pauchet@insa-rouen.fr

Plan

- 1 Introduction
- 2 Fonctionnalités basiques
- 3 Interactivité

Introduction (1/2)

Généralités

- Description

- Framework (comme PrototypeJS, Mootools, Dojo, YahooUI, ...)
- Créé en Janvier 2006
- Très utilisé : >50% des sites dans le monde
- 2 versions : 1.X (1.11.3), 2.X (2.1.4), 3.X (3.3.1)

- Objectifs

- Régler les problèmes de compatibilité entre navigateurs
- Faciliter l'écriture de scripts

Introduction (2/2)

Utilisation

- Utilisation d'une version locale
 - Téléchargement sur www.jquery.com
 - `<script type="text/javascript" src="jquery.js"></script>`
- Utilisation de version en cache
 - Google CDN :
`<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>`
 - Microsoft CDN
 - CDNJS CDN
 - jsDelivr CDN

Fonctionnalités basiques (1/13)

Sélecteur d'élément

`jQuery('selecteur')` équivalent à `$('selecteur')`

avec `selecteur` = sélecteur de type CSS, i.e. :

- `#unID`, `.uneClass`, une balise HTML, ...
- Référence : <http://api.jquery.com/category/selectors/>

Rappels : attente de la fin du chargement du DOM

- Une page HTML est un arbre dans lequel les éléments (nœuds) sont séquentiels
- Une page HTML est traitée par un navigateur comme un flux
- Pour qu'un nœud soit disponible, il est nécessaire qu'il soit entièrement chargé (balises d'ouverture et de fermeture)

```
$(document).ready( function() {  
    alert('Le DOM est prêt !');  
});
```

Fonctionnalités basiques (2/13)

Exemple

Exemple1.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemple jQuery 1</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  </head>
  <body>
    <header>
      <h1>Exemple jQuery (document ready)</h1>
    </header>
    <footer>
      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
      </script>
      <script>
        $(document).ready( function() {
          alert('Le dom est prêt ! ');
        });
      </script>
    </footer>
  </body>
</html>
```

Fonctionnalités basiques (3/13)

Parcours du DOM

- Éléments fils : `$('#selecteur1').find('selecteur2');` (équivalent à `$('#selecteur1 selecteur2');`)
- Éléments fils directs : `$('#selecteur1').children('selecteur2');`
- Élément parent : `$('#selecteur1').parent('selecteur2');`
- Élément suivant : `$('#selecteur1').next('selecteur2');`
- Élément précédent : `$('#selecteur1').prev('selecteur2');`
- Boucle sur une liste d'éléments :
`$('#selecteur').each(function(index) { ... });`

NB1 : chaque méthode a un équivalent sélecteur de CSS

(ex : `$('#selecteur1').find('selecteur2');` équivaut à `$('#selecteur1 selecteur2');`)

NB2 : `selecteur2` peut être vide

Fonctionnalités basiques (4/13)

Exemple

Exemple2.html

```
<body>
  <header>
    <h1>Exemple jQuery (sélecteurs)</h1>
  </header>
  <div id="main">
    <div>Div 1</div>
    <ul>
      <li>Li 1</li>
      <li>Li 2</li>
      <li>Li 3</li>
      <li>Li 4</li>
    </ul>
  </div>
  <footer>
    <script src="jquery-3.3.1.min.js"></script>
    <script>
      $(document).ready( function() {
        $( "#main" ).children().each(function( index ) {
          console.log( "Dans #main, on a un " + this.nodeName );
        });
      });
    </script>
  </footer>
</body>
```


Fonctionnalités basiques (5/13)

Gestion des classes

- Test d'une classe : `$('#selecteur').hasClass('classe');`
- Ajout d'une classe : `$('#selecteur').addClass('classe');`
- Suppression d'une classe : `$('#selecteur').removeClass('classe');`
- Ajout d'une classe, la supprime si elle existe :
`$('#selecteur').toggleClass('highlighted');`

Fonctionnalités basiques (6/13)

Gestion des attributs

- Valeur d'un attribut : `$('#selecteur').attr('attribut');`
NB : `undefined` est renvoyé si l'élément ne dispose pas de l'attribut

- Assignment d'un attribut : `$('#selecteur').attr('attribut', 'valeur');`

- Assignment d'un ensemble d'attributs :

```
$('#selecteur').attr( {  
    'attribut1' : 'valeur1',  
    ...  
    'attributn' : 'valeurn'  
});
```

Fonctionnalités basiques (7/13)

Gestion des CSS

- Modification d'un attribut de la CSS d'un élément :

```
$('#selecteur').css('attribut-de-css', 'valeur');
```

- Modification de plusieurs attributs de la CSS d'un élément :

```
$('#selecteur').css( {  
  'attribut-de-css1' : 'valeur1',  
  ...  
  'attribut-de-cssn' : 'valeurn'  
});
```

Fonctionnalités basiques (8/13)

Gestion des dimensions

`width()`

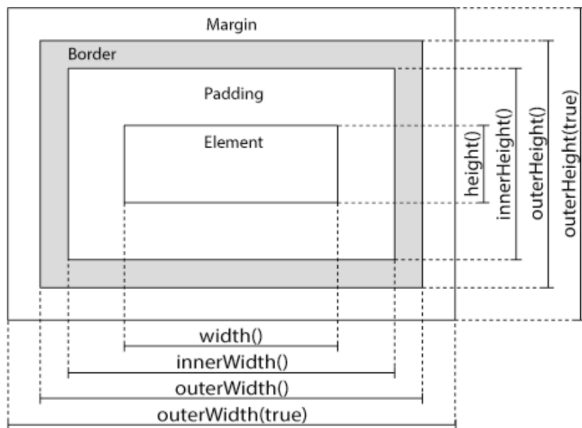
`height()`

`innerWidth()`

`innerHeight()`

`outerWidth()`

`outerHeight()`



NB : ces fonctions permettent également de fixer les valeurs

Fonctionnalités basiques (9/13)

Exemple

Exemple3.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemple jQuery 3</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <style>
      #main { width: 300px; padding: 10px; border: solid grey 2px; margin: 20px; }
    </style>
  </head>
  <body>
    <header><h1>Exemple Jquery (Dimensions)</h1></header>
    <div id="main">Contenu de la div</div>
    <input type="button" value="Élargir" onclick="$('#main').width($('#main').width()+100);affichage();"></input>
    <footer>
      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
      </script>
      <script>
        function affichage() {
          console.log( "width=" + $("#main").width() + " ; innerWidth=" + $("#main").
            innerWidth() + " ; outerWidth=" + $("#main").outerWidth() + " ; outerWidth(true)="
            + $("#main").outerWidth(true));
        }
        $(document).ready(affichage);
      </script>
    </footer>
  </body>
</html>
```

Fonctionnalités basiques (10/13)

Gestion du contenu des éléments

- Contenu textuel

- Récupération : `$('#selecteur').text();`
- Modification : `$('#selecteur').text('Nouveau contenu');`

- Contenu HTML

- Récupération : `$('#selecteur').html();`
- Modification : `$('#selecteur').html('<p>Nouveau contenu avec balises HTML!</p>');`

- Attribut `value`

- Récupération : `$('#selecteur').val();`
- Modification : `$('#selecteur').val('Nouvelle valeur');`

Fonctionnalités basiques (11/13)

Gestion du contenu des éléments

- Ajout au début d'un élément : `$('#selecteur').prepend('Contenu');`
- Ajout à la fin d'un élément : `$('#selecteur').append('Contenu');`

- Ajout avant un élément : `$('#selecteur').before('Contenu');`
- Ajout après un élément : `$('#selecteur').after('Contenu');`

NB : le contenu ajouté peut être constitué de balises HTML

- Suppression du contenu d'un élément : `$('#selecteur').empty();`
- Suppression d'un élément : `$('#selecteur').remove();`

NB : un élément supprimé du DOM peut encore être utilisé

Fonctionnalités basiques (12/13)

Exemple

Exemple4.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemple jQuery 4</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <style>
      #main{ width: 300px; padding: 10px; border: solid grey 2px; margin: 20px;
    }
    </style>
  </head>
  <body>
    <header><h1>Exemple jQuery (Ajout de contenu)</h1></header>
    <div id="main">Contenu de la div</div>
    <footer>
      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
      </script>
      <script>
        $(document).ready( function() {
          $('#main').append('<br/>Ajouté dedans à la fin');
          $('#main').prepend('Ajouté dedans au début<br/>');
          $('#main').before('Ajouté avant');
          $('#main').after('Ajouté après');
        });
      </script>
    </footer>
  </body>
</html>
```


Fonctionnalités basiques (13/13)

Chaînage des éléments

Toutes les fonctions retournent une liste d'éléments (objet \$).
Elles peuvent donc être enchaînées (**chaînables**).

Exemple

```
balises = $(".toto");  
balises = balises.append(" un texte");  
balises = balises.css("border", "1px solid red");  
balises = balises.addClass("titi");  
balises = balises.removeClass("toto");  
balises = balises.removeClass("tutu");
```

est équivalent à :

```
$(".toto").append(" un texte").css("border", "1px solid red").  
addClass("titi").removeClass("toto").removeClass("tutu");
```

Interactivité (1/7)

Animation d'éléments

- Visibilité des éléments

- Apparition : `$('#selecteur').show();`
- Masquage : `$('#selecteur').hide();`
- Apparition ou masquage : `$('#selecteur').toggle();`

- Animations avec effet

- Apparition : `$('#selecteur').slideDown();`
- Masquage : `$('#selecteur').slideUp();`
- Apparition : `$('#selecteur').fadeIn();`
- Masquage : `$('#selecteur').fadeOut();`

- Animation personnalisée

```
$('#selecteur').animate({  
    attribut-CSS1: valeur1,  
    ..  
    attribut-CSSn: valeurn  
}, vitesse );
```

Interactivité (2/7)

Exemple

Exemple5.html

```
<body>
  <header><h1>Exemple JQuery (Animations)</h1></header>

  <div id="bloquehauteur">
    <div id="main">Contenu de la div</div>
  </div>
  <a href="#" id="hide">hide</a> <a href="#" id="show">show</a> <a href="#" id="
  toggle">toggle</a> <br/>
  <a href="#" id="slideDown">slideDown</a> <a href="#" id="slideUp">slideUp</a><br/>
  <a href="#" id="fadeIn">fadeIn</a> <a href="#" id="fadeOut">fadeOut</a><br/>

  <footer>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
    </script>
    <script>
      $(document).ready( function() {
        $('#hide').click(function(){$('#main').hide();});
        $('#show').click(function(){$('#main').show();});
        $('#toggle').click(function(){$('#main').toggle();});

        $('#slideDown').click(function){$('#main').slideDown();});
        $('#slideUp').click(function){$('#main').slideUp();});

        $('#fadeIn').click(function){$('#main').fadeIn( "slow");});
        $('#fadeOut').click(function){$('#main').fadeOut( "slow");});
      });
    </script>
  </footer>
</body>
```

Interactivité (3/7)

Exemple

Exemple6.html

```
<body>
  <header><h1>Exemple JQuery (Animate)</h1></header>
  <div id="main">Contenu de la div</div>
  <footer>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
  </script>
  <script>
    $(document).ready( function() {
      $('#main').animate({
        width: "70%",
        fontSize: "2em",
        borderWidth: "10px"
      }, 1500 );
    });
  </script>
</footer>
</body>
```

ÉVÉNEMENTS

<i>click</i>	Lors d'un clic sur un élément
<i>load</i>	Quand un élément à fini de se charger
<i>change</i>	Lors du changement d'une valeur d'un champ de formulaire
<i>keydown</i>	Lors de l'appui sur une touche
<i>keyup</i>	Lorsqu'une touche est relâchée
<i>mousedown</i>	Lorsqu'un bouton de la souris est enfoncé
<i>mouseup</i>	Lorsqu'un bouton de la souris est relâché
<i>submit</i>	Lors de la validation d'un formulaire
<i>focusin</i>	Lorsque l'on met le focus sur un champs
<i>focusout</i>	Lorsque le focus sort du champs

NB : la bonne pratique est de privilégier la déclaration d'événements javascript par une fonction plutôt qu'en utilisant les attributs correspondants

Liste de compatibilité des events : <http://www.quirksmode.org/dom/events/>

«THIS»

```
$( "p" ).click(function() {  
  $( this ).toggleClass( "highlight" );});
```

Lors de l'activation d'un événement, «this» est la référence javascript de l'objet courant.

Mais pour en faire un objet jquery => \$(this)

LES FONCTIONS ANONYMES

```
$('#a.okbutton').click( function() {  
alert('Le bouton Ok');  
});
```

Manque de lisibilité et des «réutilisabilité» des
«function(){...}»

```
$('#a.okbutton').click( dookbutton);function dookbutton() {  
alert('Le bouton Ok');  
}
```

Les déclarations de fonction n'ont pas à être dans le
\$(document).ready(...);

FONCTIONS UTILES POUR LE DEBUG

```
alert('Ca marche');
```

Mieux :

```
console.log('Texte');
```

```
console.log(objet);
```

Affiché dans la console du navigateur, avec la visibilité complète sur l'objet.