



1

# JQUERY

**write less, do more**

*(écrivez moins, faites plus)*

*J-M Pecatte – S. Trouilhet*

# JQUERY

- JQuery est un framework javascript : ensemble de fonctions (bibliothèque) mais aussi une nouvelle manière de coder (nouveau langage) qui a pour objectif de simplifier les commandes communes de javascript ;
- jQuery est centré sur la manipulation des pages HTML en combinant l'utilisation du DOM (Document Object Model) et les CSS (Cascading Style Sheets), la gestion des événements et les animations ;
- jQuery uniformise le code à écrire : un script jQuery marchera de la même manière sur TOUS les navigateurs.
- jQuery a été créé à l'initiative de John Resig en 2006.
- JQuery site officiel : <http://jquery.com/>

# JQUERY

- JQuery se présente comme un fichier javascript à inclure dans la page html :
- Soit en téléchargeant le fichier et en faisant un lien local  
`<script type="text/javascript" src="jquery.min.js"></script>`
- Soit en mettant un lien vers le fichier sur internet (CDN : Content Delivery Network) [mis en cache donc plus efficace]

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js">  
</script>
```

# LA FONCTION JQUERY() OU \$()

- jQuery définit une fonction principale appelée jQuery() ou encore en abrégé \$()
- 2 utilisations principales :
  - Sélectionner un ou plusieurs nœuds DOM à l'aide de sélecteurs

`$('#h1')` → sélectionne tous les nœuds éléments h1

*Rappel javascript :*

*`document.getElementsByTagName("h1")`*

- Créer un ou plusieurs nœuds DOM à l'aide de texte HTML

`$('<h1>')` → créer un nœud élément h1

*Rappel javascript : `document.createElement("h1")`*

# RECHERCHER DANS LE DOM

- Sélection de nœuds :

- par nom d'éléments HTML \$("h1")  
*javascript : getElementsByTagName*
- par identifiant \$("#id")  
*javascript : getElementById*
- par classe \$(".classe")  
*javascript : getElementsByName*

- En tenant compte des attributs :

- Sélecteur d'attribut \$("elem[attr]")  
*javascript : getElementsByAttribute(attr, "\*")*

# RECHERCHER DANS LE DOM

- En tenant compte des attributs :
  - test de valeur  
`$("elem[attr='val']")`
  - *javascript : `getElementsByAttribute(attr,"val")`*
- En tenant compte de la hiérarchie :
  - tous les descendants `$("elem elemdesc")`
  - tous les descendants directs `$("elem > enfants")`

# OPÉRATIONS SUR DES ENSEMBLES DE NOEUDS

- Une recherche dans le DOM peut renvoyer un ensemble de nœuds (une collection)

`$("body<p")[0]`

- Le nombre d'éléments de l'ensemble : `$("td").length`

- Un élément particulier de l'ensemble :

- le premier `$("td").first()`
- le dernier `$("td").last()`
- le ième `$("td").eq(i)`

- Appliquer un traitement à chaque élément :

`$("td").each( ... $(this)....)` avec `$(this)` est l'objet courant

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

# NAVIGUER DANS LE DOM

- Accéder au contenu `.text()` `.html()`  
`$("#h1").text()` texte de la balise h1
- Accéder au parent `.parent()`
- aux ancêtres `.parents()`
- Accéder aux enfants `.children()`
- Accéder aux frères `.siblings()` `.next()` `.prev()`



# CRÉER DES ÉLÉMENTS

- Création d'un élément simple et vide `$("<h1>")`  
`$("<p>")`
- Création d'un élément avec du contenu  
`$("<p>").text("abc")`  
`<p> abc </p>`
- Création d'un élément avec des attributs  
`$("<a>", { href : "http://monsite.fr", title : "mon site" })`  
`<a href="http://monsite.fr" title="mon site"></a>`
- Rajouter un attribut à un élément  
`$("a").attr("href", "http://site.fr").attr("title", "mon site")`  
`<a href="http://site.fr" title="mon site"></a>`

# MODIFIER L'ARBRE DU DOM

- Insertion/modification du texte  
`.text(...)`
- Insertion/modification d'un sous arbre  
`.html(...)`
- insertion d'enfants `.append(...)` à la fin de la sélection  
`.prepend(...)`, au début de la sélection
- insertion de parents `.wrap(...)`
- insertion de frères `after(...)` `.before(...)`
- suppression `.remove()` `.detach()`

- Ajouter un élément fils à un élément :

`$(" p").append($("<span>"))`

*ajouter un éléments span dans les paragraphes*

## LIRE / MODIFIER LES STYLES

- lire une propriété d'un style `.css('propriété')`
- création/modification d'un style  
`.css('propriété','valeur')`

exemple : `$('#h1').css("font-weight","bold");`

- lire / modifier hauteur et largeur `.width()`  
`.height()`

# EVÈNEMENTS ET ÉCOUTEURS

- jQuery est une bibliothèque javascript : les événements sont ceux de javascript
  - positionner un écouteur sur la cible pour un type d'événement et définir le traitement associé :
    - `$(cible).typeEvenement( function (event) { ... traitement....})`
  - Un programme jQuery commence souvent par ces lignes :

```
$.ready(function() {  
    // tout le code du programme  
})
```
- ➔ ceci permet de n'exécuter le code du programme que lorsque le document est entièrement chargé.

# EVÈNEMENTS ET ÉCOUTEURS

- Exercice : mettre les titres h1 en rouge au click sur un titre

Type	click
Cible	Les balises h1
Action	Mettre en rouge le texte du titre

```
$("h1").click(function(event) {  
    $(this).css("color","red");  
});
```

`$(this)` correspond à la cible de l'événement c'est-à-dire la balise sur laquelle on a cliqué

# EVÈNEMENTS ET ÉCOUTEURS

- désactiver le comportement par défaut lié à l'événement
  - `event.preventDefault();`

# EFFETS D'ANIMATIONS

- cacher / montrer `.hide()` `.show()`
  - cacher si visible / montrer si caché `.toggle()`
  - afficher/cacher en ajustant l'opacité  
`.fadeIn()` `.fadeOut()` `.fadeToggle()`
  - afficher/cacher avec un effet de glissement sur la hauteur  
`.slideDown(...)` `.slideUp(...)` `.slideToggle()`
- ces fonctions acceptent 2 paramètres (facultatifs) :
- la vitesse en ms ou 1 des 3 constantes (slow, normal, fast)
  - une fonction qui sera appelée quand l'animation est terminée

# EFFETS D'ANIMATIONS

- animation personnalisée

`.animate()`

→ L'aspect clé de cette fonction est l'objet composé des propriétés de style sur lesquelles seront basées l'animation. Chaque paramètre de l'objet représente une propriété sur laquelle portera l'animation (exemple: "height", "top", ou "opacity").

exemple : `animate({ width: "70%",`

`opacity: 0.4,`

`marginLeft: "0.6in",`

`fontSize: "3em",`

`borderWidth: "10px"`

`}, 1500 );`