

jQuery : fondamentaux

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille
Chercheur en programmation par contrainte (IA)
Ingénieur en génie logiciel

`elmouelhi.achref@gmail.com`

Plan

- 1 Introduction
- 2 Intégration de jQuery
- 3 Un premier Hello world
- 4 Méthodes .html() et .text()
- 5 Méthodes .attr(), .val() et .prop()
- 6 Méthodes .css() et .xClass()
- 7 Méthodes de filtrage et de recherche
- 8 Méthodes .wrap() et .data()
- 9 Ajout et suppression d'un nœud
- 10 Évènements
- 11 Quelques opérations sur les tableaux
- 12 jQuery : alternatives modernes

jQuery



jQuery

jQuery

- une bibliothèque **JavaScript** lancée par **John Resig** en 2006
- conçue pour simplifier l'écriture de codes **JavaScript** et **AJAX**
- librairie front-end la plus utilisée au monde
- basée sur le concept de sélecteur **CSS 3**

jQuery

Deux solutions possibles

- Télécharger la bibliothèque **jQuery** et l'intégrer dans le fichier **HTML** comme n'importe quel fichier **JS** grâce à la balise

```
<script src = ...> </script>
```

- Faire référence à une version en ligne (CDN) du fichier **jQuery** dans la balise

```
<script src = ...> </script>
```

jQuery

Deux solutions possibles

- Télécharger la bibliothèque **jQuery** et l'intégrer dans le fichier **HTML** comme n'importe quel fichier **JS** grâce à la balise

```
<script src = ...> </script>
```

- Faire référence à une version en ligne (CDN) du fichier **jQuery** dans la balise

```
<script src = ...> </script>
```

CDN : Content Delivery Network ?

- un réseau de serveurs installés dans plusieurs villes et/ou pays sur lesquels sont mis à disposition des fichiers tels que jQuery/Bootstrap...
- en appelant un fichier **jQuery** à partir d'un fichier **HTML** sur un **CDN**, un téléchargement rapide sera effectué puisqu'on aura accès au plus proche d'où on se trouve.

Deux versions disponibles en téléchargement

- compressed version (**version de production**) : le code est compressé, le fichier est beaucoup moins volumineux (diminue la consommation de bande passante du site) mais illisible
- uncompressed version (**version de développement**) : code source lisible, pratique en cas de debug pour savoir sur quelle fonction le script a planté

Première solution

- Aller sur <https://jquery.com/download/> et télécharger une version stable compressée de jQuery
- Placer le fichier téléchargé dans le répertoire dans le répertoire de votre projet
- Faire référence à ce fichier dans la balise `<script src = ...>`
`</script>`

jQuery

Première solution

```
<!DOCTYPE html>
<html>
    <head>
        <title> jQuery test </title>
        <link rel="stylesheet" href="style.css">
    </head>
    <body>
        ...
        <script src="chemin/vers/mon/fichier/jquery-3.1.0.min
                  .js"></script>
        <script src = script.js>
            // script.js fichier contenant les scripts qui
            // utiliseront la librairie jQuery
        </script>
    </body>
</html>
```

jQuery

Pour la deuxième solution : plusieurs hébergeurs

- **jQuery** : <https://code.jquery.com/jquery-3.6.0.min.js>
- **Google** :
<https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js>
- **Microsoft** :
<https://ajax.aspnetcdn.com/ajax/jquery/jquery-3.6.0.min.js>

jQuery

Pour la deuxième solution : plusieurs hébergeurs

- **jQuery** : <https://code.jquery.com/jquery-3.6.0.min.js>
- **Google** :
<https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js>
- **Microsoft** :
<https://ajax.aspnetcdn.com/ajax/jquery/jquery-3.6.0.min.js>

Remarque

- Une version compressée de **jQuery** existe chez tous les hébergeurs (il suffit d'ajouter **.min** avant **.js**)
- Dans ce cours, on utilisera la version complète hébergée par **jQuery**

jQuery

Comment l'utiliser ?

- Copier un des liens précédents
- Coller le comme valeur de l'attribut `src` de la balise
`<script src = ...> </script>`
- Placer les balises `<script>` avant la fermeture de la balise
`</body>` de votre fichier **HTML** en commençant par celle qui importe la librairie **jQuery**

jQuery

Deuxième solution

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title> jQuery test </title>
        <link rel="stylesheet" href="style.css">
    </head>
    <body>
        ...
        <script src="https://code.jquery.com/jquery-3.6.0.min.js">
        </script>
        <script src=script.js>
            // script.js fichier contenant les scripts qui
            // utiliseront la librairie jQuery
        </script>
    </body>
</html>
```

jQuery

Le body du index.html

```
<button id=but>  
    cliquer  
</button>
```

Le fichier script.js

```
$("#but").click(  
    function(){  
        console.log("Hello World! ");  
    }  
);
```

jQuery

Le body du index.html

```
<button id=but>  
    cliquer  
</button>
```

Le fichier script.js

```
$("#but").click(  
    function(){  
        console.log("Hello World! ");  
    }  
);
```

Explication

- `$()` : la fonction magique qui prend en paramètre un sélecteur
- `$` : opérateur utilisé par certains langages et frameworks (PHP, Angular JS, EL...). En effet, `$` est l'abréviation de la fonction principale `jQuery()` (à utiliser en cas de conflit)

jQuery

Le body du index.html

```
<button id=but>  
    cliquer  
</button>
```

Le fichier script.js

```
$("#but").click(  
    function(){  
        console.log("Hello World! ");  
    }  
);
```

Explication

- `$()` : la fonction magique qui prend en paramètre un sélecteur
- `$` : opérateur utilisé par certains langages et frameworks (PHP, Angular JS, EL...). En effet, `$` est l'abréviation de la fonction principale `jQuery()` (à utiliser en cas de conflit)

Si `ReferenceError: jQuery is not defined` ou `ReferenceError: $ is not defined`, vérifier le chargement de la librairie jQuery.

jQuery

Remarque

- Avant de manipuler des éléments dans votre page, il faut attendre que cette dernière soit correctement chargée (le **DOM** est bien analysé)
- Cette opération est réalisée par la méthode `$(document).ready(...)` de jQuery

jQuery

Remarque

- Avant de manipuler des éléments dans votre page, il faut attendre que cette dernière soit correctement chargée (le **DOM** est bien analysé)
- Cette opération est réalisée par la méthode `$(document).ready(...)` de jQuery

Le fichier `script.js`

```
$(document).ready(function() {
    $("#but").click(function() {
        console.log("Hello World!");
    });
});
```

jQuery

On peut aussi utiliser le raccourci

```
$ (function() {  
    $("#but").click(function(){  
        console.log("Hello World!");  
    });  
});
```

jQuery

On peut aussi préciser la zone de recherche

```
$ (function() {  
    $("#but", "body") .click(function() {  
        console.log("Hello World!");  
    });  
});
```

Le deuxième paramètre (optionnel) correspond à la zone dans laquelle la recherche est effectuée, par défaut la recherche s'effectue dans tout le document,

jQuery

Plaçons ce code dans le body d'index.html

```
<div id=container>
    Considérons les paragraphes suivants :
    <p class=blue> bonjour </p>
    <p class=red> bonsoir, voici <a href="http://elmouelhia.free.fr/"> ma page </a></p>
    <p class="blue"> salut </p>
</div>
```

Le fichier style.css

```
.red {
    color: red;
}

.blue {
    color: blue;
}

.green {
    color: green;
}
```

Pour récupérer le contenu d'une balise

- `.html() ≡ .innerHTML en JS`
- `.text() ≡ .textContent en JS`

jQuery

Pour tester

```
$ (function() {  
    $("#but", "body").click(function(){  
        console.log($("#container").html());  
        console.log($("#container").text());  
    });  
});
```

jQuery

Pour modifier le contenu d'une balise

```
$(function() {  
    $("#but", "body").click(function() {  
        $("#container").text("<p> hello </p>");  
    });  
});
```

Ou bien

```
$(function() {  
    $("#but", "body").click(function() {  
        $("#container").html("<p> hello </p>");  
    });  
});
```

Vérifier la différence

jQuery

Exercice

Écrire un script **jQuery** qui permet d'ajouter du contenu textuel au début et à la fin de la balise `<div class=container>` sans écraser le contenu précédent.

jQuery

La méthode `.attr()`

- `.attr(nomAttribut)` : permet de récupérer la valeur de l'attribut passé en paramètre
- `.attr(nomAttribut, nouvelleValeur)` : permet de modifier la valeur de l'attribut du premier paramètre par la valeur du deuxième paramètre (si l'attribut n'existe pas, il sera créé)



Et comment supprimer un attribut ?

`removeAttr(nomAttribut)`

jQuery

Un exemple avec .attr(nomAttribut)

```
$ (function() {  
    console.log ($("p").attr("class"));  
});
```

jQuery

Un exemple avec .attr(nomAttribut)

```
$ (function() {  
    console.log ($("p").attr("class"));  
});
```

Que la valeur de la classe du premier paragraphe est affichée

jQuery

Pour faire la même chose pour tous les paragraphes, on peut utiliser la méthode `each()`

```
$ (function() {  
    $("p").each(function () {  
        console.log($(this).attr("class"));  
    });  
});
```

© Achref EL HADJ

jQuery

Pour faire la même chose pour tous les paragraphes, on peut utiliser la méthode `each()`

```
$ (function() {  
    $("p").each(function () {  
        console.log($(this).attr("class"));  
    });  
});
```

On peut faire aussi

```
$ (function() {  
    $("p").each(function (ind, elt) {  
        console.log($(elt).attr("class"));  
    });  
});
```

jQuery

On peut aussi utiliser `.length` pour déterminer le nombre d'éléments sélectionnés

```
$(function() {
    console.log($(".p").length);
});
```

jQuery

On peut aussi utiliser `.length` pour déterminer le nombre d'éléments sélectionnés

```
$(function() {
    console.log($(".p").length);
});
```

Ou `.eq(i)` pour choisir l'élément de la sélection ayant l'indice `i`

```
$(function() {
    console.log($(".p").eq(i));
});
```

jQuery

On peut aussi utiliser `.length` pour déterminer le nombre d'éléments sélectionnés

```
$(function() {
    console.log($(".p").length);
});
```

Ou `.eq(i)` pour choisir l'élément de la sélection ayant l'indice `i`

```
$(function() {
    console.log($(".p").eq(i));
});
```

Ou bien :`:eq(i)`

```
$(function() {
    console.log($(".p:eq(i)"));
});
```

jQuery

Ou aussi

```
$(function() {  
    console.log($(".p") [i]);  
});
```

De même

- p:lt (i) : les paragraphes dont l'indice est inférieur à i
- p:gt (i) : les paragraphes dont l'indice est supérieur à i
- p:even et p:odd : les paragraphes d'indice pair et impair
- p:first et p:last : le premier et le dernier paragraphes
- input:text : les input de type text

jQuery

On peut aussi utiliser le raccourci `.first()` pour choisir le premier

```
$(function() {  
    console.log($(".p").first());  
});
```

jQuery

On peut aussi utiliser le raccourci `.first()` pour choisir le premier

```
$(function() {
    console.log($(".p").first());
});
```

Ou aussi `.last()` pour choisir le dernier

```
$(function() {
    console.log($(".p").last());
});
```

jQuery

Exemple avec `.attr(nomAttribut, nouvelleValeur)` **et**
`removeAttr(nomAttribut)`

```
$(function() {
    $("p").eq(0).removeAttr("class");
    $("p").eq(0).attr("class", "red");
});
```

jQuery

Exemple avec `.attr(nomAttribut, nouvelleValeur)` **et**
`removeAttr(nomAttribut)`

```
$(function() {
    $("p").eq(0).removeAttr("class");
    $("p").eq(0).attr("class", "red");
});
```

On peut modifier la valeur sans supprimer et recréer l'attribut

```
$(function() {
    $("p").eq(0).attr("class", "red");
});
```

jQuery

Ajoutons une zone de saisie dans la page HTML : nouveau contenu

```
<div id = container>
    Considerons les paragraphes suivants :
    <p class=blue> bonjour </p>
    <p class=red> bonsoir, voici
        <a href="http://www.lsis.org/elmouelhia/"> ma page </a>
    </p>
    <p class="blue"> salut </p>
</div>
<input type=text id=nom value="initial" >
<button id=but> cliquer </button>
<script ...>
```

La méthode .val()

- `.val()` : permet de récupérer la valeur de l'attribut `value`
- `.val(nouvelleValeur)` : permet de remplacer la valeur de l'attribut `value` par la valeur de `nouvelleValeur`

jQuery

Un exemple avec .val()

```
$(function() {  
    $("#but").click(function() {  
        console.log($("#nom").val());  
    });  
});
```

jQuery

Un exemple avec .val()

```
\$(function() {  
    \$("#but").click(function() {  
        console.log(\$("#nom").val());  
    });  
});
```

Un exemple avec .val(nouvelleValeur)

```
\$(function() {  
    \$("#but").click(function() {  
        \$("#nom").val("John Wick");  
    });  
});
```

jQuery

La méthode .prop()

- `.prop(nomPropriété)` : permet de récupérer la valeur **booléenne** d'une propriété (contrairement à `.attr(nomPropriété)` qui retourne soit le nom de la propriété, si elle est définie, ou `undefined`)
- `.prop(nomPropriété, nouvelleValeur)` : permet de modifier la valeur d'une propriété



Pour supprimer une propriété

`removeProp(nomPropriété)`

jQuery

Un exemple avec .prop()

```
$(function() {  
    $("#but").click(function(){  
        $("#nom").prop("readonly", "true");  
        console.log($(this).prop("disabled"));  
        // affiche false  
    });  
});
```

jQuery

Un exemple avec .prop()

```
$(function() {  
    $("#but").click(function(){  
        $("#nom").prop("readonly", "true");  
        console.log($(this).prop("disabled"));  
        // affiche false  
    });  
});
```

Pour comprendre la différence entre .attr et .prop

```
$(function() {  
    $("#but").click(function(){  
        $("#nom").prop("readonly", true);  
        console.log($(this).prop("disabled")); // false  
        console.log($(this).attr("disabled")); // undefined  
        console.log($("#nom").prop("readonly")); // true  
        console.log($("#nom").attr("readonly")); // readonly  
    });  
});
```

La méthode .css()

- `.css(nomPropriété)` : permet de récupérer la valeur de la propriété CSS passée en paramètre
- `.css(nomPropriété, nouvelleValeur)` : permet de modifier la valeur de la propriété CSS (premier paramètre) par la valeur du deuxième paramètre (si la propriété n'existe pas, elle sera créée)

Un exemple avec .css (nomPropriété)

```
$(function() {  
    console.log($(".p").eq(0).css("color"));  
});
```

© Achref EL MOUELHI ©

Un exemple avec .css (nomPropriété)

```
$(function() {  
    console.log($("#p").eq(0).css("color"));  
});
```

Un exemple avec .css (nomPropriété, nouvelleValeur)

```
$(function() {  
    $("#but").click(function(){  
        $("p").eq(0).css("color", "green");  
    });  
});
```

Un exemple avec .css (nomPropriété)

```
$(function() {  
    console.log($("#p").eq(0).css("color"));  
});
```

Un exemple avec .css (nomPropriété, nouvelleValeur)

```
$(function() {  
    $("#but").click(function(){  
        $("p").eq(0).css("color", "green");  
    });  
});
```

Pour un changement multiple de propriétés CSS

```
$(function() {  
    $("#but").click(function(){  
        $("p").eq(0).css({"color": "green", "background-color": "red"});  
    });  
});
```

Les méthodes .xClass()

- .addClass() : pour ajouter une classe
- .removeClass() : pour supprimer une classe
- .hasClass() : pour tester l'existence d'une classe
- .toggleClass() : pour retirer ou ajouter une classe

jQuery

Exercice

En utilisant les méthodes `xClass`, écrire un code jQuery qui permet de permuter les couleurs des paragraphes suivants : (rouge \Rightarrow bleu), (bleu \Rightarrow vert) et (vert \Rightarrow rouge).

Le HTML

```
<div id=paragraphs>
  <p class=red>rouge</p>
  <p class=green>vert</p>
  <p class=blue>bleu</p>
</div>
```

Le CSS

```
.red {
  color: red;
}
.green {
  color: green;
}
.blue {
  color: blue;
}
```

jQuery

La méthode `.filter(' sélecteur')`

Retourne un ensemble d'élément filtré en fonction du sélecteur

jQuery

La méthode `.filter(' sélecteur')`

Retourne un ensemble d'élément filtré en fonction du sélecteur

Exemple

```
$ (function() {  
    $("#container > p").filter(":not(:eq(0))")  
        .css("background-color", "skyblue");  
});
```

Filtre les paragraphes ayant un indice différent de 0 parmi ceux qui sont dans container.

La méthode `.has(' sélecteur')`

Retourne un ensemble d'élément filtré en fonction du sélecteur, ne prend que les enfants associés à l'élément.

La méthode .has (' sélecteur ')

Retourne un ensemble d'élément filtré en fonction du sélecteur, ne prend que les enfants associés à l'élément.

Exemple

```
$ (function() {  
    $("#container > p") .has("a")  
        .css("background-color", "skyblue");  
});
```

La méthode `.has(' sélecteur')`

Retourne un ensemble d'élément filtré en fonction du sélecteur, ne prend que les enfants associés à l'élément.

Exemple

```
$ (function() {  
    $("#container > p") .has("a")  
        .css ("background-color", "skyblue");  
});
```

Filtre les paragraphes, définis dans `container`, ayant comme enfant un lien hypertexte.

La méthode `.has(' sélecteur')`

Retourne un ensemble d'élément filtré en fonction du sélecteur, ne prend que les enfants associés à l'élément.

Exemple

```
$(function() {  
    $("#container > p").has("a")  
        .css("background-color", "skyblue");  
});
```

Filtre les paragraphes, définis dans `container`, ayant comme enfant un lien hypertexte.

On peut aussi utiliser `has` comme pseudo-sélecteur jQuery

```
$(function() {  
    $("#container > p").filter(":has(a)")  
        .css("background-color", "skyblue");  
});
```

jQuery

La méthode `.not (' sélecteur')`

Retourne un ensemble d'élément filtré en fonction du sélecteur, retire tous les éléments qui respecte le sélecteur

jQuery

La méthode `.not (' sélecteur')`

Retourne un ensemble d'élément filtré en fonction du sélecteur, retire tous les éléments qui respecte le sélecteur

Exemple

```
$ (function() {  
    $("#container > p").not(":even")  
        .css("background-color", "skyblue");  
});
```

Filtre les paragraphes d'indice pair parmi ceux qui sont dans container.

Exercice

Écrire un code qui permet de mettre le bleu comme couleur de fond pour tous les paragraphes qui ne contiennent pas de lien hypertexte

jQuery

Première solution

```
$ (function() {  
    $("#container > p").filter(":not(:has(a))")  
        .css("background-color", "  
            skyblue");  
});
```

© Achref EL MOUADJI

jQuery

Première solution

```
\$(function() {  
    \$("#container > p").filter(":not(:has(a))")  
        .css("background-color", "  
            skyblue");  
});
```

Deuxième solution

```
\$(function() {  
    \$("#container > p").not(":has(a)")  
        .css("background-color", "  
            skyblue");  
});
```

jQuery

La méthode `.find(' sélecteur')`

- permet de chercher et de filtrer des éléments selon le sélecteur
- possible d'effectuer plusieurs recherches en utilisant la méthode `end()` pour remonter dans le chaînage

jQuery

La méthode `.find(' sélecteur')`

- permet de chercher et de filtrer des éléments selon le sélecteur
- possible d'effectuer plusieurs recherches en utilisant la méthode `end()` pour remonter dans le chaînage

Exemple

```
$(function() {  
    $("#container").find("p:even").css("background-color", "blue")  
        .end()  
        .find("p:odd").css("background-color", "pink");  
});
```

Les paragraphes d'indice pair auront une couleur de fond bleue et ceux d'indice impair auront une couleur de fond rose.

.find() vs .filter

- `.find()` cherche dans les enfants de l'élément sélectionné
- `.filter()` cherche dans les éléments sélectionnés

jQuery

La méthode `.children()`

permet de sélectionner les enfants directs de la sélection

© Achref EL MOUELHID

jQuery

La méthode `.children()`

permet de sélectionner les enfants directs de la sélection

Exemple

```
$(function() {
    $("#container").children().css("background-color",
        "skyblue");
});
```

Tous les enfants du `container` auront une couleur de fond bleue.

jQuery

Exemple avec `.children(selecteur)`

```
$ (function() {  
    $("#container").children("p").css("background-  
        color", "skyblue");  
});
```

Tous les enfants de type `p` du `container` auront une couleur de fond bleue.

jQuery

Exercice, considérons le code HTML suivant

```
<h1>Titre</h1>
<div class="about">
  <strong>Adipiscing elit.</strong>
  <div>
    Lorem ipsum dolor sit amet,
    <strong>consectetur</strong>
  </div>
</div>
<p>
  Nulla pellentesque
  <em>molestie</em> tempus.
  <strong>Curabitur</strong>
  turpis est.
</p>
```

En utilisant les méthodes précédentes, sélectionner

- tous les strongs fils de la première div
- le strong directement fils de la première div
- le strong de plus bas niveau dans la première div

jQuery

La méthode `.parent()`

permet de remonter au premier parent de la sélection

jQuery

La méthode `.parent()`

permet de remonter au premier parent de la sélection

Exemple

```
$(function() {
    $("a").parent().css("background-color", "skyblue");
});
```

Le paragraphe contenant un lien hypertexte aura une couleur de fond bleue.

jQuery

La méthode `.parents()`

permet de remonter dans les parents jusqu'à trouver le père recherché

© Achref EL MOUELLI

jQuery

La méthode `.parents()`

permet de remonter dans les parents jusqu'à trouver le père recherché

Exemple

```
$(function() {
    $("a").parents("div").css("background-color", "skyblue");
});
```

Le `div` container aura une couleur de fond bleue.

Autre méthodes

- `.next()` : le frère suivant immédiat
- `.nextAll()` : tous les éléments suivants
- `.prev()` : le frère précédent immédiat
- `.prevAll()` : tous les éléments précédents

jQuery

La méthode `.siblings()`

permet de récupérer tous les frères (précédents et suivants) de l'élément

© Achref EL MOUELHID

jQuery

La méthode `.siblings()`

permet de récupérer tous les frères (précédents et suivants) de l'élément

Exemple

```
$(function() {
    $("p.red").siblings().css("background-color", "skyblue");
});
```

Les frères (précédents et suivants) du paragraphe ayant la classe rouge auront une couleur de fond bleue.

jQuery

On peut ajouter un paramètre à `siblings` pour préciser quel type de balise on cherche

```
$ (function() {  
    $("#container").siblings("input").css("background-  
        color", "skyblue");  
});
```

Les frères (précédents et suivants) du `container` de type `input` auront une couleur de fond bleue.

jQuery

Exercice, considérons le code HTML suivant

```
<div id="menu">
  <ul>
    <li><a href="#">Lien 1</a></li>
    <li><a href="#">Lien 2</a></li>
  </ul>
</div>
<div class="contact">
  <strong>Adipiscing elit.</strong>
  <div>
    Lorem ipsum dolor sit amet, <strong>consectetur</strong>
  </div>
  <form method="POST">
    <input type="text" name="email" />
    <input type="submit" value="s'inscrire" />
  </form>
</div>
```

Que ferait le code suivant

```
$('#menu').find('li:eq(1)')  
    .siblings('li')  
    .css('border', '1px solid blue')  
    .end().end()  
    .siblings('.contact')  
    .find(':text[name=email]')  
    .parent('form')  
    .css('border', '1px solid red');
```

La méthode .wrap(baliseHTML)

permet d'entourer les éléments sélectionnés par les balises passées en paramètre

jQuery

Le body de la page index.html

```
<p>Bonjour</p>
<p>Bonsoir</p>
<button id=but>
    cliquer
</button>
```

Le CSS associé

```
div {
    color: red;
}
```

jQuery

Le body de la page index.html

```
<p>Bonjour</p>
<p>Bonsoir</p>
<button id=but>
    cliquer
</button>
```

Le CSS associé

```
div {
    color: red;
}
```

Le code qui sera exécuté en cliquant sur le bouton

```
$("#but").click(
    function() {
        $("p").wrap("<div></div>");
    }
);
```

jQuery

Le body de la page index.html

```
<p>Bonjour</p>
<p>Bonsoir</p>
<button id=but>
    cliquer
</button>
```

Le CSS associé

```
div {
    color: red;
}
```

Le code qui sera exécuté en cliquant sur le bouton

```
$("#but").click(
    function() {
        $("p").wrap("<div></div>");
    }
);
```

En cliquant sur le bouton, les paragraphes seront affichés en rouge.

jQuery

La méthode `.data(key, value)`

permet d'attacher des données aux balises sélectionnées

© Achref EL MOUADJI

jQuery

La méthode `.data(key, value)`

permet d'attacher des données aux balises sélectionnées

La méthode `.removeData(key)`

permet de supprimer des données dont la clé est passée en paramètre associées aux balises sélectionnées

jQuery

Le body de la page index.html

```
<div>
  Bonjour
  <span></span>
  <span></span>
</div>
```

jQuery

Le body de la page index.html

```
<div>
  Bonjour
  <span></span>
  <span></span>
</div>
```

Pour associer des données à une balise div

```
$(document).ready(function() {
  $("div").data("obj", {nom: "wick", prenom: "john"});
});
```

jQuery

Le body de la page index.html

```
<div>
  Bonjour
  <span></span>
  <span></span>
</div>
```

Pour associer des données à une balise div

```
$(document).ready(function() {
  $("div").data("obj", {nom: "wick", prenom: "john"});
});
```

Pour récupérer et afficher les données associées à la balise div

```
$(document).ready(function() {
  $("div").data("obj", {nom: "wick", prenom: "john"});
  $("span:first").text( $("div").data("obj").prenom);
  $("span:last").text( $("div").data("obj").nom);
});
```

Plusieurs méthodes possibles

- `.append(contenu)` : permet d'insérer le contenu en paramètre à la fin des éléments sélectionnés
- `$("contenu").appendTo(sélecteur)` : permet d'insérer le contenu en paramètre à la fin des éléments ciblés via le sélecteur
- `.prepend(contenu)` : permet d'insérer le contenu en paramètre au début des éléments sélectionnés
- `$("contenu").prependTo(sélecteur)` : permet d'insérer le contenu en paramètre au début des éléments ciblés via le sélecteur

jQuery

Exemple

```
$(function() {  
    $("#container > p.red").append("<p> hello </p>");  
});
```

© Achref EL MOUELLI

jQuery

Exemple

```
$(function() {
    $("#container > p.red").append("<p> hello </p>");
});
```

Autre utilisation de .append(\$(selecteur)) : copier le deuxième élément sélectionné à la fin du premier élément sélectionné

```
$(function() {
    $("#container").append($(".input:text"));
});
```

jQuery

Autres méthodes possibles

- `$(sélecteur).after(contenu)` : permet d'insérer le contenu en paramètre après les éléments sélectionnés
- `$(contenu).insertAfter(sélecteur)` : permet d'insérer le contenu en paramètre après les éléments sélectionnés (le contenu et le sélecteur sont inversés)
- `$(sélecteur).before(contenu)` : permet d'insérer le contenu en paramètre avant les éléments sélectionnés
- `$(contenu).insertBefore(sélecteur)` : permet d'insérer le contenu en paramètre avant les éléments sélectionnés (le contenu et le sélecteur sont inversés)

jQuery

Exemple

```
$ (function() {  
    $("#container").after("<p> hello </p>");  
});
```

© Achref EL MOUADJI

jQuery

Exemple

```
$(function() {  
    $("#container").after("<p> hello </p>");  
});
```

Pour `.insertAfter()` : il faut juste inverser le contenu et le sélecteur

```
$(function() {  
    "<p> hello </p>").insertAfter("#container");  
});
```

jQuery

Pour supprimer, on peut utiliser soit `.remove()` soit `.detach()`

```
$(function() {  
    $("p.red").remove(); // ou $("p.red").detach();  
});
```

jQuery

Pour comprendre la différence entre `.remove()` et `.detach()`

```
$(function() {  
    $("p.red").click(function(){  
        $(this).css("color", "yellow");  
    });  
    $("#but").click(function(){  
        var parRouge =$("p.red").detach();  
        $("#container").html(parRouge);  
    });  
});
```

Explication

- cliquer sur le bouton, ensuite cliquer sur le texte en rouge (il ne sera jamais affiché en jaune)
- remplacer `remove` par `detach` et refaire la même chose (le texte est affiché en jaune)
- Avec `detach`, on ne perd pas les données de nos éléments HTML (les événements...)

jQuery

Syntaxe

```
$ (function() {  
    $("selector").event(function() {  
        // code de la fonction  
    });  
});
```

jQuery

Et si on veut attacher plusieurs évènements à un sélecteur

```
$ (function() {  
    $("selector").on("event1 event2",    function() {  
        // code de la fonction  
    });  
});
```

© Achref EL MOUADJI

jQuery

Et si on veut attacher plusieurs évènements à un sélecteur

```
$ (function() {  
    $("selector").on("event1 event2",    function() {  
        // code de la fonction  
    });  
});
```

Et si les fonctions liées aux différents évènements sont différentes

```
$ (function() {  
    $("selector").on({  
        event1: function(){...},  
        event2: function(){...}  
    });  
});
```

jQuery

Et si on veut attacher plusieurs évènements à un sélecteur

```
$ (function() {  
    $("selector").on("event1 event2",    function() {  
        // code de la fonction  
    });  
});
```

Et si les fonctions liées aux différents évènements sont différentes

```
$ (function() {  
    $("selector").on({  
        event1: function(){...},  
        event2: function(){...}  
    });  
});
```

La méthode `.on()` remplace `.bind()` depuis la version 3.

jQuery

Exemple

```
$ (function() {  
    $("p.red").on({  
        mouseover: function(){  
            $("body").css("background-color", "pink");  
        },  
        mouseout: function(){  
            $("body").css("background-color", "green");  
        },  
        click: function(){  
            $("body").css("background-color", "yellow");  
        }  
    });  
});
```

jQuery

Pour supprimer un évènement

```
$(function() {  
    $("selector").off("event");  
});
```

jQuery

Pour supprimer un évènement

```
$(function() {  
    $("selector").off("event");  
});
```

Exemple

```
$(function() {  
    $("p.red").click(function() {  
        $(this).css("color", "yellow");  
    });  
    $("#but").click(function() {  
        $("p.red").off("click");  
    });  
});
```

jQuery

Et si on veut que la fonction liée à l'évènement soit exécutée une seule fois

```
$ (function() {  
    $("selector").one("event1 event2",  function() {  
        // code de la fonction  
    });  
});
```

jQuery

Et si on veut déclencher un évènement manuellement ou suite à un autre évènement

```
$ (function() {  
    $("p.red").click(function() {  
        $(this).css("color", "pink");  
    });  
    $("button").click(function() {  
        $("p.red").trigger("click");  
    });  
});
```

jQuery

Certaines méthodes comme `.hover()` acceptent deux fonctions : une sera exécutée lorsque l'élément est survolé, la deuxième quand il ne l'est plus

```
$ (function() {  
    $("p.red").hover(function() {  
        $(this).css("color", "pink");  
    },  
        function() {  
            $(this).css("color", "green");  
        } );  
});
```

Sans la deuxième fonction, l'élément reste pink lorsqu'il n'est plus survolé.

Exercice

- Créer une page **HTML** contenant deux images de grande taille
- La deuxième image n'est pas initialement affichée (vous pouvez utiliser la propriété CSS `display: none`)
- Pour atteindre le bas de la première image, il faut utiliser un **scroll**
- Lorsque le **scroll** arrive au bas de la première image, la seconde apparaît (la valeur de sa propriété **CSS** `display` reprend la valeur initiale `inline`)
- vous pouvez utiliser les propriétés suivantes :
 - `$(window).scrollTop()` : retourne le nombre de pixels à remonter pour atteindre le haut de la page
 - `$(window).height()` : retourne la hauteur de la partie de page affichée dans le navigateur
 - `$(document).height()` : retourne la hauteur totale de la page en pixels
- Lorsque le **scroll** arrive au bas du document, on a :
`$(document).height() = $(window).height() + $(window).scrollTop()`

Quelques opérations sur les tableaux

- `$.grep` : pour filtrer les éléments d'un tableau.
- `$.map` : pour appliquer une opération sur les éléments d'un tableau.
- `$.inArray` : retourne la position d'un élément dans un tableau, -1 sinon.
- `$.merge` : pour fusionner deux tableaux.

Considérons le body de la page index.html

```
<body>
  <div id=resultat></div>
  <button id=but>cliquer</button>
  <script src="http://code.jquery.com/jquery.js"></script>
  <script src="script.js"></script>
</body>
```

Considérons le body de la page index.html

```
<body>
  <div id=resultat></div>
  <button id=but>cliquer</button>
  <script src="http://code.jquery.com/jquery.js"></script>
  <script src="script.js"></script>
</body>
```

Exemple d'utilisation de .grep() : filtrer les chaînes de longueur <= 5

```
$(document).ready(function() {
    var tableau = ['Marseille', 'Barcelone', 'Madrid', 'Turin', 'Londres',
        'Milan', 'Manchester', 'Lyon'];
    $('#but').click(function() {
        tableau2 = $.grep(tableau, function(elt,ind) {
            return (elt.length > 5);
        });
        $('#resultat').text(tableau2.join(', '));
        // Marseille, Barcelone, Madrid, Londres, Manchester
    });
});
```

jQuery

Exemple d'utilisation de `.map()` : récupérer les 5 premiers caractères de chaque élément du tableau

```
$ (document) .ready(function() {
    var tableau = ['Marseille', 'Barcelone', 'Madrid',
        'Turin', 'Londres', 'Milan', 'Manchester', 'Lyon'];
    $('#but').click(function() {
        tableau2 = $.map(tableau, function(elt,ind) {
            return (elt.substring(0,5));
        });
        $('#resultat').text(tableau2.join(', '));
        // Marse, Barce, Madri, Turin, Londr, Milan,
        // Manch, Lyon
    });
});
```

On ajoute une zone de saisie dans le body de la page index.html

```
<body>
  <input type="text" id="ville" placeholder="ville à chercher">
  <div id="resultat"></div>
  <button id="but">cliquer</button>
  <script src="http://code.jquery.com/jquery.js"></script>
  <script src="script.js"></script>
</body>
```

On ajoute une zone de saisie dans le body de la page index.html

```
<body>
  <input type="text" id="ville" placeholder="ville à chercher">
  <div id="resultat"></div>
  <button id="but">cliquer</button>
  <script src="http://code.jquery.com/jquery.js"></script>
  <script src="script.js"></script>
</body>
```

Exemple d'utilisation de .inArray()

```
$(document).ready(function() {
  var tableau = ['Marseille', 'Barcelone', 'Madrid', 'Turin', 'Londres',
    'Milan', 'Manchester', 'Lyon'];
  $('#but').click(function() {
    var ville = $("#ville").val();
    position = $.inArray(ville, tableau);
    position != -1 ? $('#resultat').text("oui") : $('#resultat').text("non");
  });
});
```

jQuery

Exemple d'utilisation de `.merge()`

```
$ (document) .ready(function() {  
    var france = ['Marseille', 'Lyon', 'Paris'];  
    var autre = ['Barcelone', 'Turin', 'Manchester'];  
    $('#but').click(function() {  
        var ville = $("#ville").val();  
        resultat = $.merge(france, autre);  
        $('#resultat').text( resultat.join(', '));  
        // Marseille, Lyon, Paris, Barcelone, Turin,  
        // Manchester  
    });  
});
```

jQuery

jQuery : alternatives modernes

- **jQuery** a joué un rôle clé dans l'évolution du web.
- Aujourd'hui, il est devenu obsolète pour la majorité des cas.

© Achref EL MOUADJI

jQuery

jQuery : alternatives modernes

- **jQuery** a joué un rôle clé dans l'évolution du web.
- Aujourd'hui, il est devenu obsolète pour la majorité des cas.

Alternatives modernes :

- **React**
- **Vue.js**
- **Angular**

jQuery

Résumé

jQuery était utile pour...

Gérer le DOM facilement
Uniformiser les navigateurs
Faire de l'AJAX
Créer des UI dynamiques

Mais aujourd'hui...

JavaScript moderne le fait nativement
Les différences sont bien réduites
`fetch()` est suffisant
Les frameworks sont plus puissants