

*Crédit : O'REILLY*

# Qu'est-ce que jQuery ?

---

- jQuery = **bibliothèque JavaScript** = **ensemble de fonctions** permettant de manipuler très facilement et de façon très concise l'arbre DOM
- On peut utiliser les fonctions jQuery **dans n'importe quel programme JavaScript** !
- Comment inclure la bibliothèque jQuery dans une page web ?
  - Utiliser une version hébergée en ligne

```
<script type="text/javascript"
  src="http://code.jquery.com/jquery-1.10.1.min.js"></script>
```
  - Utiliser une version téléchargée localement

```
<script type="text/javascript" src="jquery.js"></script>
```

# Quelques références

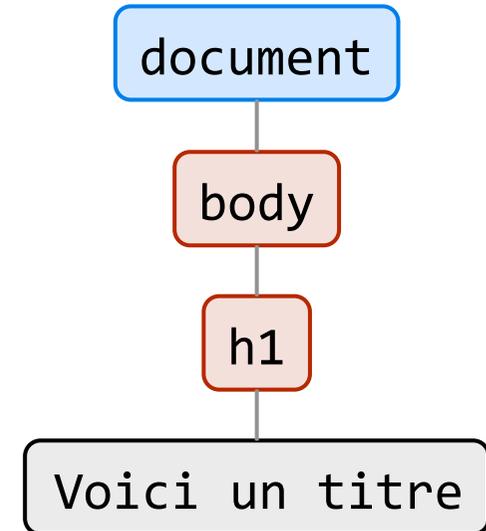
---

- Documentation de [référence sur jQuery](http://jquery.com/) :
  - <http://jquery.com/>
- Pour [tester du code](http://jsbin.com/) JavaScript et jQuery en ligne :
  - <http://jsbin.com/>
- Plugin jQuery pour les [tables](http://datatables.net/) :
  - <http://datatables.net/>
- Plugin jQuery pour valider les [formulaires](http://docs.jquery.com/Plugins/Validation) :
  - <http://docs.jquery.com/Plugins/Validation>

# Petit rappel de JavaScript & DOM

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>jQuery !</title>
    <script type="text/javascript">
// <![CDATA[
"use strict";

window.onload = function() {
  var d = window.document;
  var noeudH1 = d.createElement("h1");
  var noeudTexte = d.createTextNode("Voici un titre");
  noeudH1.appendChild(noeudTexte);
  d.body.appendChild(noeudH1);
}
// ]]>
    </script>
  </head>
  <body>
  </body>
</html>
```



# Le même exemple, avec jQuery

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>jQuery !</title>
    <script type="text/javascript"
      src="http://code.jquery.com/jquery-1.10.1.min.js"></script>
    <script type="text/javascript">
// <![CDATA[
"use strict";

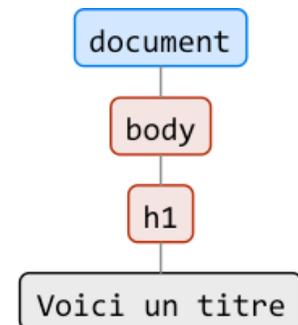
$(document).ready(function(){
  $(document.body).append( $("<h1>").text("Voici un titre") );
});

// ]]>
    </script>
  </head>
  <body>
  </body>
</html>
```

Inclusion de la bibliothèque jQuery

Fonction appelée une fois le document chargé

Sélection, création de nœuds & modification de l'arbre DOM



# La fonction \$

---

- jQuery définit une fonction principale :  
`jQuery()` aussi appelée `$()`
- 3 grands types d'utilisations :
  - Avec du texte représentant un « sélecteur » : `$("#link")`  
➔ retourne un objet jQuery qui référence un ou plusieurs nœuds DOM ayant « matché » le sélecteur = recherche
  - Avec du texte représentant du HTML : `$("<h1>")`  
➔ crée un ou plusieurs éléments DOM et retourne un objet jQuery qui le/les référence = création
  - Avec un élément DOM : `$(document.body)`  
➔ retourne un objet jQuery qui référence l'élément

# Rechercher dans le DOM : les sélecteurs

- Sélection des noeuds :

- Par nom d'élément (balise) : `$("nomElt")`

- Par identifiant : `$("#id")`

- Par classe : `$(".nomClasse")`

- Tous : `$("*")`

- En tenant compte de la hiérarchie :

- Tous les descendants : `$("nomEltAncetre nomEltDescendant")`

- Descendants directs : `$("nomEltParent > nomEltEnfant")`

- En tenant compte des attributs :

- Test de l'existence : `$("nomElt[nomAttr]")`

- Test de la valeur : `$("nomElt[nomAttr='value']")`

# Exemples

```
<h2 id="selecteurs">Sélecteurs</h2>
```

```
<h3>Définition</h3>
```

```
<p>Ce sont des expressions qui permettent de <strong>sélectionner</strong> des éléments dans l'arbre DOM. Retrouvez la documentation officielle sur <a href="http://api.jquery.com/category/selectors/">la page des sélecteurs jQuery</a>.</p>
```

```
<h3>Exemples de sélecteurs</h3>
```

```
<table class="syntax">
  <tr><th>Type</th><th>Syntaxe</th></tr>
  <tr><td>Par nom d'élément</td><td>$("nomElt")</td></tr>
  <tr><td>Par identifiant</td><td>$("#id")</td></tr>
  <tr><td>Par class</td><td>$(".nomClasse")</td></tr>
</table>
```



# Opérations sur les collections

- Obtenir la taille de la collection : `$("#td").length`
- Sélectionner des éléments de la collection :
  - Premier : `$("#td").first()`
  - Dernier : `$("#td").last()`
  - Ième : `$("#td").eq(i)`
- Appliquer une fonction à tous les éléments d'une collection :

L'argument `index` est optionnel

```
$("#td").each(function(index){  
    console.log(index + ":" + $(this));  
});
```

// affiche successivement chacun des éléments td

`this` = objet courant  
(change à chaque  
tour de « boucle »)

# Naviguer dans le DOM

- A partir d'un élément de l'arbre :

- Accéder aux attributs :

`.attr("nomAttr")`

- Accéder aux parents, aux ancêtres :

`.parent(), .parents()`

- Accéder aux enfants :

`.children()`

- Accéder au texte de l'élément : `.text()`

- Accéder à l'ensemble du sous-arbre : `.html()`

- Accéder aux frères :

`.siblings(), .next(), .prev()`

Argument optionnel :  
sélecteur, qui permet de  
« filtrer » la navigation

# Exemple

```
<h2 id="selecteurs">Sélecteurs</h2>
```

```
<h3>Définition</h3>
```

```
<p>Ce sont des expressions qui permettent de <strong>sélectionner</strong> des éléments dans l'arbre DOM. Retrouvez la documentation officielle sur <a href="http://api.jquery.com/category/selectors/">la page des sélecteurs jQuery</a>.</p>
```

```
<h3>Exemples de sélecteurs</h3>
```

```
<table class="syntax">
```

```
  <tr><th>Type</th><th>Syntaxe</th></tr>
```

```
  <tr><td>Par nom d'élément</td><td>$("nomElt")</td></tr>
```

```
  <tr><td>Par identifiant</td><td>$("#id")</td></tr>
```

```
  <tr><td>Par class</td><td>$(".nomClasse")</td></tr>
```

```
</table>
```

```
$(".td").each(function(){
  console.log($(this).text());
});
// affiche successivement le texte de chaque élément td
```

---

**LEARN**

Exercise 1



# Créer un élément DOM

- Création simple : `$("#baliseElement");`

Exemple : `$("<h1>");`  
`$("<p>");`

- Avec des attributs : `$("#baliseElement", {  
 nomAttr1: "valeurAttr1",  
 nomAttr2: "valeurAttr2",  
 ... }  
);`

Exemple : `$("<a>", {  
 href : "http://www.supelec.fr",  
 title: "Allez sur le site de l'école !",  
});`

# Modifier l'arbre DOM

- A partir d'un élément de l'arbre :
  - Insertion/modification d'attribut :  
`.attr("nomAttr", "valeurAttr")`
  - Insertion :
    - D'enfants : `.append(...)`, `.prepend(...)`
    - De frères : `.after(...)`, `.before(...)`
    - De parents : `.wrap(...)`, `.unwrap(...)`
    - De texte : `.text(...)`
    - D'un sous-arbre entier : `.html(...)`
  - Remplacement de l'élément par un autre :  
`.replaceWith(...)`
  - Suppression de l'élément :  
`.remove()`, `.detach()`

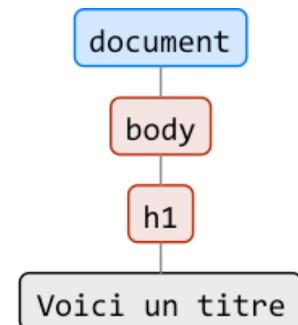
Argument(s) :  
élément DOM,  
objet jQuery,  
*texte HTML*

# Exemple I (rappel)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>jQuery !</title>
    <script type="text/javascript"
      src="http://code.jquery.com/jquery-1.10.1.min.js">
    </script>
    <script type="text/javascript">
// <![CDATA[
"use strict";

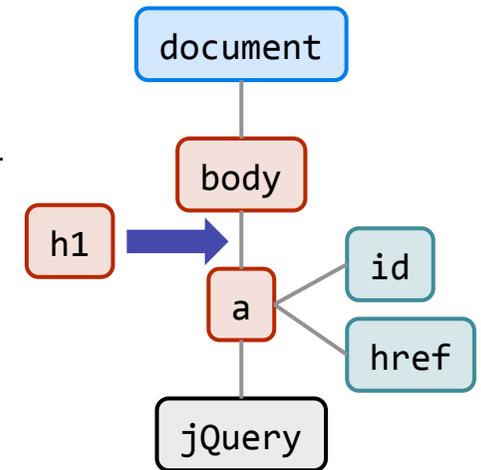
$(document).ready(function(){
  $(document.body).append( $("<h1>").text("Voici un titre") );
});

// ]]>
    </script>
  </head>
  <body>
  </body>
</html>
```



# Exemple 2

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>jQuery !</title>
    <script type="text/javascript"
      src="http://code.jquery.com/jquery-1.10.1.min.js"></script>
    <script type="text/javascript">
// <![CDATA[
"use strict";
$(document).ready(function(){
  var link = $("#link");
  $(document.body).append( $("<h1>").append(link) );
});
// ]]>
  </script>
</head>
<body>
  <a id="link" href="http://jquery.com/">jQuery</a>
</body>
</html>
```



1 Sélection du nœud <a>

4 Insertion du nœud <h1> en fils de <body>

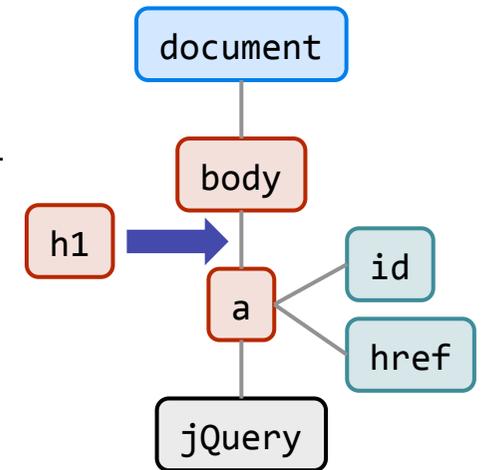
2 Création du nœud <h1>

3 Déplacement du nœud <a> en fils de <h1>

# Le même exemple, en + concis

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>jQuery !</title>
    <script type="text/javascript"
      src="http://code.jquery.com/jquery-1.10.1.min.js"></script>
    <script type="text/javascript">
// <![CDATA[
"use strict";

$(document).ready(function(){
  $("#link").wrap( $("<h1>" ) );
});
// ]]>
    </script>
  </head>
  <body>
    <a id="link" href="http://jquery.com/">jQuery</a>
  </body>
</html>
```



# Exemple complet : créer un menu

```
<div id="menu">
  <ul class="menu">
    </ul>
</div>
<div id="contenu">
  <h2 id="selecteurs">Sélecteurs</h2>
  ...
  <h2 id="operations">Opérations</h2>
  ...
</div>
```

Insertion d'éléments <li> contenant des liens cliquables vers les différents titres <h2>

```
$(document).ready(function(){
  $("#contenu h2[id]").each(function(){
    var link = $("<a>", {href : "#" + $(this).attr("id")});
    link.text($(this).text());
    $("ul.menu").append(
      $("<li>").append(link)
    );
  });
});
```

---

**LEARN**

Exercise 2



# Styles CSS

- Propriétés CSS

- Lecture :

```
.css("nomPropriete")
```

- Modification :

```
.css("nomPropriete", "valeur")
```

```
.css({"prop1" : "val1", "prop2" : "val2"})
```

Exemple :

```
$("#h2").css("background-color", "#FFDD99");
```

- Classes CSS

```
.addClass("nomClasse")
```

```
.removeClass("nomClasse")
```

Il faut que la classe CSS soit définie, soit dans le fichier HTML soit dans un fichier de style

Exemple :

```
$("#table").addClass("fondcouleur");
```

```
avec : .fondcouleur { background-color:gray; }
```

---

**LEARN**

Exercise 3



# Événements & gestionnaires

- **Événements** : déjà vus en JavaScript

Souris	Clavier	Formulaire	Fenêtre / objet HTML
click	keydown	focus	load
dblclick	keyup	blur	unload
mousedown	keypress	select	abort
mouseup		change	resize
mouseover		submit	scroll
...		reset	...

- **Gestionnaire** d'événement : fonction attachée à un objet jQuery (qui référence un élément DOM), qui sera appelée en réaction à un événement donné

```
objetjQuery.nomEvenement(function(){  
    // code à exécuter lorsque l'événement se produit  
});
```

# Exemples

**Attention, à déclarer à l'intérieur de :**

```
$(document).ready(function(){ ... });
```

- Réagir aux clics de souris :

```
$("#h2").click(function(){  
    console.log($(this).text());  
});
```

+ Désactiver le comportement par défaut de l'élément vis-à-vis des événements :

```
$("#a").click(function(event){  
    console.log($(this).text());  
    event.preventDefault();  
});
```

- Réagir au passage de la souris :

2 événements  
➔ 2 fonctions

```
$("#h2").mouseenter(function(){  
    $(this).css("text-decoration", "underline");  
});  
  
$("#h2").mouseleave(function(){  
    $(this).css("text-decoration", "none");  
});
```

# Effets d'animation

---

- jQuery permet également de **déclencher des effets d'animation** sur des éléments de l'arbre DOM :

.hide(), .show(), .toggle()

.fadeIn(), .fadeOut(), .fadeToggle()

.slideDown(), .slideUp(), .slideToggle()

– Arguments optionnels :

- Temps en millisecondes, **durée** de l'animation
- Fonction de **callback**, appelée automatiquement à la fin de l'animation

- Exemple :

```
$("#clickMe").click(function(){  
    $("#contentToHide").slideToggle();  
});
```

# Exemple complet : masquer du contenu

```
<h3>Exemples de sélecteurs</h3>
<table class="syntax">
  <tr><th>Type</th><th>Syntaxe</th></tr>
  <tr><td>Par nom d'élément</td><td>$("nomElt")</td></tr>
  <tr><td>Par identifiant</td><td>$("#id")</td></tr>
  <tr><td>Par class</td><td>$(".nomClasse")</td></tr>
</table>
```

```
$("h3").hover(function(){
  $(this).css("text-decoration", "underline");
}, function(){
  $(this).css("text-decoration", "none");
});
```

```
$("h3").click(function(){
  $(this).next("table").toggle();
});
```

Toggle :

- si la table est cachée alors elle est montrée
- si la table est montrée, alors elle est cachée

---

**LEARN**

Exercice 4



# Filtres

- jQuery définit également des **versions « raccourcies »** de certaines **expressions de sélection souvent utilisées**, notamment en ce qui concerne les formulaires

– Utilisation :

- Dans l'expression d'un sélecteur : `$("selecteur:filtre")`

- Avec la fonction « filter » :

```
$("#selecteur").filter(":filtre")
```

– Exemples de filtres :

- Type d'élément : `:input, :checkbox, :radio, :password`

- État d'élément : `:focus, :checked, :selected`

- Ou encore : `:even, :odd, :empty, :contains(...texte...)`

- Utilisation avec des **fonctions utilitaires** comme : `.not(...)`

# Exemple 1

Exemples de sélecteurs

Type	Syntaxe
Par nom d'élément	<code>\$("nomElt")</code>
Par identifiant	<code>\$("#id")</code>
Par classe	<code>\$(".nomClasse")</code>

```
<h3>Exemples de sélecteurs</h3>
<table class="syntax">
  <tr>
    <th>Type</th><th>Syntaxe</th>
  </tr>
  <tr>
    <td>Par nom d'élément</td><td>$("nomElt")</td>
  </tr>
  <tr>
    <td>Par identifiant</td><td>$("#id")</td>
  </tr>
  <tr>
    <td>Par classe</td><td>$(".nomClasse")</td>
  </tr>
</table>
```

```
$("tr:odd").each(function(){
  $(this).css("background-color", "#99CCFF");
});
```

## Exemple 2 : formulaires

### Opérations sur l'arbre DOM

Opérations à rechercher :

Navigation

Insertion

Suppression

```
<h2 id="operations">Opérations sur l'arbre DOM</h2>
<form action="" method="POST" id="operation-dom">
  <label for="operation">Opérations à rechercher :</label>
  <p>
    <input type="checkbox" name="operation" value="3"/>Navigation
  </p>
  <p>
    <input type="checkbox" name="operation" value="2"/>Insertion
  </p>
  <p>
    <input type="checkbox" name="operation" value="1"/>Suppression
  </p>
  <p><input type="submit" value="Soumettre"/></p>
</form>
```

```
$("#operation-dom").submit(function(event){
  console.log($("#input").filter(":checked").val());
  // affiche la valeur de l'attribut "value" de l'élément coché
  event.preventDefault();
});
```

---

**LEARN**

Exercise 5



---

# Conclusion



# A retenir !

---

- jQuery = **fonctions** et objets **JavaScript**  
➔ On peut utiliser jQuery **dans tout programme JavaScript !**

- Attention, pour pouvoir utiliser jQuery :



- Ne pas oublier d'**inclure la bibliothèque** jQuery dans la page HTML concernée...

- **Attendre que le DOM soit chargé** avant d'utiliser les fonctions jQuery !

- jQuery peut être **étendu** avec un mécanisme de plugins

- Pour la validation des formulaires

- Pour la mise en forme des tables

- ...