



@Moussa Camara

Introduction

La devise de jQuery: «write less, do more»

jQuery est une bibliothèque JavaScript qui permet de simplifier énormément le codage du JavaScript dans les pages Web, en particulier les appels au **DOM**(Document Object Model)et l'utilisation d'AJAX.

De nombreuses tâches complexes en JavaScript vont se réduire à une seule ligne: l'appel d'une méthode jQuery, qui aura le même comportement sur l'ensemble des navigateurs Web.

jQuery fournit une «couche d'abstraction» au développement Web, qui évite au développeur de se préoccuper des différences entre les navigateurs, et simplifie ses tests.

Elle permet de:

- manipuler les éléments et attributs HTML, d'une façon plus simple et intuitive qu'avec le DOM
- manipuler les styles CSS
- gérer les événements HTML en leur associant des méthodes événementielles
- Prendre en charge AJAX
- effectuer des effets graphiques et des animations (simples).

La bibliothèque jQuery fournit aussi des utilitaires et elle est évolutive (nombreux « plugins »)



Avantages

JQUERY EST UN BIBLIOTHÈQUE OPEN-SOURCE JAVASCRIPT

DÉFINITION DE L'OBJET '\$'

- ✓ IL SUFFIT DE CONNAITRE UN SEUL OBJET RÉFÉRENCÉ PAR LA VARIABLE "\$"
- ✓ OBJET QUI EST UNE FONCTION JAVASCRIPT POINT D'ENTRÉE DU FRAMEWORK
- ✓ SON RETOUR EST L'OBJET LUI-MÊME, CE QUI PERMET L'ENCHAINEMENT D'ACTIONS

POSSIBILITÉS DE JQUERY

- ✓ CRÉER DES ANIMATIONS
- ✓ MODIFIER ATTRIBUTS CSS
- ✓ GÉRER DES ÉVÉNEMENTS JAVASCRIPT

UTILISÉ AUSSI AVEC DES PLUGINS

Installation

POUR DÉMARRER

- TÉLÉCHARGER LA DERNIÈRE VERSION SUR [HTTP://JQUERY.COM](http://jquery.com)
- SAUVER LE FICHIER JAVASCRIPT SOUS JQUERY-X-X-X.JS
- INSÉREZ LE DANS LA PAGE HTML COMME SUIT :
- `<SCRIPT SRC="JQUERY-X.X.X.MIN.JS" TYPE="TEXT/JAVASCRIPT"></SCRIPT>`

jQuery est fournie sous deux versions : une version compressée qui sera plus rapide en production, et une version décompressée, conseillée pour la mise au point pendant la phase de développement.

`$.fn.jquery` affiche la version de jQuery

JQUERY - NOCONFLICT

JQUERY.NOCONFLICT

- ✓ PERMET D'ÉVITER LES CONFLITS AVEC L'ALIAS '\$' AVEC UNE AUTRE LIBRAIRIE.
- ✓ CETTE MÉTHODE EST UTILE POUR LA CRÉATION DE PLUGINS.
- ✓ noConflict () méthode va libérer le contrôle sera \$ identifier, de sorte que d'autres scripts peuvent l'utiliser.
- ✓ Enfaite jQuery.noConflict sert si tu utilises plusieurs framework JavaScript qui utilise le \$ comme jquery.
- ✓ <https://api.jquery.com/jquery.noConflict/>

```
jQuery.noConflict();  
(function($) {  
    $(function() {  
        // code utilisant $ comme l'alias de jQuery  
    });  
})(jQuery);  
// code utilisant $ comme l'alias d'une autre librairie
```

Utilisation: Syntaxe

jQuery a une syntaxe intuitive :

\$(selector).action ()

- \$ désigne la bibliothèque jQuery
- selector : **critère de recherche pour** « sélectionner » les éléments HTML sur lesquels va porter la méthode **action**
- action : ce qu'il faut faire sur le/les éléments sélectionnés

Utilisation: Syntaxe

Un sélecteur peut être :

- l'élément HTML courant désigné par le mot clé *this* :

- `$(this).action()` ;

- un élément HTML unique désigné par son id : comme dans un fichier CSS, l'id sera précédé de #

Par exemple, pour l'élément *idtest* :

- `$("#idtest").action()` ;

- tous les éléments HTML d'un certain type : par exemple, toutes les *div*

- `$("div").action()` ;

- tous les éléments HTML qui possèdent une certaine classe CSS : comme dans un fichier CSS, la classe sera précédée de .

Par exemple, pour la classe *titre*

- `$(".titre").action()` ;

Utilisation: Syntaxe

- jQuery fournit un moyen pratique pour résoudre ces problèmes : l'événement *ready* se déclenche lorsque le document courant (ou un autre élément au choix) est complètement chargé. Comme les autres événements, il est géré par la méthode du même nom :

```
$(document).ready ( function()  
{  
    // traitements à effectuer après le chargement complet du document :  
    // initialisation d'éléments HTML, création dynamique d'éléments par le DOM, association  
    // des événements et des méthodes événementielles  
});
```




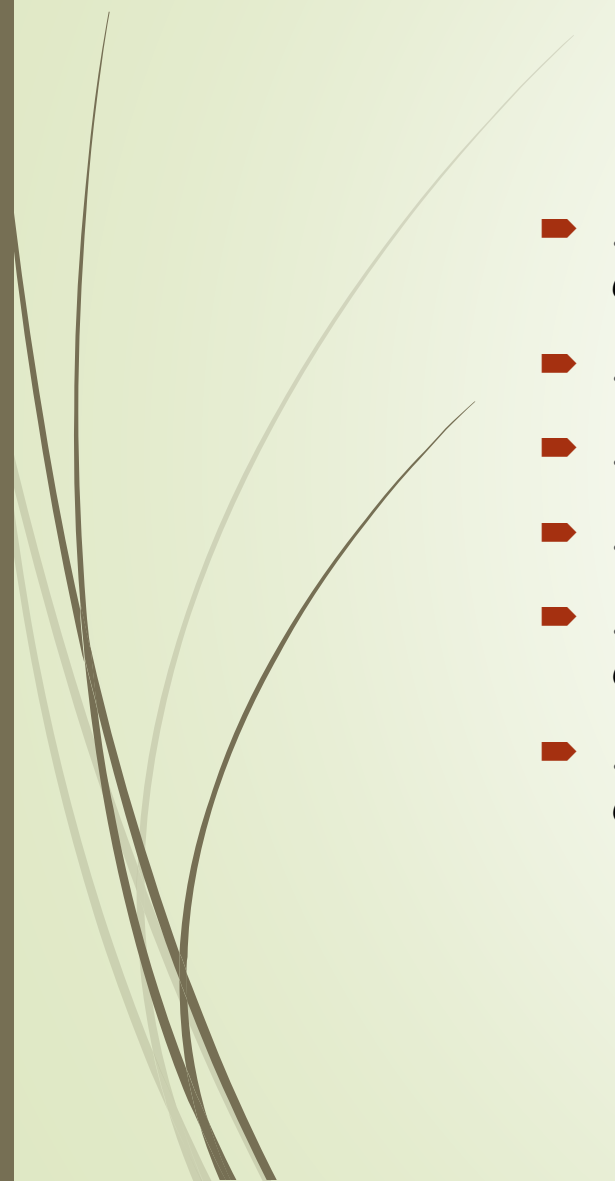

Exemple



```
$(document).ready ( function()
{
    $("p").click ( function()
    {      // traitement à effectuer
        alert ("cliqué") ;
        $(this).hide() ;
    } ) ;
});
```

Utilisation: Syntaxe

➤ Une action:

- [hide / show](#) : cache ou affiche un/plusieurs éléments HTML
- [text](#) : retourne le texte des éléments sélectionnés (sans les balises HTML)
- [html](#) : retourne le contenu des éléments sélectionnés (y compris les balises HTML)
- [val](#) : retourne le contenu des champs texte, dans un formulaire
- [append](#) : ajoute un élément fils à un container (comme *div*, *span*, *p*, *h1*, *table* etc.)

- 
- 
- [.html](#)('[contenu]') : remplacement du contenu d'un élément (les balises sont considérées comme des balises)
 - [.text](#)('[contenu]') : remplacement du contenu d'un élément
 - [.after](#)('[contenu]') : insertion du contenu après l'élément sélectionné
 - [.before](#)('[contenu]') : insertion du contenu avant l'élément sélectionné
 - [.append](#)('[contenu]') : insertion du contenu dans l'élément sélectionné à la suite des éléments existants
 - [.prepend](#)('[contenu]') : insertion du contenu dans l'élément sélectionné avant les éléments existants

- 
- 
- `.addClass('laClasse')` Ajoute la classe CSS `laClasse` à l'élément.
 - `.removeClass('laClasse')` Retire la classe CSS `laClasse` à l'élément.
 - `.toggleClass('laClasse')` Ajoute la classe CSS `laClasse` à l'élément s'il ne l'a pas, la lui retire sinon.
 - `.hasClass('laClasse')` Renvoie `true` si l'élément a la classe CSS `laClasse`, `false` sinon.

Exemple

```
<button class="cache">Cache moi</button>  
<button class="affiche">Affiche moi</button>
```

```
<p>
```

Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.

```
</p>
```

```
$("button.cache").click(function(){  
    $("p").hide();  
});
```

```
$("button.affiche").click(function(){  
    $("p").show();  
});
```

Comparaison js /jQuery

Javascript

```
var links = document.querySelectorAll('.links a')

for (var i = 0; i <= links.length; i++) {
  var link = links[i];
  link.style.backgroundColor="red";
}
```

jQuery

- `jQuery('.links a').css('backgroundColor', 'red')`
- Autre façon
- `$('.links a').css('backgroundColor', 'red')`

JQUERY – LES FONCTIONS UTILES

✓ `$("P");`

CET APPEL RETOURNE TOUS LES ÉLÉMENTS P (PARAGRAPHES) DE VOTRE DOCUMENT .

✓ `$(".MACCLASS");`

CET APPEL RETOURNE TOUS LES ÉLÉMENTS DE VOTRE DOCUMENT QUI ONT LA CLASSE CSS "MACCLASS".

✓ `$("#MONID");`

CET APPEL RETOURNE L'ÉLÉMENT DE VOTRE DOCUMENT, S'IL EXISTE, DONT L'ATTRIBUT ID EST « MONID".

✓ `$("#IDENTIFIANT > SPAN");`

CET APPEL RETOURNE TOUS LES ÉLÉMENTS SPAN CONTENUS DANS L'ÉLÉMENT DONT L'IDENTIFIANT EST « IDENTIFIANT".
ON POURRAIT ÉCRIRE CET APPEL DE CETTE FAÇON ÉGALEMENT : `$("SPAN", $("#IDENTIFIANT"));`

✓ `$("TABLE#IDENTIF > TR:EVEN");`

CET APPEL RETOURNE TOUS LES ÉLÉMENTS TR PAIRS (POUR LES LIGNES IMPAIRES REMPLACER "EVEN" PAR "ODD")
CONTENUS DANS L'ÉLÉMENT TABLE DONT L'IDENTIFIANT EST « IDENTIF"

CET APPEL PEUT ÊTRE EXTRÊMEMENT UTILE POUR APPLIQUER UNE COULEUR DE FOND A TOUTES LES LIGNES PAIRES PAR EXEMPLE.

JQUERY

```
<div>
<button class="btn btn-primary">Cachée
?</button>
</div>
<figure class="figure">

</figure>
```

```
$(document).ready(function () {
    $('button').click(function () {
        console.log('ok')
        $('img').toggle(5000);
        $('button').text(
            ($('button').text() == 'Afficher' ? 'Cachée'
            : 'Afficher') )
    })
})
```

JQUERY – LES FONCTIONS UTILES


EACH

- ✓ FONCTION QUI BOUCLE SUR CHAQUE ÉLÉMENT TROUVÉ ET EXÉCUTE À CHAQUE PASSAGE UNE FONCTION DÉFINIE.
- ✓ LA FONCTION EXÉCUTÉE DISPOSE D'UN ARGUMENT QUI EST UN ENTIER REPRÉSENTANT LA POSITION DE L'ÉLÉMENT EN COURS DE TRAITEMENT.
- ✓ UN "RETURN FALSE" À L'INTÉRIEUR DE LA FONCTION DÉFINIE STOPPERA L'EXÉCUTION DE LA BOUCLE. EN REVANCHE "RETURN TRUE" FORCERA LE PASSAGE À L'ITÉRATION SUIVANTE, DE LA MÊME MANIÈRE QUE L'INSTRUCTION "CONTINUE" DANS UNE BOUCLE NORMALE.

```
$("#img").each(function(i){  
  this.src = "image" + i + ".jpg";  
});
```

Ceci va transformer

en



```
var tab = ['aaa','bbb','ccc','ddd'];  
$.each (tab, function (index, valeur)  
{  
    console.log (index+':'+valeur)  
});
```

```
$.each ($ ('.container'), function ( index, valeur) {  
    console.log (index + ':' + $ (valeur) .text ());  
});
```

```
// OBJECTS  
const obj = {  
    un: 1,  
    deux: 2,  
    trois: 3,  
    quatre: 4,  
    cinq: 5  
};
```

```
$.each (obj, function (index, valeur) {  
    console.log (index+':'+ valeur);  
});
```

JQUERY – LES FONCTIONS UTILES

EQ

- ✓ RÉDUIT LE RÉSULTAT DE LA RECHERCHE À UN ÉLÉMENT, CORRESPONDANT À UNE POSITION DONNÉE.
- ✓ L'INTERVALLE DES POSITIONS COMMENCE À 0 ET SE TERMINE À TAILLE DE L'INDEX-1.

`$("p").eq(1)`

Ceci va transformer

`<p>Premier Test</p><p>Second Test</p>`

en

`<p>Second Test</p>`

JQUERY – LES FONCTIONS UTILES

GET

✓ PERMET D'ACCÉDER À TOUS LES ÉLÉMENTS RECHERCHÉS.

```
$("#img").get();
```

Ceci va transformer

```
 
```

en

```
 
```


JQUERY – LES FONCTIONS UTILES

LENGTH

✓ RETOURNE LE NOMBRE D'ÉLÉMENTS TROUVÉS.

```
$(“img”).length;
```

Ceci va donner ‘3’ pour les éléments

```
  
```



TP -

- Voir exercices—dom.pdf: refaire l'exercice 4 sur les images en jquery.
- Afficher des photos de petite taille sur la zone de droite, lorsqu'une des photos est cliquée, elle s'affiche automatiquement en grand dans la zone de gauche.



TP- Palindrome

- Faire un script permettant de vérifier si le saisi utilisateur est un palindrome.
- Si Oui, on affiche le message 'saisi est un palindrome'
- Sinon on affiche le message 'Saisi n'est pas un palindrome'

JQUERY – LES EVENEMENTS

Event Method	Description
<code>\$(document).ready(function)</code>	Lie une fonction à l'événement "ready" du document (Quand le document est chargée)
<code>\$(selector).click(function)</code>	Lie une fonction à l'événement "click" des événements sélectionnés
<code>\$(selector).dblclick(function)</code>	Lie une fonction à l'événement "double click" des événements sélectionnés
<code>\$(selector).focus(function)</code>	Lie une fonction à l'événement "focus" des événements sélectionnés
<code>\$(selector).mouseover(function)</code>	Lie une fonction à l'événement "mouseover" des événements sélectionnés
<code>\$(item).change(function)</code>	Lie une fonction à l'événement "modification de valeur" d'un objet
<code>\$(item).select(function)</code>	Lie une fonction à l'événement "selection" d'un objet

JQUERY – LES EVENEMENTS

LA GESTION DES ÉVÈNEMENTS EST UNE PARTIE MAJEURE DE JQUERY

✓ **LES 'EVENT HANDLERS' SONT LES FONCTIONS APPELÉS PAR UNE ACTION**

✓ **IL EST CLASSIQUE DE METTRE LE CODE JQUERY QUI GÈRE LES ÉVÉNEMENTS ENTRE LES TAGS**

<HEAD> ET </HEAD>

```
<HTML>
<HEAD>
<SCRIPT TYPE="TEXT/JAVASCRIPT" SRC="JQUERY.JS"></SCRIPT>
<SCRIPT TYPE="TEXT/JAVASCRIPT">
    $(DOCUMENT).READY(FUNCTION){
        $("BUTTON").CLICK(FUNCTION){
            $("P").HIDE();
        };
    };
</SCRIPT>
</HEAD>
<BODY>
    <H2><HEADING DE LA PAGE</H2>
    <P>MON PARAGRAPHE.</P>
    <P>UN AUTRE PARAGRAPHE.</P>
    <BUTTON>CLICK MOI</BUTTON>
</BODY>
</HTML>
```



JQUERY – LES EVENEMENTS

- Un événement correspond à une action qu'on va pouvoir intercepter et à laquelle on va pouvoir réagir. Un événement est un signal qui indique que “quelque chose vient de se passer”.
- Nos gestionnaires d'événements vont être des fonctions de rappel, c'est-à-dire des fonctions (généralement anonymes) qui vont être appelées uniquement lorsque l'événement qu'elles gèrent va être déclenché.

JQUERY – LES EVENEMENTS

L'OBJET JQUERY.EVENT

- ✓ DÉFINIT L'ENSEMBLE DES PROPRIÉTÉS DE L'ÉVÉNEMENT
- ✓ EVENT.TARGET CONTIENT LA DESTINATION POUR CET ÉVÉNEMENT
- ✓ EVENT.TYPE CONTIENT LE TYPE D'ÉVÉNEMENT

```
$("#a").click(function(event) {  
    alert(event.type);  
});
```

Va afficher

« click »

```
$("#a[href=http://moussa.com]").click(function(event)  
) {  
    alert(event.target.href);  
});
```

Va afficher

"http://moussa.com"

Les ancetres

- La méthode `.parent()` retourne l'élément parent direct de l'élément sélectionné. Cette méthode ne traverse qu'un seul niveau dans l'arborescence DOM.

```
<body>
  <section>
    <div>
      <div class='ma_classe'>Ceci est un test</div>
    </div>
  </section>
</body>
```

```
$('.ma_classe').parent().addClass('moussa');
```

Va afficher

```
<body> <section> <div class='moussa'> <div class='ma_classe'>...</div> </div> </section> </body>
```

Les descendants


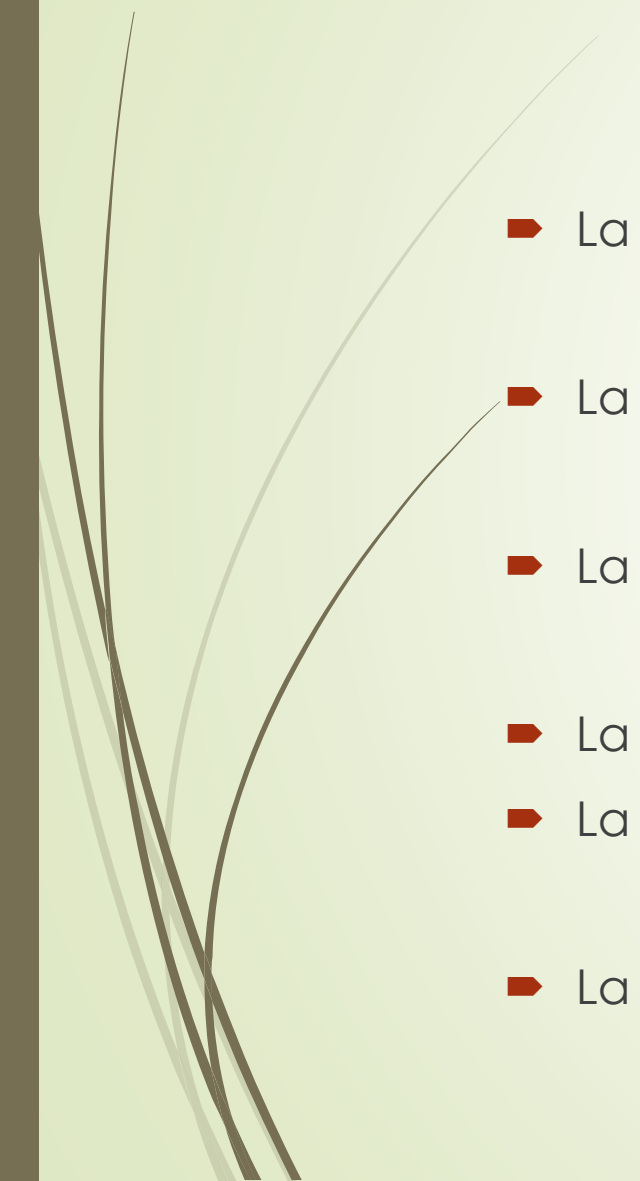
- La méthode `.children()` retourne tous les enfants directs de l'élément sélectionné. Cette méthode ne traverse qu'un seul niveau dans l'arbre DOM.

```
<body>
  <section>
    <div class='ma_classe'>
      <div>Ceci est un test</div>
    </div>
  </section>
</body>
```

```
$('.ma_classe').children().addClass('moussa');
```

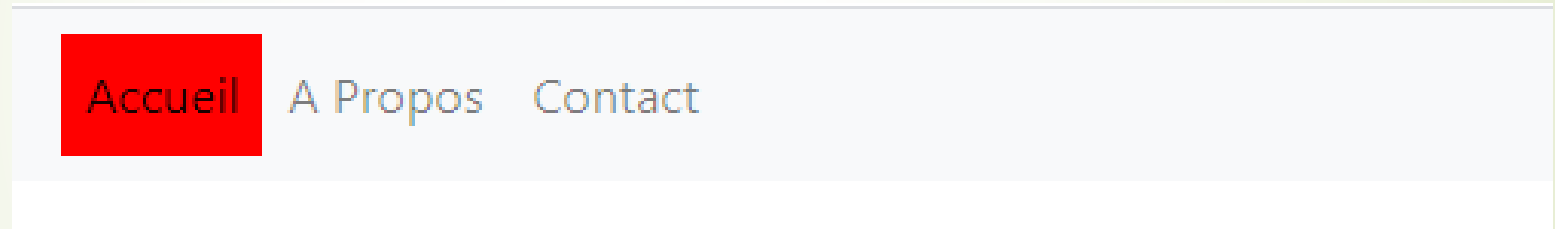
Va afficher

```
<body> <section> <div class='ma_classe'> <div class='moussa'>...</div> </div> </section> </body>
```

- 
- 
- La méthode `.find()` retourne tous les enfants passés en paramètre.
 - La méthode `.siblings()` retourne tous les éléments frères de l'élément sélectionné.
 - La méthode `.next()` retourne le frère suivant de l'élément sélectionné.
 - La méthode `.nextAll()` retourne tous les frères suivants de l'élément sélectionné.
 - La méthode `.prev()` retourne le frère prédécent l'élément sélectionné.
 - La méthode `.prevAll()` retourne tous les frères prédécents de l'élément sélectionné.

Exemple de menu actif

- Dans toutes applications web, nous trouverons une barre de menu par exemple:



- Nous pouvons rendre un lien actif ou pas en fonctions de la navigation de l'utilisateur.

Exemple de code

Index.html

```
<div class="collapse navbar-collapse"
id="navbarSupportedContent">
  <ul class="navbar-nav mr-auto liste">
    <li class="nav-item active"> <a
class="nav-link" href="#">Accueil</a></li>
    <li class="nav-item"> <a class="nav-link"
href="#">A Propos</a> </li>
    <li class="nav-item"> <a class="nav-link"
href="#">Contact</a></li>
  </ul>
</div>
```

Accueil A Propos Contact

Style.js

```
var $links = $('.liste a');
$links.click ( function (e) {
  e.preventDefault();
  $(this).parent().siblings().removeClass('
active');
  $(this).parent().addClass('active')
})
```


EFFETS VISUELS POUR JQUERY

- ENSEMBLE D'EFFETS QUI SONT DISPONIBLES POUR RENDRE PLUS AGRÉABLE L'EXPÉRIENCE UTILISATEUR
- POSSIBILITÉ DE RAJOUTER DES PLUGIN POUR AMÉLIORER CES EFFETS
- POSSIBILITÉ DE MODIFIER TRÈS FACILEMENT LA VITESSE LES COULEURS DES EFFETS
- EFFETS PRINCIPAUX QUI PEUVENT ÊTRE UTILISÉS:

BASIQUES

- ✓ SHOW SHOW (SPEED, CALLBACK)
- ✓ HIDE HIDE (SPEED, CALLBACK)
- ✓ TOGGLE
- ✓ SLIDEDOWN(SPEED, [CALLBACK])

EFFETS DE GLISSEMENT

- ✓ SLIDEDOWN(SPEED, [CALLBACK])
- ✓ SLIDEUP(SPEED, [CALLBACK])
- ✓ SLIDETOGGLE(SPEED, [CALLBACK])

EFFETS DE FONDU

- ✓ FADEIN(SPEED, CALLBACK)
- ✓ FADEOUT(SPEED, CALLBACK)
- ✓ FADETO(SPEED, CALLBACK)

EFFETS VISUELS POUR JQUERY

❖ SHOW() - SHOW(SPEED, CALLBACK)

PERMET SOIT DE RENDRE VISIBLE UN SET D'ÉLÉMENTS DOM SÉLECTIONNÉS, SOIT DE FAIRE LA MÊME CHOSE MAIS AVEC LA POSSIBILITÉ D'AJOUTER UNE OPTION "SPEED" (EN MILLISECONDES) ET DES FONCTIONS "CALLBACK".

```
$("section .btn-warning").click(function(){  
    //$("p").show(2000);  
    $("p").show(2000);  
});
```

EFFETS VISUELS POUR JQUERY

❖ HIDE() - HIDE(SPEED, CALLBACK)

PERMET SOIT DE RENDRE NON-VISIBLE UN SET D'ÉLÉMENTS DOM SÉLECTIONNÉS, SOIT DE FAIRE LA MÊME CHOSE MAIS AVEC LA POSSIBILITÉ D'AJOUTER UNE OPTION "SPEED" (EN MILLISECONDES) ET DES FONCTIONS "CALLBACK".

```
$("section .btn-primary").click(function(){  
    $("p").hide("slow");  
});
```

EFFETS VISUELS POUR JQUERY

❖ TOGGLE()

CETTE INSTRUCTION PERMET D'EFFECTUER UN "TOOGLE" , D'INTERVERTIR LE STATUT DE VISIBILITÉ D'UN SET D'ÉLÉMENTS DOM SÉLECTIONNÉS.

SI LES ÉLÉMENTS SONT VISIBLES ILS DEVIENNENT NON-VISIBLES ET INVERSEMENT.

```
$( "section .btn" ).click(function() {  
    //$( "p" ).toggle( "slow" );  
    // $( "p").slideToggle("slow");  
    $( "p").slideToggle(5000);  
  
});
```

EFFETS VISUELS POUR JQUERY

❖ SLIDEDOWN(SPEED, [CALLBACK])

CETTE INSTRUCTION PERMET DE FAIRE APPARAÎTRE UN SET D'ÉLÉMENTS SÉLECTIONNÉS ET D'AJOUTER ÉVENTUELLEMENT UNE FONCTION "CALLBACK". SEULE LA HAUTEUR EST AJUSTÉE POUR CET EFFET CE QUI DONNE UNE IMPRESSION D'APPARITION DE TYPE "SLIDE".

```
function demo04(){  
    $("#myTest03").click(function () {  
        if ($("#myTest03 div:first").is(":hidden")) {  
            $("#myTest03 div").slideDown("slow");  
        } else {  
            $("#myTest03 div").hide();  
        }  
    });  
}
```

EFFETS VISUELS POUR JQUERY

❖ SLIDEUP(SPEED, [CALLBACK])

CETTE INSTRUCTION PERMET DE FAIRE DISPARAÎTRE UN SET D'ÉLÉMENTS SÉLECTIONNÉS ET D'AJOUTER ÉVENTUELLEMENT UNE FONCTION "CALLBACK". SEULE LA HAUTEUR EST AJUSTÉE POUR CET EFFET CE QUI DONNE UNE IMPRESSION DISPARITION DE TYPE "SLIDE".

```
function demo04(){  
    $("#myTest04").click(function () {  
        if ($("#myTest04 div:first").is(":hidden")) {  
            $("#myTest04 div").show("slow");  
        } else {  
            $("#myTest04 div").slideUp();  
        }  
    });  
}
```

EFFETS VISUELS POUR JQUERY

❖ SLIDETOGGLE(SPEED, [CALLBACK])

INSTRUCTION QUI PERMET DE FAIRE UN "TOOGLE" , INTERVERTIR LE STATUT DE VISIBILITÉ D'UN SET D'ÉLÉMENTS DOM SÉLECTIONNÉS.

SEULE LA HAUTEUR EST AJUSTÉE POUR OBTENIR CET EFFET CE QUI DONNE L'IMPRESSION D'UNE APPARITION DE TYPE "SLIDE".

```
$( "section .btn" ).click(function() {  
  //$( "p" ).toggle( "slow" );  
  // $( "p").slideToggle("slow");  
  $( "p").slideToggle(5000);  
});
```


EFFETS VISUELS POUR JQUERY

❖ FADEIN(SPEED, [CALLBACK])

INSTRUCTION QUI PERMET D'EFFECTUER UN "FADEIN" SUR UN SET D'ÉLÉMENTS DOM SÉLECTIONNÉS AVEC LA POSSIBILITÉ D'AJOUTER UNE FONCTION "CALLBACK". SEULE L'OPACITÉ EST AJUSTÉE POUR CRÉER CET EFFET CE QUI SIGNIFIE QUE LA LARGEUR ET LA HAUTEUR SERONT RAPIDEMENT IDENTIFIABLES DURANT L'EFFET.

```
$( "section .btn" ).click(function() {  
    $( "p" ).fadeIn("slow");  
});
```

EFFETS VISUELS POUR JQUERY

❖ FADEOUT(SPEED, CALLBACK)

INSTRUCTION QUI PERMET D'EFFECTUER UN "FADEOUT" SUR UN SET D'ÉLÉMENTS DOM SÉLECTIONNÉS AVEC LA POSSIBILITÉ D'AJOUTER UNE FONCTION "CALLBACK".

SEULE L'OPACITÉ EST AJUSTÉE POUR CRÉER CET EFFET CE QUI SIGNIFIE QUE LA LARGEUR ET LA HAUTEUR SERONT RAPIDEMENT IDENTIFIABLES DURANT L'EFFET.

```
function demo07(){  
    $("#myTest07 p").click(function () {  
        $("#myTest07 p").fadeOut("slow");  
    });  
}
```

EFFETS VISUELS POUR JQUERY

❖ FADETO(SPEED, OPACITY, CALLBACK)

INSTRUCTION QUI PERMET D'EFFECTUER UN "FADETO" SUR UN SET D'ÉLÉMENTS DOM SÉLECTIONNÉS AVEC LA POSSIBILITÉ D'AJOUTER UNE FONCTION "CALLBACK".

SEULE L'OPACITÉ EST AJUSTÉE POUR CRÉER CET EFFET CE QUI SIGNIFIE QUE LA LARGEUR ET LA HAUTEUR SERONT RAPIDEMENT IDENTIFIABLES DURANT L'EFFET.

```
function demo08(){  
    $("#myTest08 p:first").click(function () {  
        $(this).fadeTo("slow", 0.33);  
    });  
}
```

EXERCICE

- Développer une page web qui calcule l'indice de masse corporelle :
 $IMC = \text{masse} / (\text{taille en m})^2$
- La page web doit contenir 2 champs de texte pour insérer la masse et la taille d'un individu, un bouton pour déclencher le calcul de l'indice et l'afficher dans la page web.

Votre taille (en cm)

Votre poids (en kg)

JE CALCULE MON IMC

Résultat

Votre IMC (indice de masse corporelle) est de :

24.8

18.5 < IMC < 24.9 : corpulence normale



TP - Formulaire

- Voir TP-JQUERY.docx.pdf
- Compteur du nombre de mots et de caractères



QUESTIONS



QUESTION (S) ?

Reponse



**KEEP
CALM
AND
ASK
THE GOOGLE**