Collection EXPERT IT

jQuery

http://www.free-livres.com/

Le framework JavaScript du Web 2.0

Luc VAN LANCKER

Fichiers à télécharger

www.editions-eni.fr

INFORMATIQUE TECHNIQUE

eni Editions

Introduction

Depuis 1995, soit dès le début de l'ouverture du web au grand public, le JavaScript est le compagnon fidèle du HTML. Néanmoins, il faut admettre que sa reconnaissance par les développeurs a été lente. Cependant, depuis l'apparition d'AJAX, la situation a totalement évolué. Les designers lui découvrent des qualités jusque là inexploitées et le JavaScript acquiert enfin ses lettres de noblesse.

JavaScript est cependant un langage de programmation à part entière avec toutes les difficultés d'apprentissage que cela implique. L'idée de concevoir un framework dédié à JavaScript pour accroître la productivité de développement s'est ainsi imposée. C'est alors qu'apparaît le framework jQuery.

Mais jQuery, grâce à l'approche innovante de son jeune inventeur John Resig, reconsidère complètement l'apport et l'écriture du JavaScript. En effet, il était impératif de tenir compte du DOM et des feuilles de style CSS apparus depuis la naissance du JavaScript. Il importait également de revenir à une syntaxe plus intuitive car de la version 1.1 initiale à la version 1.7 actuelle, l'évolution du JavaScript s'est réalisée au prix d'une complexité croissante.

Les grands noms de l'informatique et d'Internet comme Google, Nokia et Microsoft ne sont pas restés en reste et ont rapidement apporté leur reconnaissance à jQuery. Ainsi, la librairie jQuery est maintenant incluse dans la suite d'outils de développement Visual Studio de Microsoft.

Le contexte ainsi planté, il semble évident qu'une connaissance approfondie du HTML (ou XHTML) et des feuilles de style CSS est indispensable. De bonnes notions de JavaScript sont également souhaitées. JQuery s'adresse en effet à un public au fait de ces différentes techniques.

Dans cet ouvrage, nous allons parcourir les différents thèmes abordés par jQuery. L'auteur a privilégié une approche structurée et progressive. Chaque point de jQuery est illustré par un exemple avant de passer à des applications plus pointues. Attirons l'attention du lecteur sur le chapitre consacré aux sélecteurs (chapitre Les sélecteurs en jQuery). Il est non seulement l'illustration de la diversité de jQuery pour atteindre aisément n'importe quel élément de la page mais un élément essentiel dans l'apprentissage de jQuery. Suivront ensuite des sujets plus interactifs comme la manipulation des feuilles de style ou du DOM, les effets visuels, l'AJAX revu par jQuery et les plug-ins.

Votre étude terminée, vos applications Web seront à n'en pas douter, plus interactives, plus riches et plus innovantes, le tout avec une facilité d'écriture du JavaScript insoupçonnée.

© ENI Editions - All rigths reserved - Jonifar lina

Le retour du JavaScript

Les applications Web actuelles se situent dans un environnement dynamique. Jamais encore les sites n'ont été aussi riches en présentations visuelles et autres fonctions soutenant la gestion des informations.

Dès les premières heures du Web, le JavaScript a été un partenaire privilégié dans la conception des pages Html par l'interactivité qu'il permettait d'ajouter à celles-ci. Mais sa présence et son influence restèrent longtemps limitées, principalement à cause des difficultés de réalisation de scripts vraiment compatibles avec les différents navigateurs de l'époque.

L'apparition du DOM (*Document Object Model*), permettant d'accéder ou de mettre à jour le contenu, la structure et le style des documents Html, fut le premier moteur du renouveau du JavaScript. D'autant plus que le DOM, recommandation du W3C, fut largement adopté par tous les navigateurs. Ce qui atténua les problèmes d'interopérabilité des scripts.

Vinrent ensuite AJAX (*Asynchronous JavaScript and XML*) et les requêtes XMLHttpRequest associées qui donnèrent naissance au JavaScript asynchrone offrant la possibilité de modifier une partie des pages Web sans devoir recharger l'entièreté de celles-ci. La porte était ouverte à des applications JavaScript beaucoup plus riches répondant au mieux au souci d'interactivité des applications Web. Ici aussi la compatibilité était gagnante.

Le concept du Web 2.0, dans ses objectifs d'une meilleure usabilité et d'une plus grande ergonomie, a quant à lui encore renforcé l'interactivité des pages et la demande d'applications plus étendues. Et voilà le JavaScript propulsé comme un élément incontournable dans le développement des applications Web.

La meilleure preuve de ce repositionnement du JavaScript est assurément l'apparition de nouveaux moteurs JavaScript dans les navigateurs récents. Que ce soit Google Chrome avec son moteur JavaScript Open Source V8, Opera avec le projet Carakan, Safari dans sa version 4 ou Firefox 3.5 avec TraceMonkey, tous cherchent à améliorer (et parfois de façon sensible) le traitement du JavaScript. Seul Internet Explorer 8 est un peu à la traîne mais le renouvellement de son moteur JavaScript fait partie des priorités d'Internet Explorer 9.

© ENI Editions - All rigths reserved - Jonifar lina

Présentation de JQuery

jQuery est un framework JavaScript libre et Open Source, implanté côté client, qui porte sur l'interaction entre le DOM, JavaScript, AJAX et le Html. Cette librairie JavaScript a pour but de simplifier les commandes communes du JavaScript. La devise de jQuery est en effet, "Écrire moins pour faire plus" (*write less do more*).

jQuery, du moins à l'origine, est l'œuvre d'un seul homme : John Resig. Ce jeune surdoué de JavaScript développa la première version de jQuery en janvier 2006. Il avait alors à peine vingt ans ! S'il reste l'élément moteur de jQuery, il peut maintenant se faire aider par une communauté de passionnés.

Les spécificités de jQuery sont nombreuses mais l'essentielle est assurément la souplesse qu'il apporte pour accéder à tous les éléments du document Html grâce à la multitude de sélecteurs mis en place (voir chapitre Les sélecteurs en jQuery). Cette caractéristique fut d'ailleurs retenue pour donner un nom à ce framework : j pour JavaScript et Query pour chercher ou accéder aux éléments.

Notons également que cette librairie jQuery est en constante évolution. Les mises à jour et nouvelles versions se succèdent à un rythme régulier :

Août 2006 : version stable de jQuery 1.0.

Janvier 2007 : jQuery 1.1.

Septembre 2007 : jQuery 1.2.

Janvier 2009 : jQuery 1.3.

C'est sur la version 1.3. que porte cet ouvrage.

La qualité de jQuery a été reconnue par les grands comptes du Web et de l'informatique. Citons Google, Mozilla, Dell, IBM, WordPress, Nokia et bien d'autres. Microsoft l'a incorporé dernièrement à son logiciel Visual Studio. Sa croissance est rapide et il se pose en concurrent sérieux à d'autres framework comme Prototype, Dojo Toolkit et Sciptaculous pour ne citer que ceux-là.

© ENI Editions - All rigths reserved - Jonifar lina

Les points forts de jQuery

Le framework jQuery est de plus en plus reconnu et adopté par les développeurs car les apports de cet environnement sont nombreux.

L'approche de jQuery ne consiste pas seulement en un codage des scripts plus intuitifs et concis mais sa philosophie première est concentrée sur l'ensemble des éléments pris en compte par le DOM. Le JavaScript traditionnel, dans son évolution historique, a dû s'accommoder du Document Object Model. John Resig avec jQuery, a en quelque sorte reconsidéré le JavaScript en le percevant comme un véritable langage de programmation axé en priorité sur le DOM. Ce qui modifie totalement la façon d'appréhender et d'écrire le JavaScript.

Tous les éléments du DOM sont aisément accessibles avec jQuery. Les méthodes `getElementById`, `getElementsByName` et `getElementsByTagName` du JavaScript montrent très rapidement leurs limites, spécialement lorsque le concepteur souhaite accéder aux attributs et autres propriétés de style. Avec jQuery, tous les éléments du document peuvent être accédés facilement et surtout intuitivement. Le chapitre suivant, consacré aux sélecteurs, illustrera la diversité apportée par jQuery en la matière.

L'approche de jQuery est complète. Les méthodes et fonctions de jQuery ne se limitent pas à quelques animations d'ordre esthétique. Par quelques lignes de code, jQuery peut modifier du texte, insérer des images, trier des tableaux ou réorganiser l'entièreté de la structure du document Html. Ce framework pose un regard nouveau sur le JavaScript et, après un bref apprentissage, simplifie grandement la conception et l'écriture des scripts. Nous ne manquerons pas d'attirer votre attention, dans la suite de cet ouvrage, sur la concision du code ainsi produit.

Le code jQuery est compatible avec les différents navigateurs. La déviance des navigateurs, infime ou plus marquée, par rapport aux standards du DOM et des feuilles de style, est une regrettable réalité du développement d'applications Web évoluées. Grâce à l'interface logicielle ajoutée par jQuery, le code des applications se révèle compatible avec les principaux navigateurs du marché. Il faut déjà fouiller les tréfonds des forums spécialisés pour trouver quelques accrochages portant sur des détails sommes toutes peu utilisés. Cette compatibilité ressort en particulier sur les éléments de feuilles de style CSS3 qui ne sont encore reprises que de façon très fragmentaire par les navigateurs. Citons comme exemple l'opacité des éléments. Les méthodes jQuery `fadeIn()`, `fadeOut()` et `fadeTo()` permettent de faire varier cette opacité de manière compatible avec tous les navigateurs.

Une communauté dynamique de développeurs soutient jQuery. Cette communauté, initiée selon les principes historiques de passion et de partage d'Internet, fournit une multitude de plug-ins, soit des extensions de jQuery, dédiées à des tâches très spécifiques. Ces plug-ins, souvent des merveilles de programmation, sont disponibles librement sur la toile et sont très prisés par les concepteurs de site. Un carrousel d'images ou un tableau triable implémenté en quelques minutes et en quelques lignes de code, simplifie grandement leur travail.

© ENI Editions - All rigths reserved - Jonifar lina

Mise en place de jQuery

jQuery étant une librairie côté client, sa mise en place est extrêmement simple.

Dans un premier temps, il faut télécharger jQuery. L'adresse la plus indiquée pour le réaliser est celle du site de jQuery lui-même soit http://jquery.com/. Sous la rubrique **Download**, il vous est proposé une version compressée dite de production (production) de 19 KB et une version de développement (development) de 120 KB. Cette dernière est bien adaptée à notre étude car elle permet au plus curieux de jeter un œil sur le code utilisé par jQuery. La version de production est utilisée une fois le travail de développement terminé et vos pages mises en ligne.

Au moment de l'écriture de cet ouvrage, le fichier téléchargé porte le nom de jquery-1.3.2.js. Pour la suite de notre étude, nous l'avons renommé en jquery.js. Une fois ce fichier téléchargé, il devra être présent dans le même dossier que celui de la page Html (ou Xhtml) qui comportera un script jQuery.

Les pages faisant appel à du JavaScript traité par jQuery doivent comporter dans l'en-tête du document, soit entre les balises `<head>` … `</head>`, un appel vers ce fichier JavaScript externe : `<script type="text/javascript" src="jquery.js"></script>`

Commentaires

La librairie jQuery est ainsi chargée à chaque fois qu'elle est nécessaire au bon déroulement de la page. Il faut à ce propos souligner le caractère compact de la librairie jQuery. Avec ses 19 KB dans sa version de production, elle n'a quasi aucune conséquence sur le temps de téléchargement de la page. En effet, 19 KB correspond au téléchargement d'une petite icône présente dans le document.

Il a été dit ci-avant, que le fichier jquery.js devait être présent dans le même dossier que la page Html. Cette façon de procéder est idéalement adaptée à l'apprentissage de jQuery. Dans la pratique, jquery.js sera souvent inclus dans un sous-répertoire par exemple js. On y accède alors par `<script type="text/javascript" src="js/jquery.js"></script>`.

Il est possible d'utiliser directement les versions de jQuery hébergées par Google sur AJAX Libraries API. Le lien vers le fichier jQuery devient alors :

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min.js">
</script>
```

ou

```
<script type="text/javascript"
src=" http://ajax.googleapis.com/ajax/libs/jquery/latest/jquery.min.js">
</script>
```

Les tenants de cette méthode avancent une charge réduite du serveur de votre site et une connexion à jQuery aussi (sinon plus) rapide qu'à partir d'un serveur quelconque. De plus, cela permet aux navigateurs de ne pas télécharger inutilement plusieurs fois la même librairie sur chaque site qui l'utilise, puisqu'une fois jQuery téléchargé à partir des serveurs de Google et mis en cache, le navigateur n'aura plus à le télécharger à nouveau.

© ENI Editions - All rigths reserved - Jonifar lina

Initier un script jQuery

Le framework mis en place et prêt à être utilisé, il faut au préalable parcourir quelques notions théoriques.

Toutes les instructions ou pour être plus précis les scripts jQuery s'articulent autour d'un modèle unique. Soit,

```
jQuery(function(){
//contenu exécuté lorsque le document sera chargé
});
```

Pour économiser des frappes de clavier, le signe dollar ($) qui fonctionne comme alias de jQuery est presque toujours utilisé.

```
$(function(){
//contenu exécuté lorsque le document sera chargé
});
```

Se référant à ce modèle, tout script jQuery commence par :

```
<script type="text/javascript">
$(document).ready(function() {
// code jQuery
});
</script>
```

Soit, selon une restitution assez libre, créer un objet jQuery (`$`) à partir du document (`document`) quand celui-ci est prêt (`ready`).

La particularité de cette fonction est de charger les éléments du DOM dès que ceux-ci sont disponibles, soit bien avant le chargement complet de la page.

Ce en quoi jQuery se différencie du JavaScript classique. Celui-ci utilise, par exemple, le classique `window.onload = function()` qui attend que la page et tous les éléments qu'elle contient soient complètement chargés. Ce qui peut être très long spécialement lorsqu'il y a des images d'une taille conséquente. C'est une particularité essentielle de jQuery qui se base, rappelons-le, de façon native sur les éléments du DOM.

Ce mode de fonctionnement présente de nombreux avantages :

Tous les éléments de la page sont inclus dans un objet que les sélecteurs, méthodes et fonctions de jQuery pourront manipuler.

Le code Html est dépouillé de toutes mentions JavaScript comme par exemple les notations `<a href="javascript:void(0);">lien</a>`. Tout le code JavaScript/jQuery se situe dans une partie séparée de la page Html (entre les balises `<head> ... </head>`) ou dans un fichier js externe. Ce qui respecte pleinement le principe de la séparation du contenu et de la présentation.

Avec `$(document).ready()`, les éléments de la page sont à la disposition du développeur, avant le chargement complet et l'affichage de celle-ci. Ce qui est bien pratique pour activer des effets d'apparition ou de disparition dès l'affichage de la page par le navigateur.

Vous rencontrerez sur la toile des scripts jQuery qui débutent avec l'écriture raccourcie :

```
$(function () {
// code jQuery
});
```

Nous garderons tout au long de cet ouvrage, l'instruction `$(document).ready()` plus académique et plus parlante.

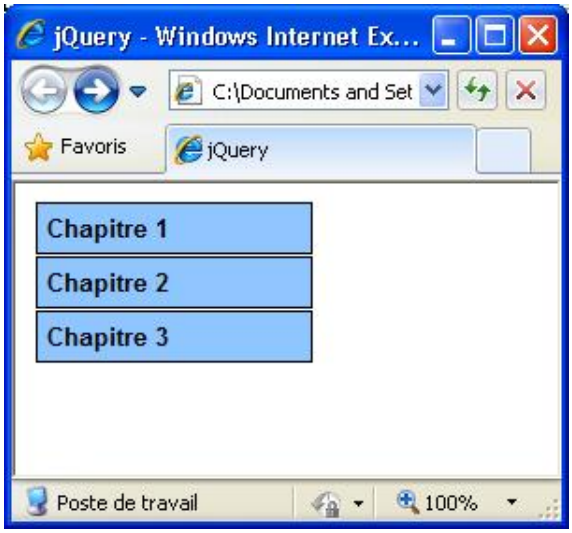
Le signe `$` est également utilisé par d'autres framework comme par exemple Prototype. La `méthode jquery.noconflict` (voir le chapitre Quelques méthodes utilitaires - Éviter les conflits) permet d'éviter les conflits à l'appel de l'alias `$` avec une autre librairie qui utiliserait également ce nommage pour une de ses fonctions.

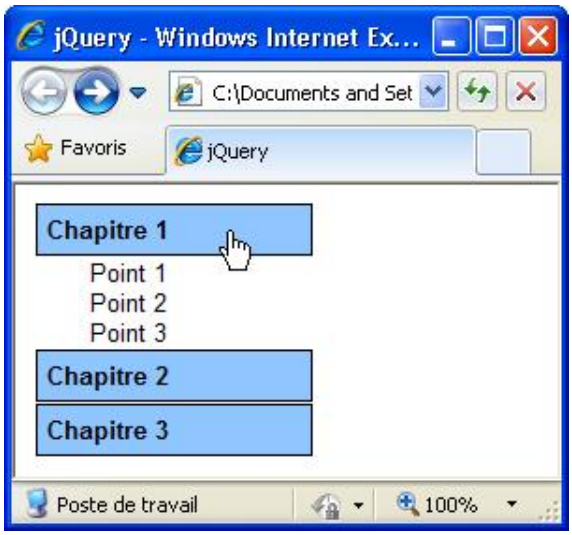
© ENI Editions - All rigths reserved - Jonifar lina

Une première application jQuery

Réalisons comme entrée en matière, un menu de navigation vertical déroulant.

Le but ici n'est pas de comprendre les méthodes jQuery qui seront étudiées plus avant mais d'avoir un aperçu de l'implantation et l'organisation générale des scripts jQuery. Cette première application a aussi comme objectif de familiariser le lecteur avec l'approche suivie par l'auteur dans notre exploration de jQuery.



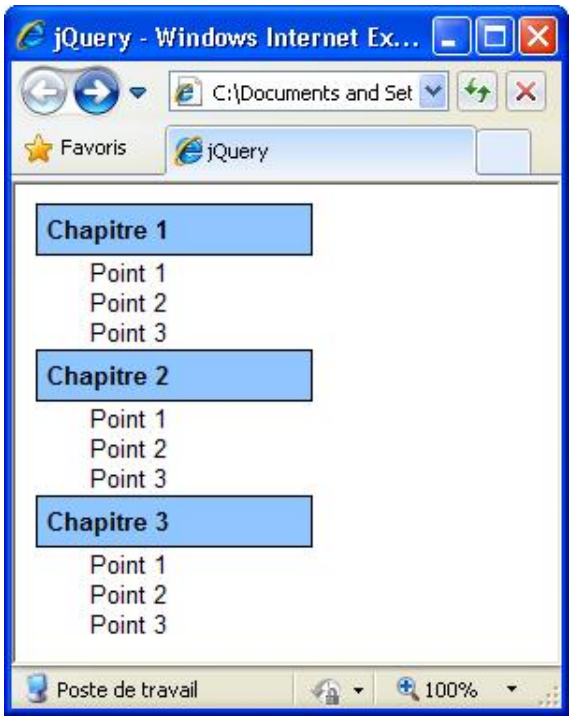


Le fichier Html de départ se présente ainsi :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
body { margin: 10px;
     font: 0.8em Arial, Helvetica, sans-serif;}
.menu { width: 150px;}
.menu_chapitre { padding: 5px;
                 cursor: pointer;
                 position: relative;
                 margin:1px;
                 font-weight:bold;
                 background: #9cf;
```

© ENI Editions - All rigths reserved - Jonifar lina

```
                 border: 1px solid black;}
.menu_point a { display:block;
                color:black;
                background-color:white;
                padding-left:30px;
                text-decoration:none;}
.menu_point a:hover { color: black;
                      text-decoration:underline;}
</style>
</head>
<body>
<div>
<div class="menu">
<p class="menu_chapitre">Chapitre 1</p>
<div class="menu_point">
<a href="#">Point 1</a>
<a href="#">Point 2</a>
<a href="#">Point 3</a>
</div>
<p class="menu_chapitre">Chapitre 2</p>
<div class="menu_point">
<a href="#">Point 1</a>
<a href="#">Point 2</a>
<a href="#">Point 3</a>
</div>
<p class="menu_chapitre">Chapitre 3</p>
<div class="menu_point">
<a href="#">Point 1</a>
<a href="#">Point 2</a>
<a href="#">Point 3</a>
</div>
</div>
</div>
</body>
</html>
```



Élaborons le script jQuery pas à pas.

Nous supposons que la librairie jquery.js a été téléchargée et incorporée dans le dossier contenant le fichier Html précédent.

Appelons le jQuery en plaçant entre les balises <head> et </head>, la balise d'en-tête suivante.

© ENI Editions - All rigths reserved - Jonifar lina

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

Le script en lui-même peut débuter.

```
<script type="text/javascript">
$(document).ready(function(){
...
});
</script>
```

Avec cette première instruction jQuery, l'arbre du DOM relatif au document est chargé dans un objet jQuery.

Une première opération à effectuer consiste à ne pas afficher les sous-menus, soit les divisions dont la classe est `menu_point`.

```
<script type="text/javascript">
$(document).ready(function(){
$("div.menu_point").hide();
...
});
</script>
```

Ainsi, jQuery (`$`) sélectionne ces divisions `<div>` dont la classe est `menu_point` (`("div.menu_point")`) et les fait disparaître (`hide()`).

Passons maintenant à la partie qui prend en charge le déroulement du sous-menu lorsqu'un des chapitres a été cliqué.

```
<script type="text/javascript">
$(document).ready(function(){
$("div.menu_point").hide();
$("p.menu_chapitre").click(function(){
$(this).next("div.menu_point").slideDown(300)
.siblings("div.menu_point").slideUp("slow");
});
});
```

Détaillons cette partie :

```
$("p.menu_chapitre").click(function(){
```

Le clic d'un des paragraphes dont la classe est `menu_chapitre` (soit de façon codée `$("p.menu_chapitre")`) engendre une fonction qui sera détaillée par les lignes suivantes (`click(function()`).

```
$(this).next("div.menu_point").slideDown(300)
```

À partir de cet élément (`this`), le script demande de rechercher l'élément `<div>` qui le suit et qui a la classe `menu_point` (soit `next("div.menu_point")`). On demande alors de faire apparaître cette division selon un effet de glissement vers les bas (`slideDown(300)`).

```
.siblings("div.menu_point").slideUp("slow");
```

Il faut encore refermer les divisions ouvertes des autres chapitres. Il est demandé à jQuery de sélectionner les frères immédiats (`siblings`) de chaque division `<div>` qui dispose d'une classe `menu_point` (`siblings("div.menu_point")`) et de refermer ces divisions par un effet de glissement vers le haut (`slideUp("slow")`).

Il faut noter que de multiples actions ont été ainsi encodées dans une même instruction. Ces différentes opérations sont simplement reliées par un point.

```
});
```

Fin de cette fonction relative au clic d'un paragraphe.

```
});
```

Fin du ready et fin de script.

Vous remarquerez déjà la concision du code jQuery. Quelques lignes suffissent pour développer ce script.

© ENI Editions - All rigths reserved - Jonifar lina

Au final, le fichier Html se présente ainsi :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("div.menu_point").hide();
$("p.menu_chapitre").click(function(){
$(this).next("div.menu_point").slideDown(300)
.siblings("div.menu_point").slideUp("slow");
});
});
</script>
<style type="text/css">
body { margin: 10px;
       font: 0.8em Arial, Helvetica, sans-serif;}
.menu { width: 150px;}
.menu_chapitre { padding: 5px;
                 cursor: pointer;
                 position: relative;
                 margin:1px;
                 font-weight:bold;
                 background: #9cf;
                 border: 1px solid black;}
.menu_point a { display:block;
                color:black;
                background-color:white;
                padding-left:30px;
                text-decoration:none;}
.menu_point a:hover { color: black;
                      text-decoration:underline;}
</style>
</head>
<body>
<div>
<div class="menu">
<p class="menu_chapitre">Chapitre 1</p>
<div class="menu_point">
<a href="#">Point 1</a>
<a href="#">Point 2</a>
<a href="#">Point 3</a>
</div>
<p class="menu_chapitre">Chapitre 2</p>
<div class="menu_point">
<a href="#">Point 1</a>
<a href="#">Point 2</a>
<a href="#">Point 3</a>
</div>
<p class="menu_chapitre">Chapitre 3</p>
<div class="menu_point">
<a href="#">Point 1</a>
<a href="#">Point 2</a>
<a href="#">Point 3</a>
</div>
</div>
</div>
</body>
</html>
```


© ENI Editions - All rigths reserved - Jonifar lina

La documentation relative à jQuery

La documentation relative à jQuery est abondante sur la toile mais est majoritairement en anglais.

Une documentation en ligne très complète de jQuery, est disponible sur le site officiel : http://docs.jquery.com/

Il est également possible de consulter cette documentation hors ligne :

- Sous forme d'un fichier Adobe Air à télécharger sur http://api.jquery.com/
- Sous forme d'un fichier chm à télécharger à l'adresse : http://charupload.wordpress.com/2007/12/07/jquery-documentation-chm/
- Sous forme d'une API : http://jquery.bassistance.de/api-browser/

Un aide-mémoire précieux au format Adobe pdf ou Microsoft xls ou image png est disponible à l'url http://www.javascripttoolbox.com/jquery/cheatsheet/

Citons aussi, Visual jQuery consultable en ligne (http://visualjquery.com/) ou téléchargeable au format zip à l'adresse : http://support.aptana.com/asap/browse/STU-3495

Pour une documentation complète en français, il faut se reporter :

- Au site Developpeur Web 2 qui propose une traduction assez fidèle de la documentation officielle (jquery.developpeur-web2.com/documentation.php).
- Au site très sympathique et d'excellente facture d'un passionné à l'adresse : jquery.jarodxxx.com/.

Lorsque vous consultez les documentations et tutoriels relatifs à jQuery, assurez-vous qu'ils correspondent bien à la version 1.3. Il y a en effet des différences sensibles entre la version 1.2 et 1.3. Pour exemple, les sélecteurs d'attributs comportant le signe @ ne sont plus reconnus par la version 1.3. Les notations du style a [@href] ou $("input[@type=radio][@checked]") n'ont plus cours et indiquent que la documentation consultée est obsolète.

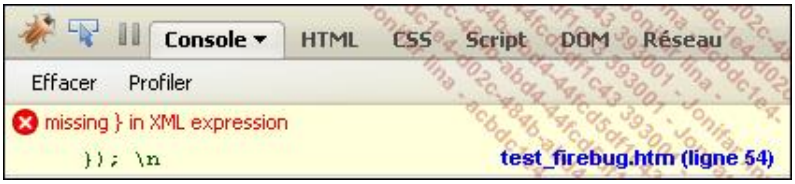
© ENI Editions - All rigths reserved - Jonifar lina

Outils de développement et de débogage

Rappelons que jQuery est un framework côté client. Ainsi, il ne nécessite qu'un strict minimum d'outils ; un navigateur et un éditeur de texte. Il faut cependant noter que pour les scripts AJAX (voir chapitre AJAX vu par jQuery), l'installation d'un serveur Web local est à prévoir mais nous y reviendrons lors de l'étude de ce chapitre.

Pour ce qui est du débogage, il n'existe pas de solution miracle pour debugger le code JavaScript. Pourtant, les navigateurs proposent depuis quelque temps, des solutions innovantes permettant d'inspecter et de débugger l'environnement des pages Web, soit le Html, les CSS, les scripts et le DOM.

Le plus reconnu par les développeurs est Firebug. Cette extension de Firefox permet, en un coup d'œil, de contrôler votre Html, CSS, JavaScript, DOM et AJAX.



Pour en tirer le meilleur profit, ne manquez pas de consulter l'excellent tutorial (en français) à l'adresse :

http://eric-pommereau.developpez.com/tutoriels/outil-web/firebug/

Il existe également une solution multiplate-forme pour les autres navigateurs soit Firebug Lite, qui est le pendant "script" de l'extension pour Firefox. Cette interopérabilité se paie (pour l'instant) par des performances inférieures à la solution réservée à Firefox.

Pour sa part, Internet Explorer 8 propose de façon native, des outils de développement qui ne sont pas sans ressembler à ceux de Firebug. On y accède par le menu **Outils - Outils de développement** (touche de raccourci [F12]).

Citons encore Dragonfly debugger pour Opera et WebInspector pour Safari.

© ENI Editions - All rigths reserved - Jonifar lina

Introduction

Les sélecteurs sont l'un des aspects le plus important de la librairie jQuery. Ceux-ci utilisent une syntaxe qui n'est pas sans rappeler celle des feuilles de style CSS. Ils permettent aux concepteurs d'identifier rapidement et aisément n'importe quel élément de la page et d'y appliquer les méthodes spécifiques à jQuery.

La bonne compréhension de ces sélecteurs jQuery est un élément clé pour la bonne compréhension et utilisation de jQuery.

Un site Web illustre à merveille le rôle de ces sélecteurs. Il est sûrement utile de le consulter à l'adresse www.codylindley.com/jqueryselectors/

© ENI Editions - All rigths reserved - Jonifar lina

Les sélecteurs de base

Ces sélecteurs basiques de jQuery ne sont en fait qu'une reformulation des méthodes `getElementById` et `getElementsByTagName` du JavaScript traditionnel.

La notation reprise par jQuery présente les avantages d'être plus concise et beaucoup plus intuitive.

Ces sélecteurs basiques sont très fréquemment utilisés dans les scripts jQuery et leur bonne compréhension se révèle indispensable à l'apprentissage de jQuery et à la suite de cet ouvrage.

1. Sélection par l'identifiant

#identifiant

Sélectionne l'élément (unique) spécifié par l'attribut `id="identifiant"`.

`$("#box")` : sélectionne l'élément dont l'id est box.

C'est la notation jQuery de `getElementById` du JavaScript classique.

Rappelons que cet identifiant doit être unique dans la page. Si par erreur, ce n'était pas le cas, jQuery reprend le premier élément doté de cet identifiant.

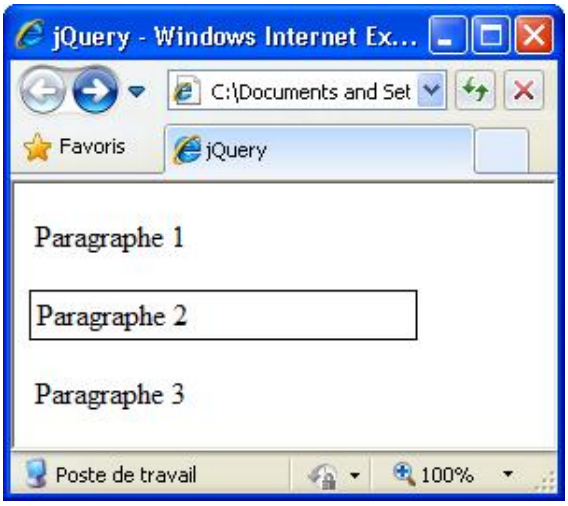
Exemple

Entourons d'une bordure, le second paragraphe dont l'identifiant est "deux".

Soit le fichier Xhtml suivant :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js" ></script>
<script type="text/javascript">
$(document).ready(function(){
$("#deux").css("border","1px solid black");
});
</script>
<style type="text/css">
p { width: 200px;
    padding: 3px;}
</style>
</head>
<body>
<p>Paragraphe 1</p>
<p id="deux">Paragraphe 2</p>
<p>Paragraphe 3</p>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina



Détaillons le script jQuery.

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

Initialisation de jQuery. Le script est prêt à opérer dès le chargement du DOM.

```
$("#deux").css("border","1px solid black");
```

Il est appliqué à l'élément dont l'id est "deux" ($("#deux")), la méthode jQuery qui permet de modifier les propriétés CSS, soit ici d'ajouter une bordure (`css("border","1px solid black")`). Cette méthode jQuery `css()` sera étudiée en détail au chapitre Manipulation des feuilles de style CSS.

```
});
```

Fin du script.

2. Sélection par le nom de la balise

élément

Sélectionne tous les éléments (ou balises) dont le nom est spécifié.

`$("div")` : sélectionne toutes les divisions `<div>` de la page.

C'est la notation jQuery de `getElementsByTagName` du JavaScript classique.

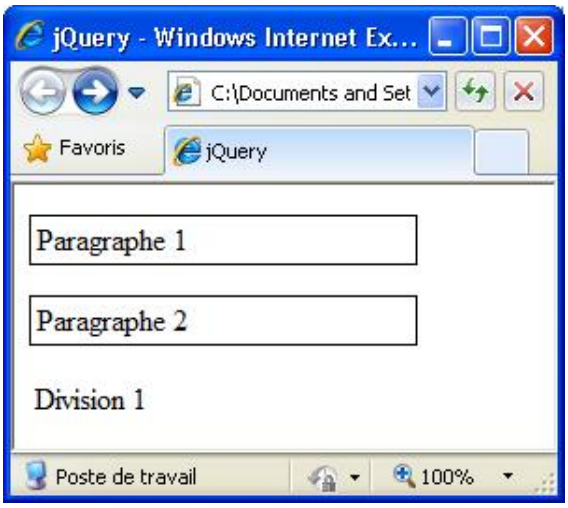
Exemple

Entourons d'une bordure tous les paragraphes du document Xhtml.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js" ></script>
<script type="text/javascript">
$(document).ready(function(){
$("p").css("border","1px solid black");
});
</script>
<style type="text/css">
p { width: 200px;
    padding: 3px;}
div { padding: 3px;}
```


© ENI Editions - All rigths reserved - Jonifar lina

```
</style>
</head>
<body>
<p>Paragraphe 1</p>
<p>Paragraphe 2</p>
<div>Division 1</div>
</body>
</html>
```




```
$("p").css("border","1px solid black");
```

jQuery sélectionne toutes les balises de paragraphe `<p>` (`$("p")`) et applique à celles-ci, une bordure par la méthode `css()`.

3. Sélection par la classe

.classe

Sélectionne tous les éléments (ou balises) avec la classe spécifiée.

`$(".texte")` : sélectionne tous les éléments dotés de l'attribut `class="texte"`.

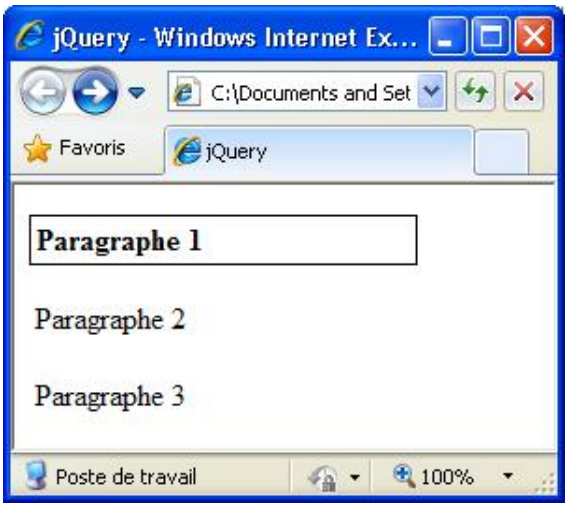
Exemple

Entourons d'une bordure le paragraphe doté de la classe `gras`.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js" ></script>
<script type="text/javascript">
$(document).ready(function(){
$(".gras").css("border","1px solid black");
});
</script>
<style type="text/css">
p { width: 200px;
    padding: 3px;}
.gras { font-weight: bold;}
</style>
</head>
<body>
<p class="gras">Paragraphe 1</p>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<p>Paragraphe 2</p>
<p>Paragraphe 3</p>
</body>
</html>
```




```
$(".gras").css("border","1px solid black");
```

La balise avec la classe `gras` est sélectionnée par jQuery (`$(".gras")`). Une bordure est alors appliquée.

Commentaires

On aurait pu noter : `$("p.gras").css("border","1px solid black")`. Ainsi, jQuery sélectionne les balises `<p>` avec la classe `gras`.

Selon les experts, cette notation serait plus efficace car jQuery peut retrouver directement les balises `<p>` dans le DOM et ensuite filtrer celles qui possèdent une classe `gras`.

La notation avec plusieurs classes est envisageable. `$(".classe1.classe2")` sélectionne tous les éléments qui ont comme nom de classe `classe1` et `classe2`.

Le sélecteur étoile * permet de sélectionner tous les éléments.

```
$("*").css("border","1px solid black");
```

jQuery permet d'associer plusieurs sélecteurs.

```
$("div,span,p.nom_classe").css("border","1px solid black");
```


© ENI Editions - All rigths reserved - Jonifar lina

Les sélecteurs hiérarchiques

La notation DOM avec ses parents, descendants, enfants, frères et sœurs (siblings) est maintenant bien ancrée dans l'écriture du JavaScript. La librairie jQuery ne pouvait ignorer cette façon de procéder.

1. Sélection des descendants

Ascendant Descendant

Sélectionne tous les descendants de l'élément noté "Descendant" par rapport à l'élément parent noté "Ascendant".

`$("#box p")` : sélectionne tous les descendants de `<p>` contenus dans l'élément parent identifié par `box`.

Les descendants peuvent être les enfants, les petits-enfants et les arrière-petits-enfants.

Exemple

Soit des divisions qui contiennent divers éléments. Retrouvons tous les descendants de l'élément identifié par `id="box"`.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js" ></script>
<script type="text/javascript">
$(document).ready(function(){
$("#box *").css("border", "1px double black");
});
</script>
<style type="text/css">
span#box { display: block;}
div { width: 80px;
      height: 100px;
      margin: 5px;
      float: left;
      background: #9cf; }
p { text-align: center;}
</style>
</head>
<body>
<span id="box">
<div></div>
<div><p>X <img src="jquery.png" alt="" /> X</p></div>
<div><p>Par<br /><span>a</span><br />graphe</p></div>
</span>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
$("#box *").css("border", "1px double black");
```

Le script jQuery sélectionne tous les descendants de l'élément avec un identifiant box (`$("#box *")`) et entoure ceux-ci d'une bordure.

Il faut remarquer que sont entourés d'une bordure, non seulement les enfants soit les divisions `<div>`, mais aussi les petits-enfants soit les balises `<p>` et les arrière-petits-enfants soit les balises `<img>` et `<span>`.

2. Sélection des enfants

Parent > Enfant

Sélectionne tous les éléments notés par "Enfant" qui sont les enfants directs de l'élément parent noté "Parent".

`$("#box > p")` : sélectionne tous les enfants immédiats de `<p>` contenus dans l'élément parent identifié par `box`.

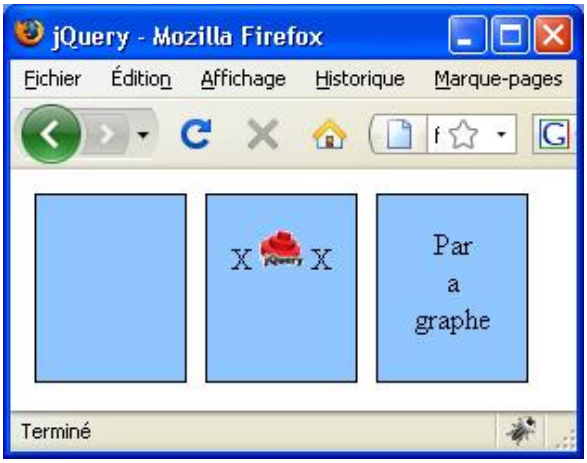
Exemple

Reprenons l'agencement de l'exemple précédent. Retrouvons les enfants (et uniquement les enfants) de l'élément identifié par `id="box"`.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js" ></script>
<script type="text/javascript">
$(document).ready(function(){
$("#box > *").css("border", "1px double black");
});
</script>
<style type="text/css">
span#box { display: block;}
div { width: 80px;
      height: 100px;
      margin: 5px;
      float: left;
      background: #9cf; }
p { text-align: center;}
</style>
</head>
<body>
<span id="box">
<div></div>
<div><p>X <img src="jquery.png" alt="" /> X</p></div>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<div><p>Par<br /><span>a</span><br />graphe</p></div>
</span>
</body>
</html>
```




```
$("#box > *").css("border", "1px double black");
```

Le script jQuery sélectionne tous les enfants directs de l'élément avec un identifiant box (`$("#box > *")`) et entoure ceux-ci d'une bordure.

Il faut remarquer que les petits-enfants `<p>` et arrière-petits-enfants `<img>` ou `<span>` ne sont plus entourés par une bordure.

On aurait pu écrire :

```
$("#box > div").css("border", "1px double black");
```

3. Sélection des frères suivants

Précédent ~ Frères

Cible les éléments frères situés après l'élément identifié par le sélecteur "Précédent" et qui répondent au sélecteur "Frères".

`$("#prev ~ div")` trouve toutes les divisions `<div>` qui sont frères après l'élément avec `#prev` comme identifiant.

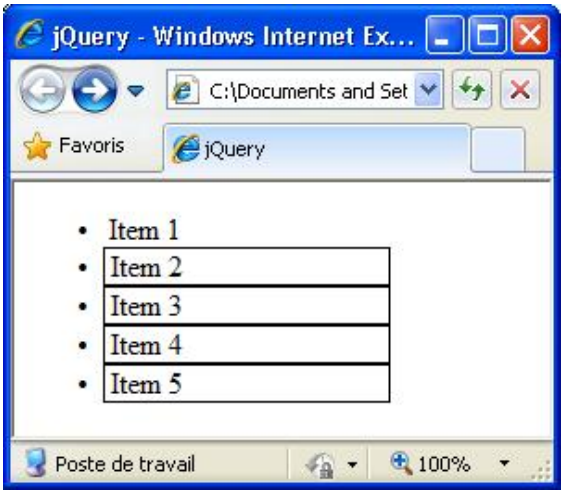
Exemple

Soit une liste non ordonnée. Retrouvons les éléments frères du premier item de la liste.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js" ></script>
<script type="text/javascript">
$(document).ready(function(){
$("li.un ~ li").css("border", "1px double black");
});
</script>
<style type="text/css">
li { width: 150px;
     padding-left: 3px;}
</style>
</head>
<body>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<ul>
<li class="un">Item 1</li>
<li class="deux">Item 2</li>
<li class="trois">Item 3</li>
<li class="quatre">Item 4</li>
<li class="cinq">Item 5</li>
</ul>
</body>
</html>
```




```
$("li.un ~ li").css("border", "1px double black");
```

Le script sélectionne tous les frères de l'élément de liste `<li>` doté de la classe un (`$("li.un ~ li")`). Tous les autres items de la liste sont ainsi repris.

4. Sélection de l'élément suivant

Précédent + Suivant

Cible l'élément immédiatement suivant situé après l'élément identifié par le sélecteur "Précédent" et qui répond au sélecteur "Suivant".

`$("#prev + div")` trouve la division `<div>` qui suit l'élément avec `#prev` comme identifiant.

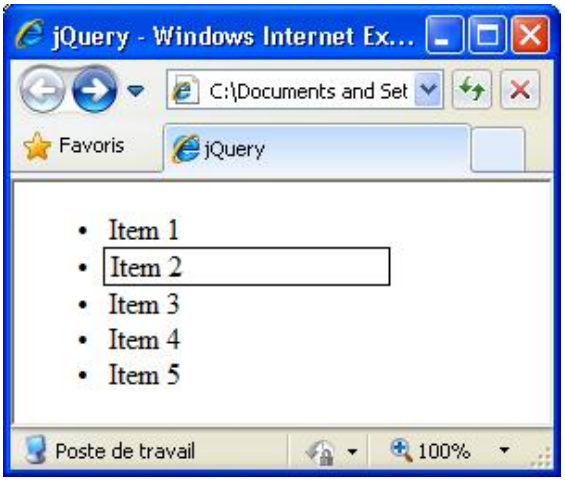
Exemple

Soit une liste non ordonnée. Retrouvons l'élément suivant du premier item de la liste.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js" ></script>
<script type="text/javascript">
$(document).ready(function(){
$("li.un + li").css("border", "1px double black");
});
</script>
<style type="text/css">
li { width: 150px;
     padding-left: 3px;}
</style>
</head>
<body>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<ul>
<li class="un">Item 1</li>
<li class="deux">Item 2</li>
<li class="trois">Item 3</li>
<li class="quatre">Item 4</li>
<li class="cinq">Item 5</li>
</ul>
</body>
</html>
```




```
$("li.un + li").css("border", "1px double black");
```

Trouve le frère immédiat parmi les frères de l'élément de liste `<li>` doté de la classe un (`$("li.un + li")`).

© ENI Editions - All rigths reserved - Jonifar lina

Les filtres jQuery de base

Tous les éléments du DOM étant répertoriés par jQuery, il devient facile de filtrer des éléments déterminés comme le premier, le dernier, etc.

1. Le premier élément

:first

Sélectionne la première instance d'un élément.

`$("li:first")` : sélectionne le premier élément de liste `<li>` du document.

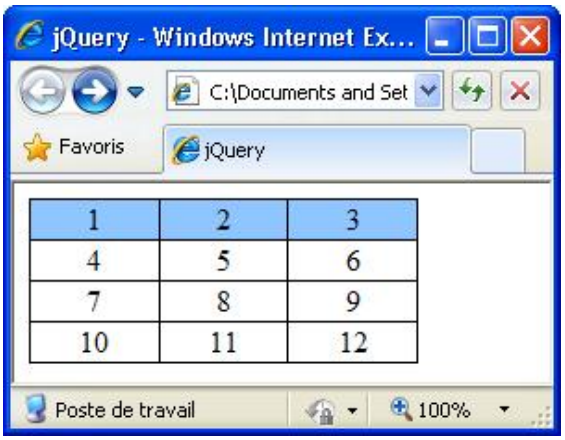
Le sélecteur `:first` sélectionne un seul élément tandis que `:first-child` (voir Les filtres enfants - Le premier élément du présent chapitre) peut en sélectionner plusieurs, soit un pour chaque parent.

Exemple

Mettre un arrière-plan de couleur à la première ligne d'un tableau.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("tr:first").css("background", "#9cf");
});
</script>
<style type="text/css">
table { width: 210px;
        border-collapse: collapse;
  border: 1px solid black;}
td { text-align: center;
     border: 1px solid black;}
</style>
</head>
<body>
<table>
<tr><td>1</td><td>2</td><td>3</td></tr>
<tr><td>4</td><td>5</td><td>6</td></tr>
<tr><td>7</td><td>8</td><td>9</td></tr>
<tr><td>10</td><td>11</td><td>12</td></tr>
</table>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina




```
$("tr:first").css("background", "#9cf");
```

`$("tr:first")` sélectionne la première balise `<tr>`.

2. Le dernier élément

:last

Sélectionne la dernière instance d'un élément.

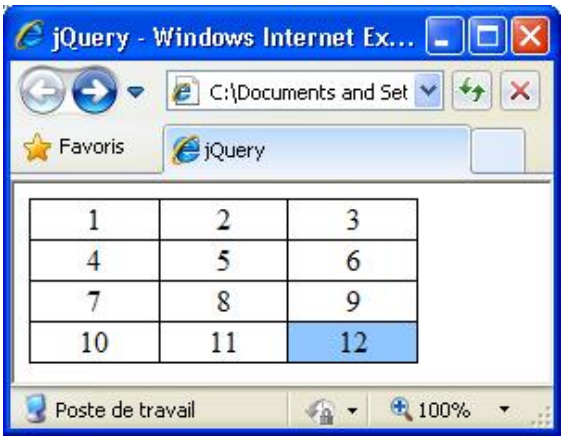
`$("li:last")` : sélectionne le dernier élément de liste `<li>` du document.

Exemple

Mettre un arrière-plan de couleur à la dernière cellule d'un tableau.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("td:last").css("background", "#9cf");
});
</script>
<style type="text/css">
table { width: 210px;
        border-collapse: collapse;
        border: 1px solid black;}
td { text-align: center;
     border: 1px solid black;}
</style>
</head>
<body>
<table>
<tr><td>1</td><td>2</td><td>3</td></tr>
<tr><td>4</td><td>5</td><td>6</td></tr>
<tr><td>7</td><td>8</td><td>9</td></tr>
<tr><td>10</td><td>11</td><td>12</td></tr>
</table>
</body>
</html>
```


© ENI Editions - All rigths reserved - Jonifar lina




```
$("td:last").css("background", "#9cf");
```

`$("td:last")` : le script sélectionne la dernière balise `<td>`.

3. Les éléments pairs

:even

Sélectionne les éléments pairs selon un index commençant à 0.

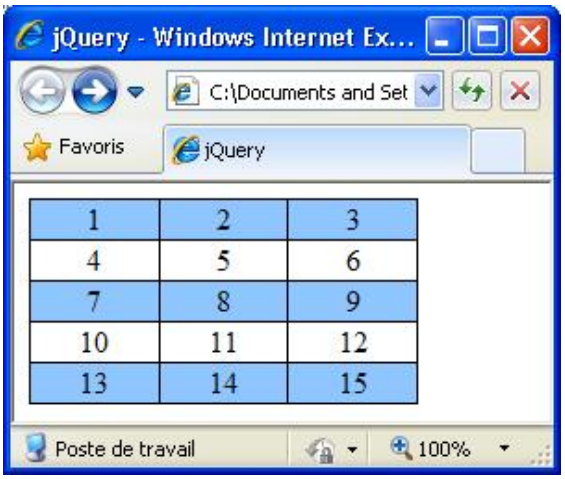
`$("tr:even")` : sélectionne les lignes d'index JavaScript 0, 2, 4 soit les lignes 1, 3, 5 à l'écran.

Exemple

Appliquons un effet listing à un tableau en dotant une ligne sur deux d'un arrière-plan.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("tr:even").css("background", "#9cf");
});
</script>
<style type="text/css">
table { width: 210px;
        border-collapse: collapse;
        border: 1px solid black;}
td { text-align: center;
     border: 1px solid black;}
</style>
</head>
<body>
<table>
<tr><td>1</td><td>2</td><td>3</td></tr>
<tr><td>4</td><td>5</td><td>6</td></tr>
<tr><td>7</td><td>8</td><td>9</td></tr>
<tr><td>10</td><td>11</td><td>12</td></tr>
<tr><td>13</td><td>14</td><td>15</td></tr>
</table>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina




```
$("tr:even").css("background", "#9cf");
```

jQuery sélectionne les balises `<tr>` paires et les agrémente d'un arrière-plan de couleur.

Cet effet qui réclamerait de nombreuses lignes de code en JavaScript classique, se programme en jQuery en une seule ligne. Cet exemple illustre bien la concision du code généré en jQuery.

---

4. Les éléments impairs

:odd

Sélectionne les éléments impairs selon un index commençant à 0.

`$("tr:odd")` : sélectionne les cellules d'index JavaScript 1, 3, 5 soit les cellules 2, 4, 6, … à l'écran.

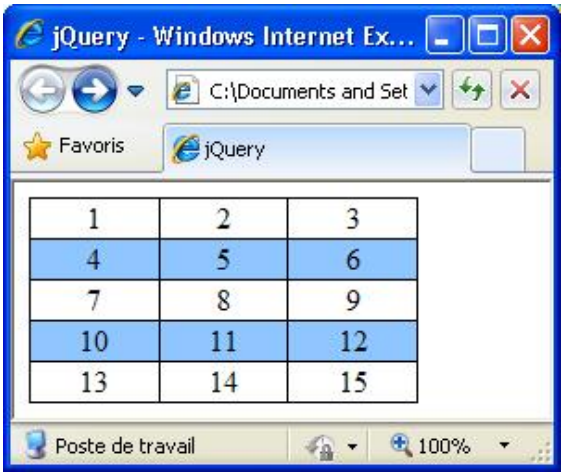
Exemple

Appliquons le même effet listing à un tableau en dotant une ligne sur deux d'un arrière-plan mais avec d'autres lignes qu'à l'exemple précédent.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("tr:odd").css("background", "#9cf");
});
</script>
<style type="text/css">
table { width: 210px;
        border-collapse: collapse;
        border: 1px solid black;}
td { text-align: center;
     border: 1px solid black;}
</style>
</head>
<body>
<table>
<tr><td>1</td><td>2</td><td>3</td></tr>
<tr><td>4</td><td>5</td><td>6</td></tr>
<tr><td>7</td><td>8</td><td>9</td></tr>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<tr><td>10</td><td>11</td><td>12</td></tr>
<tr><td>13</td><td>14</td><td>15</td></tr>
</table>
</body>
</html>
```




```
$("tr:odd").css("background", "#9cf");
```

jQuery sélectionne les balises `<tr>` impaires et les agrémente d'un arrière-plan de couleur.

5. Un élément déterminé

:eq(index)

Sélectionne l'élément déterminé par la valeur de l'index.

Comme les index JavaScript débutent à zéro, le sélecteur `:eq(0)` sélectionne donc le premier élément, `:eq(1)` le second élément et ainsi de suite.

`$("td:eq(2)")` : sélectionne la troisième cellule d'un tableau.

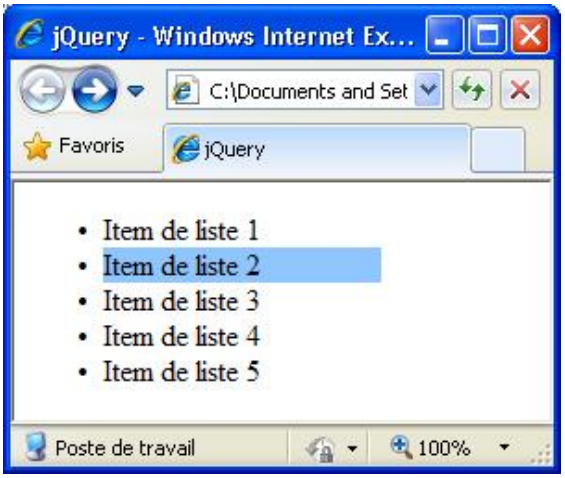
Exemple

Soit une liste numérotée. Retrouvons l'élément de liste qui apparaît en seconde position à l'écran.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("li:eq(1)").css("background", "#9cf");
});
</script>
<style type="text/css">
li { width: 150px;}
</style>
</head>
<body>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li>Item de liste 3</li>
<li>Item de liste 4</li>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<li>Item de liste 5</li>
</ul>
</body>
</html>
```




```
$("li:eq(1)").css("background", "#9cf");
```

Le second élément est accédé par `:eq(1)`. Il suffit de spécifier que c'est le second élément de liste (`$("li:eq(1)")`).

6. Les éléments suivants

:gt(index)

Sélectionne les éléments avec une valeur d'index supérieure (greater than) à la valeur fournie en paramètre.

Pour rappel, les index JavaScript débutent à 0.

`$("a:gt(1)")` : sélectionne tous les liens de la page en commençant par le troisième (soit après le second élément).

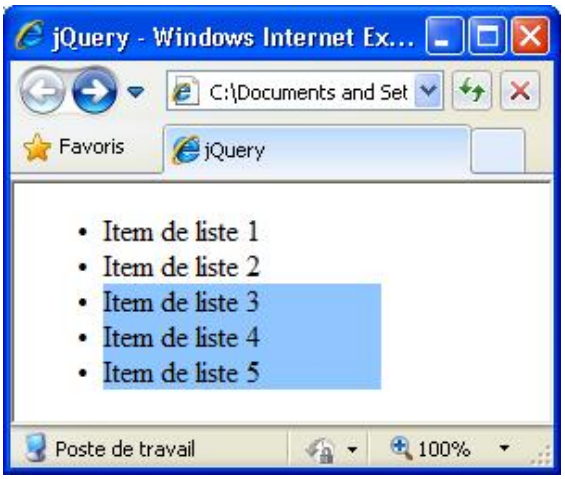
Exemple

Soit une liste numérotée. Sélectionnons les éléments de liste de l'item de liste 3 à la fin de celle-ci.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("li:gt(1)").css("background", "#9cf");
});
</script>
<style type="text/css">
li { width: 150px;}
</style>
</head>
<body>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li>Item de liste 3</li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
</body>
</html>
```




```
$("li:gt(1)").css("background", "#9cf");
```

`("li:gt(1)")` : tous les éléments de la liste qui suivent l'item 2 sont sélectionnés.

7. Les éléments précédents

:lt(index)

Sélectionne les éléments avec une valeur d'index inférieure (less than) à la valeur fournie en paramètre.

Par rappel, les index JavaScript débutent à 0.

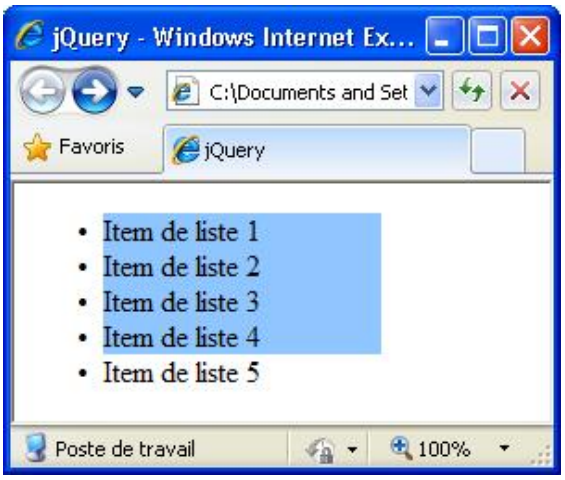
`$("p:lt(3)")` : sélectionne tous les paragraphes situés avant le quatrième, soit les trois premiers paragraphes.

Exemple

Soit une liste numérotée. Sélectionnons les quatre premiers éléments de celle-ci.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("li:lt(4)").css("background", "#9cf");
});
</script>
<style type="text/css">
li { width: 150px;}
</style>
</head>
<body>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li>Item de liste 3</li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina




```
$("li:lt(4)").css("background", "#9cf");
```

`$("li:lt(4)")` : tous les éléments de liste qui précèdent l'item 5 sont sélectionnés.

8. Les balises de titre

:header

Retourne tous les éléments qui sont des balises de titre comme `<h1>`, `<h2>`, `<h3>`, etc.

`$(":header")` : sélectionne toutes les balises de titre de la page.

Exemple

Sélectionnons toutes les balises de titre de la page.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$(":header").css("background", "#9cf");
});
</script>
<style type="text/css">
body { font-size: 10px; font-family: Arial; }
h1, h2, h3 { margin: 3px 0; }
</style>
</head>
<body>
<h1>Titre de niveau 1</h1>
<p>Contenu</p>
<h2>Titre de niveau 2</h2>
<p>Contenu</p>
<h3>Titre de niveau 3</h3>
<p>Contenu</p>
</body>
</html>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
$(":header").css("background", "#9cf");
```

`$(":header")` : sélectionne toutes les balises de titre de la page.

9. Exclusion d'un élément

:not(sélecteur)

Exclut de la sélection tous les éléments qui répondent au critère spécifié par le sélecteur.

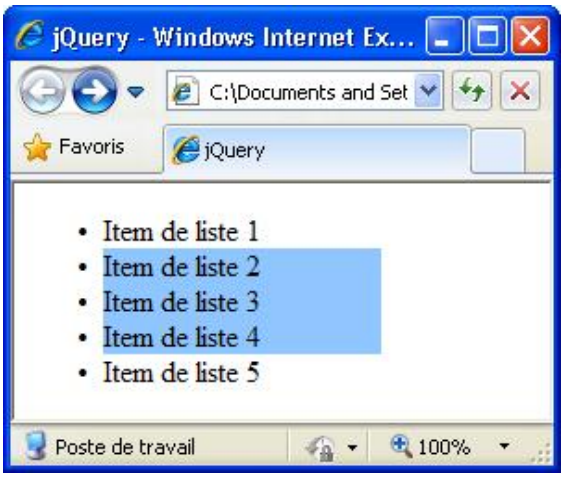
`$("div:not(#box)")` : sélectionne toutes les divisions sauf celle identifiée par box.

Exemple

Sélectionnons tous les éléments d'une liste à l'exclusion de premier et dernier item.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("li:not(:first, :last)").css("background", "#9cf");
});
</script>
<style type="text/css">
li { width: 150px;}
</style>
</head>
<body>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li>Item de liste 3</li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina




```
$("li:not(:first, :last)").css("background", "#9cf");
```

Sélectionne tous les éléments de liste `<li>` sauf la première et dernière occurrence de ceux-ci.

Il faut être attentif à l'ordre et au placement des parenthèses ouvrantes et surtout fermantes.

© ENI Editions - All rigths reserved - Jonifar lina

Les filtres enfants

Dans ce chapitre, nous avons déjà abordé les sélecteurs hiérarchiques qui permettaient de sélectionner les enfants. Les filtres enfants permettent d'effectuer un tri parmi les enfants d'un élément.

1. Le premier enfant

:first-child

Sélectionne tous les éléments qui sont le premier enfant de leur parent.

`$("ul:first-child")` : sélectionne le premier enfant (soit le premier élément de liste) de la liste non ordonnée `<ul>`.

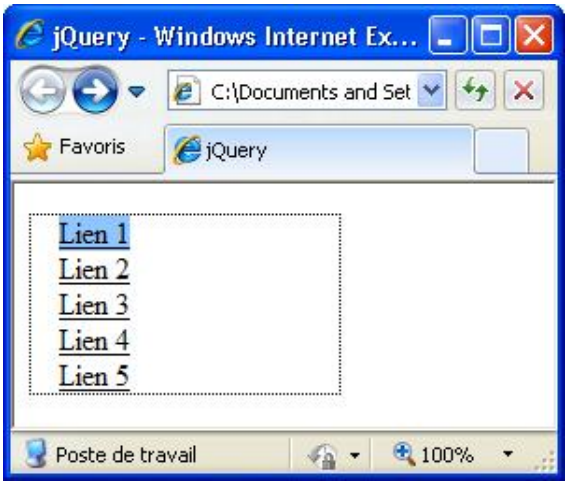
Le sélecteur `:first-child` peut sélectionner plusieurs éléments soit un pour chaque parent. À ne pas confondre avec `:first` qui ne reprend qu'un seul élément (voir la section Les filtres jQuery de base - Le premier élément du présent chapitre).

Exemple

Retrouvons le premier lien inclus dans un paragraphe.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("p a:first-child").css("background", "#9cf");
});
</script>
<style type="text/css">
a { color: black;}
p { width: 150px;
    border: 1px dotted black;
    padding-left: 15px;}
</style>
</head>
<body>
<p>
<a href="#">Lien 1</a><br />
<a href="#">Lien 2</a><br />
<a href="#">Lien 3</a><br />
<a href="#">Lien 4</a><br />
<a href="#">Lien 5</a>
</p>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina




```
$("p a:first-child").css("background", "#9cf");
```

`$("p a:first-child")` : le premier lien <a>, enfant de la balise <p> est sélectionné.

2. Le dernier enfant

:last-child

Sélectionne tous les éléments qui sont le dernier enfant de leur parent.

`$("ul:last-child")` : sélectionne le dernier enfant (soit le dernier élément de liste) de la liste non ordonnée <ul>.

Le sélecteur `:last-child` peut sélectionner plusieurs éléments soit un pour chaque parent. À ne pas confondre avec `:last` qui ne reprend qu'un seul élément (voir la section Les filtres jQuery de base - Le dernier élément du présent chapitre).

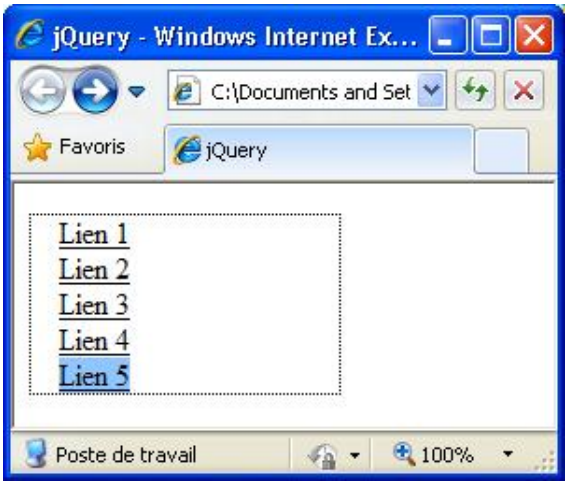
Exemple

Retrouvons le dernier lien inclus dans un paragraphe.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("p a:last-child").css("background", "#9cf");
});
</script>
<style type="text/css">
a { color: black;}
p { width: 150px;
    border: 1px dotted black;
    padding-left: 15px;}
</style>
</head>
<body>
<p>
<a href="#">Lien 1</a><br />
<a href="#">Lien 2</a><br />
<a href="#">Lien 3</a><br />
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<a href="#">Lien 4</a><br />
<a href="#">Lien 5</a>
</p>
</body>
</html>
```




```
$("p a:last-child").css("background", "#9cf");
```

`$("p a:last-child")` : le dernier lien `<a>`, enfant de la balise `<p>` est sélectionné.

3. Le énième enfant

:nth-child(index)

Sélectionne les éléments qui sont le énième enfant de leur parent. La position est fournie par le paramètre index.

Au contraire de nombreux index en jQuery, l'index ne débute pas ici à 0 mais à 1.

`$("ul li:nth-child(2)")` : sélectionne le second élément `<li>` de la liste `<ul>`.

Revenons à cette fameuse exception concernant l'index. La plupart des index utilisés par jQuery font appel à des fonctions natives de JavaScript et respectent la convention de débuter les index à 0. Le sélecteur `:nth-child`, sélecteur spécifique à jQuery, est strictement dérivé des spécifications CSS. La valeur d'index 1 signifie donc bien qu'il s'agit du premier élément.

La confusion avec `:eq(index)`, abordé à la section Les filtres jQuery de base - Un élément déterminé du présent chapitre, dont l'index commence à 0, peut être une source d'erreur difficilement décelable dans certains scripts jQuery.

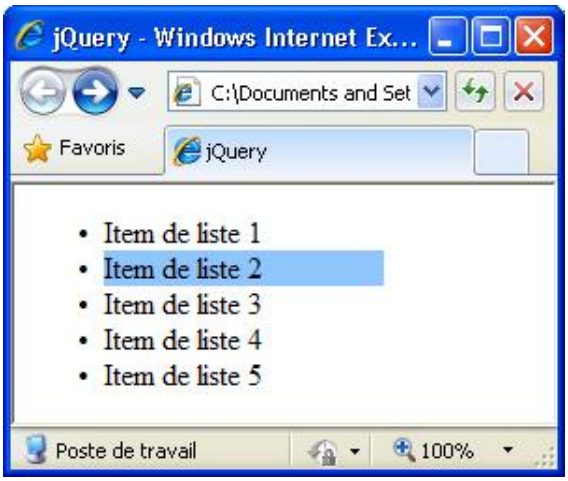
Exemple

Retrouvons l'élément de liste qui apparaît en seconde position à l'écran.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("li:nth-child(2)").css("background", "#9cf");
});
</script>
<style type="text/css">
li { width: 150px;}
```

© ENI Editions - All rigths reserved - Jonifar lina

```
</style>
</head>
<body>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li>Item de liste 3</li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
</body>
</html>
```




```
$("li:nth-child(2)").css("background", "#9cf");
```

Le script sélectionne tous les éléments de liste `<li>` qui sont en seconde position dans l'ordre des enfants.

Avec le filtre `:eq()`, nous aurions utilisé la notation :

```
$("li:eq(1)").css("background", "#9cf");
```

4. Les enfants pairs et impairs

Variante du sélecteur précédent. Le sélecteur `nth-child` peut sélectionner les éléments pairs et impairs.

:nth-child(even ou odd)

Sélectionne les éléments qui sont les nièmes enfants pairs (even) ou impairs (odd) de leur parent.

Comme `:nth-child` dont il est une variante, l'index débute à 1.

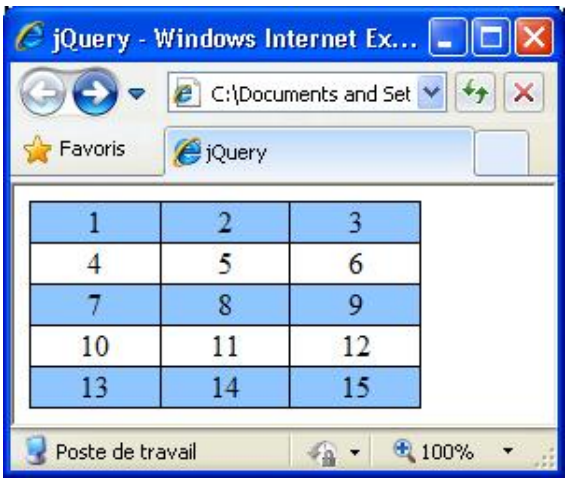
`$("table tr:nth-child(odd)")` : sélectionne les lignes impaires `<tr>` du tableau.

Ici aussi la confusion avec `:even` et `:odd` (voir les sections Les filtres jQuery de base - Les éléments pairs et Les filtres jQuery de base - Les éléments impairs du présent chapitre) est possible car l'index de ceux-ci démarre à 0.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("table tr:nth-child(odd)").css("background", "#9cf");});
</script>
<style type="text/css">
```


© ENI Editions - All rigths reserved - Jonifar lina

```
table { width: 210px;
        border-collapse: collapse;
        border: 1px solid black;}
td { text-align: center;
     border: 1px solid black;}
</style>
</head>
<body>
<table>
<tr><td>1</td><td>2</td><td>3</td></tr>
<tr><td>4</td><td>5</td><td>6</td></tr>
<tr><td>7</td><td>8</td><td>9</td></tr>
<tr><td>10</td><td>11</td><td>12</td></tr>
<tr><td>13</td><td>14</td><td>15</td></tr>
</table>
</body>
</html>
```




```
$("table tr:nth-child(odd)").css("background", "#9cf");});
```

Le script sélectionne tous les éléments de ligne `<tr>` qui sont en rang impair dans l'ordre des enfants de la balise `<table>`.

5. Les enfants uniques

:only-child

Sélectionne tous les éléments qui sont enfant unique de leur parent. Si le parent a plusieurs enfants, aucune sélection n'est effectuée.

`$("div button:only-child")` : retrouve les boutons (balise `<button>`) qui n'ont pas de frère(s) dans toutes les divisions rencontrées.

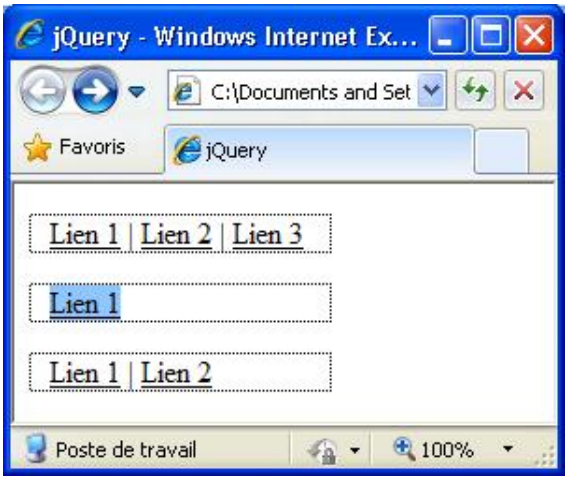
Exemple

Retrouver le lien qui est enfant unique d'une balise de paragraphe <p>.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
```

© ENI Editions - All rigths reserved - Jonifar lina

```
$("p a:only-child").css("background", "#9cf");
});
</script>
<style type="text/css">
a { color: black;}
p { width: 150px;
    border: 1px dotted black;
    padding-left: 10px;
    margin-bottom: 0px;}
</style>
</head>
<body>
<p><a href="#">Lien 1</a> | <a href="#">Lien 2</a> | <a
href="#">Lien 3</a><br /></p>
<p><a href="#">Lien 1</a></p>
<p><a href="#">Lien 1</a> | <a href="#">Lien 2</a></p>
</body>
</html>
```




```
$("p a:only-child").css("background", "#9cf");
```

`$("p a:only-child")` : jQuery passe en revue les différents paragraphes. Examine les liens `<a>` présents dans ceux-ci et ne retient que le paragraphe qui ne comporte qu'un seul lien.

© ENI Editions - All rigths reserved - Jonifar lina

Les filtres de contenu

1. Un texte donné

:contains(texte)

Sélectionne les éléments qui contiennent un texte ou un fragment de texte fourni en argument.

Il faut noter que l'argument texte est sensible à la casse (*case sensitive*).

`$("div:contains('Eni')")` : sélectionne les divisions qui contiennent le texte "Eni".

Exemple

Mettons en évidence les paragraphes qui contiennent le fragment de texte "En".

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("p:contains(En)").css("background", "#9cf");
});
</script>
<style type="text/css">
a { color: black;}
p { width: 150px;
    border: 1px dotted black;
    padding-left: 10px;
    margin-bottom: 0px;}
</style>
</head>
<body>
<p>Editions Eni</p>
<p>Spécialisé en informatique</p>
<p>En Bretagne</p>
</body>
</html>
```

```
$("p:contains(En)").css("background", "#9cf");
```

© ENI Editions - All rigths reserved - Jonifar lina

`$("p:contains(En)")` : met en évidence les paragraphes contenant les lettres "En".

2. Un contenu vide

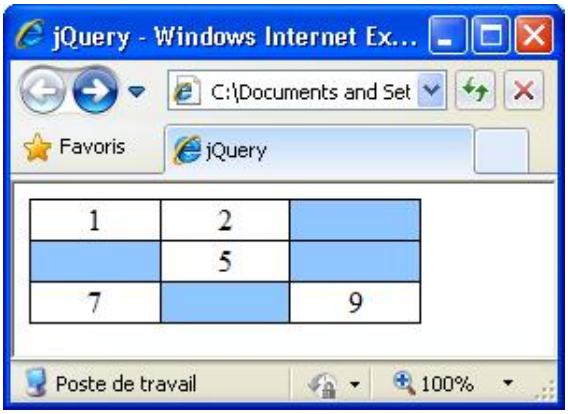
:empty

Sélectionne tous les éléments qui n'ont pas d'enfants (inclus les nœuds de texte).

`$("div:empty")` : sélectionne les divisions vides.

Exemple

Retrouvons les cellules vides d'un tableau.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("td:empty").css("background", "#9cf");});
</script>
<style type="text/css">
table { width: 210px;
        border-collapse: collapse;
        border: 1px solid black;}
td { text-align: center;
     border: 1px solid black;}
</style>
</head>
<body>
<table>
<tr><td>1</td><td>2</td><td></td></tr>
<tr><td></td><td>5</td><td></td></tr>
<tr><td>7</td><td></td><td>9</td></tr>
</table>
</body>
</html>
```

`$("td:empty").css("background", "#9cf");});`

`$("td:empty")` : retrouve les cellules de tableau `<td>` vides.

© ENI Editions - All rigths reserved - Jonifar lina

3. La qualité de parent

:parent

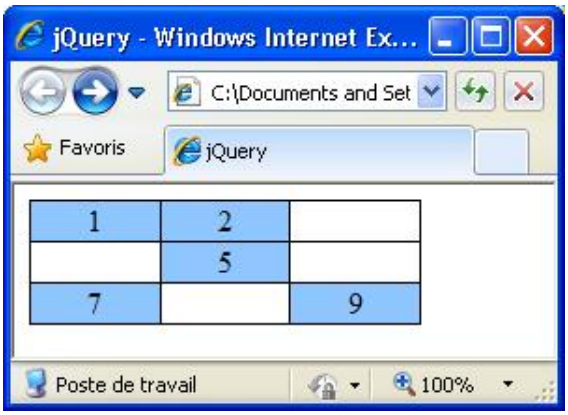
Sélectionne les éléments qui sont parent, c'est-à-dire qui ont des éléments enfants, inclus les nœuds de texte.

`$("div:parent")` : sélectionne les divisions qui ont des éléments enfants.

Ce sélecteur est en quelque sorte l'inverse du précédent.

Exemple

Reprenons notre tableau et retrouvons les cellules non vides.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("td:parent").css("background", "#9cf");});
</script>
<style type="text/css">
table { width: 210px;
        border-collapse: collapse;
        border: 1px solid black;}
td { text-align: center;
     border: 1px solid black;}
</style>
</head>
<body>
<table>
<tr><td>1</td><td>2</td><td></td></tr>
<tr><td></td><td>5</td><td></td></tr>
<tr><td>7</td><td></td><td>9</td></tr>
</table>
</body>
</html>
```

`$("td:parent").css("background", "#9cf");});`

`$("td:parent")` : retrouve les cellules de tableau `<td>` qui ont un contenu et qui sont donc parent.

4. Un sélecteur déterminé

© ENI Editions - All rigths reserved - Jonifar lina

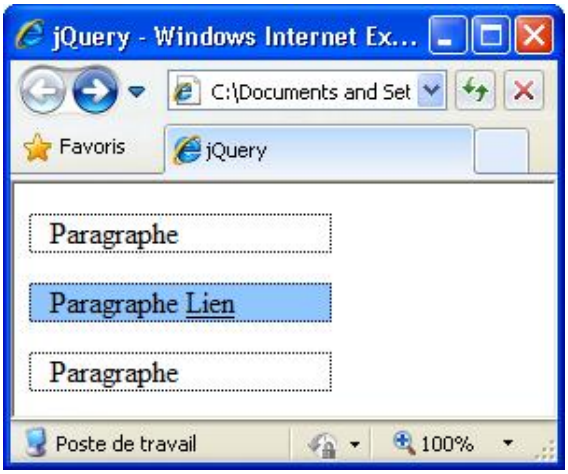
:has(sélecteur)

Sélectionne les éléments qui contiennent le sélecteur transmis en argument.

`$("div:has(p)")` : sélectionne les divisions qui contiennent un ou des paragraphe(s).

Exemple

Retrouvons le paragraphe qui comporte un lien.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("p:has(a)").css("background", "#9cf");
});
</script>
<style type="text/css">
a { color: black;}
p { width: 150px;
    border: 1px dotted black;
    padding-left: 10px;
    margin-bottom: 0px;}
</style>
</head>
<body>
<p>Paragraphe</p>
<p>Paragraphe <a href="#">Lien</a></p>
<p>Paragraphe</p>
</body>
</html>
```

`$("p:has(a)").css("background", "#9cf");`

`$("p:has(a)")` : retrouve le paragraphe qui possède un lien.

© ENI Editions - All rigths reserved - Jonifar lina

Les filtres de visibilité

1. Élément visible

:visible

Sélectionne les éléments qui sont visibles.

`$("p:visible")` : sélectionne les paragraphes visibles.

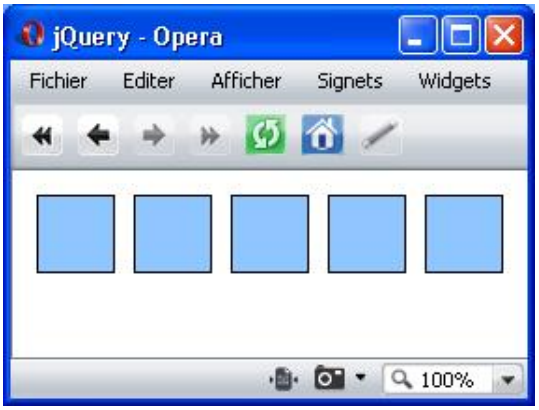
Pour ce filtre jQuery, un élément est considéré comme visible s'il consomme de l'espace dans le document. Les propriétés CSS de visibilité ne sont pas prises en compte.

Exemple

Dotons les divisions visibles d'un arrière-plan de couleur.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js" ></script>
<script type="text/javascript">
$(document).ready(function(){
$("div:visible").css("background", "#9cf");
});
</script>
<style type="text/css">
.hidden { display:none;}
div { width: 40px;
      height: 40px;
      margin: 5px;
      float: left;
      border: 1px solid black;}
p { text-align: center;}
</style>
</head>
<body>
<div></div>
<div></div>
<div></div>
<div></div>
<div></div>
<div class="hidden"></div>
<div class="hidden"></div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina




```
$("div:visible").css("background", "#9cf");
```

`$("div:visible")` : uniquement les divisions visibles sont sélectionnées.

2. Elément caché

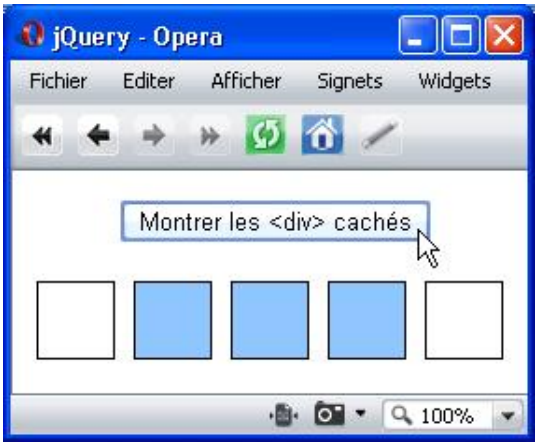
:hidden

Sélectionne les éléments qui sont cachés.

`$("p:hidden")` : sélectionne les paragraphes cachés.

Exemple

Ce script va afficher, au clic du bouton, les divisions cachées.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```


© ENI Editions - All rigths reserved - Jonifar lina

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js" ></script>
<script type="text/javascript">
$(document).ready(function(){
$("button").click(function () {
$("div:hidden").css("background", "#9cf").show();
});
});
</script>
<style type="text/css">
button {margin-left: 50px;}
.hidden { display:none; }
div { width: 40px;
      height: 40px;
      margin: 5px;
      float: left;
      border: 1px solid black;}
</style>
</head>
<body>
<p><button>Montrer les &lt;div&gt; cachés</button></p>
<div class="box"></div>
<div class="hidden"></div>
<div class="hidden"></div>
<div class="hidden"></div>
<div class="box"></div>
</body>
</html>
```

Détaillons brièvement le script.

```
$("button").click(function () {
$("div:hidden").show("fast");
});
```

Au clic du bouton (`$("button").click(function())`, les divisions cachées (`("div:hidden")`) sont affichées (`show("fast")`).

© ENI Editions - All rigths reserved - Jonifar lina

Les filtres d'attribut

Les attributs sont nombreux dans le langage Html et Xhtml. Citons `title`, `alt`, `src`, `href`, `width`, `style`, etc.

Dans certaines documentations disponibles sur Internet, vous êtes susceptibles de retrouver la notation `[@attr]` relative aux filtres d'attribut. Cette notation a été retirée de la version jQuery 1.3. Il suffit simplement de retirer le signe @ des sélecteurs pour la rendre compatible avec la spécification 1.3.

1. L'attribut

[attribut]

Sélectionne les éléments dotés de l'attribut spécifié.

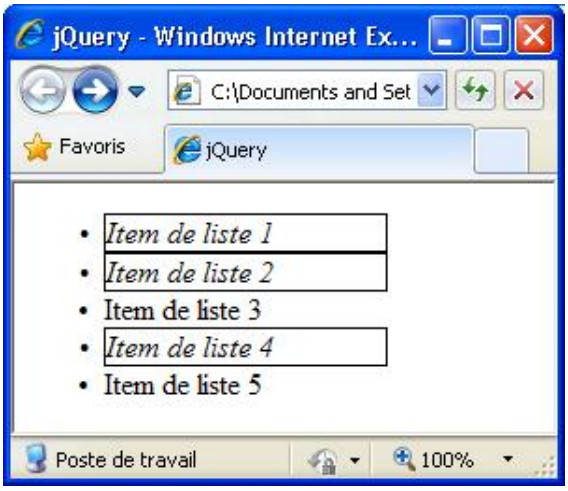
`$("div[id]")` : sélectionne les éléments qui ont un attribut `id`.

Exemple

Retrouvons les éléments de liste avec l'attribut `class`.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("li[class]").css("border", "1px solid black");
});
</script>
<style type="text/css">
li { width: 150px;}
.italique { font-style: italic; }
</style>
</head>
<body>
<ul>
<li class="italique">Item de liste 1</li>
<li class="italique">Item de liste 2</li>
<li>Item de liste 3</li>
<li class="italique">Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina




```
$("li[class]").css("border", "1px solid black");
```

`$("li[class]")` : parmi les éléments de liste `<li>`, jQuery sélectionne ceux qui possèdent un attribut de classe.

2. L'attribut avec une certaine valeur

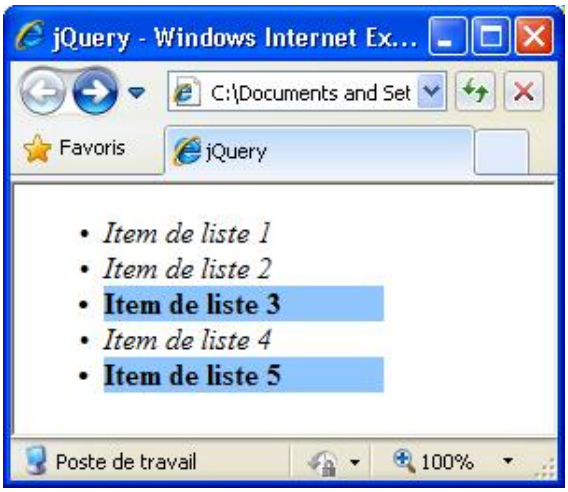
[attribut=valeur]

Sélectionne les éléments dotés d'un attribut avec une valeur déterminée.

`$("input[name='newsletter']")` : sélectionne l'élément de formulaire `<input>` avec un attribut `name="newsletter"`.

Exemple

Mettons en exergue les éléments de liste dotés de l'attribut de classe `class="gras"`.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("li[class='gras']").css("background", "#9cf");
});
```


© ENI Editions - All rigths reserved - Jonifar lina

```
</script>
<style type="text/css">
li { width: 150px;}
.italique { font-style: italic;}
.gras { font-weight: bolder;}
</style>
</head>
<body>
<ul>
<li class="italique">Item de liste 1</li>
<li class="italique">Item de liste 2</li>
<li class="gras">Item de liste 3</li>
<li class="italique">Item de liste 4</li>
<li class="gras">Item de liste 5</li>
</ul>
</body>
</html>
```

```
$("li[class='gras']").css("background", "#9cf");
```

`$("li[class='gras']")` : parmi les éléments de liste `<li>`, jQuery sélectionne ceux qui possèdent un attribut de classe `class="gras"`.

3. L'attribut qui n'a pas une certaine valeur

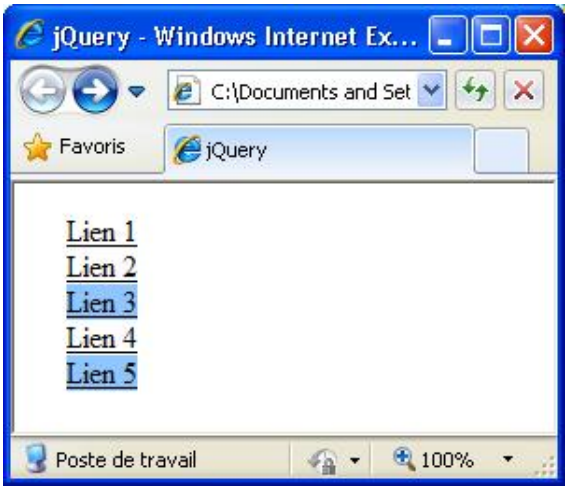
[attribut!=valeur]

Sélectionne les éléments qui n'ont pas l'attribut spécifié et ceux qui ont l'attribut spécifié mais pas avec la valeur indiquée. Valeur est case sensitive.

`$("input[name!=newsletter]")` : sélectionne les éléments de formulaire `<input>` qui n'ont pas d'attribut `name` et ceux qui ont un attribut `name` avec une autre valeur que `newsletter`.

Exemple

Retrouvons les liens qui n'ont pas l'attribut `title="lien interne"`.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
```

© ENI Editions - All rigths reserved - Jonifar lina

```
$(document).ready(function(){
$("a[title!='lien interne']").css("background", "#9cf");
});
</script>
<style type="text/css">
a { color: black;}
p { margin-left: 20px;}
</style>
</head>
<body>
<p>
<a href="#" title="lien interne">Lien 1</a><br />
<a href="#" title="lien interne">Lien 2</a><br />
<a href="#" title="lien externe">Lien 3</a><br />
<a href="#" title="lien interne">Lien 4</a><br />
<a href="#" title="lien externe">Lien 5</a><br />
</p>
</body>
</html>
```

`$("a[title!='lien interne']").css("background", "#9cf");`

`$("a[title!='lien interne']")` : parmi les liens `<a>`, retenons ceux qui n'ont pas l'attribut `title="lien interne"`.

4. L'attribut dont la valeur commence par

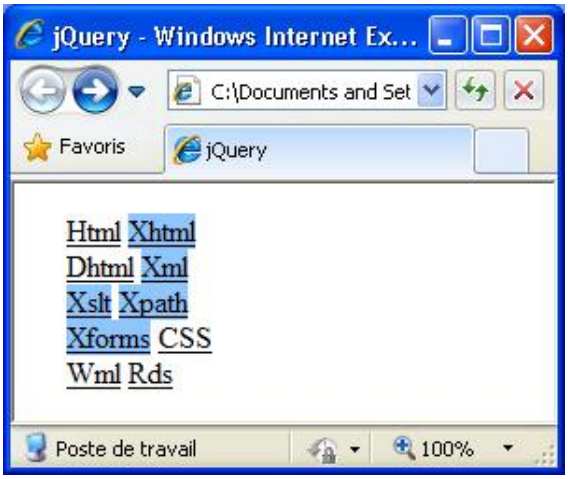
[attribut^=value]

Sélectionne les éléments qui possèdent l'attribut spécifié et dont la valeur commence par les caractères indiqués.

`$("input[name^='news']")` : sélectionne les éléments de formulaire `<input>` avec l'attribut `name` et dont la valeur de celui-ci commence par les caractères "news".

Exemple

Parmi les liens disponibles, retenons ceux dont l'attribut `title` commence par la lettre X.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
```


© ENI Editions - All rigths reserved - Jonifar lina

```
$("a[title^='X']").css("background", "#9cf");
});
</script>
<style type="text/css">
a { color: black;}
p { margin-left: 20px;}
</style>
</head>
<body>
<p>
<a href="#" title="Html">Html</a> <a href="#"
title="Xhtml">Xhtml</a><br />
<a href="#" title="Dhtml">Dhtml</a> <a href="#"
title="Xml">Xml</a><br />
<a href="#" title="Xslt">Xslt</a> <a href="#"
title="Xpath">Xpath</a><br />
<a href="#" title="Xforms">Xforms</a> <a href="#"
title="CSS">CSS</a><br />
<a href="#" title="Wml">Wml</a> <a href="#"
title="Rds">Rds</a></p>
</body>
</html>
```

`$("a[title^='X']").css("background", "#9cf");`

`$("a[title^='X']")` : parmi les liens `<a>`, retenons ceux dont l'attribut `title` commence par le lettre X.

5. L'attribut dont la valeur finit par

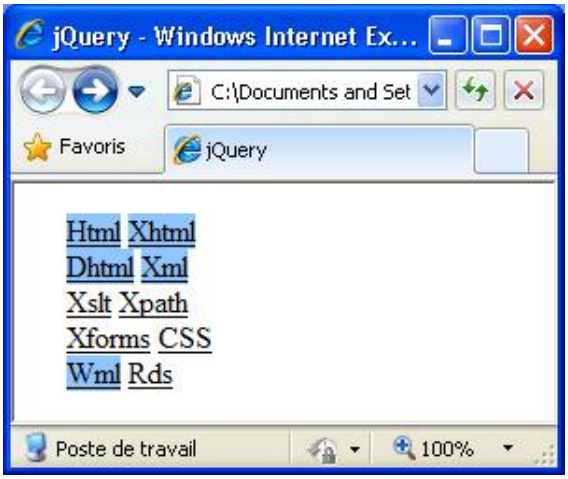
[attribut$=value]

Sélectionne les éléments qui possèdent l'attribut spécifié et dont la valeur se termine par les caractères indiqués.

`$("input[name$='letter']")` : sélectionne les éléments de formulaire `<input>` avec l'attribut `name` et dont la valeur de celui-ci se termine par les caractères "letter".

Exemple

En reprenant l'exemple précédent, retenons les liens dont l'attribut `title` se termine par les lettres "ml".




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("a[title$='ml']").css("background", "#9cf");
});
</script>
<style type="text/css">
a { color: black;}
p { margin-left: 20px;}
</style>
</head>
<body>
<p>
<a href="#" title="Html">Html</a> <a href="#"
title="Xhtml">Xhtml</a><br />
<a href="#" title="Dhtml">Dhtml</a> <a href="#"
title="Xml">Xml</a><br />
<a href="#" title="Xslt">Xslt</a> <a href="#"
title="Xpath">Xpath</a><br />
<a href="#" title="Xforms">Xforms</a> <a href="#"
title="CSS">CSS</a><br />
<a href="#" title="Wml">Wml</a> <a href="#"
title="Rds">Rds</a></p>
</body>
</html>
```

`$("a[title$='ml']").css("background", "#9cf");`

`$("a[title$='ml']")` : parmi les liens `<a>`, retenons ceux dont l'attribut `title` se termine par les lettres "ml".

6. L'attribut dont la valeur contient

[attribut*=value]

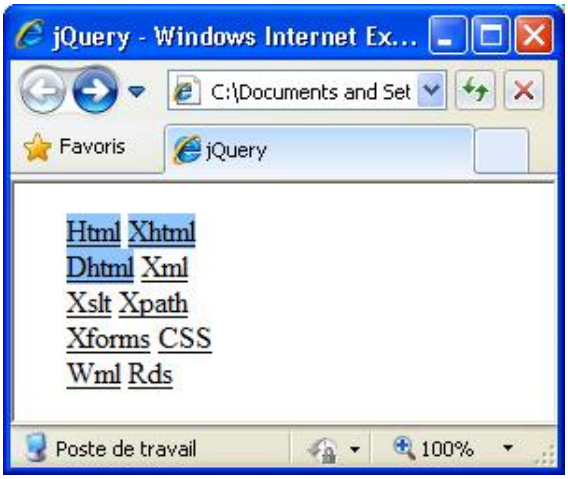
Sélectionne les éléments qui possèdent l'attribut spécifié et dont la valeur comporte les caractères indiqués.

`$("input[name*='slet']")` : sélectionne les éléments de formulaire `<input>` avec l'attribut `name` et dont la valeur de celui-ci comporte les caractères "slet".

La séquence (ou l'ordre) des lettres est respectée.

Exemple

Toujours en reprenant l'exemple précédent, retenons les liens dont l'attribut `title` comporte les lettres "tm".




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
```


© ENI Editions - All rigths reserved - Jonifar lina

```
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("a[title*='tm']").css("background", "#9cf");
});
</script>
<style type="text/css">
a { color: black;}
p { margin-left: 20px;}
</style>
</head>
<body>
<p>
<a href="#" title="Html">Html</a> <a href="#"
title="Xhtml">Xhtml</a><br />
<a href="#" title="Dhtml">Dhtml</a> <a href="#"
title="Xml">Xml</a><br />
<a href="#" title="Xslt">Xslt</a> <a href="#"
title="Xpath">Xpath</a><br />
<a href="#" title="Xforms">Xforms</a> <a href="#"
title="CSS">CSS</a><br />
<a href="#" title="Wml">Wml</a> <a href="#"
title="Rds">Rds</a></p>
</body>
</html>
```

`$("a[title*='tm']").css("background", "#9cf");`

`$("a[title*='tm']")` : parmi les liens `<a>`, retenons ceux dont l'attribut `title` comporte les lettres "tm".

7. Les filtres multiples d'attribut

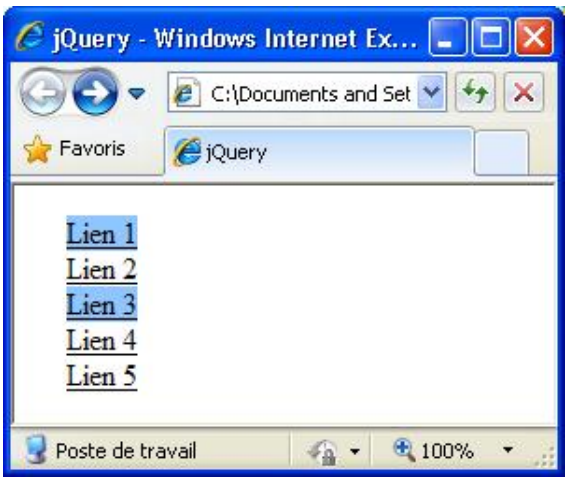
[filtre d'attribut 1][filtre d'attribut 2][filtre d'attribut 3]...

Sélectionne les éléments qui répondent à tous les filtres d'attribut spécifiés.

`$("input[id][name$='man']")` : sélectionne les balises `<input>` qui possèdent un identifiant `id` et dont l'attribut `name` se termine par "man".

Exemple

Retenons les liens dont l'attribut `title` commence par "lien", se termine par "interne" et comporte "chapitre 1".



© ENI Editions - All rigths reserved - Jonifar lina

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("a[title^='lien'][title$='interne'][title*='chapitre1']")
.css("background", "#9cf");
});
</script>
<style type="text/css">
a { color: black;}
p { margin-left: 20px;}
</style>
</head>
<body>
<p>
<a href="#" title="lien chapitre1 interne">Lien 1</a><br />
<a href="#" title="lien chapitre2 interne">Lien 2</a><br />
<a href="#" title="lien chapitre1 interne">Lien 3</a><br />
<a href="#" title="lien chapitre3 interne">Lien 4</a><br />
<a href="#" title="lien externe">Lien 5</a><br />
</p>
</body>
</html>
```

```
$("a[title^='lien'][title$='interne'][title*='chapitre1']")
```

Parmi les liens `<a>`, retenons ceux dont l'attribut `title` commence par "lien", se termine par "interne" et comporte "chapitre1".

© ENI Editions - All rigths reserved - Jonifar lina

Les sélecteurs et filtres de formulaires

Les formulaires en jQuery méritent une place à part dans notre étude du jQuery. Ainsi les sélecteurs et filtres relatifs aux formulaires seront abordés au chapitre Les formulaires.

© ENI Editions - All rigths reserved - Jonifar lina

Les sélecteurs et les caractères spéciaux

Les symboles repris dans la syntaxe de jQuery constituent un problème lorsqu'ils sont utilisés dans la partie littérale du code.

Il faut alors indiquer que ces caractères ne sont pas des symboles jQuery. Pour ce faire, on fera précéder les caractères spéciaux de deux barres obliques inverses \\ (*backslashes*)

Par exemple :

```
#foo\\:bar
#foo\\[bar\\]
#foo\\.bar
```

La liste complète des caractères spéciaux qui appartiennent à la syntaxe de jQuery sont : `# ; & , . + * ~ ' : " ! ^ $ [ ] ( ) = > | /`

Il est plus raisonnable de les éviter purement et simplement.

© ENI Editions - All rigths reserved - Jonifar lina

Introduction

Les nombreux sélecteurs de jQuery mis en place, nous entamons avec ce chapitre, l'aspect dynamique du JavaScript et de jQuery qui est de modifier les éléments.

© ENI Editions - All rigths reserved - Jonifar lina

Ajouter ou supprimer une classe

addClass(classe)

Ajoute la classe spécifiée à tous les éléments sélectionnés.

`$("p:last").addClass("selected")` : ajoute la classe `selected` au dernier paragraphe.

Cette méthode retourne un objet jQuery.

Commentaires

Il faut noter que cette méthode ne remplace pas une classe. Elle ajoute simplement une classe.

Il est possible d'ajouter plus d'une classe à la fois. Elles sont notées les unes à la suite des autres, séparées par un espace, soit `addClass(classe1 classe2 classe3)`.

Cette méthode `addClass()` est souvent associée à la méthode `removeClass()` pour créer un effet de commutation.

removeClass(classe)

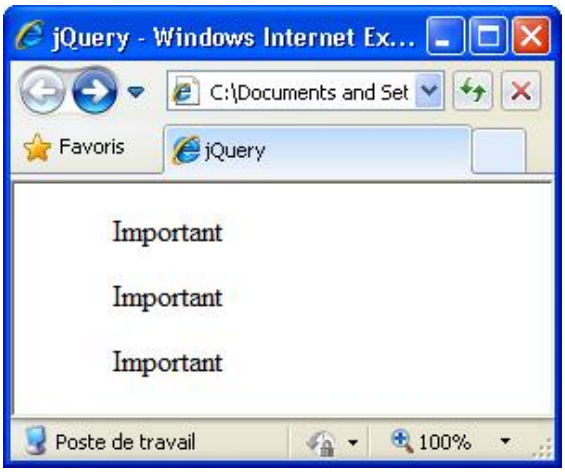
Supprime la classe spécifiée à tous les éléments sélectionnés.

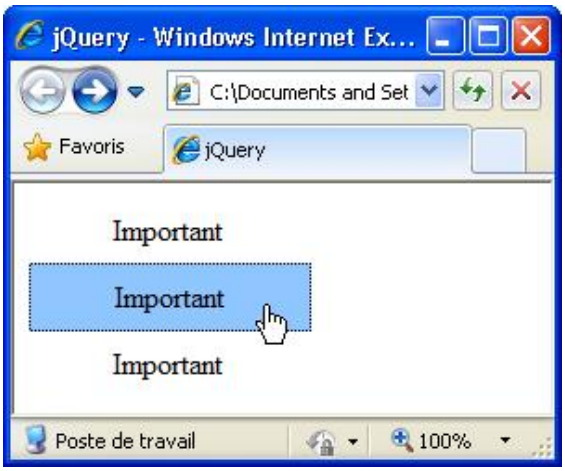
`$("p:last").removelass("selected")` : supprime la classe `selected` au dernier paragraphe.

Cette méthode retourne un objet jQuery.

Exemple

Au survol d'un paragraphe par le curseur, mettons celui-ci en évidence en le dotant d'un arrière-plan de couleur et d'une bordure. Cet effet se réalise en ajoutant une classe au passage de la souris.





Le document initial se présente comme suit :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
div { width: 150px;
      height: 35px;
      line-height: 35px;
      vertical-align: middle;
      text-align: center;
      cursor: pointer;}
.arriereplan { background-color: #9cf;
               border: 1px solid black;}
</style>
</head>
<body>
<div>Important</div>
<div>Important</div>
<div>Important</div>
</body>
</html>
```

Passons au script jQuery.

```
<script type="text/javascript">
$(document).ready(function(){
$("div").mouseover(function(){
$(this).addClass("arriereplan");
});
$("div").mouseout(function(){
$(this).removeClass("arriereplan");
});
});
</script>
```

Détaillons celui-ci.

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

Au chargement du DOM.

```
$("div").mouseover(function(){
$(this).addClass("arriereplan");
});
```

Au survol d'une division `<div>` par la souris (`mouseover`), le script ajoute (`addClass()`) à la division survolée (`this`), la classe `arriereplan`.

```
$("div").mouseout(function(){
$(this).removeClass("arriereplan");
});
```

Lorsque le curseur quitte la division (`mouseout`), jQuery retire (`removeClass()`) à la division survolée (`this`), la classe `arriereplan`.

```
});
```

Fin de script.

Le fichier Xhtml complet devient :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
```


© ENI Editions - All rigths reserved - Jonifar lina

```
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("div").mouseover(function(){
$(this).addClass("arriereplan");
});
$("div").mouseout(function(){
$(this).removeClass("arriereplan");
});
});
</script>
<style type="text/css">
div { width: 150px;
      height: 35px;
      line-height: 35px;
      vertical-align: middle;
      text-align: center;
      cursor: pointer;}
.arriereplan { background-color: #9cf;
               border: 1px solid black;}
</style>
</head>
<body>
<div>Important</div>
<div>Important</div>
<div>Important</div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Vérifier la présence d'une classe

hasClass(classe)

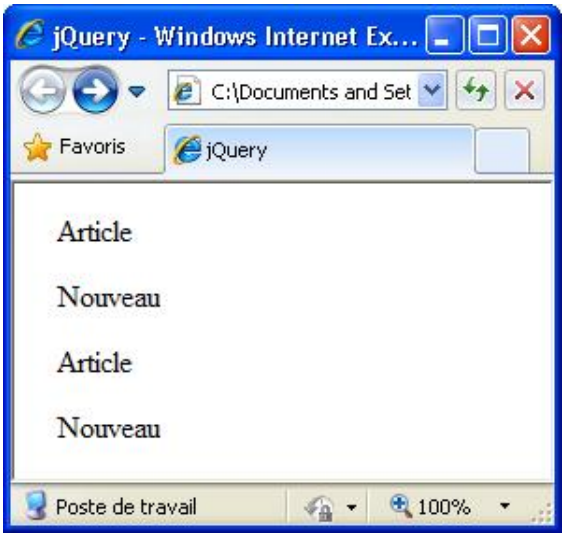
Vérifie si la classe spécifiée en argument est présente pour les éléments ciblés. Retourne *true* si la classe spécifiée est présente pour au moins un des éléments ciblés, *false* dans le cas contraire.

`$("#p1").hasClass("box")` : vérifie si l'élément identifié par `p1` possède la classe `box`.

La méthode renvoie un booléen (true ou false).

Exemple

Soit une série de paragraphes.

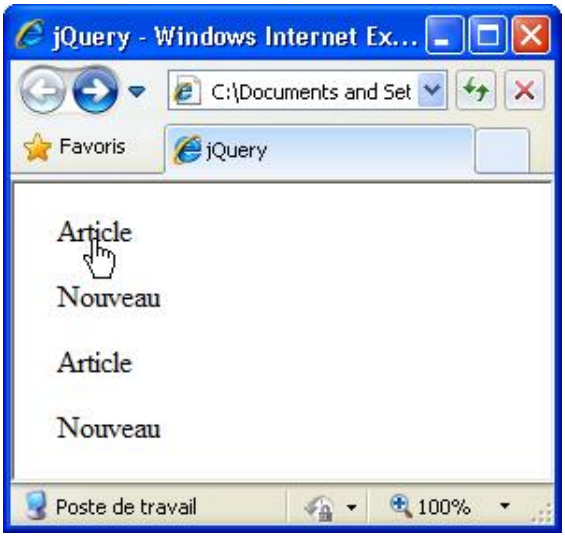


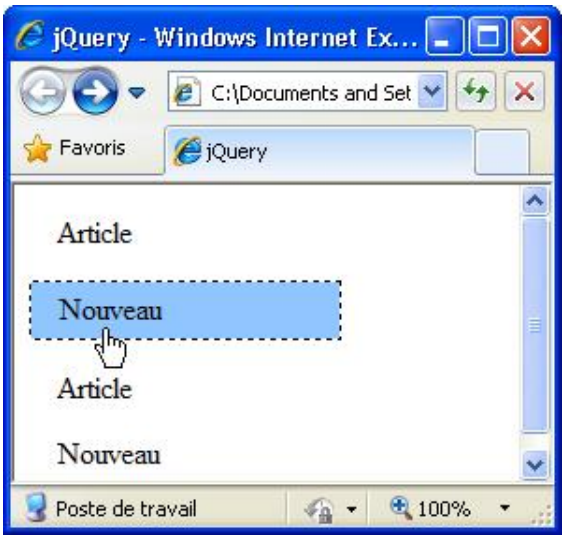
Au départ,

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
p { width: 150px;
    padding-left: 15px;
    cursor: pointer;}
.arriereplan { height: 30px;
               line-height: 30px;
               vertical-align: middle;
               background-color: #9cf;
               border: 1px dashed black;}
</style>
</head>
<body>
<p>Article</p>
<p class="new">Nouveau</p>
<p>Article</p>
<p class="new">Nouveau</p>
</body>
</html>
```

Au survol de la souris, le script jQuery ne doit doter d'un arrière-plan de couleur et d'une bordure uniquement les paragraphes avec la classe `new`.

© ENI Editions - All rigths reserved - Jonifar lina





Le script jQuery :

```
<script type="text/javascript">
$(document).ready(function(){
$("p").mouseover(function(){
if ($(this).hasClass("new") ) {
$(this).addClass("arriereplan")
};
});
});
</script>
```

Explications du script :

```
$(document).ready(function(){
$("p").mouseover(function(){
```

Au chargement du DOM et au survol des paragraphes par le curseur.

```
if ($(this).hasClass("new") ) {
$(this).addClass("arriereplan")
};
```

Le script vérifie si (`if`) l'élément (`this`) survolé possède la classe `new` (`hasClass("new")`). Dans l'affirmative, la classe `arriereplan` est ajoutée (`addClass()`) à celui-ci. Remarquons que rien n'empêche de mélanger JavaScript classique et JavaScript jQuery.

```
});
```


© ENI Editions - All rigths reserved - Jonifar lina

```
});
```

Fin de script

Au final, voici le fichier Xhtml :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("p").mouseover(function(){
if ($(this).hasClass("new") ) {
$(this).addClass("arriereplan")
};
});
});
</script>
<style type="text/css">
p { width: 150px;
    padding-left: 15px;
    cursor: pointer;}
.arriereplan { height: 30px;
               line-height: 30px;
               vertical-align: middle;
               background-color: #9cf;
               border: 1px dashed black;}
</style>
</head>
<body>
<p>Article</p>
<p class="new">Nouveau</p>
<p>Article</p>
<p class="new">Nouveau</p>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Basculer entre deux classes

La librairie jQuery propose plusieurs méthodes qui permettent de déclencher tantôt une action, tantôt une autre. Cet effet de permutation est repris sous le terme de *toggle*. Nous le rencontrerons plusieurs fois dans notre exploration de jQuery.

Outre des effets spectaculaires, ces méthodes entraînent une économie appréciable de lignes de code.

toggleClass(classe)

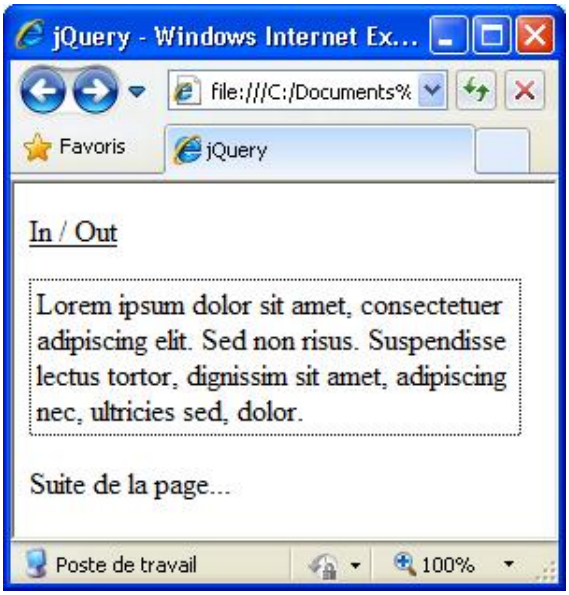
Ajoute la classe spécifiée si elle n'est pas présente, retire la classe spécifiée si elle est présente.

`$(p).toggleClass("classe1")` : applique la classe `classe1` aux paragraphes si elle n'est pas présente. Si elle l'est, enlève la classe `classe1`.

Cette méthode retourne un objet jQuery.

Exemple

Au clic du lien, faire apparaître ou disparaître une division de la page.



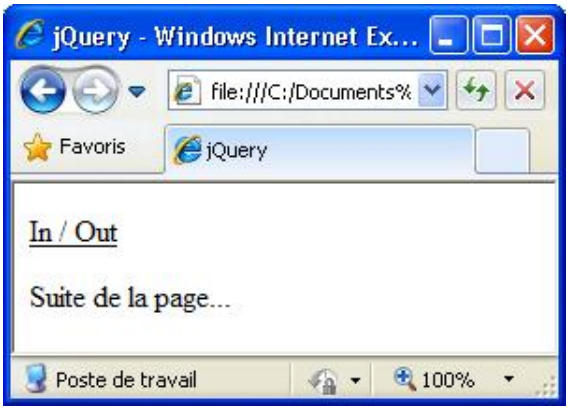
Au départ :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
a{color: black}
div { width: 255px;
      padding: 3px;
      border: 1px dotted black;
      cursor: pointer;}
.cacher { display: none;}
</style>
</head>
<body>
<p><a href="#">In / Out</a></p>
<div>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor.</div>
<p>Suite de la page...</p>
</body>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
</html>
```

Le script jQuery :




```
<script type="text/javascript">
$(document).ready(function(){
$("a").click(function(){
$("div").toggleClass("cacher");
});
});
</script>
```

Explorons le script pas à pas.

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

Au chargement de la page, du DOM pour être précis.

`$("a").click(function(){`

Au clic de la souris sur le lien `<a>`.

`$("div").toggleClass("cacher");`

Permuter (`toggleClass()`) la classe `cacher` sur la division `<div>`.

```
});
});
```

Fin de script.

Au final :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("a").click(function(){
$("div").toggleClass("cacher");
});
});
</script>
<style type="text/css">
a{color: black}
div { width: 255px;
```


© ENI Editions - All rigths reserved - Jonifar lina

```
      padding: 3px;
      border: 1px dotted black;
      cursor: pointer;}
.cacher { display: none;}
</style>
</head>
<body>
<p><a href="#">In / Out</a></p>
<div>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor.</div>
<p>Suite de la page...</p>
</body>
</html>
```

Il y a bien d'autres façons de réaliser cet effet en jQuery, par exemple, avec les méthodes `show()` et `hide()` (voir chapitre Les effets consacré aux effets).

© ENI Editions - All rigths reserved - Jonifar lina

Connaître la valeur d'un attribut

Cette méthode jQuery correspond à `getAttribute()` du JavaScript classique.

attr(nom de l'attribut)

Accède à la valeur de l'attribut mentionné.

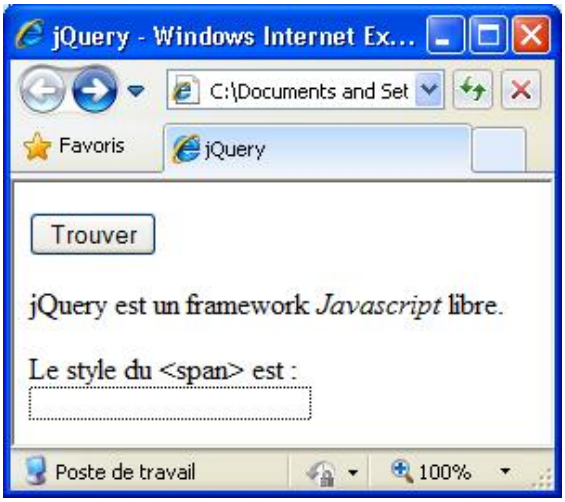
Cette méthode est assez utile pour retrouver la valeur d'un attribut de l'élément sélectionné ou du premier élément sélectionné s'il y en a plusieurs. Si l'élément n'a pas d'attribut répondant à ce nom, la valeur `undefined` est retournée.

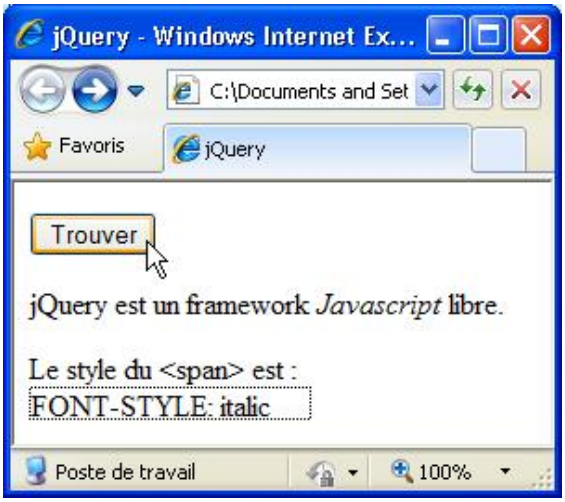
`$("a").attr("title")` : récupère la valeur de l'attribut `title` du premier lien rencontré.

Cette méthode retourne un objet jQuery.

Exemple

Au clic sur le bouton, recherchons le style attaché à la balise `<span>JavaScript</span>`. Le résultat sera affiché dans une division prévue à cet effet.





Le fichier de départ :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
```

© ENI Editions - All rigths reserved - Jonifar lina

```
div { width: 150px;
      height: 16px;
      border: 1px dotted black;}
</style>
</head>
<body>
<p><button>Trouver</button></p>
<p>
jQuery est un framework <span style="font-
style:italic">JavaScript</span> libre.
</p>
Le style du &lt;span&gt; est :<div></div>
</body>
</html>
```

Le script jQuery se présente ainsi :

```
<script type="text/javascript">
$(document).ready(function(){
$("button").click(function(){
var css = $("span").attr("style");
$("div").text(css);
});
});
</script>
```

Détaillons celui-ci.

```
$(document).ready(function(){
$("button").click(function(){
```

Au chargement du DOM et au clic du bouton.

```
var css = $("span").attr("style");
```

Le script charge dans la variable `css`, la valeur de l'attribut `style="…"` attaché à la balise `<span>`.

```
$("div").text(css);
```

Cette variable est affichée comme du texte (`text(css)`) dans la division `<div>` prévue à cet effet.

```
});
});
```

Fin de script.

Le document final devient :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("button").click(function(){
var css = $("span").attr("style");
$("div").text(css);
});
});
</script>
<style type="text/css">
div { width: 150px;
      height: 16px;
      border: 1px dotted black;}
```


© ENI Editions - All rigths reserved - Jonifar lina

```
</style>
</head>
<body>
<p><button>Trouver</button></p>
<p>
jQuery est un framework <span style="font-
style:italic">JavaScript</span> libre.
</p>
Le style du &lt;span&gt; est :<div></div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Ajouter un attribut et sa valeur

attr(attribut, valeur)

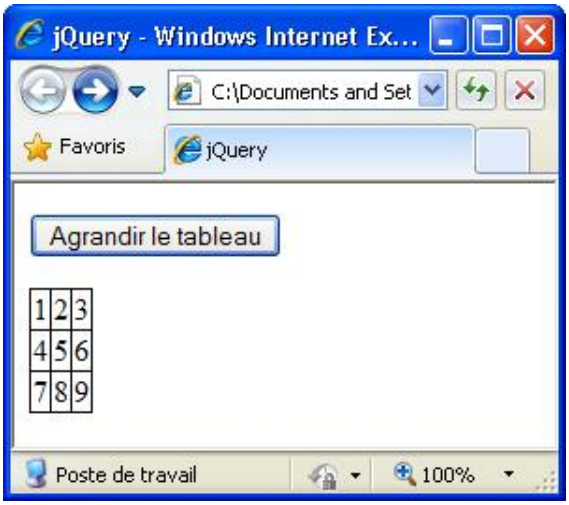
Assigne une paire attribut/valeur à tous les éléments concernés.

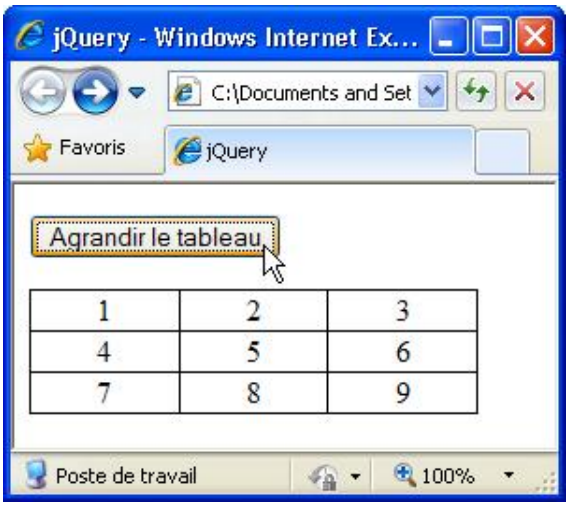
`$("#photo").attr("alt","Parc éoliennes")` : assigne à l'élément identifié par `#photo`, l'attribut `alt="Parc éoliennes"`.

Cette méthode retourne un objet jQuery.

Exemple

Au clic sur le bouton, un tableau de données sera affiché avec une largeur plus grande pour le rendre ainsi plus lisible.





Le fichier de départ :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
table {  border-collapse: collapse;
         border: 1px solid black;}
td { text-align: center;
     border: 1px solid black;}
</style>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
</head>
<body>
<p><button>Agrandir le tableau</button></p>
<table>
<tr>
<td>1</td><td>2</td><td>3</td>
</tr>
<tr>
<td>4</td><td>5</td><td>6</td>
</tr>
<tr>
<td>7</td><td>8</td><td>9</td>
</tr>
</table>
</body>
</html>
```

Le script jQuery s'écrit :

```
<script type="text/javascript">
$(document).ready(function(){
$("button").click(function(){
$("table").attr("width","240px");
});
});
</script>
```

Soit, en détail :

```
$(document).ready(function(){
$("button").click(function(){
```

Au chargement du DOM et au clic sur le bouton.

```
$("table").attr("width","240px");
```

L'attribut `width` avec une valeur de `240 px` est ajouté à la balise de tableau `<table>`.

```
});
});
```

Fin du script

Un effet spectaculaire pour un minimum de code !

Le fichier complet :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("button").click(function(){
$("table").attr("width","240px");
});
});
</script>
<style type="text/css">
table {  border-collapse: collapse;
         border: 1px solid black;}
td { text-align: center;
     border: 1px solid black;}
</style>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
</head>
<body>
<p><button>Agrandir le tableau</button></p>
<table>
<tr>
<td>1</td><td>2</td><td>3</td>
</tr>
<tr>
<td>4</td><td>5</td><td>6</td>
</tr>
<tr>
<td>7</td><td>8</td><td>9</td>
</tr>
</table>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Ajouter plusieurs attributs et leurs valeurs

attr({propriétés})

Permet d'assigner un ensemble de paires attribut/valeur aux éléments sélectionnés.

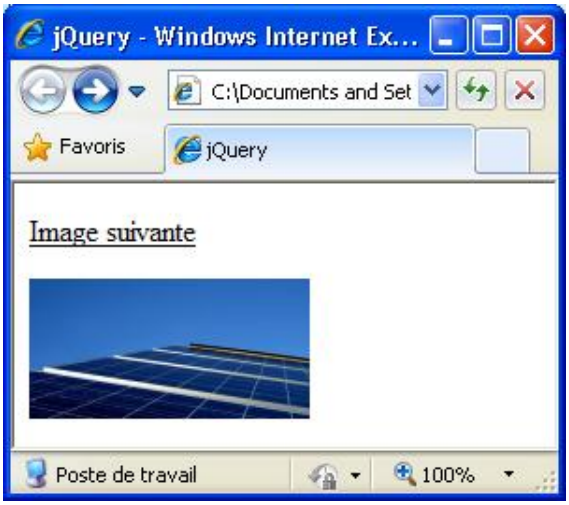
Les différentes propriétés sont séparées par une virgule.

`$("img").attr({ src: "hat.gif", alt: "Logo jQuery!" })` : assigne les attributs `src` et `alt` aux images.

Cette méthode retourne un objet jQuery.

Exemple

Passons d'une image à une autre par un simple clic sur un lien.



Les images de l'exemple sont disponibles dans l'espace de téléchargement réservé à cet ouvrage.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
a { color: black;}
</style>
</head>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<body>
<p><a href="#">Image suivante</a></p>
<div>
<img src="pansolaire1.png" alt="Panneau solaire 1"/>
</div>
</body>
</html>
```

Le script jQuery se présente comme suit.

```
<script type="text/javascript">
$(document).ready(function(){
$("a").click(function(){
$("img").attr({ src: "pansolaire2.png",
                alt: "Panneau solaire 2",
                title: "Ecologie"
});
});
});
</script>
```

Étudions-le pas à pas.

```
$(document).ready(function(){
$("a").click(function(){
```

Au chargement de jQuery (du DOM) et au clic du lien.

```
$("img").attr({ src: "pansolaire2.png",
                alt: "Panneau solaire 2",
                title: "Ecologie"
});
```

Le script ajoute d'abord à la balise `<img>`, l'attribut `src`. Ce qui permet de charger la nouvelle image. Accessoirement, les attributs `alt` et `title` sont également ajoutés.

```
});
});
```

Fin de script.

Le fichier complet devient :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("a").click(function(){
$("img").attr({ src: "pansolaire2.png",
                alt: "Panneau solaire 2",
                title: "Ecologie"
});
});
});
</script>
<style type="text/css">
a { color: black;}
</style>
</head>
<body>
<p><a href="#">Image suivante</a></p>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<div>
<img src="pansolaire1.png" alt="Panneau solaire 1"/>
</div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Supprimer un attribut

removeAttr(nom de l'attribut)

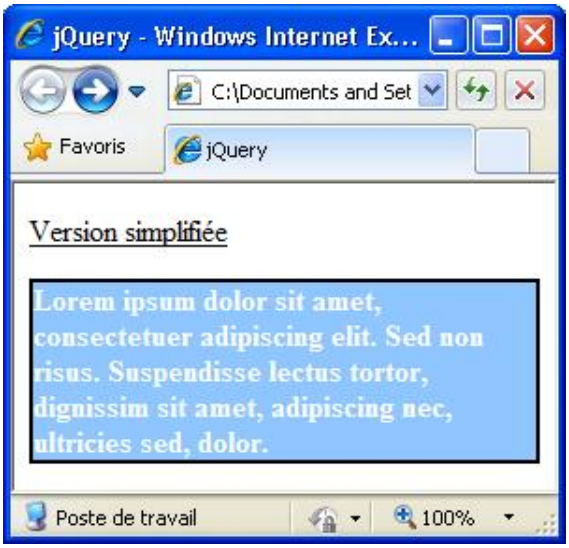
Supprime un attribut des éléments concernés.

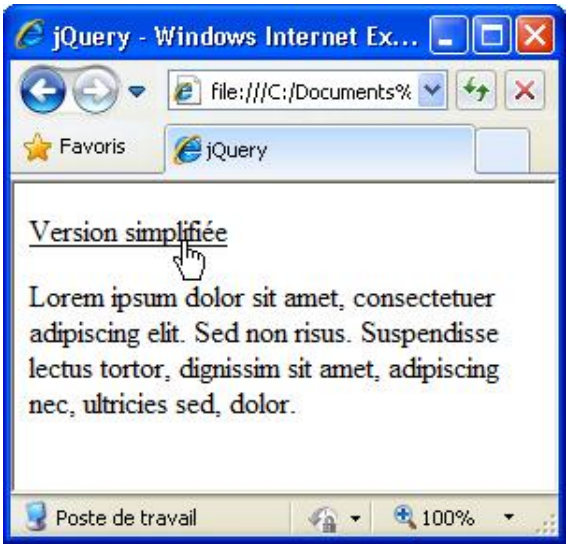
`$("#div1").removeAttr("disabled")` : supprime l'attribut `disabled` de l'élément identifié par `div1`.

Cette méthode retourne un objet jQuery.

Exemple

Supprimons l'attribut de style de la division afin de rendre la lecture plus aisée.





Le document Xhtml initial :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
a { color: black}
</style>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
</head>
<body>
<p><a href="#">Version simplifiée</a></p>
<div style="background-color: #9cf; border: 2px solid black;
color: white; font-weight: bold;">Lorem ipsum dolor sit amet,
consectetuer adipiscing elit. Sed non risus. Suspendisse lectus
tortor, dignissim sit amet, adipiscing nec, ultricies sed,
dolor.</div>
</body>
</html>
```

Le script jQuery :

```
<script type="text/javascript">
$(document).ready(function(){
$("a").click(function(){
$("div").removeAttr("style");
});
});
</script>
```

```
$(document).ready(function(){
$("a").click(function(){
```

Lorsque le DOM est chargé et le lien cliqué.

```
$("div").removeAttr("style");
```

Le script enlève l'attribut de style en ligne (voir le code Xhtml) de la division.

```
});
});
```

Fin de script.

Le fichier Xhtml final se présente alors :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("a").click(function(){
$("div").removeAttr("style");
});
});
</script>
<style type="text/css">
a { color: black}
</style>
</head>
<body>
<p><a href="#">Version simplifiée</a></p>
<div style="background-color: #9cf; border: 2px solid black;
color: white; font-weight: bold;">Lorem ipsum dolor sit amet,
consectetuer adipiscing elit. Sed non risus. Suspendisse lectus
tortor, dignissim sit amet, adipiscing nec, ultricies sed,
dolor.</div>
</body>
</html>
```


© ENI Editions - All rigths reserved - Jonifar lina

Connaître l'attribut value

Ce point a trait aux formulaires et leur attribut `value`.

val()

Récupère sous forme de chaîne de caractères, le contenu de l'attribut `value` du premier élément de la sélection.

`$("input").val()` : récupère le contenu de l'attribut `value` du premier champ de formulaire de type `<input>`.

Cette méthode retourne une chaîne de caractères.

Exemple

Récupérons la valeur du bouton radio coché.

Au départ :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
div { width: 150px;
      height: 16px;
      border: 1px dotted black;}
</style>
</head>
<body>
<form action="">
<input type="radio" name="1" value="Internet Explorer" />Internet
Explorer<br />
<input type="radio" name="1" value="Firefox" />Firefox<br />
<input type="radio" name="1" value="Safari" />Safari<br />
<input type="radio" name="1" value="Opera" />Opera<br />
<input type="radio" name="1" value="Google" />Google<br />
<input type="radio" name="1" value="Autre" />Autre
</form>
<p>Votre choix est :
<button>Trouver</button></p>
<div></div>
</body>
</html>
```

Le script jQuery.

```
<script type="text/javascript">
$(document).ready(function(){
$("button").click(function(){
var choix = $('input:radio:checked').val();
$("div").text(choix);
});
});
</script>
```

Explications :

```
$(document).ready(function(){
$("button").click(function(){
```

Au chargement du DOM et au clic sur le bouton.

```
var choix = $('input:radio:checked').val();
```

Le contenu du bouton radio sélectionné (`'input:radio:checked'`) est stocké dans la variable `choix`.

```
$("div").text(choix);
```

Le contenu de la variable `choix` est affiché (`text(choix)`) dans la division.

```
});
});
```

Fin de script.

Au final :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("button").click(function(){
var choix = $('input:radio:checked').val();
$("div").text(choix);
});
});
</script>
<style type="text/css">
div { width: 150px;
      height: 16px;
      border: 1px dotted black;}
</style>
</head>
<body>
<form action="">
<input type="radio" name="1" value="Internet Explorer" />Internet
Explorer<br />
<input type="radio" name="1" value="Firefox" />Firefox<br />
<input type="radio" name="1" value="Safari" />Safari<br />
<input type="radio" name="1" value="Opera" />Opera<br />
<input type="radio" name="1" value="Google" />Google<br />
<input type="radio" name="1" value="Autre" />Autre
</form>
<p>Votre choix est :
<button>Trouver</button></p>
<div></div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Modifier l'attribut value

Variante de `val()` étudié au point précédent. Ici, jQuery permet de modifier l'attribut `value`.

val(valeur)

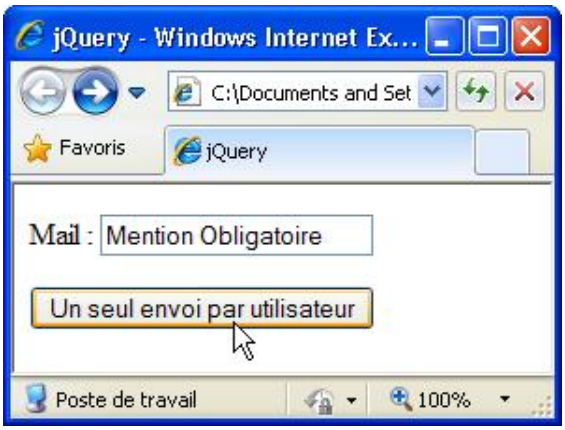
Assigne une nouvelle valeur à l'attribut `value` de l'élément sélectionné.

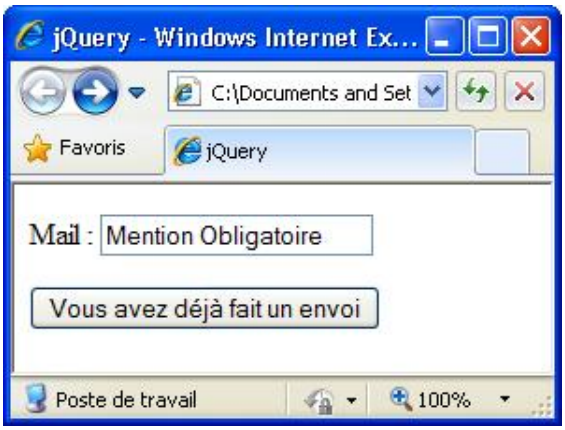
`$("#input").val("Test")` : ajoute la valeur `Test` à une ligne de texte.

Cette méthode retourne un objet jQuery.

Exemple

Après une première soumission du formulaire, le texte du bouton d'envoi est modifié.





Le fichier initial :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
a { color: black;}
</style>
</head>
<body>
<form action="">
<p>Mail : <input type="text" id="input1" value="Mention
Obligatoire" /></p>
<p><input type="submit" id="soumettre" value="Un seul envoi par
```

© ENI Editions - All rigths reserved - Jonifar lina

```
utilisateur" /></p>
</form>
</body>
</html>
```

Passons au script jQuery.

```
<script type="text/javascript">
$(document).ready(function(){
$("#soumettre").click(function(){
$("#soumettre").val("Vous avez déjà fait un envoi");
return false;
});
});
</script>
```

Détaillons celui-ci.

```
$(document).ready(function(){
$("#soumettre").click(function(){
```

Au chargement du DOM et au clic sur le bouton de soumission.

```
$("#soumettre").val("Vous avez déjà fait un envoi");
return false;
```

Une nouvelle valeur (`"Vous avez déjà fait un envoi"`) est transmise au bouton d'envoi. Il faut noter que la mention précédente est ainsi remplacée.

```
});
});
```

Fin du script.

Au final :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#soumettre").click(function(){
$("#soumettre").val("Vous avez déjà fait un envoi");
});
});
</script>
<style type="text/css">
a { color: black;}
</style>
</head>
<body>
<form action="">
<p>Mail : <input type="text" id="input1" value="Mention
Obligatoire" /></p>
<p><input type="submit" id="soumettre" value="Un seul envoi par
utilisateur" /></p>
</form>
</body>
</html>
```


© ENI Editions - All rigths reserved - Jonifar lina

Introduction

Est-il nécessaire de souligner l'importance prise par les feuilles de style dans l'écriture du code des pages Web ? Avec jQuery, la modification dynamique des propriétés de style CSS se révèle facile à implémenter.

La méthode `css()` étudiée aux trois premiers points de ce chapitre, n'est pas sans rappeler la méthode `attr()` abordée au chapitre précédent.

© ENI Editions - All rigths reserved - Jonifar lina

Accéder à une propriété de style

La méthode `css()` se décline de trois façons. La première permet uniquement d'accéder à la propriété de style CSS d'un élément donné.

css(nom)

Permet d'accéder à une propriété de style du premier élément trouvé.

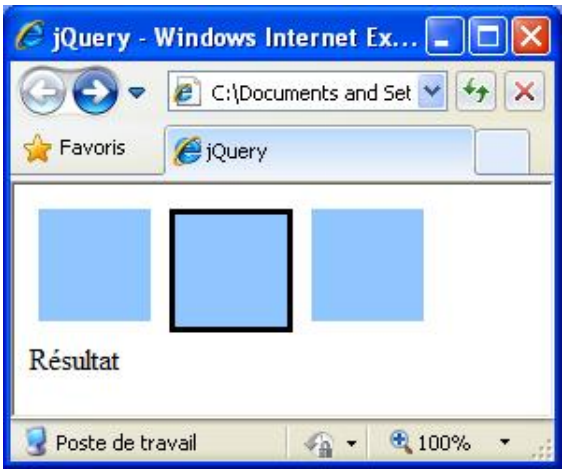
Le nom est une chaîne de caractères qui reprend la propriété de style à accéder.

```
$("p").css("color");
```

Cette méthode renvoie une chaîne de caractères (String).

Exemple

Soit une page avec 3 éléments de type bloc.

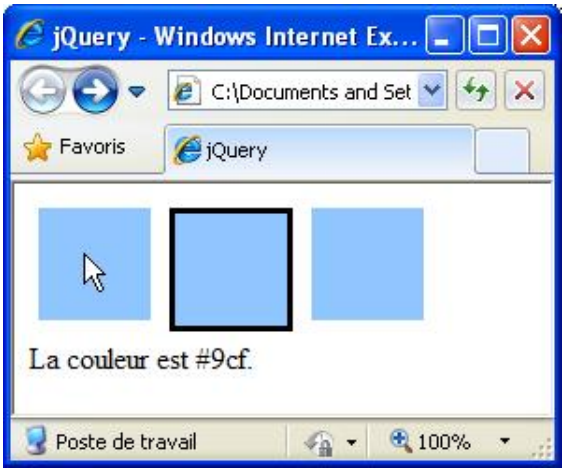


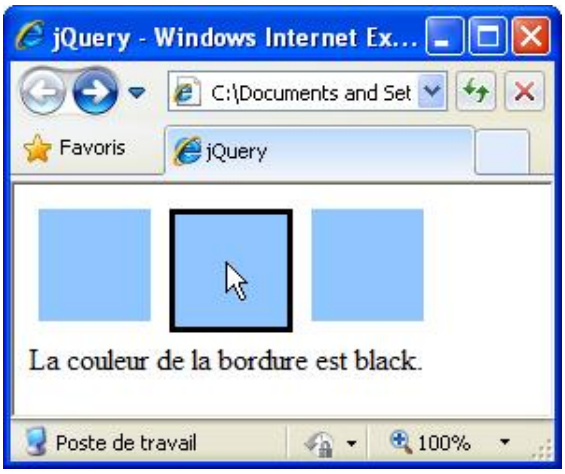
Le document Html :

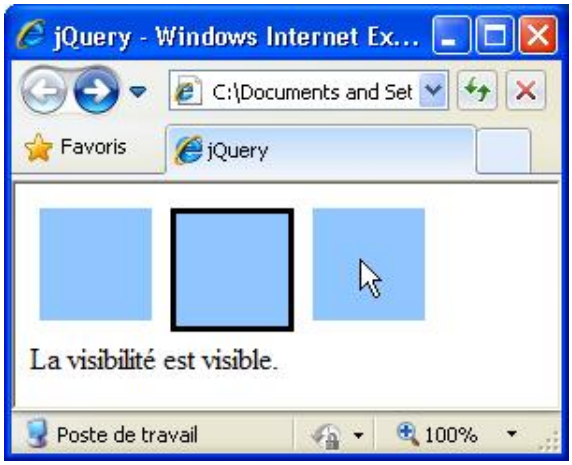
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
div { width: 60px;
      height: 60px;
      margin: 5px;
      float: left;
      background-color:#9cf;"}
</style>
</head>
<body>
<div id="div1"></div>
<div id="div2" style="border: 3px solid black;"></div>
<div id="div3" style="visibility: visible;"></div>
<p style="clear: left;" id="resultat">Résultat </p>
</body>
</html>
```

Au clic sur un élément de type bloc, le script affichera respectivement, de gauche à droite, la couleur d'arrière-plan, la couleur de la bordure et la visibilité.

© ENI Editions - All rigths reserved - Jonifar lina







Soit le script jQuery :

```
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#div1").click(function () {
var couleur = $(this).css("background-color");
$("#resultat").html("La couleur est <span>" + couleur +
"</span>.");
});
$("#div2").click(function () {
var bordure = $(this).css("border-color");
$("#resultat").html("La couleur de la bordure est <span>" +
bordure + "</span>.");
});
$("#div3").click(function () {
```


© ENI Editions - All rigths reserved - Jonifar lina

```
var visibility = $(this).css("visibility");
$("#resultat").html("La visibilité est <span>" + visibility +
"</span>.");
});
});
//]]>
</script>
```

Détaillons celui-ci.

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

Dès que le DOM est chargé.

```
$("#div1").click(function () {
var couleur = $(this).css("background-color");
```

Le clic de la souris sur le premier carré (`id="div1"`) charge dans la variable `couleur` la propriété de la couleur d'arrière-plan de cet élément par `css("background-color")`.

```
$("#resultat").html("La couleur est <span>" + couleur + "</span>.");
});
```

La valeur de la propriété de style est alors affichée comme du Html dans la balise `<p>` identifiée par `resultat`.

```
$("#div2").click(function () {
var border = $(this).css("border-color");
```

Le clic de la souris sur le second carré (`id="div2"`) charge dans la variable `bordure` la couleur de la bordure de cet élément par `css("border-color")`.

```
$("#resultat").html("La couleur de la bordure est <span>" + border +
"</span>.");
});
```

La valeur est affichée dans la page Html.

```
$("#div3").click(function () {
var visibility = $(this).css("visibility");
```

Le clic de la souris sur le troisième carré (`id="div3"`) charge dans la variable `visibility`, l'état de visibilité de cet élément par `css("visibility")`.

```
$("#resultat").html("La visibilité est <span>" + visibility + "</span>.");
});
```

La valeur est affichée dans la page.

```
});
```

Fin de script.

Le fichier final est :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js" ></script>
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#div1").click(function () {
```

© ENI Editions - All rigths reserved - Jonifar lina

```
var couleur = $(this).css("background-color");
$("#resultat").html("La couleur est <span>" + couleur +
"</span>.");
});
$("#div2").click(function () {
var bordure = $(this).css("border-color");
$("#resultat").html("La couleur de la bordure est <span>" +
bordure + "</span>.");
});
$("#div3").click(function () {
var visibility = $(this).css("visibility");
$("#resultat").html("La visibilité est <span>" + visibility +
"</span>.");
});
});
//]]>
</script>
<style type="text/css">
div { width:60px;
      height:60px;
      margin:5px;
      float:left;
      background-color:#9cf;"}
</style>
</head>
<body>
<div id="div1"></div>
<div id="div2" style="border: 3px solid black;"></div>
<div id="div3" style="visibility: visible;"></div>
<p style="clear: left;" id="resultat">Résultat </p>
</body>
</html>
```


© ENI Editions - All rigths reserved - Jonifar lina

Modifier les propriétés de style

La fonction `css()`, dotée de certains paramètres, permet également de modifier des propriétés de style d'éléments de la page.

css({propriété de style})

Modifie les propriétés de style d'un élément donné en utilisant la notation CSS clé/valeur pour les propriétés de style à transformer.

```
$("p").css({ color: "red", background: "blue" });
```

Il faut remarquer la présence des accolades habituelles pour les déclarations de style.

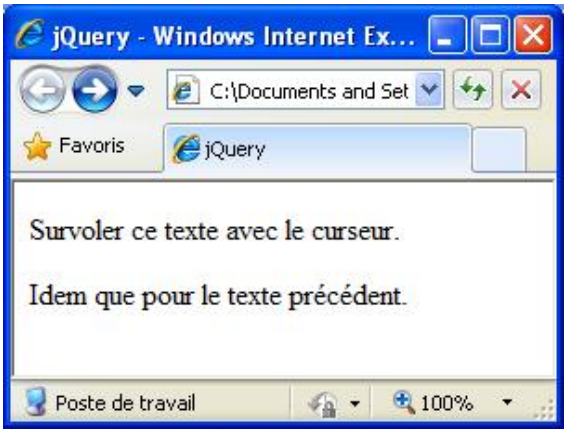
Si la clé contient un trait d'union comme par exemple `background-color`, celle-ci doit être placée entre des guillemets (soit `"background-color"`).

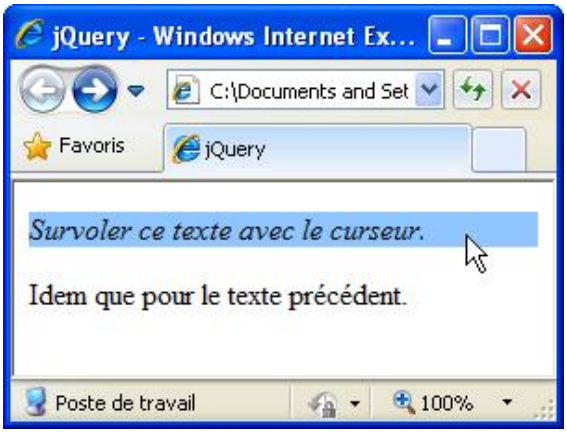
La notation JavaScript (CamelCase) peut également être adoptée soit `backgroundColor` au lieu de `background-color`.

Cette méthode renvoie un objet jQuery.

Exemple

Illustrons cette méthode jQuery par un exemple. Au survol d'un paragraphe, celui-ci est doté d'un arrière-plan et affiche la police de caractères en italique.






```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
```

© ENI Editions - All rigths reserved - Jonifar lina

```
p { font: Arial sans-serif;}
</style>
</head>
<body>
<p>Survoler ce texte avec le curseur.</p>
<p>Idem que pour le texte précédent.</p>
</body>
</html>
```

Le script jQuery :

```
<script type="text/javascript">
$(document).ready(function(){
$("p").mouseover(function () {
$(this).css({'background-color' : '#9cf', 'font-style':
'italic'});
});
$("p").mouseout(function () {
$(this).css({'background-color' : '', 'font-style' : ''});
});
});
</script>
```

Explications :

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

Chargement du DOM.

```
$("p").mouseover(function () {
$(this).css({'background-color' : '#9cf', 'font-style': 'italic'});
});
```

Au survol du paragraphe `<p>` par la souris (`mouseover()`), la fonction `css()` modifie la couleur de l'arrière-plan et met la police de caractères en italique de celui-ci (`css({'background-color' : '#9cf', 'font-style': 'italic'})`).

```
$("p").mouseout(function () {
$(this).css({'background-color' : '', 'font-style' : ''});
});
```

Lorsque le curseur quitte le paragraphe (`mouseout()`), les modifications sont annulées.

`});`

Fin du script.

Les propriétés notées selon les prescriptions CSS peuvent également être reprises selon la notation JavaScript. Le script deviendrait alors :

```
<script type="text/javascript">
$(document).ready(function(){
$("p").mouseover(function () {
$(this).css({backgroundColor : '#9cf', fontStyle: 'italic'});
});
$("p").mouseout(function () {
$(this).css({backgroundColor : '', fontStyle : ''});
});
});
</script>
```

Le document complet :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<title>jQuery</title>
<script type="text/javascript" src="jquery.js" ></script>
<script type="text/javascript">
$(document).ready(function(){
$("p").mouseover(function () {
$(this).css({'background-color' : '#9cf', 'font-style': 'italic'});
});
$("p").mouseout(function () {
$(this).css({'background-color' : '', 'font-style' : ''});
});
});
</script>
<style type="text/css">
p { font: Arial sans-serif;}
</style>
</head>
<body>
<p>Survoler ce texte avec le curseur.</p>
<p>Idem que pour le texte précédent.</p>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Attribuer des propriétés de style

La fonction `css()` de jQuery propose une dernière façon de noter les transformations des propriétés de style.

css(clé,valeur)

Modifie les propriétés de style d'un élément donné en utilisant la notation clé/valeur pour les propriétés de style à transformer.

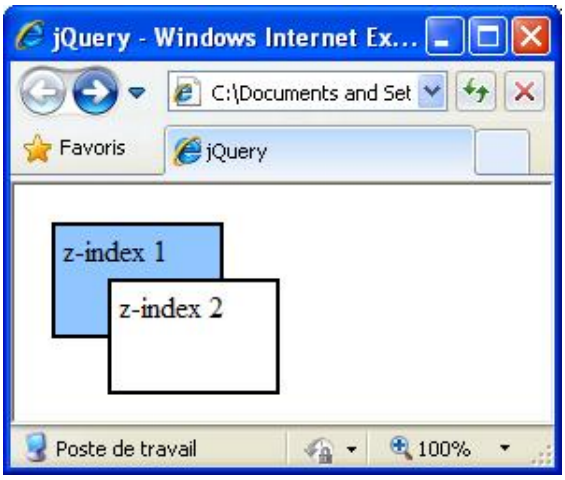
- clé (chaîne de caractère) correspond au nom de la propriété de style à modifier.
- valeur (chaîne de caractère ou nombre) est la nouvelle valeur de la propriété. Si un nombre est spécifié, jQuery le converti automatiquement en pixel.

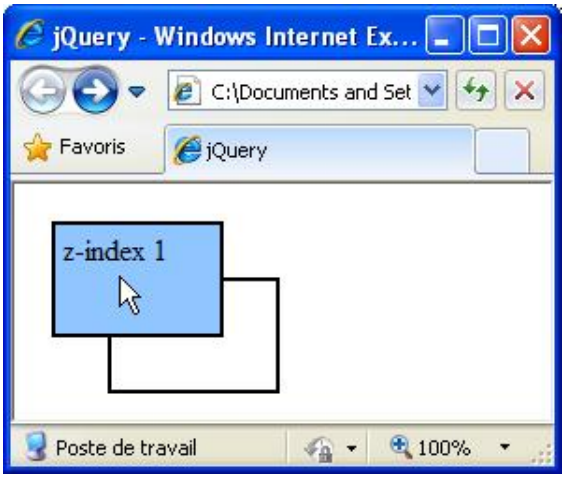
```
$("p").css("color","red");
```

Cette méthode renvoie un objet jQuery.

Exemple

Prenons deux divisions superposées. Par un script jQuery, modifions l'ordre de celles-ci au survol de la souris.






```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<style type="text/css">
#boite1 { position: absolute;
          left: 20px;
          top: 20px;
          width: 80px;
          height: 50px;
          padding: 4px;
          border: 2px solid black;
          background-color: #9cf;}
#boite2 { position: absolute;
          left: 50px;
          top: 50px;
          width: 80px;
          height: 50px;
          padding: 4px;
          border: 2px solid black;
          background-color: white;}
</style>
</head>
<body>
<div id="boite1">
z-index 1
</div>
<div id="boite2">
z-index 2
</div>
</body>
</html>
```

Remarquons qu'aucune propriété de style relative à la superposition des divisions (`z-index`) n'est présente dans le code du fichier de départ.

Le script jQuery :

```
<script type="text/javascript">
$(document).ready(function(){
$("#boite1").mouseover(function () {
$(this).css('z-index' , '10');
});
$("#boite1").mouseout(function () {
$(this).css('z-index' , '');
});
});
</script>
```

Explications :

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

Au chargement du DOM.

```
$("#boite1").mouseover(function () {
$(this).css('z-index' , '10');
});
```

Au survol de la souris (`mouseover()`), la division dont l'identifiant est `boite1`, se voit attribuer une valeur de `z-index` de `10`, ce qui la place au premier plan par rapport à la division identifiée par `boite2`.

```
$("#boite1").mouseout(function () {
$(this).css('z-index' , '');
});
```

Lorsque le curseur quitte la division `boite1`, la valeur par défaut de `z-index` est restituée.

```
});
```

Fin de script.

© ENI Editions - All rigths reserved - Jonifar lina

Le document complet devient :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js" ></script>
<script type="text/javascript">
$(document).ready(function(){
$("#boite1").mouseover(function () {
$(this).css('z-index' , '10');
});
$("#boite1").mouseout(function () {
$(this).css('z-index' , '');
});
});
</script>
<style type="text/css">
#boite1 { position: absolute;
          left: 20px;
          top: 20px;
          width: 80px;
          height: 50px;
          padding: 4px;
          border: 2px solid black;
          background-color: #9cf;}
#boite2 { position: absolute;
          left: 50px;
          top: 50px;
          width: 80px;
          height: 50px;
          padding: 4px;
          border: 2px solid black;
          background-color: white;}
</style>
</head>
<body>
<div id="boite1">
z-index 1
</div>
<div id="boite2">
z-index 2
</div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Le dimensionnement

JQuery propose une série de méthodes relatives à la dimension des éléments.

height()

Renvoie la hauteur, exprimée en pixels, d'un élément.

```
$("p").height();
```

Cette méthode renvoie une chaîne de caractères (String).

height(valeur)

Assigne une hauteur aux éléments spécifiés. Si aucune unité n'est spécifiée (comme `em` ou `%`), l'unité `px` sera choisie par défaut.

- valeur (chaîne de caractères ou entier) : valeur de la hauteur assignée.

```
$("p").height(120);
```

Cette méthode renvoie un objet jQuery.

Après la hauteur, la largeur…

width()

Renvoie la largeur, exprimée en pixels, d'un élément

```
$("p").width();
```

Cette méthode renvoie une chaîne de caractères (String).

width(valeur)

Assigne une largeur aux éléments spécifiés. Si aucune unité n'est spécifiée (comme `em` ou `%`), l'unité `px` sera choisie par défaut.

- valeur (chaîne de caractères ou entier) : valeur de la largeur assignée.

```
$("p").width(120);
```

Cette méthode renvoie un objet jQuery.

Citons encore les méthodes :

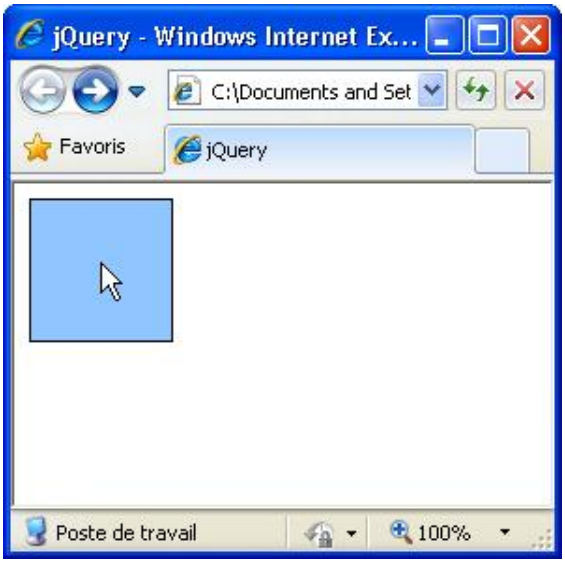
- **innerHeight()** : retourne la hauteur intérieure (la bordure exclue mais le padding inclus) du premier élément trouvé dans la sélection.
- **innerWidth()** : récupère la largeur intérieure (la bordure exclue mais le padding inclus) du premier élément trouvé répondant à la sélection.
- **outerHeight(options)** : retourne la hauteur extérieure (la bordure incluse et le padding par défaut) pour le premier élément trouvé répondant à la sélection.
- **outerWidth(options)** : retourne la largeur extérieure (inclut la bordure et le padding par défaut) pour le premier élément trouvé répondant à la sélection.

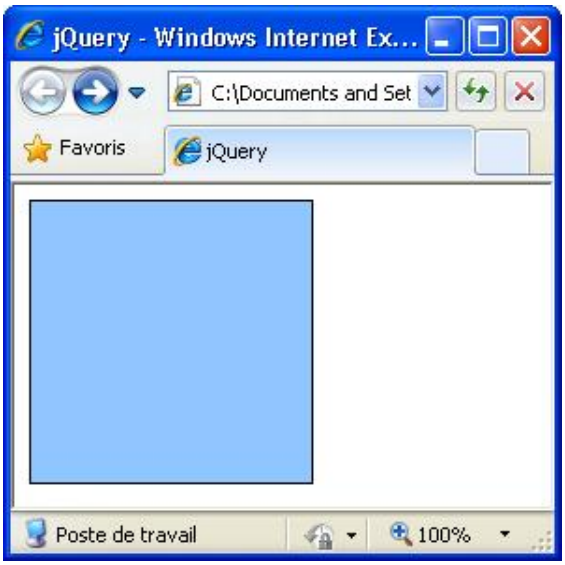
Ces méthodes fonctionnent pour les éléments visibles et non visibles.

Exemple

Au clic sur un élément boîte, un script jQuery détecte la largeur et la hauteur. Ces dimensions sont alors doublées.

© ENI Editions - All rigths reserved - Jonifar lina






```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
div { position: relative;
      width: 75px;
      height: 75px;
      border: 1px solid black;
      background-color: #9cf;}
</style>
</head>
<body>
<div></div>
</body>
</html>
```

Le script :

```
<script type="text/javascript">
$(document).ready(function(){
$("div").click(function () {
hauteur=$(this).height()
```


© ENI Editions - All rigths reserved - Jonifar lina

```
largeur=$(this).width()
$(this).height(hauteur*2).width(largeur*2);
});
});
</script>
```

Explications :

```
$(document).ready(function(){
$("div").click(function () {
```

Au chargement et au clic sur la boîte.

```
hauteur=$(this).height()
largeur=$(this).width()
```

La hauteur et la largeur de l'élément sélectionné sont stockées dans une variable.

```
$(this).height(hauteur*2).width(largeur*2);
```

La hauteur et la largeur de l'élément sont multipliées par 2.

```
});
});
```

Fin du script.

Le document final se présente alors :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js" ></script>
<script type="text/javascript">
$(document).ready(function(){
$("div").click(function () {
hauteur=$(this).height()
largeur=$(this).width()
$(this).height(hauteur*2).width(largeur*2);
});
});
</script>
<style type="text/css">
div { position: relative;
      width: 75px;
      height: 75px;
      border: 1px solid black;
      background-color: #9cf;}
</style>
</head>
<body>
<div></div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Le positionnement

Parallèlement aux méthodes jQuery sur la dimension des éléments, jQuery propose des méthodes pour déterminer la position de ceux-ci.

position()

Renvoie la valeur `top` et `left` de la position d'un élément relative à son élément parent.

```
$("p:first").position();
```

Cette méthode renvoie un objet de type `object{top, left}`.

offset()

Renvoie la valeur `top` et `left` de la position d'un élément relative au document.

```
$("p:first").offset();
```

Cette méthode renvoie un objet de type `object{top, left}`.

Plus particulières sont les méthodes `scrollTop(valeur)` et `scrollLeft(valeur)` qui permettent de modifier pour un élément le décalage par rapport au bord supérieur ou au bord gauche. D'une certaine façon, ces méthodes permettent de contrôler l'amplitude de la barre de défilement verticale et horizontale.

scrollTop(valeur)

Modifie le décalage (en pixels) entre le bord supérieur du document (`top`) et l'élément sélectionné, en prenant la valeur passée en argument.

valeur : nombre positif représentant le nouveau décalage désiré.

```
$("div").scrollTop(300);
```

Cette méthode renvoie un objet jQuery.

scrollLeft(valeur)

Modifie le décalage (en pixels) entre le bord gauche du document (`left`) et l'élément sélectionné, en prenant la valeur passée en argument.

valeur : nombre positif représentant le nouveau décalage désiré.

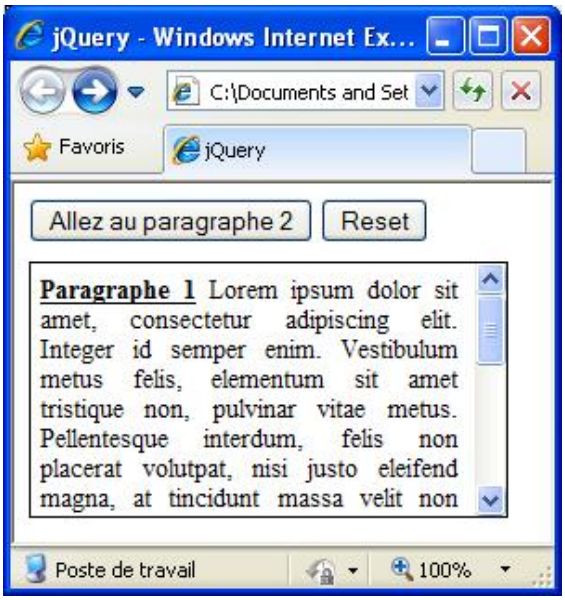
```
$("div").scrollLeft(300);
```

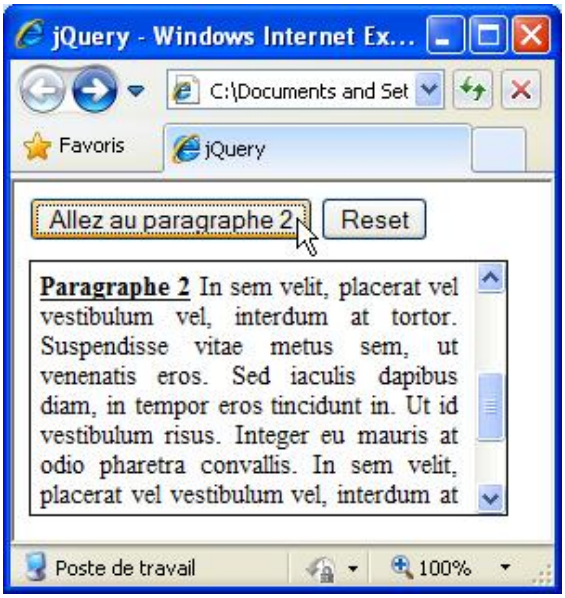
Cette méthode renvoie un objet jQuery.

Exemple

Illustrons ceci par un exemple. Par le clic sur un bouton, nous allons permettre à l'utilisateur d'atteindre directement le second paragraphe d'un texte.

© ENI Editions - All rigths reserved - Jonifar lina






```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
div { position: relative;
      width: 250px;
      height: 130px;
      border: 1px solid black;
      margin-top: 10px;
      padding: 3px;
      font-size: 0.9em;
      overflow: auto;}
p { margin:2px;
    width:225px;
    text-align: justify;}
span { font-weight: bold;
       text-decoration: underline}
</style>
</head>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<body>
<button id="go">Allez au paragraphe 2</button>
<button id="reset">Reset</button>
<div class="demo">
<p><span>Paragraphe 1</span> Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Integer id semper enim. Vestibulum
metus felis, elementum sit amet tristique non, pulvinar vitae
metus. Pellentesque interdum, felis non placerat volutpat, nisi
justo eleifend magna, at tincidunt massa velit non dolor</p>
<p><span>Paragraphe 2</span> In sem velit, placerat vel vestibulum
vel, interdum at tortor. Suspendisse vitae metus sem, ut venenatis
eros. Sed iaculis dapibus diam, in tempor eros tincidunt in. Ut id
vestibulum risus. Integer eu mauris at odio pharetra convallis.
In sem velit, placerat vel vestibulum vel, interdum at tortor.
Suspendisse vitae metus sem, ut venenatis eros. Sed iaculis
dapibus diam, in tempor eros tincidunt in. Ut id vestibulum risus.
Integer eu mauris at odio pharetra convallis.</p>
</div>
</body>
</html>
```

Le script jQuery :

```
<script type="text/javascript">
$(document).ready(function(){
$("#go").click(function () {
$("div").scrollTop(148);
});
$("#reset").click(function () {
$("div").scrollTop(0);
});
});
</script>
```

Explication du script :

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

Au chargement du DOM.

```
$("#go").click(function () {
$("div").scrollTop(148);
});
```

Au clic du bouton, la position du texte par rapport au bord supérieur de la division est modifiée pour atteindre le second paragraphe.

```
$("#reset").click(function () {
$("div").scrollTop(0);
});
```

Le bouton reset remet le décalage vers le haut dans sa position initiale pour faire apparaître à nouveau le premier paragraphe.

```
});
```

Fin de script.

Le document complet se présente alors comme suit.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js" ></script>
```

© ENI Editions - All rigths reserved - Jonifar lina

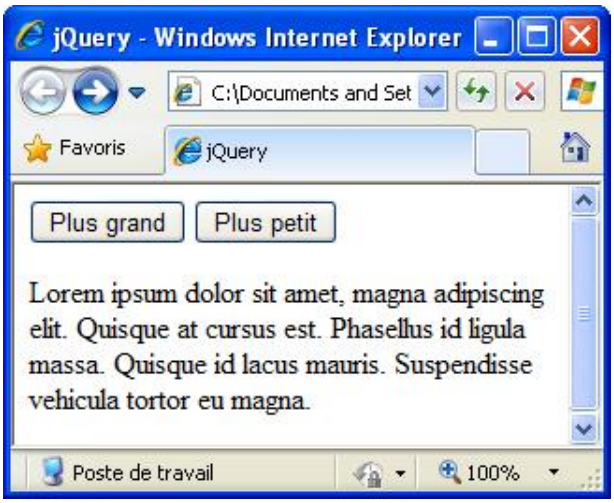
```
<script type="text/javascript">
$(document).ready(function(){
$("#go").click(function () {
$("div").scrollTop(148);
});
$("#reset").click(function () {
$("div").scrollTop(0);
});
});
</script>
<style type="text/css">
div { position: relative;
      width: 250px;
      height: 130px;
      border: 1px solid black;
      margin-top: 10px;
      padding: 3px;
      font-size: 0.9em;
      overflow: auto;}
p { margin:2px;
    width:225px;
    text-align: justify;}
span { font-weight: bold;
       text-decoration: underline}
</style>
</head>
<body>
<button id="go">Allez au paragraphe 2</button>
<button id="reset">Reset</button>
<div class="demo">
<p><span>Paragraphe 1</span> Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Integer id semper enim. Vestibulum
metus felis, elementum sit amet tristique non, pulvinar vitae
metus. Pellentesque interdum, felis non placerat volutpat, nisi
justo eleifend magna, at tincidunt massa velit non dolor</p>
<p><span>Paragraphe 2</span> In sem velit, placerat vel vestibulum
vel, interdum at tortor. Suspendisse vitae metus sem, ut venenatis
eros. Sed iaculis dapibus diam, in tempor eros tincidunt in. Ut id
vestibulum risus. Integer eu mauris at odio pharetra convallis.
In sem velit, placerat vel vestibulum vel, interdum at tortor.
Suspendisse vitae metus sem, ut venenatis eros. Sed iaculis
dapibus diam, in tempor eros tincidunt in. Ut id vestibulum risus.
Integer eu mauris at odio pharetra convallis.</p>
</div>
</body>
</html>
```


© ENI Editions - All rigths reserved - Jonifar lina

Applications

1. Redimensionner la taille des caractères

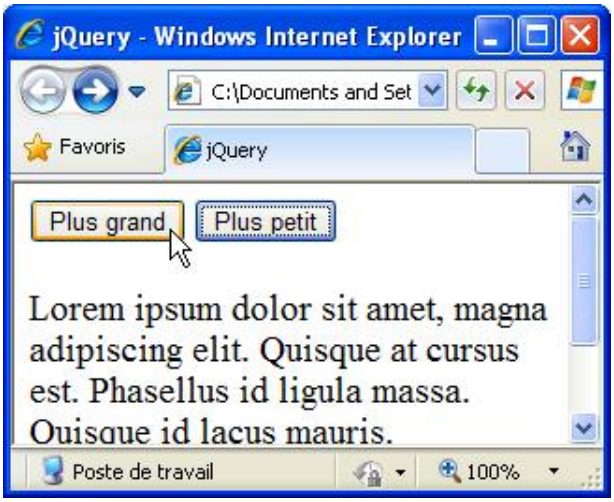
Nous allons permettre à l'utilisateur de modifier, à son gré, la taille de la police de caractères pour un meilleur confort de lecture.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
p { font-size: 1em;}
</style>
</head>
<body>
<input type="button" value="Plus grand" id="plus" />
<input type="button" value="Plus petit" id="moins" />
<p>Lorem ipsum dolor sit amet, magna adipiscing elit. Quisque at
cursus est. Phasellus id ligula massa. Quisque id lacus mauris.
Suspendisse vehicula tortor eu
magna.</p>
</body>
</html>
```

Le script jQuery va permettre au clic d'un premier bouton d'agrandir la taille des caractères. Un second bouton est prévu pour diminuer celle-ci.

© ENI Editions - All rigths reserved - Jonifar lina



```
<script type="text/javascript">
$(document).ready(function(){
$('input').click(function(){
var texte = $('p');
var taille = texte.css('fontSize');
var nombre = parseFloat(taille, 10);
var unite = taille.slice(-2);
if(this.id == 'plus') {
nombre *= 1.2;
}
else if (this.id == 'moins'){
nombre /=1.2;
}
texte.css('fontSize', nombre + unite);
});
});
</script>
```

Le détail de celui-ci :

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

Au chargement du DOM.

```
$('input').click(function(){
```

Au clic sur un des boutons.

```
var texte = $('p');
var taille = texte.css('fontSize');
```


© ENI Editions - All rigths reserved - Jonifar lina

La taille de caractères (`fontSize`) du texte est stockée dans une variable (`taille`).

```
var nombre = parseFloat(taille, 10);
```

La fonction JavaScript `parseFloat()` convertit cette chaîne de caractères sous forme de nombre (base 10).

```
var unite = taille.slice(-2);
```

Avec la fonction `slice()` de jQuery, nous obtenons l'unité utilisée pour la taille de caractères (`px` ou `em`). La valeur négative permet de commencer à partir de la fin de la sélection.

```
if(this.id == 'plus')
{nombre *= 1.2;
}
```

Si le bouton "plus" est cliqué, la taille des caractères est augmentée de 20 %.

```
else if (this.id == 'moins')
{nombre /=1.2;
}
```

Si le bouton "moins" est activé, la taille de caractères est diminuée dans la même proportion.

```
texte.css('fontSize', nombre + unite);
```

La nouvelle valeur de la taille de caractères est attribuée.

```
});
});
```

Fin de script.

Notre fichier devient alors :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js" ></script>
<script type="text/javascript">
$(document).ready(function(){
$('input').click(function(){
var texte = $('p');
var taille = texte.css('fontSize');
var nombre = parseFloat(taille, 10);
var unite = taille.slice(-2);
if(this.id == 'plus') {
nombre *= 1.2;
}
else if (this.id == 'moins'){
nombre /=1.2;
}
texte.css('fontSize', nombre + unite);
});
});
</script>
<style type="text/css">
p { font-size: 1em;}
</style>
</head>
<body>
<input type="button" value="Plus grand" id="plus" />
<input type="button" value="Plus petit" id="moins" />
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<p>Lorem ipsum dolor sit amet, magna adipiscing elit. Quisque at
cursus est. Phasellus id ligula massa. Quisque id lacus mauris.
Suspendisse vehicula tortor eu
magna.</p>
</body>
</html>
```

2. Zoom sur image avec une légende

Soit une page qui présente des images miniatures. Au survol de la souris, l'image est affichée dans sa taille réelle. Le script offre la possibilité de joindre une légende à celle-ci.

Les images de cette application sont disponibles dans l'espace de téléchargement consacré à cet ouvrage.

Le fichier Xhtml de départ :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<style type="text/css">
body { margin: 0px;
       padding: 10px;
       font:80% Arial, sans-serif;}
h2 { clear:both;
     font-size:160%;
     font-weight:normal;
     margin: 0px;
     padding-top: 10px;
     padding-bottom: 15px;}
a { text-decoration:none;
    color:black;}
img { border: 1px solid black;}
ul,li { margin: 0px;
        padding: 0px;}
li { list-style: none;
     float: left;
     display: inline;
     margin-right: 10px;}
#zoom { position: absolute;
        border: 1px solid #333;
        background: #333;
        padding: 3px;
        display: none;
        color: #fff;}
/style>
</head>
<body>
<h2>Effet de Zoom avec légende</h2>
<ul>
<li><a href="ours1.jpg" class="zoom" title="Ours polaire en
danger"><img src="pt_ours1.jpg" alt="" /></a></li>
<li><a href="ours2.jpg" class="zoom" title="Ours polaire en
danger"><img src="pt_ours2.jpg" alt="" /></a></li>
<li><a href="ours3.jpg" class="zoom" title="Ours polaire en
danger"><img src="pt_ours3.jpg" alt="" /></a></li>
</ul>
</body>
</html>
```

Commentaires :

La légende qui apparaît dans l'image agrandie est reprise dans l'attribut title du lien.

Grâce au lien vers l'image dans sa dimension initiale, la page reste accessible aux utilisateurs qui auraient désactivé le JavaScript.

Le script jQuery :

```
<script type="text/javascript">
//<![CDATA[
this.zoom_image = function(){
xOffset = 10;
yOffset = 30;
$("a.zoom").hover(function(e){
this.texte = this.title;
this.title = "";
var legende = (this.texte != "") ? "<br/>" + this.texte : "";
$("body").append("<p id='zoom'><img src='"+ this.href +"'
alt='Visualisation image' />"+ legende +"</p>");
$("#zoom")
.css("top",(e.pageY - xOffset) + "px")
.css("left",(e.pageX + yOffset) + "px")
.fadeIn("slow");
},
function(){
this.title = this.texte;
$("#zoom").remove();
});
$("a.zoom").mousemove(function(e){
```

© ENI Editions - All rigths reserved - Jonifar lina

```
$("#zoom")
.css("top",(e.pageY - xOffset) + "px")
.css("left",(e.pageX + yOffset) + "px");
});
};
$(document).ready(function(){
zoom_image();
});
//]]>
</script>
```

Ce script réclame bien entendu des explications.

```
this.zoom_image = function(){
xOffset = 10;
yOffset = 30;
```

Avant de lancer le jQuery et l'effet qu'il apportera, il faut d'abord mettre en place la fonction `zoom_image`. Commençons par créer deux variables (`xOffset` et `yOffset`) qui définissent le déport horizontal et vertical du pop-up de l'image par rapport au curseur. Ces valeurs peuvent être ajustées par le concepteur de la page.

```
$("a.zoom").hover(function(e){
```

Le script applique la méthode `hover()` aux liens dotés de la classe `zoom`. L'événement associé au curseur est transmis en paramètre à la fonction (`function(e)`).

```
this.texte = this.title;
this.title = "";
```

Au survol de la miniature, le texte de l'attribut `title` est repris dans la variable texte.

```
var legende = (this.texte != "") ? "<br/>" + this.texte : "";
```

Si l'attribut `title` est non vide, la légende de l'image (variable `legende`) est construite à partir du texte de celui-ci.

```
$("body").append("<p id='zoom'><img src='"+ this.href +"'
alt='Visualisation image' />"+ legende +"</p>");
```

Le script insère (`append()`) alors dans le body, un paragraphe avec l'identifiant `zoom` contenant l'image dont l'adresse est fournie par le lien (`img src='"+ this.href`) ainsi que la légende.

```
$("#zoom")
.css("top",(e.pageY - xOffset) + "px")
.css("left",(e.pageX + yOffset) + "px")
.fadeIn("slow");
```

Ce paragraphe (`id='zoom'`) est alors affiché selon les propriétés de style `top` et `left` en ajoutant le déport défini plus haut, à la position du curseur. Un effet de `fadeIn()` a été ajouté.

```
},
```

Fin de la première fonction de la méthode `hover()`.

```
function(){
this.title = this.texte;
$("#zoom").remove();
```

Lorsque le curseur quitte la miniature (`$("#zoom")`), le pop-up de l'image disparaît.

```
});
```

Fin de la seconde fonction de la méthode `hover()`.

```
$("a.zoom").mousemove(function(e){
```

Il faut encore prévoir que l'utilisateur, tout en survolant la miniature (`$("a.zoom")`), est susceptible de faire bouger le

© ENI Editions - All rigths reserved - Jonifar lina

curseur de la souris (`mousemove()`). L'événement associé au curseur est transmis en paramètre à la fonction (`function (e)`).

```
$("#zoom")
.css("top",(e.pageY - xOffset) + "px")
.css("left",(e.pageX + yOffset) + "px");
});
```

Le pop-up contenant l'image suit les mouvements du curseur.

```
};
```

Fin du hover().

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

Toutes les fonctions étant à présent définies, jQuery peut mettre en œuvre la fonction `zoom_image()` au chargement du DOM.

Le fichier Html final est alors :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script src="jquery.js" type="text/javascript"></script>
<script type="text/javascript">
//<![CDATA[
this.zoom_image = function(){
xOffset = 10;
yOffset = 30;
$("a.zoom").hover(function(e){
this.texte = this.title;
this.title = "";
var legende = (this.texte != "") ? "<br/>" + this.texte : "";
$("body").append("<p id='zoom'><img src='"+ this.href +"'
alt='Visualisation image' />"+ legende +"</p>");
$("#zoom")
.css("top",(e.pageY - xOffset) + "px")
.css("left",(e.pageX + yOffset) + "px")
.fadeIn("slow");
},
function(){
this.title = this.texte;
$("#zoom").remove();
});
$("a.zoom").mousemove(function(e){
$("#zoom")
.css("top",(e.pageY - xOffset) + "px")
.css("left",(e.pageX + yOffset) + "px");
});
};
$(document).ready(function(){
zoom_image();
});
//]]>
</script>
<style type="text/css">
body { margin: 0px;
       padding: 10px;
       font:80% Arial, sans-serif;}
h2 { clear:both;
```

```
     font-size:160%;
     font-weight:normal;
     margin: 0px;
     padding-top: 10px;
     padding-bottom: 15px;}
a { text-decoration:none;
    color:black;}
img { border: 1px solid black;}
ul,li { margin: 0px;
    padding: 0px;}
li { list-style: none;
     float: left;
     display: inline;
     margin-right: 10px;}
#zoom { position: absolute;
        border: 1px solid #333;
        background: #333;
        padding: 3px;
        display: none;
        color: #fff;}
</style>
</head>
<body>
<h2>Effet de Zoom avec légende</h2>
<ul>
<li><a href="ours1.jpg" class="zoom" title="Ours polaire en
danger"><img src="pt_ours1.jpg" alt="" /></a></li>
<li><a href="ours2.jpg" class="zoom" title="Ours polaire en
danger"><img src="pt_ours2.jpg" alt="" /></a></li>
<li><a href="ours3.jpg" class="zoom" title="Ours polaire en
danger"><img src="pt_ours3.jpg" alt="" /></a></li>
</ul>
</body>
</html>
```

3. Une infobulle avec jQuery

© ENI Editions - All rigths reserved - Jonifar lina

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
body { margin: 0;
       padding-left: 10px;
       padding-right: 10px;
       font: 90% Arial, sans-serif;}
a { text-decoration: none;
    border-bottom: 1px dashed black;
    color: black;}
#infobulle { position: absolute;
             border: 1px solid black;
             background: #9cf;
             padding:2px 5px;
             display: none;}
</style>
</head>
<body>
<p>jQuery est un framework <a href="#" class="infobulle"
title="Langage de scripts utilisé dans les pages web
interactives">JavaScript</a> libre qui porte sur l'interaction
entre le Html et le JavaScript (comprenant le <a href="#"
class="infobulle" title="Document Object Model">DOM</a> et l'<a
href="#" class="infobulle" title="Asynchronous JavaScript and
XML">AJAX </a>).</p>
<p><img border="0" src="jquery-logo.gif" /></p>
<p>Source : <a href="#" class="infobulle" title="Encyclopédie
universelle disponible sur le Web">Wikipedia</a></p>
</body>
</html>
```

Remarquons que l'attribut title du lien contient le texte de l'infobulle.

Le script :

```
<script type="text/javascript">
//<![CDATA[
this.popup = function(){
xOffset = 10;
```

© ENI Editions - All rigths reserved - Jonifar lina

```
yOffset = 20;
$("a.infobulle").hover(function(e){
this.texte = this.title;
this.title = "";
$("body").append("<p id='infobulle'>"+ this.texte +"</p>");
$("#infobulle")
.css("top",(e.pageY - xOffset) + "px")
.css("left",(e.pageX + yOffset) + "px")
.fadeIn("fast");
},
function(){
this.title = this.texte;
$("#infobulle").remove();
});
$("a.infobulle").mousemove(function(e){
$("#infobulle")
.css("top",(e.pageY - xOffset) + "px")
.css("left",(e.pageX + yOffset) + "px");
});
};
$(document).ready(function(){
popup();
});
//]]>
</script>
```

Détaillons ce script qui est assez proche de l'application précédente.

```
this.popup = function(){
xOffset = 10;
yOffset = 20;
```

Mise en place de la fonction `popup` qui définit le déport horizontal et vertical.

```
$("a.infobulle").hover(function(e){
```

Au survol d'un lien avec une classe infobulle (`$("a.infobulle")`), une méthode jQuery `hover()` est appliquée.

```
this.texte = this.title;
this.title = "";
$("body").append("<p id='infobulle'>"+ this.texte +"</p>");
$("#infobulle")
.css("top",(e.pageY - xOffset) + "px")
.css("left",(e.pageX + yOffset) + "px")
.fadeIn("fast");
},
```

Lorsque le curseur est positionné sur le lien, le contenu de l'attribut `title` du lien est repris dans la variable texte. Le script ajoute (`append()`) au corps (`body`) du document, un paragraphe dont l'identifiant est `infobulle` contenant la variable `texte`. Celui-ci est ensuite affiché dans la page avec les coordonnées `top` et `left`. Cet affichage s'effectue avec un effet (`fadeIn("fast")`).

```
function(){
this.title = this.texte;
$("#infobulle").remove();
});
```

Lorsque le curseur quitte le lien, l'infobulle est enlevée (`remove()`).

```
$("a.infobulle").mousemove(function(e){
$("#infobulle")
.css("top",(e.pageY - xOffset) + "px")
.css("left",(e.pageX + yOffset) + "px");
});
};
```

Prévoyons le cas où l'utilisateur bouge légèrement le curseur (`mousemove()`) alors qu'il est positionné sur le lien. Ceci

© ENI Editions - All rigths reserved - Jonifar lina

permet au script d'accompagner ces mouvements.

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

Toutes les fonctions étant définies, le script jQuery peut activer la fonction `popup()`.

Le document final :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script src="jquery.js" type="text/javascript"></script>
<script type="text/javascript">
//<![CDATA[
this.popup = function(){
xOffset = 10;
yOffset = 20;
$("a.infobulle").hover(function(e){
this.texte = this.title;
this.title = "";
$("body").append("<p id='infobulle'>"+ this.texte +"</p>");
$("#infobulle")
.css("top",(e.pageY - xOffset) + "px")
.css("left",(e.pageX + yOffset) + "px")
.fadeIn("fast");
},
function(){
this.title = this.texte;
$("#infobulle").remove();
});
$("a.infobulle").mousemove(function(e){
$("#infobulle")
.css("top",(e.pageY - xOffset) + "px")
.css("left",(e.pageX + yOffset) + "px");
});
};
$(document).ready(function(){
popup();
});
//]]>
</script>
<style type="text/css">
body { margin: 0;
       padding-left: 10px;
       padding-right: 10px;
       font: 90% Arial, sans-serif;}
a { text-decoration: none;
    border-bottom: 1px dashed black;
    color: black;}
#infobulle { position: absolute;
             border: 1px solid black;
             background: #9cf;
             padding:2px 5px;
             display: none;}
</style>
</head>
<body>
<p>jQuery est un framework <a href="#" class="infobulle"
title="Langage de scripts utilisé dans les pages web
interactives">JavaScript</a> libre qui porte sur l'interaction
entre le Html et le JavaScript (comprenant le <a href="#"
class="infobulle" title="Document Object Model">DOM</a> et l'<a
```

© ENI Editions - All rigths reserved - Jonifar lina

```
href="#" class="infobulle" title="Asynchronous JavaScript and
XML">AJAX </a>).</p>
<p><img border="0" src="jquery-logo.gif" alt="" /></p>
<p>Source : <a href="#" class="infobulle" title="Encyclopédie
universelle disponible sur le Web">Wikipedia</a></p>
</body>
</html>
```


© ENI Editions - All rigths reserved - Jonifar lina

Introduction

Les événements sont les déclencheurs de l'interactivité apportée par le JavaScript. Ceux-ci permettent d'associer à des actions de l'utilisateur, des fonctions et des méthodes. Avec la librairie jQuery, le principe reste le même mais certains gestionnaires d'événements sont adaptés et des méthodes nouvelles introduites.

© ENI Editions - All rigths reserved - Jonifar lina

Les gestionnaires d'événements

La librairie jQuery met à disposition des gestionnaires d'événements assez similaires à ceux du JavaScript traditionnel. Ainsi au `onmouseover` de JavaScript correspond `mouseover` en jQuery. Le préfixe "on" a simplement disparu.

1. Au clic de la souris

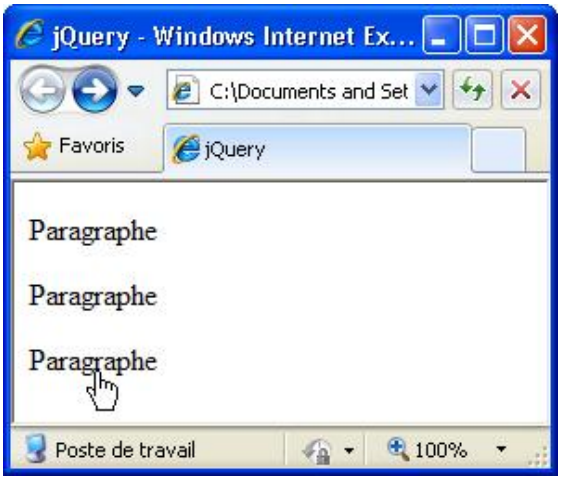
click()

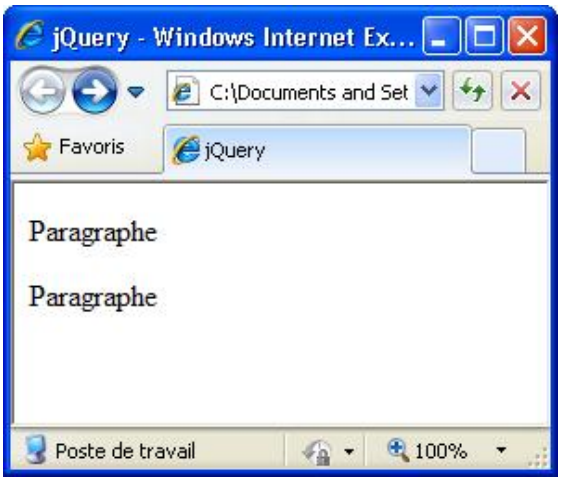
Associe une fonction à l'événement clic des éléments de la sélection.

```
$("p").click();
```

Exemple

Au clic d'un paragraphe, celui-ci disparaît.






```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("p").click(function () {
```

© ENI Editions - All rigths reserved - Jonifar lina

```
$(this).slideUp();
});
});
</script>
<style type="text/css">
p { cursor: pointer;}
</style>
</head>
<body>
<p>Paragraphe</p>
<p>Paragraphe</p>
<p>Paragraphe</p>
</body>
</html>
```

Voyons le script jQuery :

```
$(document).ready(function(){
$("p").click(function () {
```

Au clic sur les paragraphes, la fonction suivante est exécutée.

```
$(this).slideUp();
```

L'élément cliqué (`this`) disparaît en glissant vers le haut.

```
});
});
```

Fin de script.

2. Au double clic

dblclick( )

Associe une fonction à l'événement double clic des éléments de la sélection.

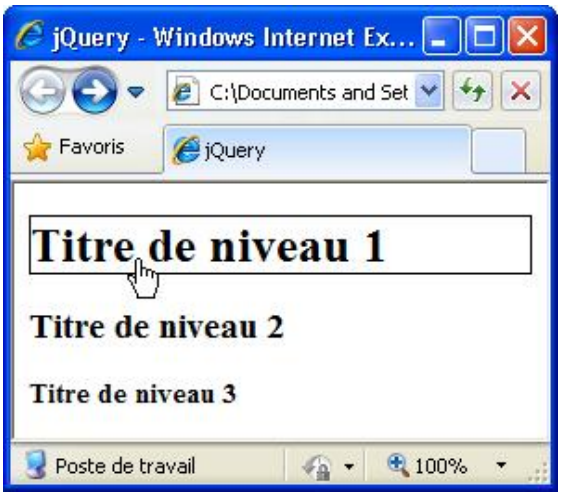
```
$("div").dblclick();
```

Le Web étant le royaume de clic simple, le double clic n'est finalement que peu répandu sur la toile.

Exemple

Au clic d'un titre, celui-ci est entouré d'une bordure.

© ENI Editions - All rigths reserved - Jonifar lina




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("h1, h2").dblclick(function () {
$(this).toggleClass('bordure');
});
});
</script>
<style type="text/css">
body { font: arial, sans-serif;
       font-size: 0.8em; }
.bordure { border: 1px solid black;}
h1,h2 { cursor: pointer;}
</style>
</head>
<body>
<h1>Titre de niveau 1</h1>
<h2>Titre de niveau 2</h2>
<h3>Titre de niveau 3</h3>
</body>
</html>
```

Un coup d'œil au script :

```
$(document).ready(function(){
$("h1, h2").dblclick(function () {
```

Au double clic d'un titre de niveau 1 et 2.

```
$(this).toggleClass('bordure');
```

La classe `bordure` est tantôt activée, tantôt désactivée (`toggleClass()`).

```
});
});
```

Fin du script.

3. Le focus

© ENI Editions - All rigths reserved - Jonifar lina

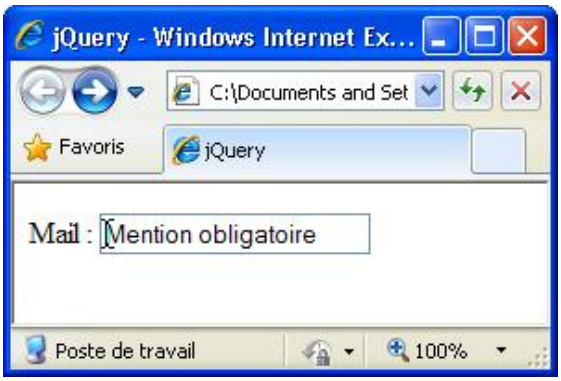
focus()

Associe une fonction à l'événement au focus des éléments spécifiés.

```
$("p").focus();
```

Exemple

Au focus d'une ligne de texte, ajoutons la valeur `Mention obligatoire`.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#mail").focus(function () {
$(this).val("Mention obligatoire");
});
});
</script>
</head>
<body>
<p>Mail : <input id="mail" type="text" value="" /></p>
</body>
</html>
```

Quelques explications concernant le script :

```
$(document).ready(function(){
$("#mail").focus(function () {
```

Au focus de la ligne de texte `mail`.

© ENI Editions - All rigths reserved - Jonifar lina

```
$(this).val("Mention obligatoire");
```

Ajoutons à celle-ci la valeur (`val("Mention obligatoire")`).

```
});
});
```

Fin de script.

4. La perte du focus

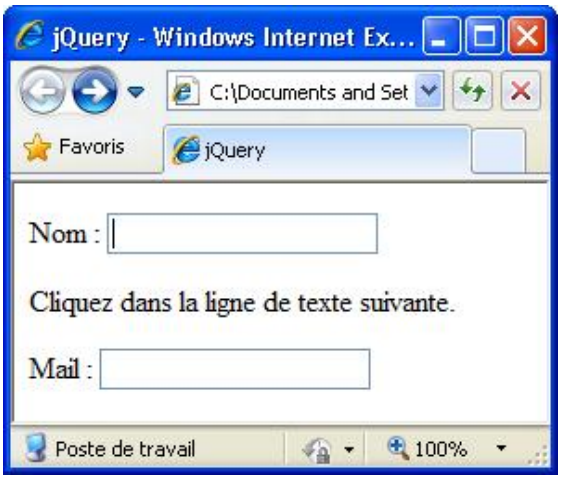
blur()

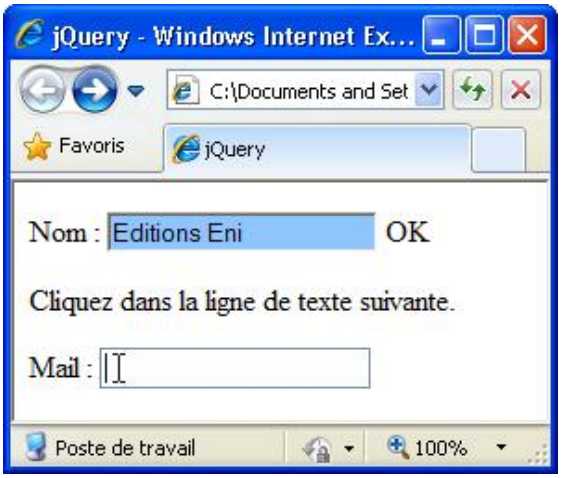
Déclenche l'événement qui se produit lorsque l'élément perd le focus. Cela a pour effet de déclencher toutes les fonctions associées à cet événement pour les éléments sélectionnés.

```
$("input").blur();
```

Exemple

Après l'encodage du champ relatif au nom et la perte du focus de celui-ci, l'arrière-plan est colorié et la mention "OK" est ajoutée.






```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#nom").focus();
$("#nom").blur(function () {
$(this).css({"background-color":"#9cf"});
$("p:first").append(" OK");
});
});
</script>
</head>
<body>
<p>Nom : <input id="nom" type="text" value="" /></p>
<p>Cliquez dans la ligne de texte suivante.</p>
<p>Mail : <input id="mail" type="text" value="" /></p>
</body>
</html>
```

Quelques détails relatifs au script :

```
$(document).ready(function(){
$("#nom").focus();
```

Le script donne d'abord le focus à la ligne de texte relative au nom.

```
$("#nom").blur(function () {
$(this).css({"background-color":"#9cf"});
$("p:first").append(" OK");
```

À la perte du focus de celle-ci, l'arrière-plan de la ligne de texte est colorié et la mention "OK" est ajoutée.

```
});
});
```

Fin de script.

5. La barre de défilement

scroll()

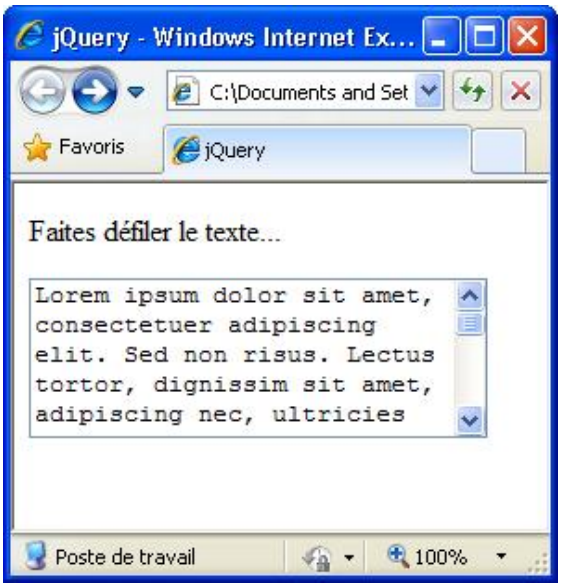
Associe une fonction à l'utilisation de la barre de défilement d'un élément.

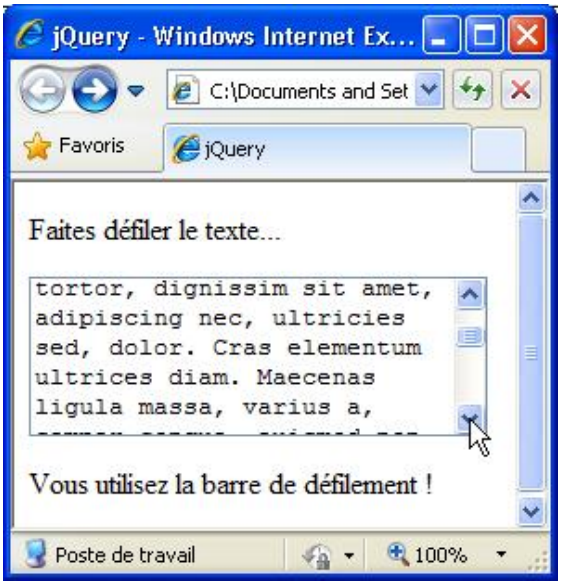
```
$(window).scroll();
```

Exemple

Un texte apparaît lorsque l'utilisateur utilise la barre de défilement.

© ENI Editions - All rigths reserved - Jonifar lina






```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("textarea").scroll(function () {
$("span").css({"display": "inline"}).fadeOut("slow");
});
});
</script>
<style type="text/css">
span { display:none;}
</style>
</head>
<body>
<p>Faites défiler le texte...</p>
<textarea cols="28" rows="5">Lorem ipsum dolor sit amet,
consectetuer adipiscing elit. Sed non risus. Lectus tortor,
dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras
```

© ENI Editions - All rigths reserved - Jonifar lina

```
elementum ultrices diam. Maecenas ligula massa, varius a, semper
congue, euismod non, mi. Proin porttitor, orci nec nonummy
molestie, enim est eleifend mi, non fermentum diam nisl sit amet
erat. Duis semper.</textarea>
<p><span>Vous utilisez la barre de défilement !</span></p>
</body>
</html>
```

Le script expliqué :

```
$(document).ready(function(){
$("textarea").scroll(function () {
```

À l'utilisation de la barre de défilement de la zone de texte (`<textarea>`),

```
$("span").css({"display": "inline"}).fadeOut("slow");
```

Le texte de la balise `<span>` est affiché avec un effet.

```
});
});
```

Fin de script.

6. Le bouton de la souris

mousedown()

Associe une fonction pour les éléments sélectionnés lorsque l'utilisateur presse une touche de la souris.

```
$("p").mousedown();
```

mouseup()

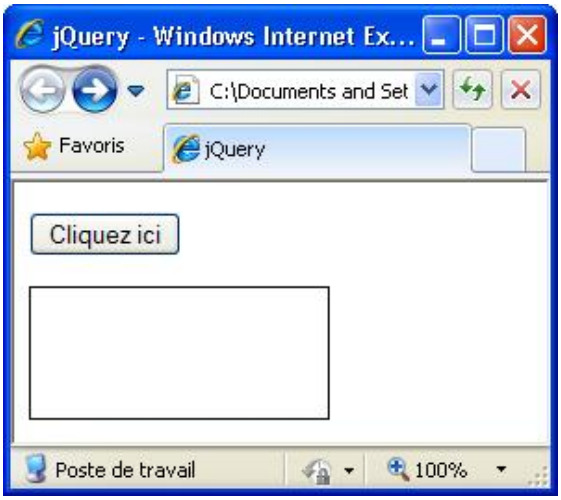
Associe une fonction pour les éléments de la sélection lorsque l'utilisateur relâche le bouton de la souris.

```
$("p").mouseup();
```

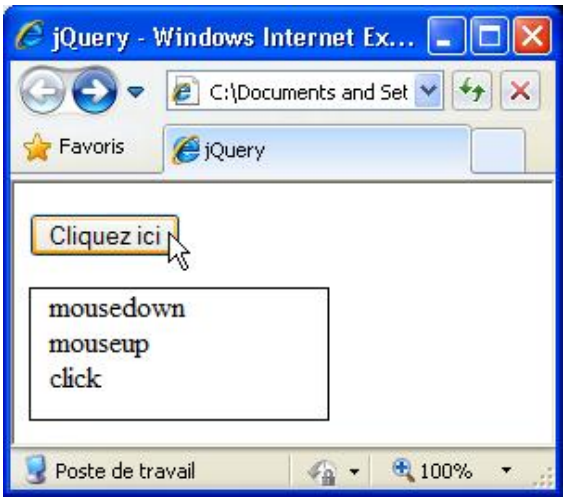
L'événement clic est activé lorsqu'un `mousedown` et un `mouseup` est détecté.

Exemple

Au clic dans une division, détaillons les événements `mousedown`, `mouseup` et `click`.



© ENI Editions - All rigths reserved - Jonifar lina




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
//<![CDATA[
$(document).ready(function() {
$("input").mousedown(function(){
$("#sortie").append("mousedown<br>");
});
$("input").mouseup(function(){
$("#sortie").append("mouseup<br>");
});
$("input").click(function(){
$("#sortie").append("click<br>");
});
});
//]]>
</script>
<style type="text/css">
div { width: 150px;
      height: 70px;
      overflow: auto;
      border: solid 1px black;
      padding-left: 10px;}
</style>
</head>
<body>
<p><input type="button" value="Cliquez ici" /></p>
<div id="sortie"></div>
</body>
</html>
```

Explorons le script jQuery :

```
$(document).ready(function() {
$("input").mousedown(function(){
$("#sortie").append("mousedown<br>");
});
```

À la pression du bouton de la souris (`mousedown()`), le mot "mousedown" est inséré dans la division identifiée par `sortie`.

```
$("input").mouseup(function(){
$("#sortie").append("mouseup<br>");
```

© ENI Editions - All rigths reserved - Jonifar lina

```
});
```

Au relâchement du bouton de la souris (`mouseup()`), le mot "mouseup" est inséré dans la division.

```
$("input").click(function(){
$("#sortie").append("click<br>");
});
```

Lorsque l'événement `click` est détecté, c'est le mot "clic" qui est ajouté.

```
});
```

Fin de script.

7. Le déplacement du curseur

Détecter le moindre mouvement du curseur est un événement intéressant mais il faut être conscient que cela entraîne une forte consommation des ressources du système. En effet, les actions associées au mouvement de la souris se déclenchent dès que le curseur bouge d'un pixel. Ce processus peut diminuer l'usabilité d'un site, en particulier pour les ordinateurs d'ancienne génération.

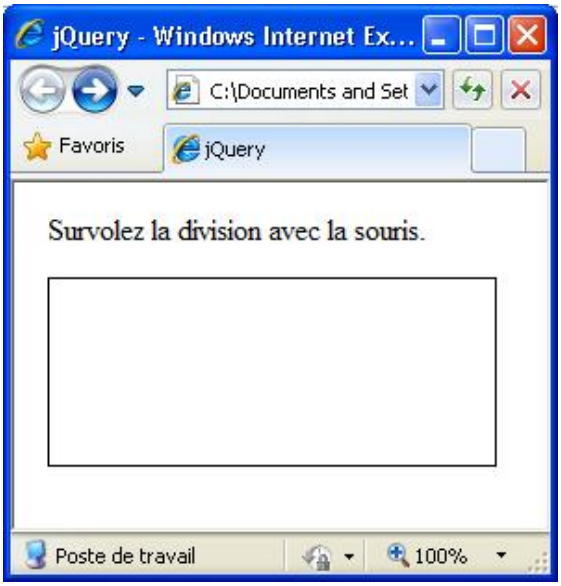
mousemove()

Associe une fonction lorsque l'utilisateur bouge le curseur de la souris.

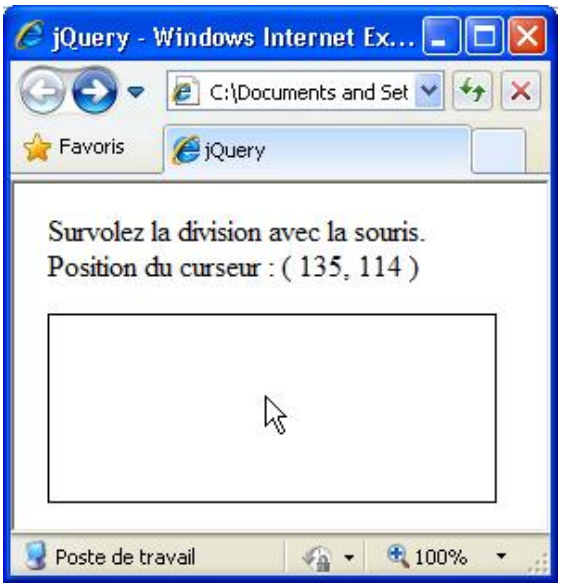
```
$("div").mousemove();
```

Exemple

Retrouvons les coordonnées du curseur de la souris à chacun de ses mouvements.



© ENI Editions - All rigths reserved - Jonifar lina




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("div").mousemove(function(e){
var pageCoords = "( " + e.pageX + ", " + e.pageY + " )";
$("span:first").text("Position du curseur :  " + pageCoords);
});
$("div").mouseout(function(e){
$("span:first").text("");
});
});
</script>
<style type="text/css">
div { width:240px;
      height:100px;
      margin-left:10px;
      border:1px solid black; }
p { margin-left:10px; }
span { display:block; }
</style>
</head>
<body>
<p>Survolez la division avec la souris.
<span> </span>
</p>
<div></div>
</body>
</html>
```

Le script :

```
$(document).ready(function(){
$("div").mousemove(function(e){
```

Au mouvement du curseur dans la division.

```
var pageCoords = "( " + e.pageX + ", " + e.pageY + " )";
```

La position horizontale et verticale du curseur est stockée dans la variable `pageCoords`.

© ENI Editions - All rigths reserved - Jonifar lina

```
$("span:first").text("Position du curseur : " + pageCoords);
});
```

La position est affichée comme du texte dans la première balise .

```
$("div").mouseout(function(e){
$("span:first").text("");
});
```

Lorsque le curseur quitte la division (`mouseout()`), plus rien n'est affiché (`text("")`).

```
});
```

Fin de script.

8. L'entrée et la sortie du curseur

Les habitués du JavaScript retrouvent les deux "compères" ; `onmouseover` et `onmouseout`.

mouseover()

Associe une fonction lorsque l'utilisateur positionne le curseur de la souris au-dessus d'un élément.

```
$("div").mouseover();
```

mouseout()

Associe une action à l'événement lorsque le curseur de la souris quitte un élément.

```
$("div").mouseout();
```

De façon (a priori) assez similaire, jQuery ajoute les événements `mouseenter` et `mouseleave`.

mouseenter()

Exécute une fonction lorsque le pointeur entre dans un élément.

```
$("div").mouseenter();
```

mouseleave()

Déclenche une fonction lorsque le pointeur quitte un élément.

```
$("div").mouseleave();
```

Cependant le fonctionnement de ces gestionnaires d'événements est assez différent. En effet, `mouseover`, hérité du JavaScript `onmouseover`, propage l'événement à la hiérarchie des objets de la page. Ce phénomène est connu sous le terme propagation des événements ou débordement des événements. Au contraire, `mouseenter` évite cette propagation de l'événement.

Il en est de même pour `mouseout` qui se déclenche à chaque fois que le pointeur se déplace vers ou à partir d'un élément enfant. À l'inverse, `mouseleave` ne se déclenche qu'une fois, soit lorsque le pointeur quitte l'élément en cours.

Dans certaines situations de programmation, cette propagation peut se révéler gênante et jQuery y apporte une solution.

Illustrons cette différence, assez ténue, entre le `mouseover` et le `mouseenter` par des exemples.

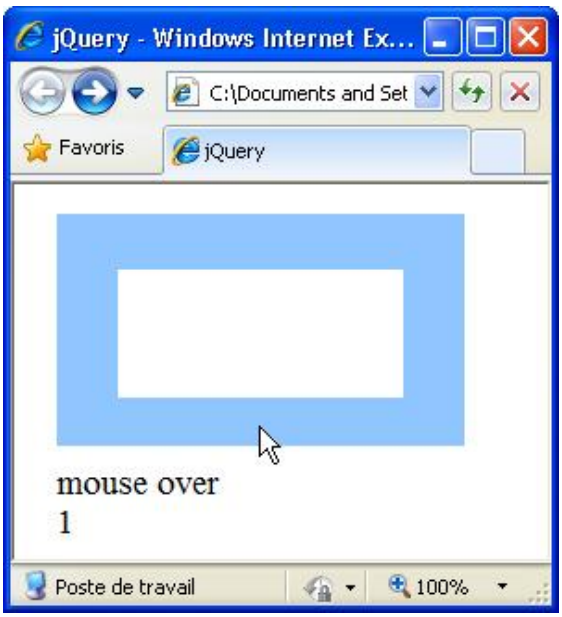
Prenons tout d'abord le cas de `mouseover`.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
```


© ENI Editions - All rigths reserved - Jonifar lina

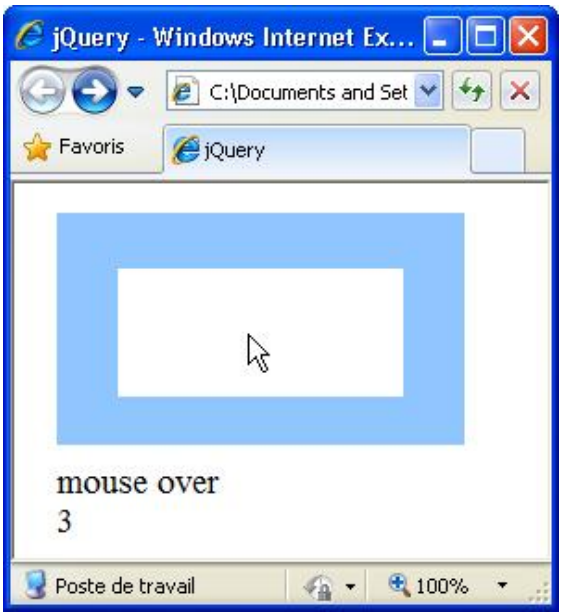
```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
var i = 0;
$("div.overout").mouseover(function(){
$("#afficher").text("mouse over");
$("#compteur").text(++i);
})
.mouseout(function(){
$("#afficher").text("mouse out");
});
});
</script>
<style type="text/css">
div.out { width: 220px;
          height: 125px;
          margin-left: 15px;
          background-color: #9cf;}
div.in { width: 70%;
         height: 55%;
         background-color: white;
         margin: 15px auto;}
p { line-height:0.9em;
    padding:0;}
#afficher { margin-top: 10px;
            margin-left: 15px;
            font-size: 1.2em;}
#compteur { margin-left: 15px;
            font-size: 1.2em;}
</style>
</head>
<body>
<div class="out overout"><p> </p>
<div class="in overout"></div>
</div>
<div id="afficher"></div>
<div id="compteur"></div>
</body>
</html>
```

Lorsque le curseur de la souris survole (`mouseover`) la division coloriée, le compteur implémenté dans le script, indique tout naturellement le chiffre 1.



Par contre, lorsque le curseur de la souris survole ensuite la division blanche (élément enfant de la division coloriée), le compteur marque non pas 2 mais le chiffre 3. L'événement du survol de l'élément enfant s'est propagé à l'événement du survol de l'élément parent. On dénombre ainsi 3 événements de survol.

© ENI Editions - All rigths reserved - Jonifar lina



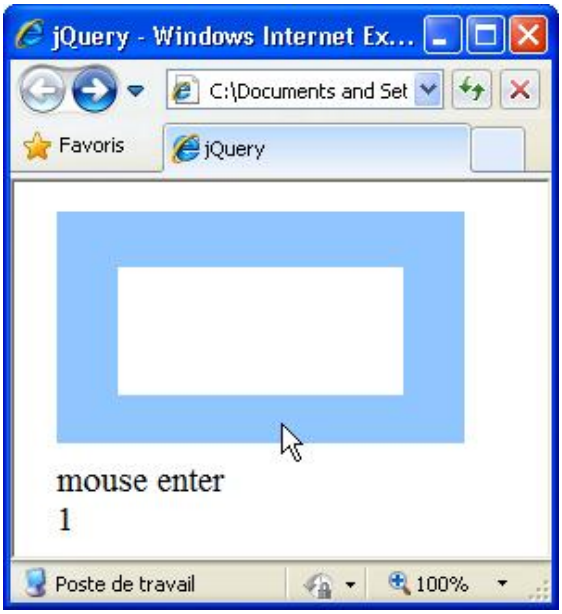
Reprenons le même code, mais cette fois-ci avec `mouseenter` et `mouseleave`.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
var i = 0;
$("div.overout").mouseenter(function(){
$("#afficher").text("mouse enter");
$("#compteur").text(++i);
})
.mouseleave(function(){
$("#afficher").text("mouse leave");
});
});
</script>
<style type="text/css">
div.out { width: 220px;
          height: 125px;
          margin-left: 15px;
          background-color: #9cf;}
div.in { width: 70%;
         height: 55%;
         background-color: white;
         margin: 15px auto;}
p { line-height:0.9em;
    padding:0;}
#afficher { margin-top: 10px;
            margin-left: 15px;
            font-size: 1.2em;}
#compteur { margin-left: 15px;
            font-size: 1.2em;}
</style>
</head>
<body>
<div class="out overout"><p> </p>
<div class="in overout"></div>
</div>
<div id="afficher"></div>
<div id="compteur"></div>
```

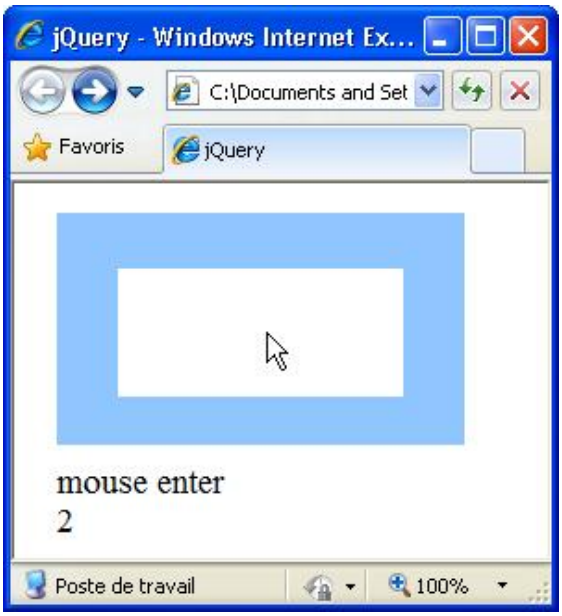

© ENI Editions - All rigths reserved - Jonifar lina

```
</body>
</html>
```

Lorsque le curseur de la souris entre (`mouseenter`) dans la division coloriée, le compteur implémenté dans le script, indique tout naturellement le chiffre 1.



Et lorsque le curseur de la souris entre ensuite dans la division blanche (élément enfant de la division coloriée), le compteur marque le chiffre 2. L'événement `mouseenter` ne crée pas de propagation d'événement.



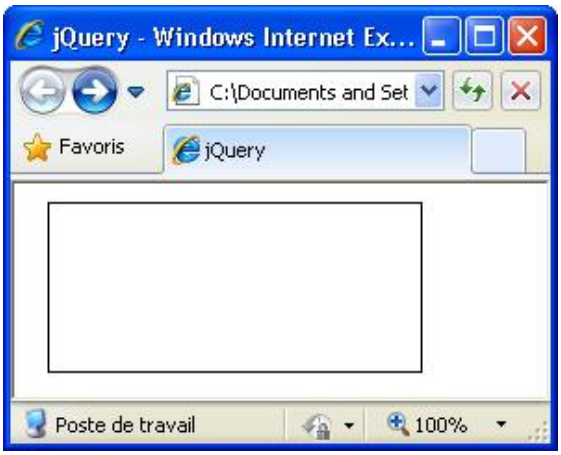
Ce débordement d'événement induit des effets indésirables lorsqu'il y a des éléments parents avec des enfants et qu'un `mouseover` ou `mouseout` est appliqué à l'élément parent. Le mouvement de la souris sur les éléments enfants peut déclencher un événement `mouseout` sur leurs éléments parent. L'événement commence par l'élément enfant et remonte de parent en parent. La propagation est ascendante.

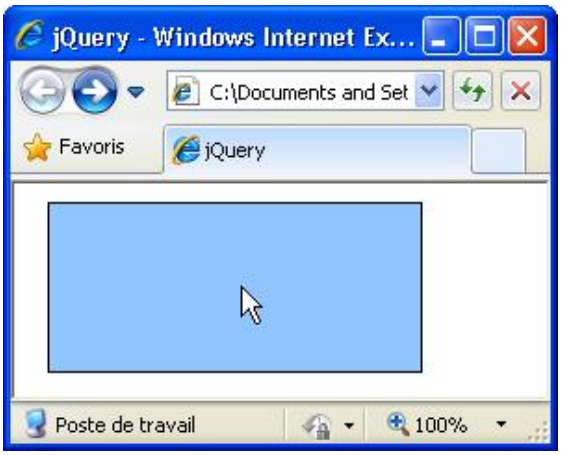
Pour en savoir plus, consultez Google avec les mots-clés "propagation d'événement", "débordement d'événement" ou "event bubling".

Revenons à nos événements jQuery par un exemple de `mouseenter` et `mouseleave`.

Au survol de la division, un arrière-plan de couleur est appliqué. Lorsque le curseur quitte celle-ci, la situation initiale est rétablie.

© ENI Editions - All rigths reserved - Jonifar lina






```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("div").mouseenter(function(){
$(this).css({"background-color":"#9cf"});
})
.mouseleave(function(){
$(this).css({"background-color":"white"});
});
});
</script>
<style type="text/css">
div { width: 200px;
      height: 90px;
      margin: 10px;
      border: 1px solid black;}
</style>
</head>
<body>
<div></div>
</body>
</html>
```

Explications du script.

```
$(document).ready(function(){
$("div").mouseenter(function(){
```


© ENI Editions - All rigths reserved - Jonifar lina

```
$(this).css({"background-color":"#9cf"});
})
```

Au survol de la division par la souris (mouseenter()), la propriété CSS d'arrière-plan de celle-ci est modifiée (css ({"background-color":"#9cf"}))

```
.mouseleave(function(){
$(this).css({"background-color":"white"});
});
```

Lorsque le curseur quitte la division (mouseleave()), la couleur d'arrière-plan est remise sur le blanc (css ({"background-color":"white"})).

```
});
```

Fin de script.

Commentaire

Comme dans cet exemple, la division n'a pas d'élément enfant, l'utilisation de mouseover et mouseout aurait eu le même résultat.

9. Soumettre une requête

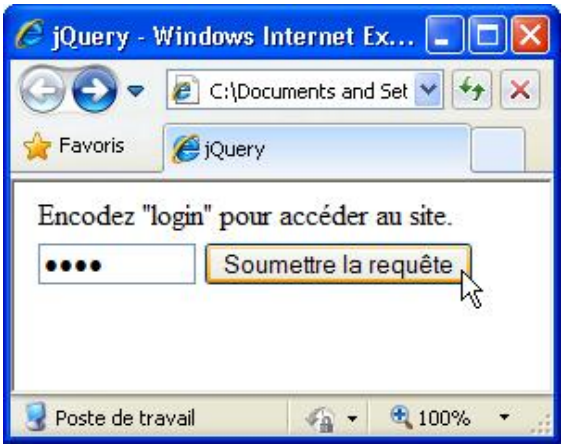
submit()

Déclenche l'événement d'envoi du formulaire.

```
$("form").submit();
```

Exemple

Un formulaire réclame un mot de passe (login en l'occurrence) pour accéder au site. À la soumission, si le mot de passe est correct, l'utilisateur est dirigé vers l'adresse souhaitée. Sinon, un message d'erreur est affiché.



© ENI Editions - All rigths reserved - Jonifar lina

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("form").submit(function() {
if ($("input:first").val() == "login") {
return true;
}
$("span").html("<b>Mot de passe non valide.</b><br>Recommencez
!").show();
return false;
});
});
//]]>
</script>
<style type="text/css">
div,p,span { margin: 5px 0 0 5px;}
span { display: block;}
</style>
</head>
<body>
<p>Encodez "login" pour accéder au site.</p>
<form action="javascript:window.location.href='http://www.google.fr';">
<div>
<input type="password" size="10" />
<input type="submit" />
</div>
</form>
<span></span>
</body>
</html>
```

Passons le script en revue.

```
$(document).ready(function(){
$("form").submit(function() {
```

À la soumission du formulaire.

```
if ($("input:first").val() == "login") {
return true;
}
```

Si le mot de passe introduit est conforme, l'action définie dans le code est effectuée.

© ENI Editions - All rigths reserved - Jonifar lina

```
$("span").html("<b>Mot de passe non valide.</b><br>Recommencez !").show();
return false;
```

Si le mot de passe n'est pas conforme, un message d'erreur est inséré comme du Html dans la balise et celle-ci est affichée.

```
});
});
```

Fin du script.

10. Les autres événements

Il existe encore d'autres événements mais d'un emploi plus rare. Parcourons ceux-ci plus rapidement.

- **change()** : déclenche un événement lorsqu'un contrôle de formulaire est modifié, par exemple lorsqu'une case à cocher de formulaire est activée.
- **error()** : cet événement est déclenché lorsqu'une erreur survient dans le script.
- **keydown()**, **keyup()** et **keypress()** : déclenchent un événement lorsqu'une touche du clavier est pressée (vers le bas), lorsqu'une touche du clavier est relâchée (vers le haut) et lorsqu'un caractère est encodé.
- **load()** et **unload()** : se produit lorsque la page en cours est chargée ou quittée.
- **resize()** : associe un événement lorsque la taille d'un élément, généralement la fenêtre du navigateur, est modifiée.
- **select()** : se produit lorsque l'utilisateur sélectionne un texte (ou une partie de celui-ci). Parfois appliqué aux champs de formulaire du type ligne de texte ou zone de texte (textarea).

Et n'oublions pas l'événement à la base de tout script jQuery, soit **ready()** qui associe une fonction lorsque le DOM est prêt à être traversé et manipulé.

© ENI Editions - All rigths reserved - Jonifar lina

Méthodes ou gestionnaires d'événements avancés

Vous pouvez programmer en jQuery pendant de longues années en utilisant simplement les gestionnaires d'événements passés en revue dans la section Les gestionnaires d'événements de ce chapitre. Cependant, si vous désirez tirer le maximum des possibilités de jQuery et vous lancer dans des scripts plus complexes, il vous faudra alors assimiler les méthodes et gestionnaires d'événements avancés de jQuery.

1. Lier un événement à un objet

La méthode `bind()` est plus flexible et plus puissante que les événements spécifiques comme `click()` ou `mouseover()` parcourus ci-avant. La méthode permet non seulement d'affecter un ou plusieurs événements à un objet jQuery sur lequel sera exécutée une fonction passée en paramètre mais également de transmettre des données à cette fonction. Ainsi un clic sur un lien ou le survol d'une image peut attribuer des informations différentes au gestionnaire d'événement. La fonction liée à l'événement pourra alors s'exécuter différemment selon le contexte fourni par les données.

bind(événement, [données], fonction)

Affecte l'événement à un élément donné.

- événement (chaîne de caractères) : l'événement associé. Si plusieurs événements sont spécifiés, ils seront simplement séparés par un espace.
- données (optionnel) : les données éventuellement fournies à la fonction.
- fonction : le code à exécuter au déclenchement de l'événement.

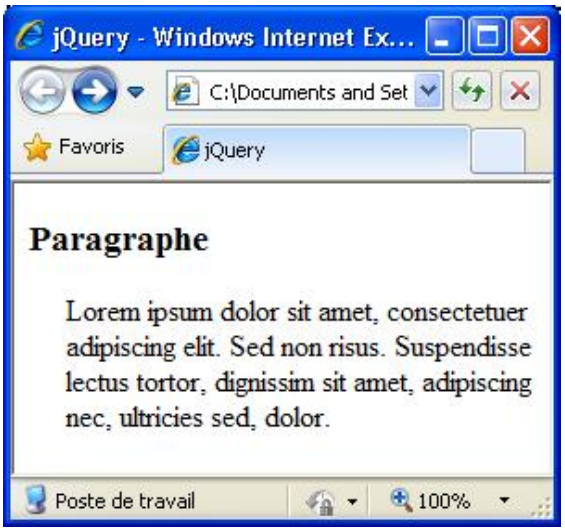
```
$("a#lien3").bind("mouseover mouseout", function(e){
$(this).text( "-- " + $(this).text() + " --" );
});
```

Cette méthode retourne un objet jQuery.

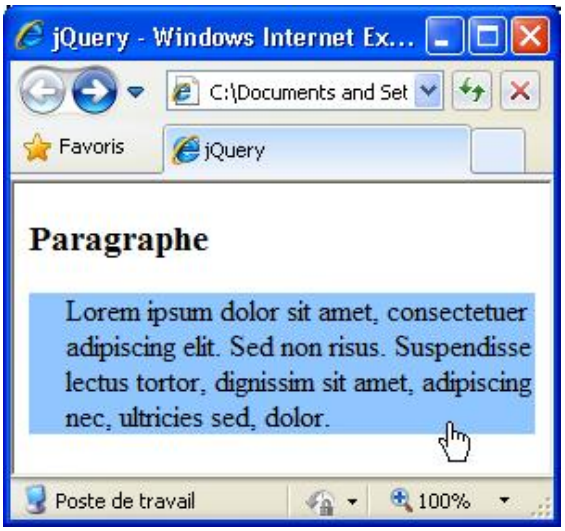
La méthode `unbind()` supprime les actions associées à un événement par la méthode `bind()`.

Exemple

Relions les événements `mouseenter` et `mouseleave` à la balise de paragraphe. Cette nouvelle association a comme action de mettre ou non un arrière-plan de couleur.



© ENI Editions - All rigths reserved - Jonifar lina




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("p").bind("mouseenter mouseleave", function(e){
$(this).toggleClass("over");
});
});
</script>
<style type="text/css">
p { cursor:pointer;
    padding-left:20px;}
p.over { background: #9cf;}
</style>
</head>
<body>
<h3>Paragraphe</h3>
<p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor.</p>
</body>
</html>
```

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

Au chargement du DOM et l'initialisation de jQuery.

`$("p").bind("mouseenter mouseleave", function(e){`

Les événements `mouseenter` et `mouseleave` sont liés (`bind()`) aux paragraphes `<p>`.

```
$(this).toggleClass("over");
});
```

L'action associée est de basculer (`toggleClass()`) la classe `over`.

`});`

Fin du ready.

© ENI Editions - All rigths reserved - Jonifar lina

2. Exécuter une fonction une seule fois

one(événement,[données],fonction)

Associe une fonction à un événement donné. La différence avec la fonction `bind()` est que la fonction associée à l'événement ne sera exécutée au maximum qu'une seule fois pour chaque élément de la sélection.

- événement (chaîne de caractères) : type d'événement concerné.
- données [optionnel] : données supplémentaires à passer au gestionnaire d'événements.
- fonction : fonction à associer à l'événement.

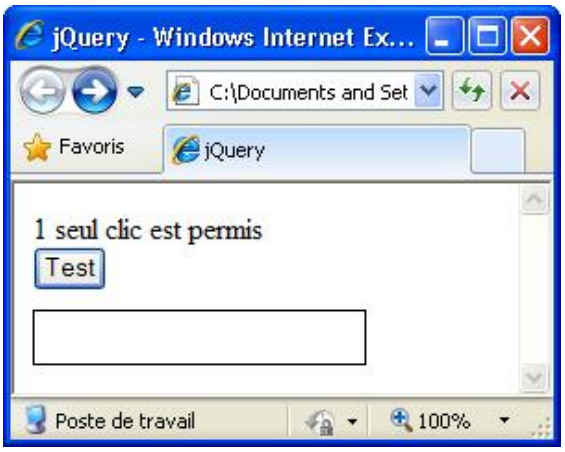
```
$("p").one("click", function(){
alert( $(this).text() );
});
```

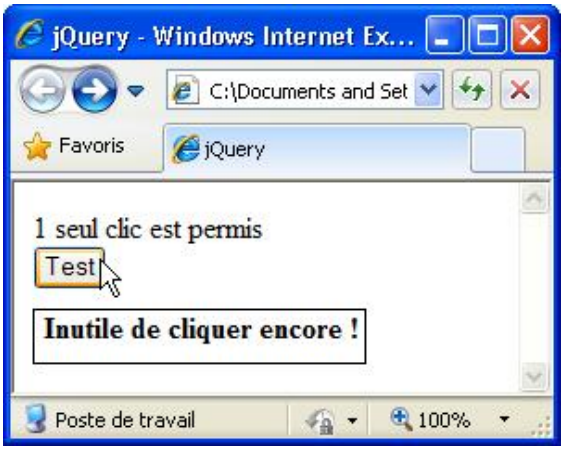
Cette méthode retourne un objet jQuery.

Cette fonction peut se révéler très utile dans certains scripts.

Exemple

Soit un bouton que l'on ne peut cliquer qu'une fois. Au premier clic, un message est affiché.






```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("button").one("click", function(){
$("div").html("<b>Inutile de cliquer encore !</b>");
});
});
//]]>
</script>
<style type="text/css">
div { width: 180px;
      height: 30px;
      margin-top:10px;
      padding-left: 5px;
      border: 1px solid black;}
</style>
</head>
<body>
1 seul clic est permis<br />
<button>Test</button>
<div></div>
</body>
</html>
```

```
$(document).ready(function(){
$("button").one("click", function(){
```

Au clic sur le bouton, la fonction suivante est exécutée.

```
$("div").html("<b>Inutile de cliquer encore !</b>");
});
```

La phrase "Inutile de cliquer encore" est affichée comme du Html dans la division. Comme la méthode `one()` a été utilisée, la phrase ne s'affiche qu'une seule fois et tout clic supplémentaire sur le bouton n'entraîne plus aucune action.

```
});
```

Fin du script.

3. Déclencher un événement particulier

trigger(événement)

Déclenche un événement particulier pour les éléments de la sélection. Cela va aussi déclencher l'action par défaut du navigateur pour ce type d'événement (si elle existe). Par exemple, utiliser le type d'événement `'submit'` dans la fonction va aussi déclencher l'envoi du formulaire par le navigateur. Cette action par défaut peut être empêchée en retournant "`false`" dans une des fonctions associées à l'événement pour un des éléments de la sélection.

Vous pouvez également utiliser la fonction `bind()`.

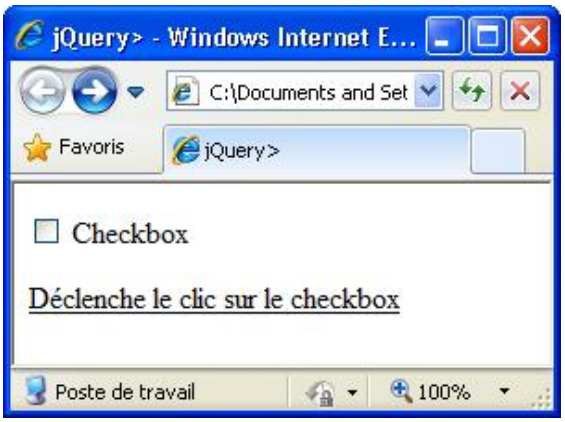
```
$("p").trigger("click");
```

Cette méthode retourne un objet jQuery.

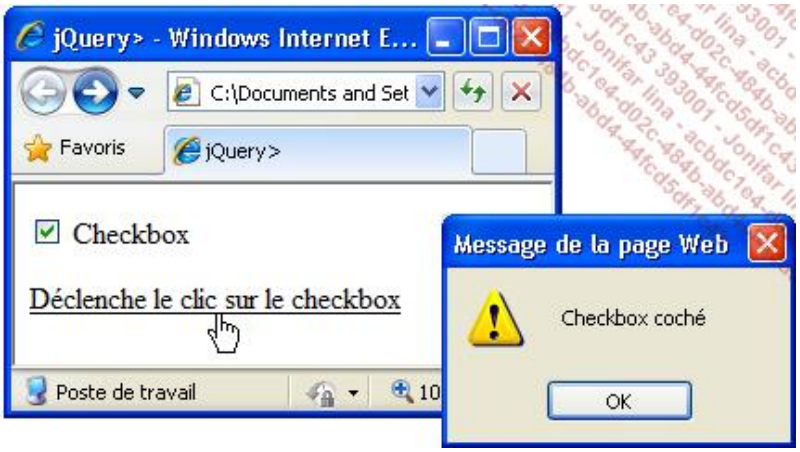
Exemple

Au clic du lien, la case à cocher `Checkbox` sera sélectionnée.

© ENI Editions - All rigths reserved - Jonifar lina



Au clic du lien, le script coche la case **Checkbox**, ce qui déclenche un message d'alerte.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function () {
$('input').bind('click', function() {
alert('Checkbox coché');
});
$('a').bind('click', function() {
$('input').trigger('click');
return false;
});
});
</script>
<style type="text/css">
a { color: black;}
</style>
</head>
<body>
<p><input type="checkbox" /> Checkbox</p>
<p><a href="#">Déclenche le clic sur le checkbox</a></p>
</body>
</html>
```

4. Déléguer un événement

© ENI Editions - All rigths reserved - Jonifar lina

Soit un événement `click` sur un élément de liste `<li>` qui génère un nouvel élément de liste. L'événement `click` doit continuer à fonctionner pour tous ces éléments de liste nouvellement créés. Ce qui n'est pas le cas de la méthode `bind()` qui doit être redéfinie sur ces nouveaux éléments. La méthode `live()` remédie à ce problème.

live(événement, fonction)

Attache une fonction à un événement donné pour tous les éléments existants et futurs trouvés.

```
$("p").live("click", function(){
alert( $(this).text() );
});
```

Affiche une boîte d'alerte au clic de chaque paragraphe de texte.

Cette méthode retourne un objet jQuery.

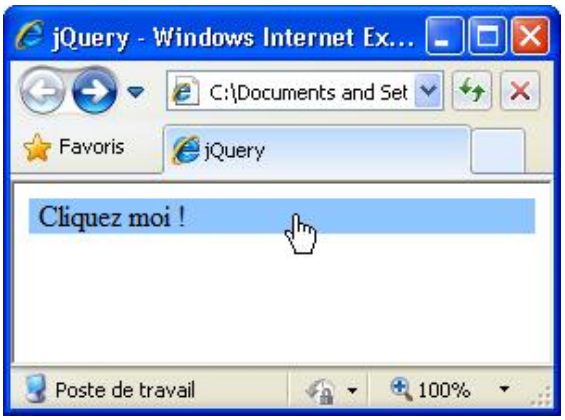
Il faut bien comprendre que jQuery fonctionne sur les éléments existant au chargement du DOM. Si un événement génère un nouvel élément, votre fonction n'agira pas sur celui-ci. La méthode `live()` permet de contourner le problème et rappelle la fonction définie après son exécution. Ce qui a pour conséquence de pouvoir agir sur les nouveaux éléments d'une page.

La méthode `die(événement, fonction)` annule `live()` en arrêtant la fonction définie en paramètre.

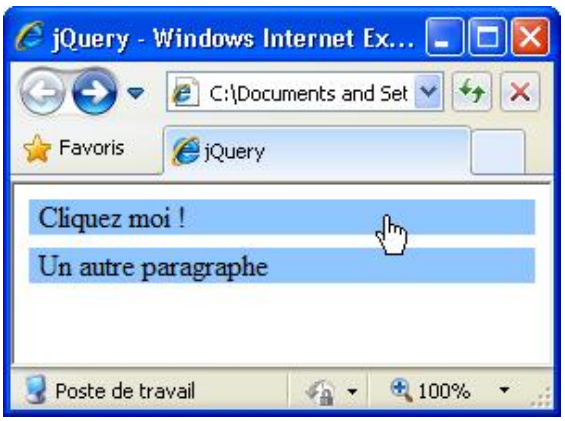
Exemple

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("p").live("click", function(){
$(this).after("<p>Un autre paragraphe</p>");
});
});
//]]>
</script>
<style type="text/css">
p { background: #9cf;
    cursor: pointer;
    padding-left: 5px;
    margin : 7px 0 7px 0;}
</style>
</head>
<body>
<p>Cliquez moi !</p>
</body>
</html>
```

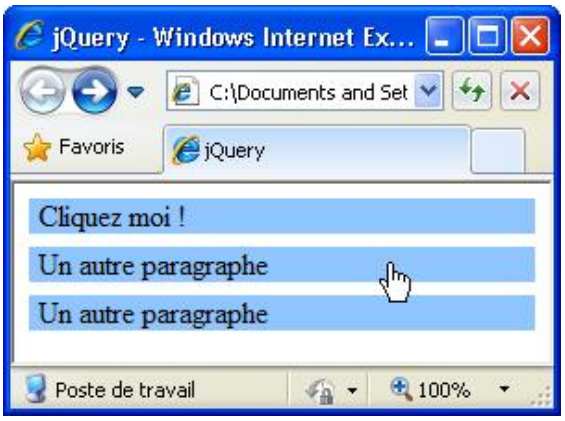

© ENI Editions - All rigths reserved - Jonifar lina



Le clic génère un nouveau paragraphe.



En cliquant sur le paragraphe initial ou sur le paragraphe nouvellement créé, la fonction ajoute à nouveau un paragraphe. Et ainsi de suite...



© ENI Editions - All rigths reserved - Jonifar lina

Méthodes propres à jQuery

1. Au passage de la souris

Cette fonction propre à jQuery reprend les événements `onmouseover` et `onmouseout` du JavaScript ou `mouseover` et `mouseout` de jQuery, soit lorsque le curseur survole un élément et quitte celui-ci.

hover(fonction 1, fonction 2)

La méthode `hover()` de jquery lie deux événements fréquemment utilisés à l'élément sélectionné ; l'un lorsque le pointeur survole et l'autre lorsqu'il le quitte.

Cette méthode s'effectue donc en deux temps. Quand le curseur se situe au-dessus d'un élément déterminé, la première fonction passée en paramètre est exécutée. Lorsque le curseur sort du cadre de l'élément, la seconde fonction est exécutée.

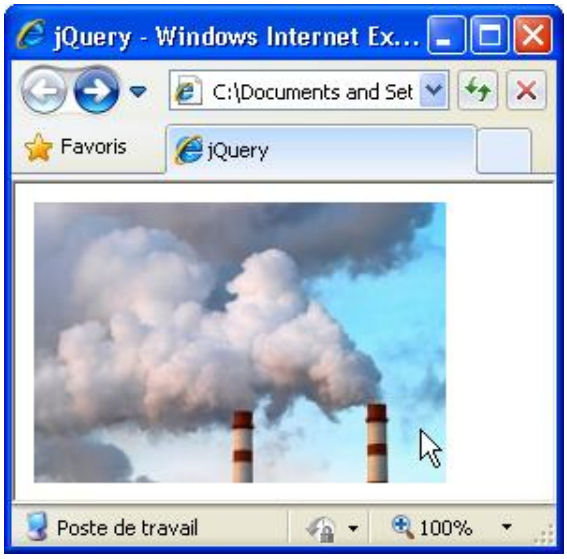
```
$("p").hover(function(){
$(this).addClass("hover");
},
function(){
$(this).removeClass("hover");
});
```

Au survol des paragraphes, la classe `hover` est ajoutée. Lorsque le curseur quitte la zone, la classe est retirée. Cette méthode retourne un objet jQuery.

Exemple

Le classique, une image qui change au survol de la souris.

© ENI Editions - All rigths reserved - Jonifar lina




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
.image {  position: absolute;
          top: 10px;
          left: 10px;}
</style>
</head>
<body>
<div class="image">
<img src="pollution1.jpg" />
</div>
</body>
</html>
```

Le script jQuery s'écrit :

```
<script type="text/javascript">
$(document).ready(function () {
$('img').hover(function () {
this.src = 'pollution2.jpg';
},
function () {
this.src = 'pollution1.jpg';
});
});
</script>
```

Détaillons celui-ci.

```
$(document).ready(function () {
$('img').hover(function () {
```

Une méthode `hover()` est appliquée aux images.

```
this.src = 'pollution2.jpg';
},
```

Dans un premier temps, au survol du curseur, l'image "pollution2.jpg" est chargée en modifiant l'attribut `src` de la balise `<img>`.

© ENI Editions - All rigths reserved - Jonifar lina

```
function () {
this.src = 'pollution1.jpg';
});
```

Dans un second temps, lorsque le curseur quitte l'image, le script revient à la situation de départ.

```
});
```

Fin de script.

Au final :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function () {
$('img').hover(function () {
this.src = 'pollution2.jpg';
},
function () {
this.src = 'pollution1.jpg';
});
});
</script>
<style type="text/css">
.image {  position: absolute;
          top: 10px;
          left: 10px;}
</style>
</head>
<body>
<div class="image">
<img src="pollution1.jpg" alt="" />
</div>
</body>
</html>
```

2. Basculer d'une situation à l'autre

Nous retrouvons ici la fonction `toggle()` déjà abordée par ailleurs.

toggle (fonction 1, fonction 2, etc.)

Permet de basculer d'une fonction à l'autre à chaque clic sur les éléments de la sélection.

Dès que l'événement `clic` est réalisé, la première fonction est exécutée. Au clic suivant, la seconde sera exécutée. Et ainsi de suite.

```
$("p").toggle(function(){
$(this).addClass("selected");
},
function(){
$(this).removeClass("selected");
});
```

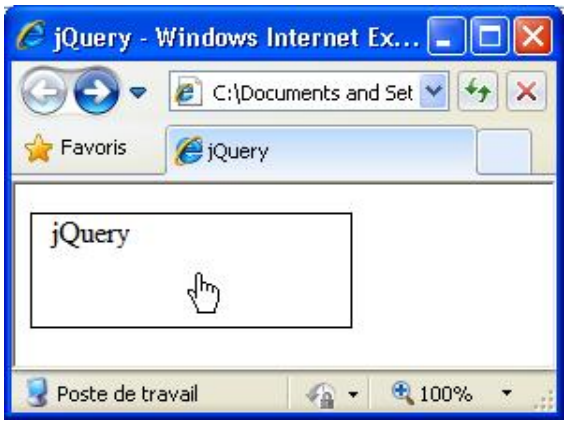
Au clic sur les paragraphes, la classe `selected` est ajoutée. Au clic suivant, la même classe est enlevée. Et ainsi de suite.

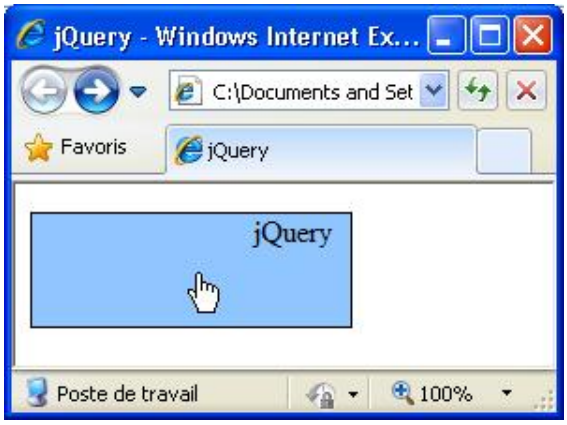
Cette méthode retourne un objet jQuery.

Exemple

© ENI Editions - All rigths reserved - Jonifar lina

Au clic sur une division, l'arrière-plan de celle-ci est colorié. Au clic suivant, celle-ci reprend sa couleur blanche.






```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
#div { width: 150px;
       height: 60px;
       cursor: pointer;
       border: 1px solid black;
       padding-left: 10px;
       padding-right: 10px;}
</style>
</head>
<body>
<div id="div">jQuery</div>
</body>
</html>
```

Le script jQuery.

```
<script type="text/javascript">
$(document).ready(function(){
$("#div").toggle( function () {
$(this).css({"background-color": "#9cf","text-align": "right"});
},
function () {
$(this).css({"background-color": "white", "text-align": "left"});
}
);
```


© ENI Editions - All rigths reserved - Jonifar lina

```
});
</script>
```

Explications du script.

```
$(document).ready(function(){
$("#div").toggle( function () {
$(this).css({"background-color": "#9cf","text-align": "right"});
},
```

Au premier clic sur la division `<div>`, celle-ci se voit dotée d'un arrière-plan de couleur (`background-color": "#9cf"`) et d'un alignement de texte à droite (`text-align": "right`).

```
function () {
$(this).css({"background-color": "white", "text-align": "left"});
}
);
```

Au second clic, l'arrière-plan et l'alignement sont rétablis.

```
});
```

Fin de script.

Au final :

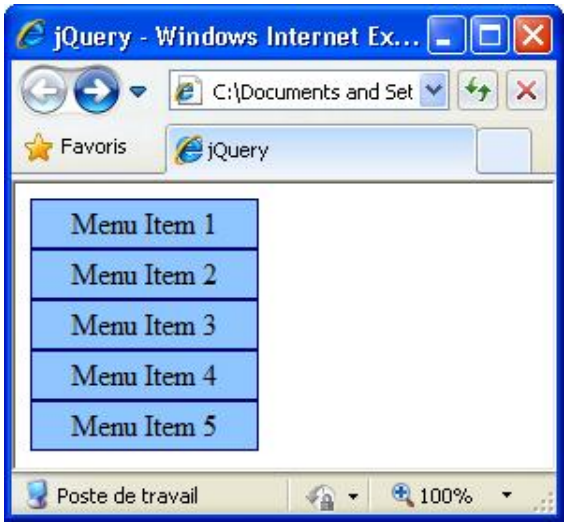
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#div").toggle( function () {
$(this).css({"background-color": "#9cf","text-align": "right"});
},
function () {
$(this).css({"background-color": "white", "text-align": "left"});
}
);
});
</script>
<style type="text/css">
#div { width: 150px;
       height: 60px;
       cursor: pointer;
       border: 1px solid black;
       padding-left: 10px;
       padding-right: 10px;}
</style>
</head>
<body>
<div id="div">jQuery</div>
</body>
</html>
```

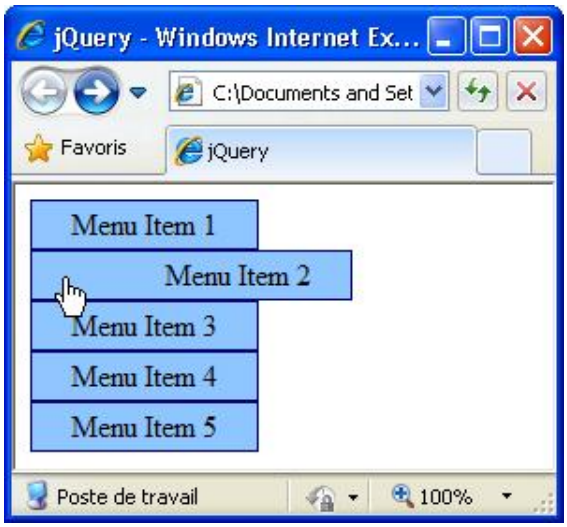
© ENI Editions - All rigths reserved - Jonifar lina

Applications

1. Un menu décalé

Au survol d'un item du menu de navigation, celui-ci se déplace vers la droite.






```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
#nav ul { list-style: none;
          padding: 0px;
          margin: 0px;}
#nav li a { display: block;
            height: 25px;
            line-height: 25px;
            width: 120px;
            background: #99ccff;
            border: 1px solid navy;
            color: black;
            text-decoration: none;
```

© ENI Editions - All rigths reserved - Jonifar lina

```
            text-align: center;}
</style>
</head>
<body>
<div id="nav">
<ul id="menu">
<li><a href="#">Menu Item 1</a></li>
<li><a href="#">Menu Item 2</a></li>
<li><a href="#">Menu Item 3</a></li>
<li><a href="#">Menu Item 4</a></li>
<li><a href="#">Menu Item 5</a></li>
</ul>
</div>
</body>
</html>
```

Le script jQuery se présente ainsi :

```
<script type="text/javascript">
$(document).ready(function() {
$('ul#menu li a').hover(function() {
$(this).stop().animate( { paddingLeft:"50px" }, 400 );
},
function() {
$(this).stop().animate( { paddingLeft:"0" }, 200 )
})
});
</script>
```

Développons celui-ci.

```
$(document).ready(function() {
$('ul#menu li a').hover(function() {
```

La méthode `hover()` est associée aux éléments de la liste constituant le menu de navigation.

```
$(this).stop().animate( { paddingLeft:"50px" }, 400 );
},
```

Au survol de chaque item, le script arrête d'abord toute animation en cours (`stop()`). Puis, met en œuvre une animation (`animate()`) qui consiste à augmenter le retrait par rapport au bord gauche (`paddingLeft:"50px"`).

```
$(this).stop().animate( { paddingLeft:"0" }, 200 )
})
```

Lorsque le curseur quitte l'item, le second volet de la méthode `hover()`, fait revenir l'item à sa position initiale.

```
});
```

Fin du ready et du script.

Le document final devient :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function() {
$('ul#menu li a').hover(function() {
$(this).stop().animate( { paddingLeft:"50px" }, 400 );
},
function() {
```


© ENI Editions - All rigths reserved - Jonifar lina

```
$(this).stop().animate( { paddingLeft:"0" }, 200 )
})
});
</script>
<style type="text/css">
#nav ul { list-style: none;
          padding: 0px;
          margin: 0px;}
#nav li a { display: block;
            height: 25px;
            line-height: 25px;
            width: 120px;
            background: #99ccff;
            border: 1px solid navy;
            color: black;
            text-decoration: none;
            text-align: center;}
</style>
</head>
<body>
<div id="nav">
<ul id="menu">
<li><a href="#">Menu Item 1</a></li>
<li><a href="#">Menu Item 2</a></li>
<li><a href="#">Menu Item 3</a></li>
<li><a href="#">Menu Item 4</a></li>
<li><a href="#">Menu Item 5</a></li>
</ul>
</div>
</body>
</html>
```

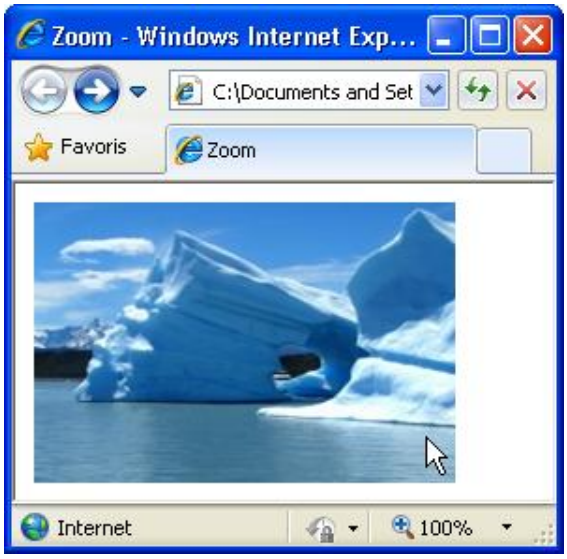
2. Zoom sur vignette

Concevons un script qui permet d'afficher une image en taille réelle au survol d'une vignette de celle-ci.

Cet exemple est une version simplifiée du superbe script de SohTanaka disponible à l'adresse : www.sohtanaka.com/web-design/examples/image-zoom/

La transcription papier ne rend pas le côté dynamique du script. Reportez-vous à l'espace de téléchargement pour en profiter pleinement.

© ENI Editions - All rigths reserved - Jonifar lina



Le fichier Xhtml initial :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>Zoom</title>
<style type="text/css">
body { padding: 3px;
       margin: 0px;}
.container { position: absolute;
             top: 50px;
             left : 10px;}
ul.thumb { padding: 0px;
           list-style-type: none;
           margin: 0px;
           width: 260px;
           float: left;}
ul.thumb li { position: relative;
              padding: 0px;
              margin: 0px;
              width: 100px;
              height: 100px;
              float: left;}
ul.thumb li img { position: absolute;
                  padding : 0px;
                  width: 90px;
                  height: 60px;
                  left: 0px;
                  border: none;}
</style>
</head>
<body>
<p> Survolez la vignette pour agrandir l'image.</p>
<div class="container">
<ul class="thumb">
<li><img alt="" src="iceberg1.jpg" /></li>
<li><img alt="" src="iceberg2.jpg" /></li>
</ul>
</div>
</body>
</html>
```

Le script :

© ENI Editions - All rigths reserved - Jonifar lina

```
<script type="text/javascript">
$(document).ready(function(){
$("ul.thumb li").hover(function() {
$(this).css({'z-index' : '10'});
$(this).find('img').addClass("hover").stop()
.animate({ marginTop: '-90px',
           marginLeft: '-50px',
           top: '50%',
           left: '50%',
           width: '225px',
           height: '150px',
           }, 200);
} ,
function() {
$(this).css({'z-index' : '0'});
$(this).find('img').removeClass("hover").stop()
.animate({ marginTop: '0',
           marginLeft: '0',
           top: '0',
           left: '0',
           width: '90px',
           height: '60px',
           }, 400);
});
});
</script>
```

Celui-ci nécessite quelques éclaircissements :

```
$(document).ready(function(){
$("ul.thumb li").hover(function() {
```

Application d'un `hover()` sur la vignette.

```
$(this).css({'z-index' : '10'});
.animate({ marginTop: '-90px',
           marginLeft: '-50px',
           top: '50%',
           left: '50%',
           width: '225px',
           height: '150px',
           }, 200);
} ,
```

Au survol de la vignette, la propriété de style `z-index` est modifiée (`css({'z-index' : '10'})`) pour faire passer l'image au premier plan. Ensuite une animation (`animate()`) modifie une série de paramètres.

```
function() {
$(this).css({'z-index' : '0'});
$(this).find('img').removeClass("hover").stop()
.animate({ marginTop: '0',
           marginLeft: '0',
           top: '0',
           left: '0',
           width: '90px',
           height: '60px',
           }, 400);
});
```

Lorsque le curseur quitte l'image, la propriété de style `z-index` est rétablie à son état initial. Ce qu'effectue également une animation (`animate()`) sur une série de paramètres.

```
});
```

Fin du script.

Le fichier final :

© ENI Editions - All rigths reserved - Jonifar lina

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>Zoom</title>
<style type="text/css">
body { padding: 3px;
       margin: 0px;}
.container { position: absolute;
             top: 50px;
             left : 10px;}
ul.thumb { padding: 0px;
           list-style-type: none;
           margin: 0px;
           width: 260px;
           float: left;}
ul.thumb li { position: relative;
              padding: 0px;
              margin: 0px;
              width: 100px;
              height: 100px;
              float: left;}
ul.thumb li img { position: absolute;
                  padding : 0px;
                  width: 90px;
                  height: 60px;
                  left: 0px;
                  border: none;}
</style>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("ul.thumb li").hover(function() {
$(this).css({'z-index' : '10'});
$(this).find('img').addClass("hover").stop()
.animate({ marginTop: '-90px',
           marginLeft: '-50px',
           top: '50%',
           left: '50%',
           width: '225px',
           height: '150px',
           }, 200);
} ,
function() {
$(this).css({'z-index' : '0'});
$(this).find('img').removeClass("hover").stop()
.animate({ marginTop: '0',
           marginLeft: '0',
           top: '0',
           left: '0',
           width: '90px',
           height: '60px',
           }, 400);
});
});
</script>
</head>
<body>
<p> Survolez la vignette pour agrandir l'image.</p>
<div class="container">
<ul class="thumb">
<li><img alt="" src="iceberg1.jpg" /></li>
<li><img alt="" src="iceberg2.jpg" /></li>
</ul>
</div>
</body>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Introduction

Les animations visuelles font partie prenante du JavaScript. Utilisées à bon escient et avec modération, elles peuvent donner plus de force à un élément du contenu. Le Web 2.0 les a d'ailleurs largement adoptées.

Il faut admettre que réaliser une animation graphiquement évoluée en pur JavaScript, tourne rapidement en un cauchemar de programmation pour tenir compte des spécifications propres à chaque type de navigateur, voire à chaque version de celui-ci. Le framework jQuery propose une série d'effets visuels faciles à encoder et parfaitement compatibles. En outre, jQuery offre aussi la possibilité de créer vos propres animations.

Ce chapitre est très visuel avec ses effets et autres animations. Il est vivement conseillé de consulter les exemples fournis dans l'espace de téléchargement pour en comprendre le fonctionnement réel.

© ENI Editions - All rigths reserved - Jonifar lina

Afficher et cacher

Les méthodes `show()` et `hide()` de jQuery permettent de faire apparaître et disparaître des éléments.

show(vitesse, fonction de rappel)

Affiche un élément sélectionné (pour autant que celui-ci soit caché).

L'animation modifie dynamiquement la hauteur, la largeur et l'opacité de l'élément. Depuis la spécification jQuery 1.3, les marges externes et internes sont également modifiées pour obtenir un effet plus fluide.

- vitesse (optionnel) : chaîne de caractères représentant une des trois vitesses prédéfinies (`'slow'`, `'normal'` ou `'fast'`) ou le nombre en millisecondes correspondant à la durée de l'effet.

  ```
  $('p').show('slow');
  ```

- fonction de rappel (callback) (optionnel) : fonction à exécuter à la fin de l'effet.

  ```
  $('p').show('slow', function(){alert("Fin");});
  ```

Cette méthode renvoie un objet jQuery.

hide(vitesse, fonction de rappel)

Cache un élément sélectionné (pour autant que celui-ci soit visible).

L'animation modifie dynamiquement la hauteur, la largeur et l'opacité de l'élément. Depuis la spécification jQuery 1.3, les marges externes et internes sont également modifiées pour obtenir un effet plus fluide.

- vitesse (optionnel) : chaîne de caractères représentant une des trois vitesses prédéfinies (`'slow'`, `'normal'` ou `'fast'`) ou le nombre en millisecondes correspondant à la durée de l'effet.

  ```
  $('p').hide('fast');
  ```

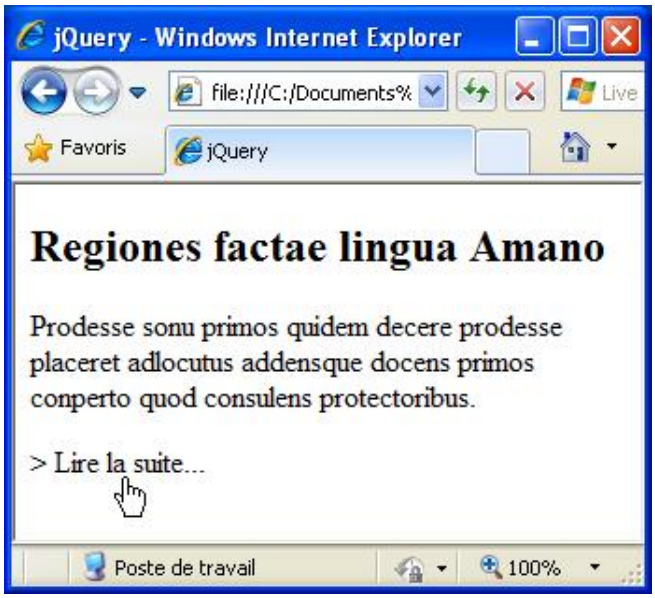
- fonction de rappel (callback) (optionnel) : fonction à exécuter à la fin de l'effet.

  ```
  $('p').hide('normal', function(){alert("Fin");});
  ```

Cette méthode renvoie un objet jQuery.

1. Afficher et cacher du texte

L'exemple suivant propose au lecteur d'afficher la suite d'un article, après avoir cliqué sur le lien.

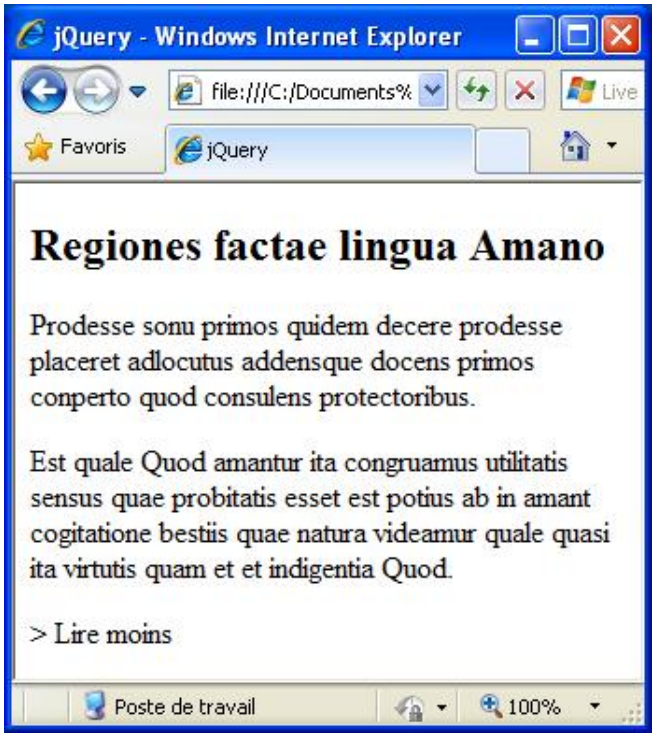


© ENI Editions - All rigths reserved - Jonifar lina

Le fichier Html de départ se présente comme suit :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
a { color: black;
    text-decoration: none;}
</style>
</head>
<body>
<h2>Regiones factae lingua Amano</h2>
<p>
Prodesse sonu primos quidem decere prodesse placeret adlocutus
addensque docens primos conperto quod consulens protectoribus.
</p>
<p>
Est quale Quod amantur ita congruamus utilitatis sensus quae
probitatis esset est potius ab in amant cogitatione bestiis quae
natura videamur quale quasi ita virtutis quam et et indigentia
Quod.
</p>
<div>
<a href="#" class="suite">&gt; Lire la suite...</a>
</div>
</body>
</html>
```

Le script jQuery :




```
<script type="text/javascript">
$(document).ready(function() {
var $paragraphe2 = $('p:eq(1)');
$paragraphe2.hide();
$('a.suite').click(function() {
if ($paragraphe2.is(':hidden')) {
```


© ENI Editions - All rigths reserved - Jonifar lina

```
$paragraphe2.show('slow');
$(this).text('> Lire moins');
} else {
$paragraphe2.hide('slow');
$(this).text('> Lire la suite...');
return false;
}
});
});
</script>
```

Détaillons celui-ci :

```
$(document).ready(function() {
var $paragraphe2 = $('p:eq(1)');
```

Au chargement du DOM, le second paragraphe est chargé dans la variable `$paragraphe2`. Rappelons que la méthode `eq()` débute à 0, comme souvent en JavaScript.

Les programmeurs utilisent souvent la convention de faire débuter les variables dans un script jQuery par le signe `$`.

```
$paragraphe2.hide();
```

Le second paragraphe est caché au chargement de la page.

```
$('a.suite').click(function() {
if ($paragraphe2.is(':hidden')) {
$paragraphe2.show('slow');
$(this).text('> Lire moins');
```

Au clic sur le lien **> Lire la suite...**, si le second paragraphe est bien caché (`hidden`), il est alors affiché à une vitesse lente prédéfinie et le texte du lien est modifié en **> Lire moins**.

```
} else {
$paragraphe2.hide('slow');
$(this).text('> Lire la suite...');
return false;
}
```

Sinon, le second paragraphe est caché et le texte du lien modifié en "> Lire la suite...".

```
});
});
```

Fin de script.

Le fichier complet devient alors :

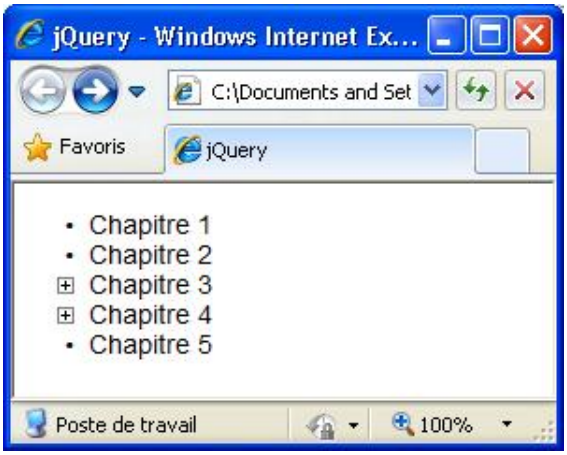
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script src="jquery.js" type="text/javascript"></script>
<script type="text/javascript">
$(document).ready(function() {
var $paragraphe2 = $('p:eq(1)');
$paragraphe2.hide();
$('a.suite').click(function() {
if ($paragraphe2.is(':hidden')) {
$paragraphe2.show('slow');
$(this).text('> Lire moins');
} else {
$paragraphe2.hide('slow');
$(this).text('> Lire la suite...');
```

© ENI Editions - All rigths reserved - Jonifar lina

```
return false;
}
});
});
</script>
<style type="text/css">
a { color: black;
    text-decoration: none;}
</style>
</head>
<body>
<h2>Regiones factae lingua Amano</h2>
<p>
Prodesse sonu primos quidem decere prodesse placeret adlocutus
addensque docens primos conperto quod consulens protectoribus.
</p>
<p>
Est quale Quod amantur ita congruamus utilitatis sensus quae
probitatis esset est potius ab in amant cogitatione bestiis quae
natura videamur quale quasi ita virtutis quam et et indigentia
Quod.
</p>
<div>
<a href="#" class="suite">&gt; Lire la suite...</a>
</div>
</body>
</html>
```

2. Dérouler des listes imbriquées

Nous allons permettre de dérouler des listes imbriquées par un clic sur une image.



Les images plus.gif ⊞ et moins.gif ⊟ sont disponibles dans l'espace de téléchargement réservé à cet ouvrage.

Le fichier Html de départ se présente comme suit :

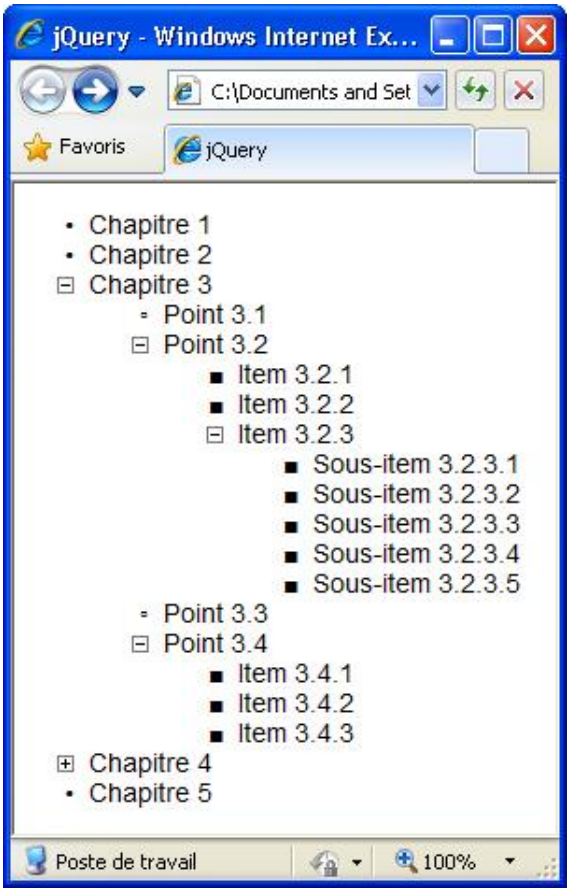
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script src="jquery.js" type="text/javascript"></script>
<style type="text/css">
body{ font-family: Arial,sans-serif;
      font-size: 0.9em;
      margin-left: 0px;}
</style>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
</head>
<body>
<ul>
<li>Chapitre 1</li>
<li>Chapitre 2</li>
<li>Chapitre 3
<ul>
<li>Point 3.1</li>
<li>Point 3.2
<ul>
<li>Item 3.2.1</li>
<li>Item 3.2.2</li>
<li>Item 3.2.3
<ul>
<li>Sous-item 3.2.3.1</li>
<li>Sous-item 3.2.3.2</li>
<li>Sous-item 3.2.3.3</li>
<li>Sous-item 3.2.3.4</li>
<li>Sous-item 3.2.3.5</li>
</ul>
</li>
</ul>
</li>
<li>Point 3.3</li>
<li>Point 3.4
<ul>
<li>Item 3.4.1</li>
<li>Item 3.4.2</li>
<li>Item 3.4.3</li>
</ul>
</li>
</ul>
</li>
<li>Chapitre 4
<ul>
<li>Point 4.1</li>
<li>Point 4.2
<ul>
<li>Item 4.2.1</li>
<li>Item 4.2.2</li>
</ul>
</li>
</ul>
</li>
<li>Chapitre 5</li>
</ul>
</body>
</html>
```

Le script jQuery :

© ENI Editions - All rigths reserved - Jonifar lina




```
<script type="text/javascript">
$(document).ready(function() {
$('li:has(ul)')
.click(function(){
if ($(this).children().is(':hidden')) {
$(this).css('list-style-image','url(minus.gif)')
       .children().show();
}
else {
$(this).css('list-style-image','url(plus.gif)')
       .children().hide();
}
return false;
})
.css('cursor','pointer')
.click();
$('li:not(:has(ul))').css({cursor: 'default',
                          'list-style-image':'none'
});
});
</script>
```

Détaillons celui-ci.

```
$(document).ready(function() {
$('li:has(ul)')
```

Au chargement du DOM, le script sélectionne tous les éléments de liste (`<li>`) qui possèdent une liste imbriquée (`<ul>`).

```
.click(function(){
```

Au clic sur ces éléments.

```
if ($(this).children().is(':hidden')) {
```


© ENI Editions - All rigths reserved - Jonifar lina

```
$(this).css('list-style-image','url(minus.gif)')
       .children().show();
}
```

Si les listes imbriquées de cet élément (soit les descendants) sont cachées, l'image est changée en minus.gif et celles-ci sont déployées.

```
else {
$(this).css('list-style-image','url(plus.gif)')
       .children().hide();
}
return false;
})
```

Sinon, l'image plus.gif est affichée et les listes imbriquées sont cachées.

```
.css('cursor','pointer')
.click();
```

Il nous reste à nous occuper de la forme du curseur de la souris. Pour les éléments sélectionnés par les lignes de code précédentes (`li:has(ul)`), le curseur prend la forme d'une main (`pointer`).

```
$('li:not(:has(ul))').css({cursor: 'default',
                           'list-style-image':'none'
});
```

Pour les éléments qui ne possèdent pas de liste imbriquée (`'li:not(:has(ul))'`), la forme du curseur par défaut est retenue et il n'y a pas de petite icône affichée.

```
});
```

Fin du script.

Le fichier complet devient :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
body{ font-family: Arial,sans-serif;
      font-size: 0.9em;
      margin-left: 0px;}
</style>
<script src="jquery.js" type="text/javascript"></script>
<script type="text/javascript">
$(document).ready(function() {
$('li:has(ul)')
.click(function(){
if ($(this).children().is(':hidden')) {
$(this).css('list-style-image','url(minus.gif)')
       .children().show();
}
else {
$(this).css('list-style-image','url(plus.gif)')
       .children().hide();
}
return false;
})
.css('cursor','pointer')
.click();
$('li:not(:has(ul))').css({cursor: 'default',
                        'list-style-image':'none'
});
```

© ENI Editions - All rigths reserved - Jonifar lina

```
});
</script>
</head>
<body>
<ul>
<li>Chapitre 1</li>
<li>Chapitre 2</li>
<li>Chapitre 3
<ul>
<li>Point 3.1</li>
<li>Point 3.2
<ul>
<li>Item 3.2.1</li>
<li>Item 3.2.2</li>
<li>Item 3.2.3
<ul>
<li>Sous-item 3.2.3.1</li>
<li>Sous-item 3.2.3.2</li>
<li>Sous-item 3.2.3.3</li>
<li>Sous-item 3.2.3.4</li>
<li>Sous-item 3.2.3.5</li>
</ul>
</li>
</ul>
</li>
<li>Point 3.3</li>
<li>Point 3.4
<ul>
<li>Item 3.4.1</li>
<li>Item 3.4.2</li>
<li>Item 3.4.3</li>
</ul>
</li>
</ul>
</li>
<li>Chapitre 4
<ul>
<li>Point 4.1</li>
<li>Point 4.2
<ul>
<li>Item 4.2.1</li>
<li>Item 4.2.2</li>
</ul>
</li>
</ul>
</li><li>Chapitre 5</li>
</ul>
</body>
</html>
```


© ENI Editions - All rigths reserved - Jonifar lina

Glisser verticalement

Les fonctions `slideDown()` et `slideUp()` permettent de jouer dynamiquement sur la hauteur d'un élément, généralement une division `<div> ... </div>`.

slideDown(vitesse, fonction de rappel)

Fait glisser vers le bas (down) un élément sélectionné.

L'animation modifie seulement la hauteur. Depuis la spécification jQuery 1.3, les marges verticales, externes et internes sont également modifiées pour obtenir un effet plus fluide.

- vitesse (optionnel) : chaîne de caractères représentant une des trois vitesses prédéfinies (`'slow'`, `'normal'` ou `'fast'`) ou le nombre en millisecondes correspondant à la durée de l'effet.

  ```
  $('div').slideDown('fast');
  ```

- fonction de rappel (callback) (optionnel) : fonction à exécuter à la fin de l'effet.

  ```
  $('div').slideDown('fast', function(){alert("Fin");});
  ```

Cette méthode renvoie un objet jQuery.

slideUp(vitesse, fonction de rappel)

Fait glisser vers le haut (up) un élément sélectionné.

L'animation modifie seulement la hauteur. Depuis la spécification jQuery 1.3, les marges verticales, externes et internes sont également modifiées pour obtenir un effet plus fluide.

- vitesse (optionnel) : chaîne de caractères représentant une des trois vitesses prédéfinies (`'slow'`, `'normal'` ou `'fast'`) ou le nombre en millisecondes correspondant à la durée de l'effet.

  ```
  $('div').slideUp('fast');
  ```

- fonction de rappel (callback) (optionnel) : fonction à exécuter à la fin de l'effet.

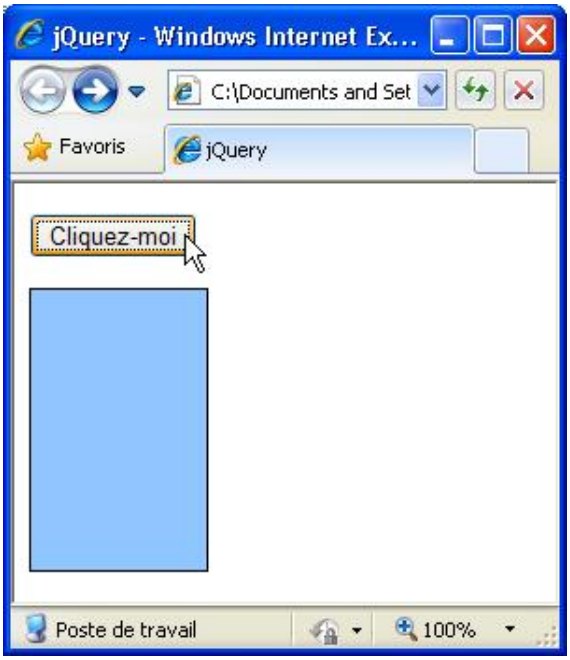
  ```
  $('div').slideUp('fast', function(){alert("Fin");});
  ```

Cette méthode renvoie un objet jQuery.

1. Faire glisser une division

Au clic sur le bouton, nous allons dérouler une division.

© ENI Editions - All rigths reserved - Jonifar lina




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-
8"/>
<title>jQuery</title>
<style type="text/css">
div { width:94px;
      height:150px;
      background:#9cf;
      border: 1px solid black;
      display:none; }
</style>
</head>
<body>
<p>
<button>Cliquez-moi</button>
</p>
<div></div>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
</body>
</html>
```

Le script jQuery :

```
<script type="text/javascript">
$(document).ready(function(){
$("button").click(function () {
if ($("div").is(":hidden")) {
$("div").slideDown("slow");
}
else
{$("div").slideUp("slow");
}
});
});
</script>
```

Explications :

```
$(document).ready(function(){
$("button").click(function () {
```

Après le chargement du DOM, au clic de la souris sur la balise `<button>`.

```
if ($("div").is(":hidden")) {
$("div").slideDown("slow");
}
```

Si la division (`<div>`) est cachée, celle-ci glisse vers le bas.

```
else
{$("div").slideUp("slow");
}
```

Sinon, la division glisse vers le haut.

```
});
});
```

Fin du script.

La page complète :

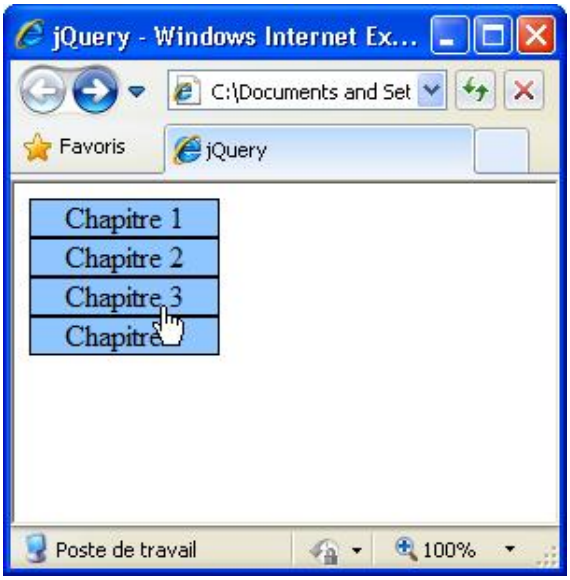
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-
8"/>
<title>jQuery</title>
<script src="jquery.js" type="text/javascript"></script>
<script type="text/javascript">
$(document).ready(function(){
$("button").click(function () {
if ($("div").is(":hidden")) {
$("div").slideDown("slow");
}
else
{$("div").slideUp("slow");
}
});
});
</script>
<style type="text/css">
div { width:94px;
      height:150px;
```

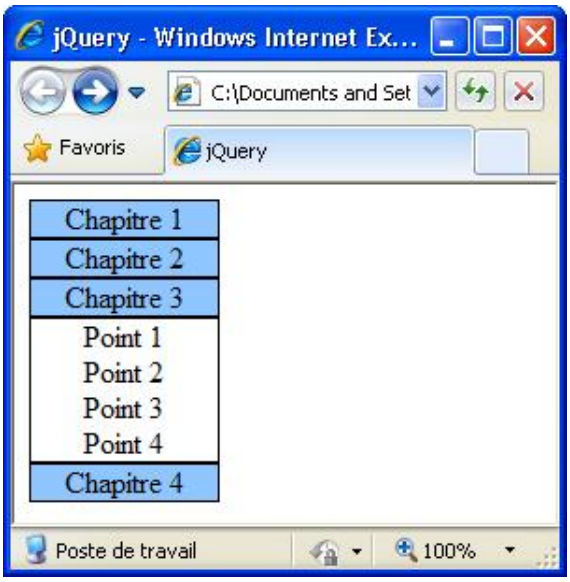
© ENI Editions - All rigths reserved - Jonifar lina

```
      background:#9cf;
      border: 1px solid black;
      display:none; }
</style>
</head>
<body>
<p>
<button>Cliquez-moi</button>
</p>
<div></div>
</body>
</html>
```

2. Un menu déroulant vertical

Élaborons un menu déroulant vertical assez simple sinon simpliste.





Le fichier Html initial :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
```


© ENI Editions - All rigths reserved - Jonifar lina

```
8859-1" />
<title>jQuery</title>
<style type="text/css">
a { text-decoration: none;
    color: black;}
.chapitre {   width: 100px;
              background: #9cf;
              border-bottom: 1px solid black;
              cursor: pointer;
              text-align: center;
              border: 1px solid black;}
.items { width: 100px;
         text-align: center;
         border: 1px solid black;
         display: none;}
</style>
</head>
<body>
<div class="chapitre">Chapitre 1</div>
<div class="items">
<a href="">Point 1</a><br />
<a href="">Point 2</a><br />
<a href="">Point 3</a>
</div>
<div class="chapitre">Chapitre 2</div>
<div class="items">
<a href="">Point 1</a><br />
<a href="">Point 2</a>
</div>
<div class="chapitre">Chapitre 3</div>
<div class="items">
<a href="">Point 1</a><br />
<a href="">Point 2</a><br />
<a href="">Point 3</a><br />
<a href="">Point 4</a>
</div>
<div class="chapitre">Chapitre 4</div>
<div class="items">
<a href="">Point 1</a><br />
<a href="">Point 2</a><br />
<a href="">Point 3</a>
</div>
</body>
</html>
```

Le script JQuery :

```
<script type="text/javascript">
$(document).ready(function() {
$('div.chapitre').click(function() {
$('div.items').slideUp('normal');
$(this).next().slideDown('normal');
});
$("div.items").hide();
});
</script>
```

```
$(document).ready(function() {
$('div.chapitre').click(function() {
```

Au chargement du DOM, au clic sur la division dont la classe est `chapitre`.

```
$('div.items').slideUp('normal');
```

Les divisions `items` se referment (glissent vers le haut).

```
$(this).next().slideDown('normal');
});
```

© ENI Editions - All rigths reserved - Jonifar lina

Le script demande d'atteindre les éléments suivants (`next`) de l'élément sélectionné (`this`) par le clic sur la division `chapitre`.

```
$("div.items").hide();
```

Les divisions `items` sont cachées afin que le menu se présente de façon refermée à l'ouverture de la page.

```
});
```

Fin de script.

Le fichier complet :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script src="jquery.js" type="text/javascript"></script>
<script type="text/javascript">
$(document).ready(function() {
$('div.chapitre').click(function() {
$('div.items').slideUp('normal');
$(this).next().slideDown('normal');
});
$("div.items").hide();
});
</script>
<style type="text/css">
a { text-decoration: none;
    color: black;}
.chapitre { width: 100px;
            background: #9cf;
            border-bottom: 1px solid black;
            cursor: pointer;
            text-align: center;
            border: 1px solid black;}
.items { width: 100px;
         text-align: center;
         border: 1px solid black;
         display: none;}
</style>
</head>
<body>
<div class="chapitre">Chapitre 1</div>
<div class="items">
<a href="">Point 1</a><br />
<a href="">Point 2</a><br />
<a href="">Point 3</a>
</div>
<div class="chapitre">Chapitre 2</div>
<div class="items">
<a href="">Point 1</a><br />
<a href="">Point 2</a>
</div>
<div class="chapitre">Chapitre 3</div>
<div class="items">
<a href="">Point 1</a><br />
<a href="">Point 2</a><br />
<a href="">Point 3</a><br />
<a href="">Point 4</a>
</div>
<div class="chapitre">Chapitre 4</div>
<div class="items">
<a href="">Point 1</a><br />
<a href="">Point 2</a><br />
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<a href="">Point 3</a>
</div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Réaliser un effet de fondu

Cet effet d'apparition ou de disparition d'un élément en modifiant progressivement son opacité est peut-être la plus belle réalisation de jQuery dans ce chapitre consacré aux effets.

fadeIn(vitesse, fonction de rappel)

Fait apparaître l'élément sélectionné selon un effet de fondu.

Cette animation est obtenue en ajustant uniquement l'opacité. L'élément sélectionné doit ainsi déjà avoir une largeur et une hauteur spécifiée.

- vitesse (optionnel) : chaîne de caractères représentant une des trois vitesses prédéfinies (`'slow'`, `'normal'` ou `'fast'`) ou le nombre en millisecondes correspondant à la durée de l'effet.

  ```
  $('p:first').fadeIn(4000);
  ```

- fonction de rappel (callback) (optionnel) : fonction à exécuter à la fin de l'effet.

  ```
  $('p:first').fadeIn(4000, function(){alert("Fin");});
  ```

Cette méthode renvoie un objet jQuery.

fadeOut(vitesse, fonction de rappel)

Fait disparaître l'élément sélectionné selon un effet de fondu.

Cette animation est obtenue en ajustant uniquement l'opacité. L'élément sélectionné doit ainsi déjà avoir une largeur et une hauteur spécifiée.

- vitesse (optionnel) : chaîne de caractères représentant une des trois vitesses prédéfinies (`'slow'`, `'normal'` ou `'fast'`) ou le nombre en millisecondes correspondant à la durée de l'effet.

  ```
  $('p:first').fadeIn(4000);
  ```

- fonction de rappel (callback) (optionnel) : fonction à exécuter à la fin de l'effet.

  ```
  $('p:first').fadeIn(4000, function(){alert("Fin");});
  ```

Cette méthode renvoie un objet jQuery.

fadeTo(vitesse, opacité, fonction de rappel)

Modifie l'opacité de l'élément sélectionné jusqu'à une valeur fournie dans le script.

- vitesse (optionnel) : chaîne de caractères représentant une des trois vitesses prédéfinies (`'slow'`, `'normal'` ou `'fast'`) ou le nombre en millisecondes correspondant à la durée de l'effet.

  ```
  $( 'p:first').fadeTo('slow', 0.33);
  ```

- opacity(nombre) : détermine la valeur de l'opacité à atteindre (nombre compris entre 0 et 1).

- fonction de rappel (callback) (optionnel) : fonction à exécuter à la fin de l'effet.

  ```
  $('p:first').fadeTo('slow', 0.33, function(){alert("Fin");});
  ```

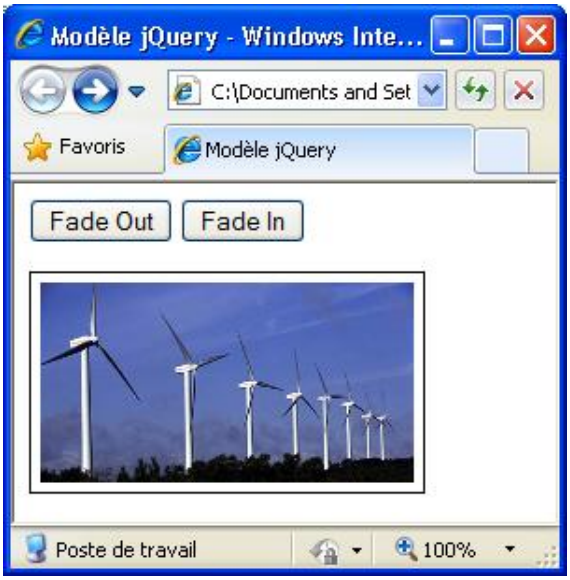
Cette méthode renvoie un objet jQuery.

1. Une apparition et disparition progressive

Mettons en œuvre ces méthodes jQuery sur une image.

Les images de cet exemple sont présentes dans l'espace de téléchargement.

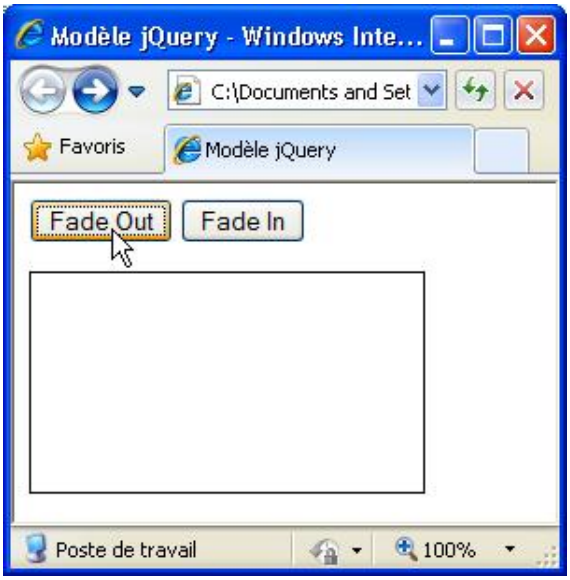
© ENI Editions - All rigths reserved - Jonifar lina

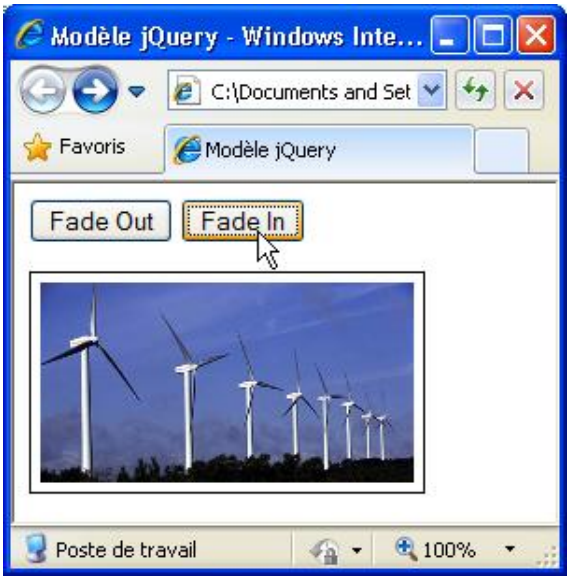



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>Modèle jQuery</title>
<style type="text/css">
#bloc { width: 210px;
        height: 117px;
        border: 1px solid black;
        margin-top: 15px;}
#image { margin: 5px;}
</style>
</head>
<body>
<div>
<input type="button" id="boutonFadeOut" value="Fade Out" />
<input type="button" id="boutonFadeIn" value="Fade In" />
</div>
<div id="bloc">
<img src="eole3.jpg" alt="" width="200" height="107" id="image" />
</div>
</body>
</html>
```

La partie script :

© ENI Editions - All rigths reserved - Jonifar lina






```
<script type="text/javascript">
$(document).ready(function(){
$("#boutonFadeOut").click(function () {
$("#image").fadeOut();
});
$("#boutonFadeIn").click(function () {
$("#image").fadeIn();
});
});</script>
```

Détaillons celui-ci.

```
$(document).ready(function(){
$("#boutonFadeOut").click(function () {
$("#image").fadeOut();
});
```

Après chargement du DOM, le clic sur le bouton **Fade Out**, fait disparaître l'image avec un effet de fondu.

```
$("#boutonFadeIn").click(function () {
$("#image").fadeIn();
});
```

Le clic sur le bouton **Fade In** fait apparaître l'image avec un effet de fondu.

© ENI Editions - All rigths reserved - Jonifar lina

```
});
```

Fin de script.

Le fichier final :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>Modèle jQuery</title>
<script src="jquery.js" type="text/javascript"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#boutonFadeOut").click(function () {
$("#image").fadeOut();
});
$("#boutonFadeIn").click(function () {
$("#image").fadeIn();
});
});
</script>
<style type="text/css">
#bloc { width: 210px;
        height: 117px;
        border: 1px solid black;
        margin-top: 15px;}
#image { margin: 5px;}
</style>
</head>
<body>
<div>
<input type="button" id="boutonFadeOut" value="Fade Out" />
<input type="button" id="boutonFadeIn" value="Fade In" />
</div>
<div id="bloc">
<img src="eole3.jpg" alt="" width="200" height="107" id="image" />
</div>
</body>
</html>
```

2. Jouer sur l’opacité

Faisons apparaître en clair, une image présentée en grisé au chargement de la page.

Les images de cet exemple sont présentes dans l’espace de téléchargement.

© ENI Editions - All rigths reserved - Jonifar lina

Le fichier Html initial :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
p { font-family:Arial, Helvetica, sans-serif;
    margin-left: 12px;
    font-size: 12px;}
.image { border: 1px solid black;
         margin-left: 8px;}
</style>
</head>
<body>
<p>Survolez l'image avec le curseur de la souris</p>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<div>
<img src="eole1.jpg" alt="" width="120" height="180" class="image" />
<img src="eole2.jpg" alt="" width="120" height="180" class="image" />
</div>
</body>
</html>
```

Le script jQuery ;

```
<script type="text/javascript">
$(document).ready(function(){
$(".image").fadeTo("slow", 0.5);
$(".image").hover(function(){
$(this).fadeTo("slow", 1.0);
},function(){
$(this).fadeTo("fast", 0.5);
});
});
</script>
```

Explicitons le script.

```
$(document).ready(function(){
$(".image").fadeTo("slow", 0.5);
```

Au chargement du DOM, les images sont affichées en grisé avec une opacité de 0,5.

```
$(".image").hover(function(){
$(this).fadeTo("slow", 1.0);
```

Au survol de l'image par le curseur de la souris, l'image en question est affichée normalement (opacité 1). L'effet de fondu est lent.

```
},function(){
$(this).fadeTo("fast", 0.5);
```

Lorsque le curseur quitte l'image, celle-ci revient à son état initial (opacité 0,5).

```
});
});
```

Fin de script.

Le fichier complet est alors :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$(".image").fadeTo("slow", 0.5);
$(".image").hover(function(){
$(this).fadeTo("slow", 1.0);
},function(){
$(this).fadeTo("fast", 0.5);
});
});
</script>
<style type="text/css">
p { font-family:Arial, Helvetica, sans-serif;
    margin-left: 12px;
    font-size: 12px;}
.image { border: 1px solid black;
```


© ENI Editions - All rigths reserved - Jonifar lina

```
        margin-left: 8px;}
</style>
</head>
<body>
<p>Survolez l’image avec le curseur de la souris</p>
<div>
<img src="eole1.jpg" alt="" width="120" height="180" class="image" />
<img src="eole2.jpg" alt="" width="120" height="180" class="image" />
</div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Basculer d'un effet à l'autre

Ce basculement d'un état à l'autre ou d'une fonction à l'autre est un classique de jQuery et a déjà été abordé dans le chapitre consacré aux événements.

toggle()

Permet de basculer l'état d'affichage de l'élément sélectionné. Si l'élément est affiché, la fonction le fait disparaître (avec la fonction `hidden()`) et inversement (avec la fonction `show()`).

```
$("p").toggle();
```

Cette méthode renvoie un objet jQuery.

toggle(fonction 1,fonction2)

Permet de basculer (switch) entre deux fonctions à chaque clic sur l'élément sélectionné. Lors du clic initial, la première fonction est exécutée. Lors du clic suivant, la seconde est alors exécutée. Lors d'un autre clic, à nouveau la première fonction et ainsi de suite.

```
$("p").toggle(function(){
$(this).addClass("selected");
},function(){
$(this).removeClass("selected");
});
```

Cette méthode renvoie un objet jQuery.

Cet effet de basculement est aussi appliqué à la fonction de glissement (voir la section Glisser verticalement du présent chapitre).

slideToggle(vitesse, fonction de rappel)

Cette fonction fait glisser vers le bas un élément qui est en état "Up" et fait glisser vers le haut un élément qui est en état "Down".

L'animation modifie seulement la hauteur. Depuis la spécification jQuery 1.3, les marges verticales, externes et internes sont également modifiées pour obtenir un effet plus fluide.

- vitesse (optionnel) : chaîne de caractères représentant une des trois vitesses prédéfinies (`'slow'`, `'normal'` ou `'fast'`) ou le nombre en millisecondes correspondant à la durée de l'effet.

  ```
  $('div').slideToggle('fast');
  ```

- fonction de rappel (callback) (optionnel) : fonction à exécuter à la fin de l'effet.

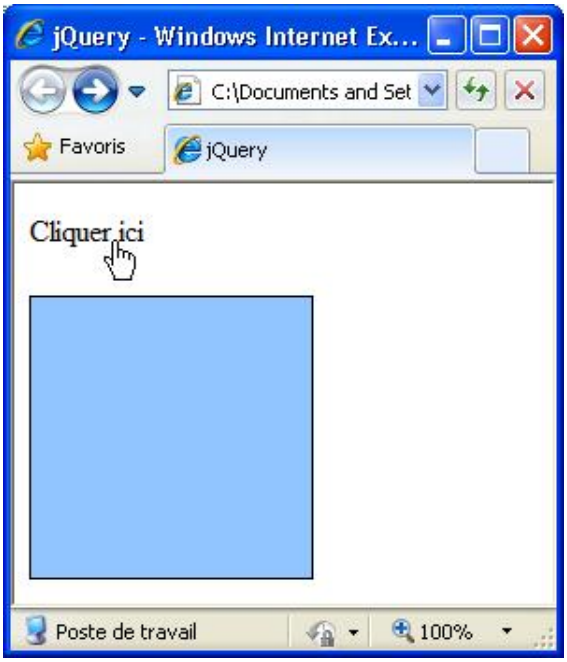
  ```
  $('div').slideToggle('fast', function(){alert("Fin");});
  ```

Cette méthode renvoie un objet jQuery.

1. Illustration du basculement

Un script simple pour décrire cette fonction `toggle()`.

© ENI Editions - All rigths reserved - Jonifar lina



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css" media="screen">
#div_toggle { background-color: #9cf;
              width: 150px;
              height: 150px;
              border: 1px solid black;}
p { cursor: pointer;
    margin-bottom: 25px;}
</style>
</head>
<body>
<p id="start">
Cliquer ici
```


© ENI Editions - All rigths reserved - Jonifar lina

```
</p>
<div id="div_toggle"></div>
</body>
</html>
```

Le script jQuery :

```
<script type="text/javascript">
$(document).ready(function() {
$("#start").click(function () {
$("#div_toggle").toggle("slow");
});
});
</script>
```

Détails du script :

```
$(document).ready(function() {
$("#start").click(function () {
```

Au clic sur le paragraphe identifié par `id="start"`.

```
$("#div_toggle").toggle("slow");
```

La fonction de basculement (`toggle`) est appliquée à la division.

```
});
});
```

Fin de script.

Le document final :

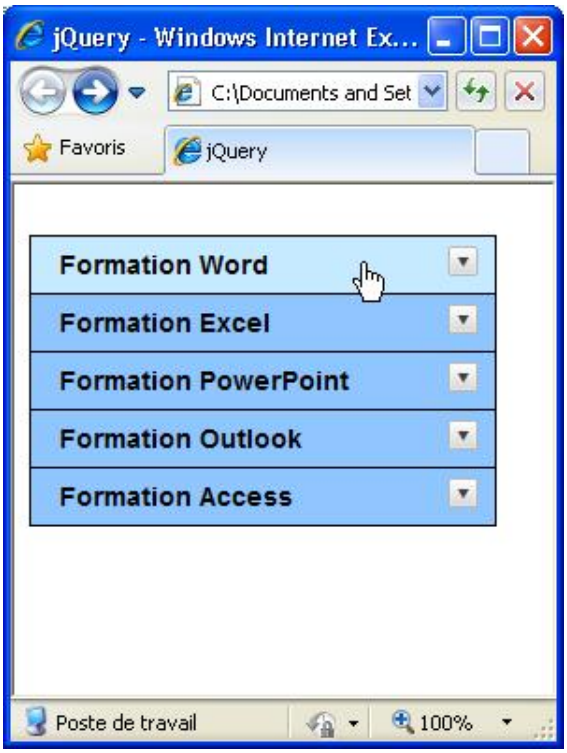
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
jQuery(document).ready(function() {
$("#start").click(function () {
$("#div_toggle").toggle("slow");
});
});
</script>
<style type="text/css" media="screen">
#div_toggle { background-color:#9cf;
              width:150px;
              height:150px;
              border: 1px solid black;}
p { cursor: pointer;
    margin-bottom: 25px;}
</style>
</head>
<body>
<p id="start">
Cliquer ici
</p>
<div id="div_toggle"></div>
</body>
</html>
```

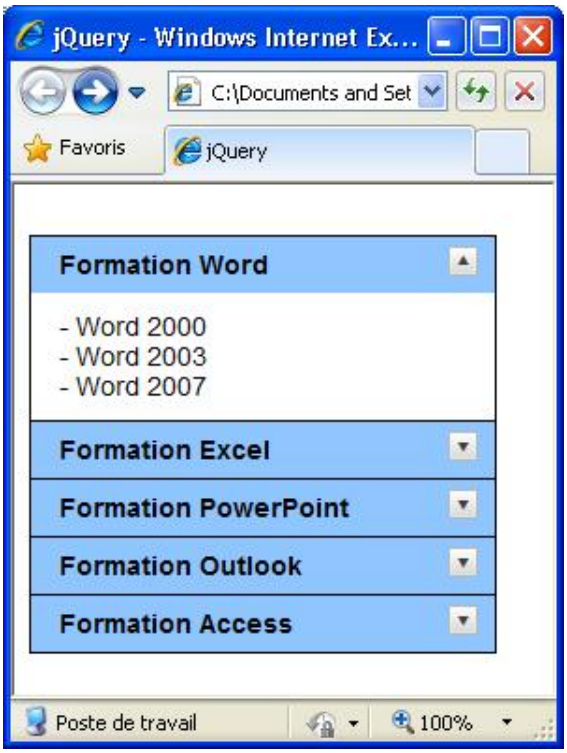
© ENI Editions - All rigths reserved - Jonifar lina

2. Un menu accordéon

Présentons un menu vertical qui se déroule et se rétracte au clic de la souris.

L'image arrow.gif de ce menu est disponible dans l'espace de téléchargement.





Le fichier de départ :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
```


© ENI Editions - All rigths reserved - Jonifar lina

```
8859-1" />
<title>jQuery</title>
<style type="text/css">
.accordeon { width: 250px;
             border-bottom: solid 1px black;}
.accordeon h3 { margin: 0;
                background: #9cf url(arrow.gif) no-repeat right -51px;
                padding: 7px 15px;
                font: bold 0.9em Arial, sans-serif;
                border: solid 1px black;
                border-bottom: none;
                cursor: pointer;}
.accordeon h3:hover { background-color: #ccecff;}
.accordeon h3.active { background-position: right 5px;}
.accordeon p { margin: 0;
               font: 0.9em Arial, sans-serif;
               padding: 10px 15px 10px;
               border-left: solid 1px black;
               border-right: solid 1px black;}
</style>
</head>
<body>
<br />
<div class="accordeon">
<h3>Formation Word</h3>
<p>- Word 2000<br />- Word 2003<br />- Word 2007</p>
<h3>Formation Excel</h3>
<p>- Excel 2000<br />- Excel 2003<br />- Excel 2007</p>
<h3>Formation PowerPoint</h3>
<p>- PowerPoint 2000<br />- PowerPoint 2003<br />- PowerPoint
2007</p>
<h3>Formation Outlook</h3>
<p>- Outlook 2000<br />- Outlook 2003<br />- Outlook 2007</p>
<h3>Formation Access</h3>
<p>- Access 2000<br />- Access 2003<br />- Access 2007</p>
</div>
</body>
</html>
```

Il faut remarquer l'utilisation d'une seule image pour afficher tantôt la flèche vers le haut et tantôt la flèche vers le bas. La flèche vers le bas est obtenue avec une position dans l'arrière-plan de -51 pixels tandis que la flèche vers le haut (la classe active) a une position de 5 pixels.

Le script jQuery :

```
<script type="text/javascript">
$(document).ready(function(){
$(".accordeon p").hide();
$(".accordeon h3").click(function(){
$(this).next("p").slideToggle("slow")
.siblings("p:visible").slideUp("slow");
$(this).toggleClass("active");
$(this).siblings("h3").removeClass("active");
});
});
</script>
```

Détaillons celui-ci.

© ENI Editions - All rigths reserved - Jonifar lina

```
$(document).ready(function(){
$(".accordeon p").hide();
```

Au chargement de la page (du DOM pour être précis), tous les sous-menus sont cachés par la méthode `hide()`.

```
$(".accordeon h3").click(function(){
$(this).next("p").slideToggle("slow")
.siblings("p:visible").slideUp("slow");
```

Au clic de la souris sur un élément du menu (`.accordeon h3`), le script appelle l'élément suivant (soit une balise `<p>`) et déroule le paragraphe qui contient les sous-menus. Les éléments frères de la balise `<p>` qui sont visibles soit les autres sous-menus, sont glissés vers le haut pour ainsi disparaître.

```
$(this).toggleClass("active");
$(this).siblings("h3").removeClass("active");
```

Ces deux lignes de code veillent à l'affichage de la petite flèche vers le haut ou vers le bas des éléments du menu. En suite du clic effectué précédemment, la flèche pointe vers le haut comme défini par la classe active. Pour les autres éléments de menu, la flèche pointera vers le bas par la suppression de cette classe `active`.

```
});
});
```

Fin du script.

Au final :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$(".accordeon p").hide();
$(".accordeon h3").click(function(){
$(this).next("p").slideToggle("slow")
.siblings("p:visible").slideUp("slow");
$(this).toggleClass("active");
$(this).siblings("h3").removeClass("active");
});
});
</script>
<style type="text/css">
.accordeon { width: 250px;
             border-bottom: solid 1px black;}
.accordeon h3 { margin: 0;
                background: #9cf url(arrow.gif) no-repeat right -51px;
                padding: 7px 15px;
                font: bold 0.9em Arial, sans-serif;
                border: solid 1px black;
                border-bottom: none;
                cursor: pointer;}
.accordeon h3:hover { background-color: #ccecff;}
.accordeon h3.active { background-position: right 5px;}
.accordeon p { margin: 0;
               font: 0.9em Arial, sans-serif;
               padding: 10px 15px 10px;
               border-left: solid 1px black;
               border-right: solid 1px black;}
</style>
</head>
<body>
<br />
<div class="accordeon">
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<h3>Formation Word</h3>
<p>- Word 2000<br />- Word 2003<br />- Word 2007</p>
<h3>Formation Excel</h3>
<p>- Excel 2000<br />- Excel 2003<br />- Excel 2007</p>
<h3>Formation PowerPoint</h3>
<p>- PowerPoint 2000<br />- PowerPoint 2003<br />- PowerPoint
2007</p>
<h3>Formation Outlook</h3>
<p>- Outlook 2000<br />- Outlook 2003<br />- Outlook 2007</p>
<h3>Formation Access</h3>
<p>- Access 2000<br />- Access 2003<br />- Access 2007</p>
</div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Créer une animation

La fonction `animate()` permet de créer et de paramétrer vos propres animations au gré de votre créativité.

animate(paramètres, vitesse, easing, fonction de rappel)

L'aspect clé de cette fonction est l'objet composé des propriétés de style sur lesquelles sera basée l'animation. Chaque paramètre de l'objet représente une propriété sur laquelle portera l'animation (exemple: `height`, `top` ou `opacity`). La valeur associée à la clé indique comment la propriété sera animée. Si la valeur est un nombre, le style de la propriété passera de sa valeur actuelle à la valeur spécifiée. Si la valeur `hide`, `show` ou `toogle` est spécifiée, une animation par défaut sera construite pour cette propriété.

À noter que ces propriétés devront être spécifiées selon la notation JavaScript (CamelCase) soit par exemple `marginLeft` au lieu de la notation CSS soit `margin-left`.

- paramètres : conteneurs d'attributs de style que vous souhaitez animer et à quelle valeur.
- vitesse (optionnel) : chaîne de caractères représentant une des trois vitesses prédéfinies (`'slow'`, `'normal'` ou `'fast'`) ou le nombre en millisecondes correspondant à la durée de l'effet.

  ```
  animate( {fontSize:"24px", left:300, width: "200px", opacity: 0.5} , 1000 )
  ```

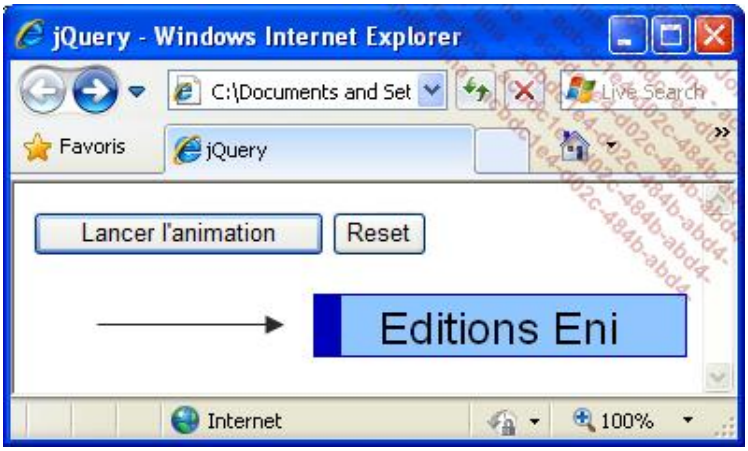
  ou

  ```
  $("p").animate({ height: 'toggle', opacity: 'toggle' }, "slow");
  ```

- easing (optionnel) : nom de l'effet customisé que vous souhaitez utiliser (plugin requis).
- fonction de rappel (optionnel) : fonction à exécuter à la fin de l'animation.

Cette méthode renvoie un objet jQuery.

1. Une animation sur une division




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
#box { position: relative;
       width: 110px;
       height: 25px;
       background-color: #9cf;
       margin: 20px 0px;
       padding-left: 20px;
       padding-right: 20px;
       padding-top: 5px;
       font: 0.75em Arial, sans-serif;
       border: 1px solid #0000c0;}
</style>
</head>
<body>
<input type="button" id="start" value="Lancer l'animation" />
<input type="button" id="reset" value="Reset" />
<div id="box">Editions Eni</div>
</body>
</html>
```

Le script jQuery :

```
<script type="text/javascript">
$(document).ready(function(){
$('#start').click(function(){
$('#box').animate({left:150,
          width: "200px",
          opacity: 0.5
          }, 1500)
.animate( { fontSize:"24px" } , 1000 )
.animate( { borderLeftWidth:"15px" }, 1000)
.animate( { opacity: 1 }, 1000);
$("#reset").click(function(){
$("#box").css({width:"",left:"",fontSize:"",opacity:"",borderWidth:""});
});
});
});
</script>
```

Détaillons cette animation.

```
$(document).ready(function(){
$('#start').click(function(){
```


© ENI Editions - All rigths reserved - Jonifar lina

Au clic sur le bouton **Lancer l'animation**.

```
$('#box').animate({left: 150,
          width: "200px",
          opacity: 0.5
          }, 1500)
```

La boîte box est déplacée vers la droite de 150 pixels (`left: 150`), sa largeur est portée à 200 pixels et son opacité diminuée de moitié. L'animation dure 1500 millisecondes.

```
.animate( { fontSize:"24px" } , 1000 )
```

Puis, la taille de la police de caractères est agrandie.

```
.animate( { borderLeftWidth:"15px" }, 1000)
```

Ensuite, l'épaisseur de la bordure gauche est portée à 15 pixels.

```
.animate( { opacity: 1 }, 1000);
```

Et pour terminer, l'opacité est rétablie à la valeur 1.

```
$("#reset").click(function(){
$("#box").css({width:"",left:"",fontSize:"",opacity:"",borderWidth:""});
```

Le clic sur le bouton **Reset** remet la boîte dans sa position initiale en annulant toutes les modifications réalisées sur la largeur, la position, l'opacité ainsi que la largeur de la bordure gauche.

```
});
});
});
```

Fin du script.

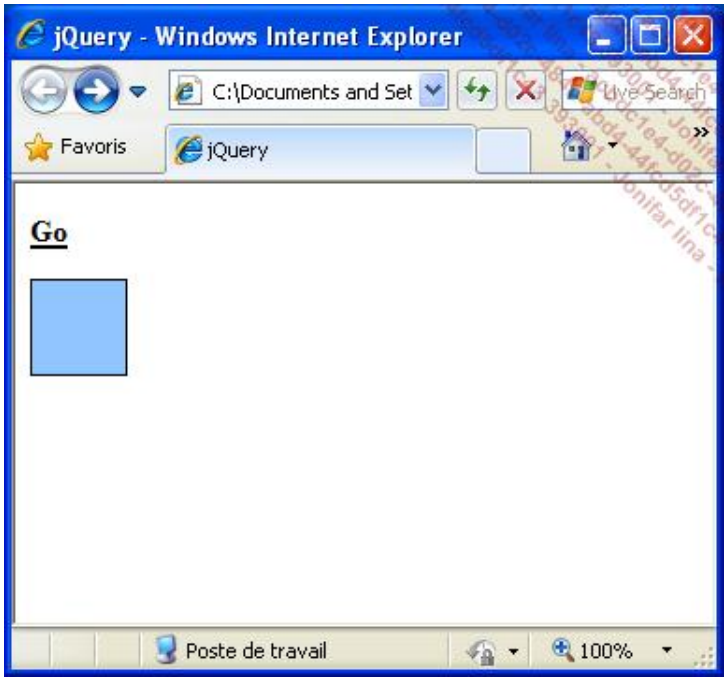
La page finale est alors :

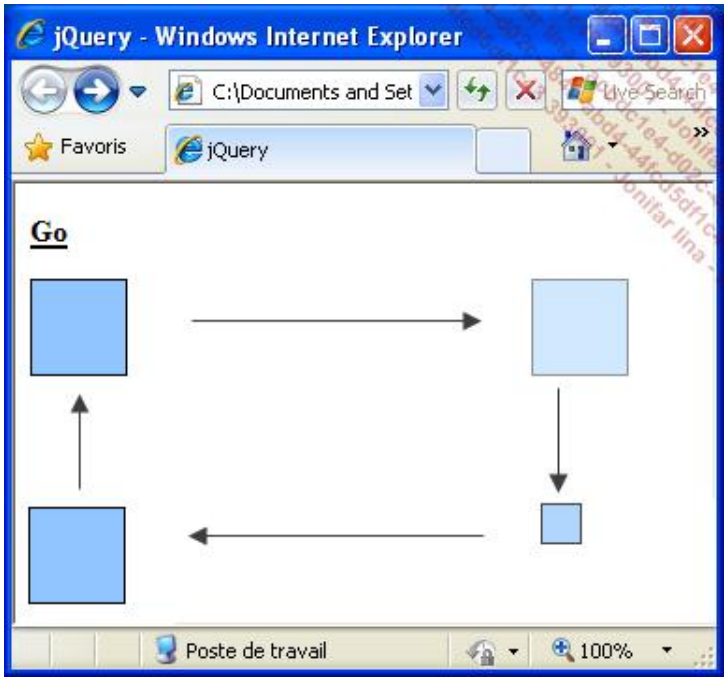
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$('#start').click(function(){
$('#box').animate({left:150,
          width: "200px",
          opacity: 0.5
          }, 1500)
.animate( { fontSize:"24px" } , 1000 )
.animate( { borderLeftWidth:"15px" }, 1000)
.animate( { opacity: 1 }, 1000);
$("#reset").click(function(){
$("#box").css({width:"",left:"",fontSize:"",opacity:"",borderWidth:""});
});
});
});
</script>
<style type="text/css">
#box { position: relative;
       width: 110px;
       height: 25px;
       background-color: #9cf;
       margin: 20px 0px;
```

© ENI Editions - All rigths reserved - Jonifar lina

```
        padding-left: 20px;
        padding-right: 20px;
        padding-top: 5px;
        font: 0.75em Arial, sans-serif;
        border: 1px solid #0000c0;}
</style>
</head>
<body>
<input type="button" id="start" value="Lancer l'animation" />
<input type="button" id="reset" value="Reset" />
<div id="box">Editions Eni</div>
</body>
</html>
```

2. Une animation évoluée





© ENI Editions - All rigths reserved - Jonifar lina

Le fichier htm de départ :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
a { color: black;
    font-weight: bold;
    font: Arial 10px;}
#box { position: relative;
       width: 50px;
       height: 50px;
       background: #9cf;
       border: 1px solid black;}
</style>
</head>
<body>
<p>
<a class="go" href="#">Go</a></p>
<div id="box"></div>
</body>
</html>
```

Le script jQuery

```
<script type="text/javascript">
$(document).ready(function(){
$(".go").click(function(){
$("#box").animate({left: "+=270", opacity: "0.4"}, 1200)
.animate({top: "+=120", opacity: "0.7", height: "20", width:
"20"}, "slow")
.animate({left: "0", opacity: "1", height: "50", width: "50"},
"slow")
.animate({top: "0"}, "fast")
.slideUp()
.slideDown()
return false;
});
});
</script>
```

Explication du script :

```
$(document).ready(function(){
$(".go").click(function(){
```

Au clic sur le lien.

```
$("#box").animate({left: "+=270", opacity: "0.4"}, 1200)
```

La boîte (`box`) se déplace vers la droite de 270 pixels (`left: "+=270`) et voit son opacité diminuée à la valeur de 0.4. Cette animation a une durée de 1200 millisecondes.

```
.animate({top: "+=120", opacity: "0.7", height: "20", width: "20"},
"slow")
```

Elle descend alors de 120pixels, son opacité passe à 0.7, sa largeur se réduit à 20 pixels ainsi que sa hauteur.

```
.animate({left: "0", opacity: "1", height: "50", width: "50"}, "slow")
```

Elle revient à sa position horizontale de départ et retrouve son aspect initial (opacité 1, hauteur et largeur de 50 pixels).

© ENI Editions - All rigths reserved - Jonifar lina

```
.animate({top: "0"}, "fast")
```

La boîte revient à sa position verticale de départ.

```
.slideUp()
.slideDown()
return false;
```

Une animation finale fait glisser la boîte vers le haut pour descendre aussitôt.

```
});
});
```

Fin de script.

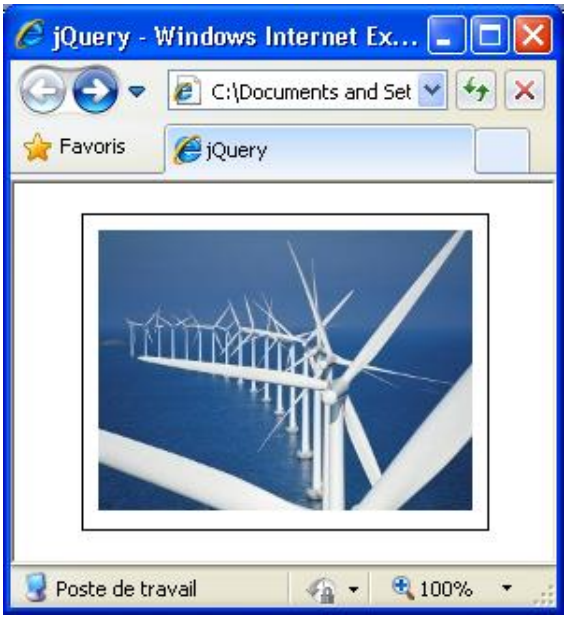
La page Html finale est ainsi :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$(".go").click(function(){
$("#box").animate({left: "+=270", opacity: "0.4"}, 1200)
.animate({top: "+=120", opacity: "0.7", height: "20", width:
"20"}, "slow")
.animate({left: "0", opacity: "1", height: "50", width: "50"},
"slow")
.animate({top: "0"}, "fast")
.slideUp()
.slideDown()
return false;
});
});
</script>
<style type="text/css">
a { color: black;
    font-weight: bold;
    font: Arial 10px;}
#box { position: relative;
       width: 50px;
       height: 50px;
       background: #9cf;
       border: 1px solid black;}
</style>
</head>
<body>
<p>
<a class="go" href="#">Go</a></p>
<div id="box"></div>
</body>
</html>
```

3. Un effet original au survol de la souris

Au survol de l’image, celle-ci glisse vers la droite pour laisser apparaître la légende.

© ENI Editions - All rigths reserved - Jonifar lina

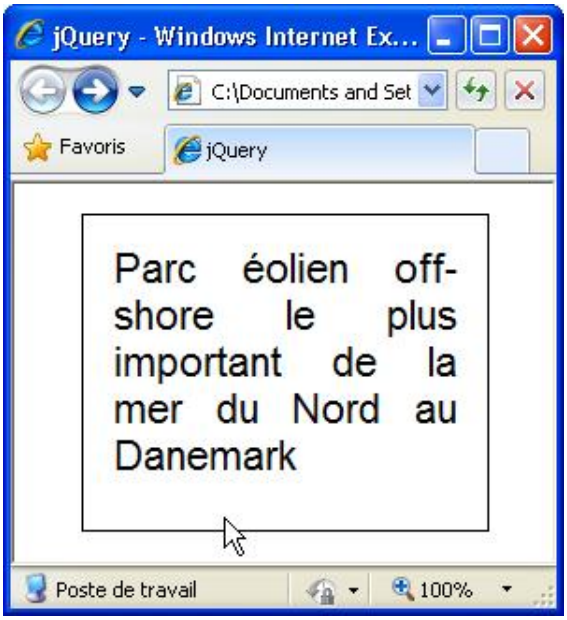


Au départ :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css" media="screen">
ul.hover_bloc li { position: relative;
                   width: 184px;
                   height: 136px;
                   padding: 16px;
                   list-style-type: none;
                   font: 1.4em Arial, sans-serif;
                   text-align: justify;
                   color: black;
                   overflow: hidden;
                   border : 1px solid black;}
ul.hover_bloc li img { position: absolute;
                       top: 8px;
                       left: 8px;
                       border: 0px;}
</style>
</head>
<body>
<ul class="hover_bloc">
<li>
<img alt="" src="parc_eole.jpg" />
Parc éolien off-shore le plus important de la mer du Nord au
Danemark
</li>
</ul>
</body>
</html>
```

Le script jQuery :

© ENI Editions - All rigths reserved - Jonifar lina




```
<script type="text/javascript">
$(document).ready(function(){
$('ul.hover_bloc li').hover(function(){
$(this).find('img').animate({left:'300px'},{duration:500});
}, function(){
$(this).find('img').animate({left:'8px'},{duration:500});
});
});
</script>
```

Détaillons celui-ci.

```
$(document).ready(function(){
$('ul.hover_bloc li').hover(function(){
```

Au survol par la souris de la zone.

```
$(this).find('img').animate({left:'300px'},{duration:500});
```

Le script trouve l'image (`find('img')`) et la fait glisser vers la droite de 300 pixels (`left:'300px'`). L'animation dure 500 millisecondes.

© ENI Editions - All rigths reserved - Jonifar lina

```
}, function(){
$(this).find('img').animate({left:'8px'},{duration:500});
```

Lorsque le curseur quitte la zone, l'image revient à sa position initiale.

```
});
});
```

Fin du script.

Il faut admirer au passage, la concision du code engendré par jQuery.

La page finale est alors :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$('ul.hover_bloc li').hover(function(){
$(this).find('img').animate({left:'300px'},{duration:500});
}, function(){
$(this).find('img').animate({left:'8px'},{duration:500});
});
});
</script>
<style type="text/css" media="screen">
ul.hover_bloc li { position: relative;
                   width: 184px;
                   height: 136px;
                   padding: 16px;
                   list-style-type: none;
                   font: 1.4em Arial, sans-serif;
                   text-align: justify;
                   color: black;
                   overflow: hidden;
                   border : 1px solid black;}
ul.hover_bloc li img { position: absolute;
                       top: 8px;
                       left: 8px;
                       border: 0px;}
</style>
</head>
<body>
<ul class="hover_bloc">
<li>
<img alt="" src="parc_eole.jpg" />
Parc éolien off-shore le plus important de la mer du Nord au
Danemark
</li>
</ul>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Introduction

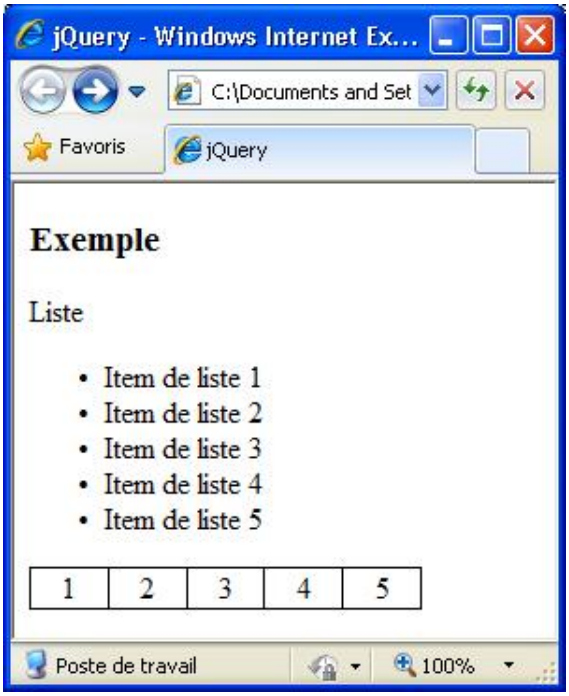
En permettant au concepteur d'accéder à chacun des éléments de la page, le DOM a relancé le JavaScript. Mais il faut bien admettre que le traçage de parents à enfants et autres frères n'est pas toujours très aisé à mettre en place. En outre, une simple modification entraîne souvent une réécriture complète du code. La librairie jQuery remédie grandement à ces inconvénients par ses nombreux sélecteurs (chapitre Les sélecteurs en jQuery) et par des méthodes spécifiques pour traverser et manipuler les éléments du DOM.

Pour ce chapitre, nous utiliserons une page type. Celle-ci comporte une liste non ordonnée avec 5 items et un tableau d'une ligne et 5 colonnes.

Il faut remarquer aussi les divisions `exemple` et `contenu` car elles interviennent plus loin dans notre étude.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
table { border: 1px solid black;
        border-collapse:collapse;}
td { border: 1px solid black;
     text-align: center;}
.contenu { width: 210px;}
.bordure { border: 1px solid black;}
.arriereplan { background-color: #9cf;}
</style>
</head>
<body>
<div id="exemple">
<h3>Exemple</h3>
<div class="contenu">
<p>Liste</p>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li>Item de liste 3</li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
<table width="210">
<tr>
<td>1</td><td>2</td><td>3</td><td>4</td><td>5</td>
</tr>
</table>
</div>
</div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina



© ENI Editions - All rigths reserved - Jonifar lina

Trouver les enfants

children()

Récupère un groupe d'éléments contenant les enfants immédiats de chacun des éléments concernés par la sélection.

```
$("div").children()
```

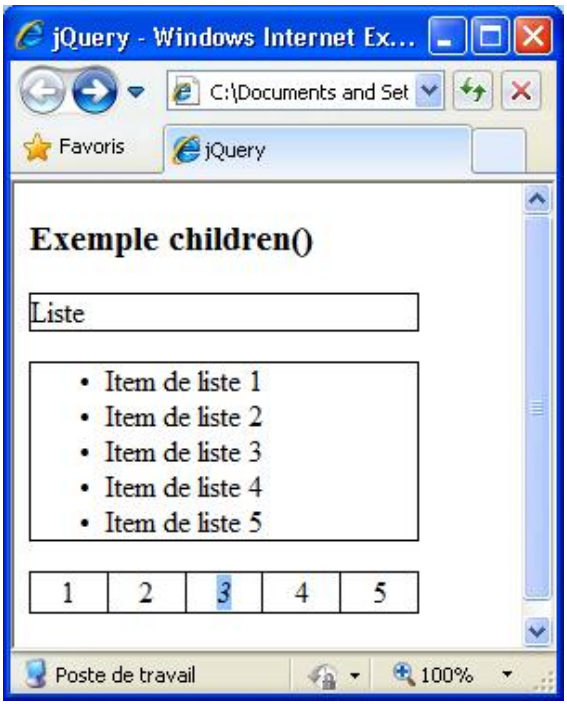
Cette méthode renvoie un objet jQuery.

Exemple

Ajoutons une bordure aux enfants de la division `<div class="contenu">` et un arrière-plan aux enfants de la cellule 3 du tableau.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$(".contenu").children().addClass("bordure");
$("#select_table").children().addClass("arriereplan");
});
</script>
<style type="text/css">
table { border: 1px solid black;
        border-collapse:collapse;}
td { border: 1px solid black;
     text-align: center;}
.contenu { width: 210px;}
.bordure { border: 1px solid black;}
.arriereplan { background-color: #9cf;}
</style>
</head>
<body>
<div id="exemple">
<h3>Exemple children()</h3>
<div class="contenu">
<p>Liste</p>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li>Item de liste 3</li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
<table width="210">
<tr>
<td>1</td><td>2</td><td
id="select_table"><i>3</i></td><td>4</td><td>5</td>
</tr>
</table>
</div>
</div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina




```
$(".contenu").children().addClass("bordure");
```

La méthode `children()` appliquée à la division dont la classe est `contenu` a entouré d'une bordure les enfants immédiats de celle-ci, soit le paragraphe `<p>Liste</p>` et la liste non ordonnée `<ul> ... </ul>`.

```
$("#select_table").children().addClass("arriereplan");
```

La méthode `children()` appliquée à la cellule dont l'identifiant est `select_table` a ajouté un arrière-plan de couleur aux enfants immédiats de celle-ci, soit son contenu (le chiffre 3).

Commentaire

Il est possible de filtrer les éléments retournés par jQuery en utilisant une expression optionnelle. La méthode `children()` prend alors la forme de `children(expression)` qui permet de ne retenir que les éléments enfants qui répondent à l'expression introduite.

Exemple :

Soit l'extrait de code suivant :

```
<ul>
<li class="selected">Item de liste 1</li>
<li>Item de liste 2</li>
<li class="selected">Item de liste 3</li>
<li>Item de liste 4</li>
<li class="selected">Item de liste 5</li>
</ul>
```

Il est possible de ne retenir parmi les enfants de l'élément `<ul>` que les éléments dont la classe est `selected`.

Le code jQuery est alors :

```
$("ul").children(.selected);
```


© ENI Editions - All rigths reserved - Jonifar lina

jQuery

Le framework JavaScript du Web 2.0

Luc VAN LANCKER

Résumé

Ce livre sur jQuery s'adresse à des **experts ou des candidats experts** dans la création de sites Web. La connaissance, sinon la maîtrise du **JavaScript**, des feuilles de style **CSS**, du **DOM** et de **l'AJAX** sont des pré-requis indispensables à la compréhension et à la mise en pratique de cet ouvrage.
Dans ce livre, l'auteur a privilégié une **approche structurée et progressive**. Chaque thème de jQuery est illustré par un exemple avant de passer à une mise en pratique sur des applications plus pointues.
Après une présentation du **framework**, l'auteur consacre un chapitre aux **sélecteurs**, qui non seulement illustrent la diversité de jQuery pour atteindre aisément n'importe quel élément de la page mais qui sont aussi un concept essentiel dans l'apprentissage de jQuery. Dans les chapitres suivants le lecteur découvre les **éléments d'interactivité** apportés par jQuery d'abord par la manipulation des attributs (ajout, modification ou suppression à la volée) puis par l'application aux **feuilles de style CSS**. Suivent les **événements** classiques du **JavaScript** mais surtout ceux apportés par jQuery. Après la présentation des **effets visuels** aussi nombreux qu'originaux, l'étude du **DOM** et de **l'AJAX revisité par jQuery** est longuement détaillée. Le chapitre final passe en revue quelques-uns des nombreux **plug-ins** développés par la **communauté jQuery** qui permettent d'apporter, en quelques lignes de code, des effets pour le moins spectaculaires.
Sa lecture terminée, le lecteur sera à même de développer des applications web **plus interactives**, **plus riches et plus innovantes**, le tout avec une facilité d'écriture du JavaScript insoupçonnée.
Les exemples du livre ainsi que les illustrations utilisées, la librairie jQuery et les fichiers relatifs aux plugins étudiés sont disponibles en téléchargement sur cette page.

Les chapitres du livre :
Démarrer avec jQuery - Les sélecteurs en jQuery - Manipuler les attributs - Manipulation des feuilles de style CSS - Les événements - Les effets - Traverser le DOM - Manipuler le DOM - Filtrer le DOM - AJAX - Quelques méthodes utilitaires - Les formulaires - Les plug-ins jQuery

L'auteur

Dès les débuts d'Internet, **Luc Van Lancker**, enthousiasmé par l'idée de communication universelle que véhiculait ce concept, s'est complètement investi dans ce domaine. C'est un formateur passionné, très au fait des nouvelles technologies liées au web et grand pédagogue. Luc Van Lancker est auteur aux Editions ENI de livres sur HTML, XHTML, AJAX, des CSS au DHTML dans différentes collections.

Ce livre numérique a été conçu et est diffusé dans le respect des droits d'auteur. Toutes les marques citées ont été déposées par leur éditeur respectif. La loi du 11 Mars 1957 n'autorisant aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les "copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective", et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, "toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayant cause, est illicite" (alinéa 1er de l'article 40). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal. Copyright Editions ENI

Trouver les parents directs

parent()

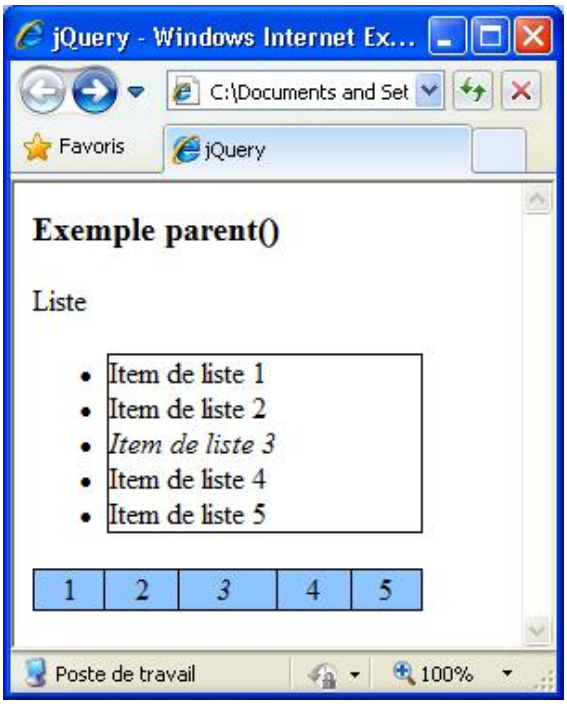
Récupère un groupe d'éléments contenant les parents immédiats de chacun des éléments concernés par la sélection.

```
$("span").parent()
```

Cette méthode renvoie un objet jQuery.

Exemple

Entourons d'une bordure les parents immédiats du troisième élément de la liste et ajoutons un arrière-plan de couleur aux parents immédiats de la troisième cellule du tableau.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#select_li").parent().addClass("bordure");
$("#select_table").parent().addClass("arriereplan");
});
</script>
<style type="text/css">
table { border: 1px solid black;
        border-collapse:collapse;}
td { border: 1px solid black;
     text-align: center;}
.contenu { width: 210px;}
.bordure { border: 1px solid black;}
.arriereplan { background-color: #9cf;}
</style>
</head>
<body>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<div id="exemple">
<h3>Exemple parent()</h3>
<div class="contenu">
<p>Liste</p>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li id="select_li"><i>Item de liste 3</i></li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
<table width="210">
<tr>
<td>1</td><td>2</td><td
id="select_table"><i>3</i></td><td>4</td><td>5</td>
</tr>
</table>
</div>
</div>
</body>
</html>
```

```
$("#select_li").parent().addClass("bordure");
```

La méthode `parent()` appliquée au troisième item de la liste (`id="select_li"`) a entouré d'une bordure les parents directs de celui-ci, soit la balises `<ul>` et uniquement celle-ci.

```
$("#select_table").parent().addClass("arriereplan");
```

La méthode `parent()` appliquée à la troisième cellule du tableau (`select_table`) a ajouté un arrière-plan aux parents directs de celle-ci, soit la balise de tableau `<table>` et uniquement celle-ci.

Commentaire

Avec `parent(expression)`, les éléments retournés peuvent être filtrés de sorte que seuls les éléments qui répondent à l'expression soient retenus.

© ENI Editions - All rigths reserved - Jonifar lina

Trouver tous les parents

parents()

Récupère un groupe d'éléments contenant tous les parents de chacun des éléments concernés par la sélection.

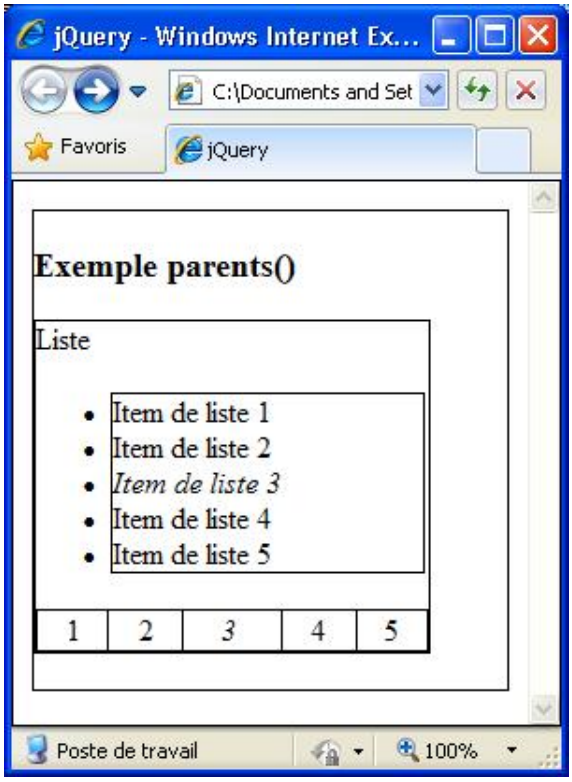
```
$("li").parents()
```

Cette méthode renvoie un objet jQuery.

La méthode `parents()` renvoie tous les ascendants alors que `children()` ne prend en compte que les éléments enfants immédiats.

Exemple

Entourons d'une bordure les parents du troisième élément de la liste.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#select_li").parents().addClass("bordure");
});
</script>
<style type="text/css">
table { border: 1px solid black;
        border-collapse:collapse;}
td { border: 1px solid black;
     text-align: center;}
```

© ENI Editions - All rigths reserved - Jonifar lina

```
.contenu { width: 210px;}
.bordure { border: 1px solid black;}
.arriereplan { background-color: #9cf;}
</style>
</head>
<body>
<div id="exemple">
<h3>Exemple parents()</h3>
<div class="contenu">
<p>Liste</p>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li id="select_li"><i>Item de liste 3</i></li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
<table width="210">
<tr>
<td>1</td><td>2</td><td>3</td><td>4</td><td>5</td>
</tr>
</table>
</div>
<br />
</div>
</body>
</html>
```

```
$("#select_li").parents().addClass("bordure");
```

La méthode `parents()` appliquée au troisième item de la liste (`id="select_li"`) a entouré d'une bordure tous les parents de celui-ci, soit la balises `<ul>`, la division contenu et la balise `<body>`.

Commentaire

Avec `parents(expression)`, les éléments retournés peuvent être filtrés de sorte que seuls les éléments qui répondent à l'expression soient retenus.

© ENI Editions - All rigths reserved - Jonifar lina

Trouver les frères

siblings()

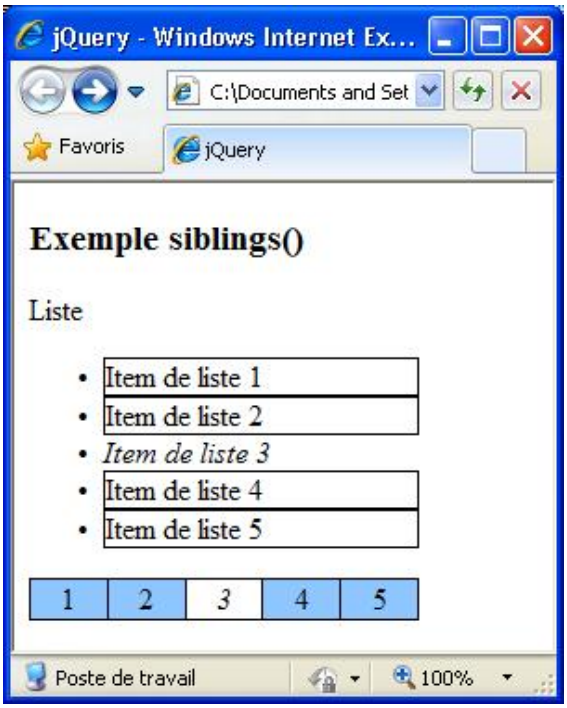
Retourne la liste des frères immédiats de chaque élément de la sélection.

```
$("div").siblings()
```

Cette méthode renvoie un objet jQuery.

Exemple

Entourons d'une bordure les frères du troisième élément de la liste et ajoutons un arrière-plan de couleur aux frères de la troisième cellule du tableau.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#select_li").siblings().addClass("bordure");
$("#select_table").siblings().addClass("arriereplan");
});
</script>
<style type="text/css">
table { border: 1px solid black;
        border-collapse:collapse;}
td { border: 1px solid black;
     text-align: center;}
.contenu { width: 210px;}
.bordure { border: 1px solid black;}
.arriereplan { background-color: #9cf;}
</style>
</head>
<body>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<div id="exemple">
<h3>Exemple siblings()</h3>
<div class="contenu">
<p>Liste</p>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li id="select_li"><i>Item de liste 3</i></li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
<table width="210">
<tr>
<td>1</td><td>2</td><td
id="select_table"><i>3</i></td><td>4</td><td>5</td>
</tr>
</table>
</div>
</div>
</body>
</html>
```

```
$("#select_li").siblings().addClass("bordure");
```

La méthode `siblings()` appliquée au troisième item de la liste (`id="select_li"`) a entouré d'une bordure les frères immédiats de celui-ci, soit les autres balises `<li>`.

```
$("#select_table").siblings().addClass("arriereplan");
```

La méthode `siblings()` appliquée à la troisième cellule du tableau (`select_table`) a ajouté un arrière-plan aux frères immédiats de celle-ci, soit les autres balises `<td>`.

Commentaire

Avec `siblings(expression)`, les éléments retournés peuvent être filtrés de sorte que seuls les éléments qui répondent à l'expression soient retenus.

© ENI Editions - All rigths reserved - Jonifar lina

Trouver le frère précédent

prev()

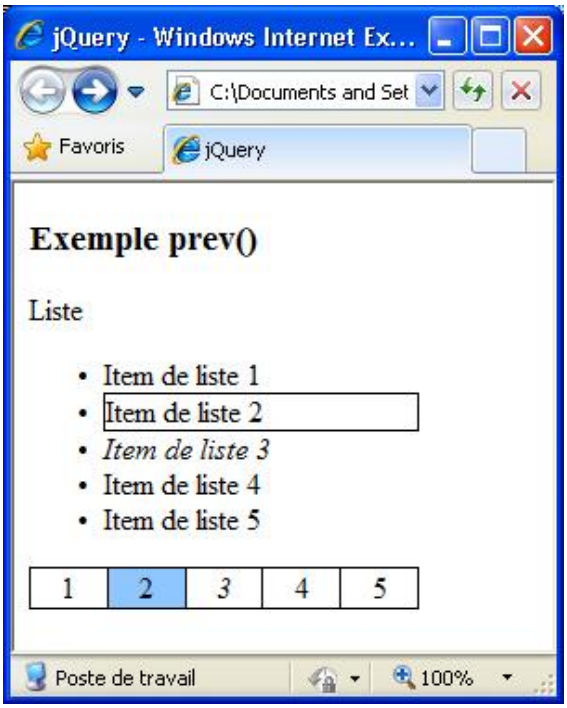
Retourne le frère immédiat précédent de chaque élément de la sélection.

```
$("td").prev()
```

Cette méthode renvoie un objet jQuery.

Exemple

Entourons d'une bordure le frère précédent du troisième élément de la liste et ajoutons un arrière-plan de couleur au frère précédent de la troisième cellule du tableau.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#select_li").prev().addClass("bordure");
$("#select_table").prev().addClass("arriereplan");
});
</script>
<style type="text/css">
table { border: 1px solid black;
        border-collapse:collapse;}
td { border: 1px solid black;
     text-align: center;}
.contenu { width: 210px;}
.bordure { border: 1px solid black;}
.arriereplan { background-color: #9cf;}
</style>
</head>
<body>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<div id="exemple">
<h3>Exemple prev()</h3>
<div class="contenu">
<p>Liste</p>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li id="select_li"><i>Item de liste 3</i></li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
<table width="210">
<tr>
<td>1</td><td>2</td><td
id="select_table"><i>3</i></td><td>4</td><td>5</td>
</tr>
</table>
</div>
</div>
</body>
</html>
```

```
$("#select_li").prev().addClass("bordure");
```

La méthode `prev()` appliquée au troisième item de la liste (`id="select_li"`) a entouré d'une bordure le frère immédiatement précédent de celui-ci, soit la balise `<li>Item de liste 2</li>`.

```
$("#select_table").prev().addClass("arriereplan");
```

La méthode `prev()` appliquée à la troisième cellule du tableau (`select_table`) a ajouté un arrière-plan au frère immédiatement précédent de celle-ci, soit la balise `<td>2</td>`.

Commentaire

Avec `prev(expression)`, les éléments retournés peuvent être filtrés de sorte que seuls les éléments qui répondent à l'expression soient retenus.

© ENI Editions - All rigths reserved - Jonifar lina

Trouver les frères précédents

prevAll()

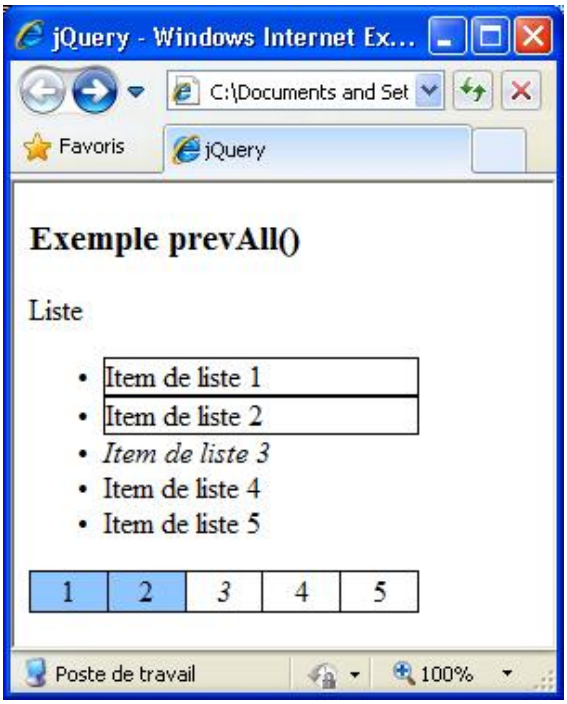
Retourne les frères immédiats précédents de chaque élément de la sélection.

```
$("td").prevAll()
```

Cette méthode renvoie un objet jQuery.

Exemple

Entourons d'une bordure le frère précédent du troisième élément de la liste et ajoutons un arrière-plan de couleur au frère précédent de la troisième cellule du tableau.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#select_li").prevAll().addClass("bordure");
$("#select_table").prevAll().addClass("arriereplan");
});
</script>
<style type="text/css">
table { border: 1px solid black;
        border-collapse:collapse;}
td { border: 1px solid black;
     text-align: center;}
.contenu { width: 210px;}
.bordure { border: 1px solid black;}
.arriereplan { background-color: #9cf;}
</style>
</head>
<body>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<div id="exemple">
<h3>Exemple prevAll()</h3>
<div class="contenu">
<p>Liste</p>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li id="select_li"><i>Item de liste 3</i></li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
<table width="210">
<tr>
<td>1</td><td>2</td><td
id="select_table"><i>3</i></td><td>4</td><td>5</td>
</tr>
</table>
</div>
</div>
</body>
</html>
```

```
$("#select_li").prevAll().addClass("bordure");
```

La méthode `prevAll()` appliquée au troisième item de la liste (`id="select_li"`) a entouré d'une bordure les frères précédents de celui-ci, soit les balises `<li>Item de liste 1</li>` et `<li>Item de liste 2</li>`.

```
$("#select_table").prevAll().addClass("arriereplan");
```

La méthode `prevAll()` appliquée à la troisième cellule du tableau (`select_table`) a ajouté d'un arrière-plan les frères précédents de celle-ci, soit les balises `<td>1</td>` et `<td>2</td>`.

Commentaire

Avec `prevAll(expression)`, les éléments retournés peuvent être filtrés de sorte que seuls les éléments qui répondent à l'expression soient retenus.

© ENI Editions - All rigths reserved - Jonifar lina

Trouver le frère suivant

next()

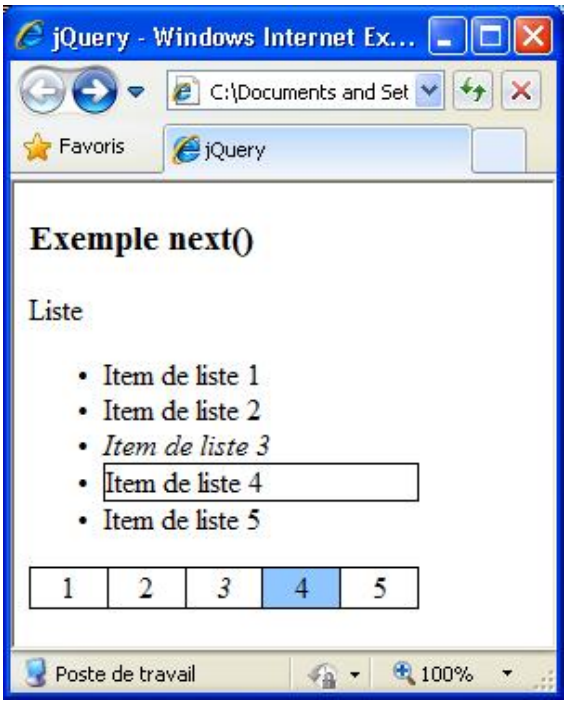
Retourne le frère immédiat suivant de chaque élément de la sélection.

```
$("td").prev()
```

Cette méthode renvoie un objet jQuery.

Exemple

Entourons d'une bordure le frère suivant du troisième élément de la liste et ajoutons un arrière-plan de couleur au frère suivant de la troisième cellule du tableau.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#select_li").next().addClass("bordure");
$("#select_table").next().addClass("arriereplan");
});
</script>
<style type="text/css">
table { border: 1px solid black;
        border-collapse:collapse;}
td { border: 1px solid black;
     text-align: center;}
.contenu { width: 210px;}
.bordure { border: 1px solid black;}
.arriereplan { background-color: #9cf;}
</style>
</head>
<body>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<div id="exemple">
<h3>Exemple next()</h3>
<div class="contenu">
<p>Liste</p>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li id="select_li"><i>Item de liste 3</i></li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
<table width="210">
<tr>
<td>1</td><td>2</td><td
id="select_table"><i>3</i></td><td>4</td><td>5</td>
</tr>
</table>
</div>
</div>
</body>
</html>
```

```
$("#select_li").next().addClass("bordure");
```

La méthode `next()` appliquée au troisième item de la liste (`id="select_li"`) a entouré d'une bordure le frère immédiatement suivant de celui-ci, soit la balise `<li>Item de liste 4</li>`.

```
$("#select_table").next().addClass("arriereplan");
```

La méthode `next()` appliquée à la troisième cellule du tableau (`select_table`) a ajouté un arrière-plan au frère immédiatement suivant de celle-ci, soit la balise `<td>4</td>`.

Commentaire

Avec `next(expression)`, les éléments retournés peuvent être filtrés de sorte que seuls les éléments qui répondent à l'expression soient retenus.

© ENI Editions - All rigths reserved - Jonifar lina

Trouver les frères suivants

nextAll()

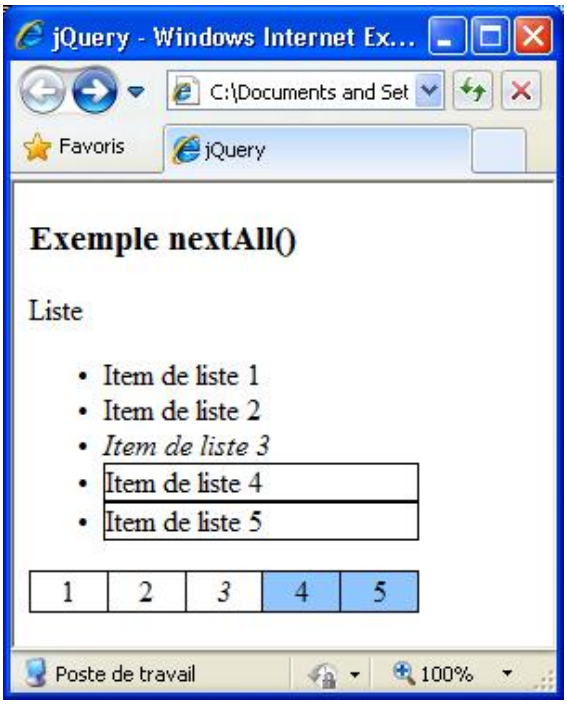
Retourne les frères immédiats suivants de chaque élément de la sélection.

```
$("td").nextAll()
```

Cette méthode renvoie un objet jQuery.

Exemple

Entourons d'une bordure les frères immédiatement suivants du troisième élément de la liste et ajoutons un arrière-plan de couleur aux frères immédiatement suivants de la troisième cellule du tableau.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#select_li").nextAll().addClass("bordure");
$("#select_table").nextAll().addClass("arriereplan");
});
</script>
<style type="text/css">
table { border: 1px solid black;
        border-collapse:collapse;}
td { border: 1px solid black;
     text-align: center;}
.contenu { width: 210px;}
.bordure { border: 1px solid black;}
.arriereplan { background-color: #9cf;}
</style>
</head>
<body>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<div id="exemple">
<h3>Exemple nextAll()</h3>
<div class="contenu">
<p>Liste</p>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li id="select_li"><i>Item de liste 3</i></li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
<table width="210">
<tr>
<td>1</td><td>2</td><td
id="select_table"><i>3</i></td><td>4</td><td>5</td>
</tr>
</table>
</div>
</div>
</body>
</html>
```

```
$("#select_li").nextAll().addClass("bordure");
```

La méthode `nextAll()` appliquée au troisième item de la liste (`id="select_li"`) a entouré d'une bordure, les frères suivants de celui-ci, soit les balises `<li>Item de liste 4</li>` et `<li>Item de liste 5</li>`.

```
$("#select_table").nextAll().addClass("arriereplan");
```

La méthode `nextAll()` appliquée à la troisième cellule du tableau (`select_table`) a ajouté un arrière-plan aux frères suivants de celle-ci, soit les balises `<td>4</td>` et `<td>5</td>`.

Commentaire

Avec `nextAll(expression)`, les éléments retournés peuvent être filtrés de sorte que seuls les éléments qui répondent à l'expression soient retenus.

© ENI Editions - All rigths reserved - Jonifar lina

Trouver le contenu

contents()

Trouve tous les nœuds enfants situés dans les éléments de la sélection (incluant les nœuds texte).

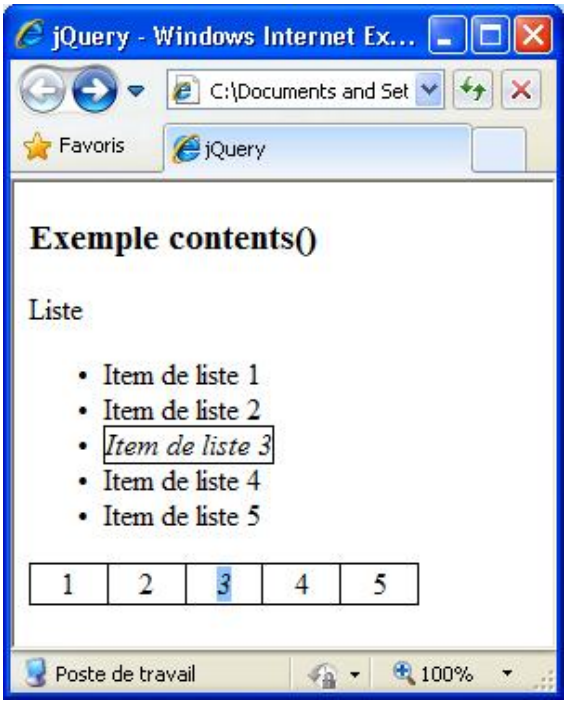
Si l'élément spécifié est une balise `<iframe>`, `contents()` trouve le contenu du document.

```
$("p").contents()
```

Cette méthode renvoie un objet jQuery.

Exemple

Entourons d'une bordure, le contenu du troisième élément de la liste et ajoutons un arrière-plan de couleur au contenu de la troisième cellule du tableau.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#select_li").contents().addClass("bordure");
$("#select_table").contents().addClass("arriereplan");
});
</script>
<style type="text/css">
table { border: 1px solid black;
        border-collapse:collapse;}
td { border: 1px solid black;
     text-align: center;}
.contenu { width: 210px;}
.bordure { border: 1px solid black;}
.arriereplan { background-color: #9cf;}
</style>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
</head>
<body>
<div id="exemple">
<h3>Exemple contents()</h3>
<div class="contenu">
<p>Liste</p>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li id="select_li"><i>Item de liste 3</i></li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
<table width="210">
<tr>
<td>1</td><td>2</td><td
id="select_table"><i>3</i></td><td>4</td><td>5</td>
</tr>
</table>
</div>
</div>
</body>
</html>
```

```
$("#select_li").contents().addClass("bordure");
```

La méthode `contents()` appliquée au troisième item de la liste (`id="select_li"`) a entouré d'une bordure, le nœud texte enfant de celui-ci, soit les mots `"Item de liste 3"`.

```
$("#select_table").contents().addClass("arriereplan");
```

La méthode `contents()` appliquée à la troisième cellule du tableau (`select_table`) a ajouté un arrière-plan au nœud texte enfant, soit le chiffre 3.

Commentaire

Avec `contents(expression)`, les éléments retournés peuvent être filtrés de sorte que seuls les éléments qui répondent à l'expression soient retenus.

© ENI Editions - All rigths reserved - Jonifar lina

Trouver certains parents

closest(sélecteur)

Retourne l'ensemble d'éléments contenant le parent le plus proche de l'élément sélectionné répondant au sélecteur, l'élément de départ inclus.

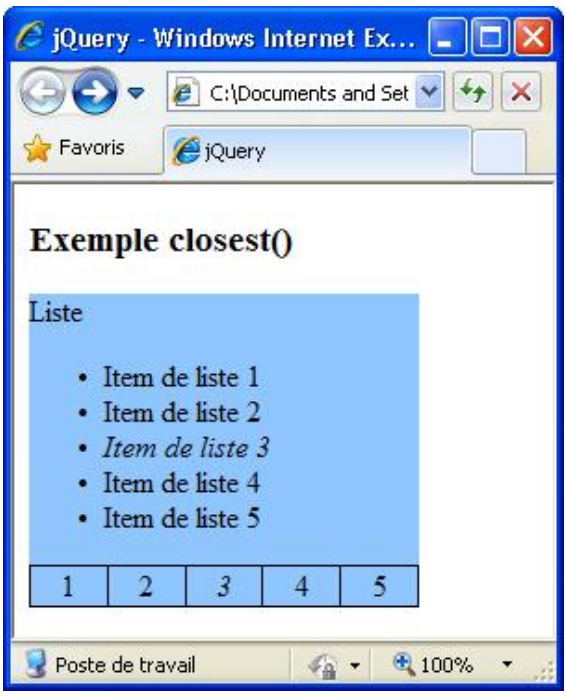
La méthode `closest()` vérifie d'abord si l'élément courant répond à l'expression spécifiée. Dans l'affirmative, il retourne simplement l'élément spécifié. Sinon, il continue alors de traverser le document vers le haut, parent par parent, jusqu'à ce qu'il trouve un élément répondant à la condition de l'expression. Si aucun élément n'est trouvé, la méthode ne retourne rien.

```
$("div").closest("p")
```

Cette méthode renvoie un objet jQuery.

Exemple

Retrouvons les parents du troisième élément de la liste jusqu'à la division dont la classe est `contenu` et ajoutons un arrière-plan de couleur.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#select_li").closest(".contenu").addClass("arriereplan");
});
</script>
<style type="text/css">
table { border: 1px solid black;
        border-collapse:collapse;}
td { border: 1px solid black;
     text-align: center;}
.contenu { width: 210px;}
.bordure { border: 1px solid black;}
```

© ENI Editions - All rigths reserved - Jonifar lina

```
.arriereplan { background-color: #9cf;}
</style>
</head>
<body>
<div id="exemple">
<h3>Exemple closest()</h3>
<div class="contenu">
<p>Liste</p>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li id="select_li"><i>Item de liste 3</i></li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
<table width="210">
<tr>
<td>1</td><td>2</td><td
id="select_table"><i>3</i></td><td>4</td><td>5</td>
</tr>
</table>
</div>
</div>
</body>
</html>
```

```
$("#select_li").closest(".contenu").addClass("arriereplan");
```

Avec la méthode `closest()`, nous allons remonter le DOM jusqu'à la division dont la classe est contenu.

D'autres manipulations sont possibles :

```
$("#select_li").closest("#exemple").addClass("arriereplan");
```

ou

```
$("#select_li").closest("ul").addClass("arriereplan");
```

Commentaire

Avec `closest(expression)`, les éléments retournés peuvent être filtrés de sorte que seuls les éléments qui répondent à l'expression soient retenus.

© ENI Editions - All rigths reserved - Jonifar lina

Trouver certains descendants

find(expression ou sélecteur)

Recherche les éléments descendants répondant aux conditions du sélecteur spécifié.

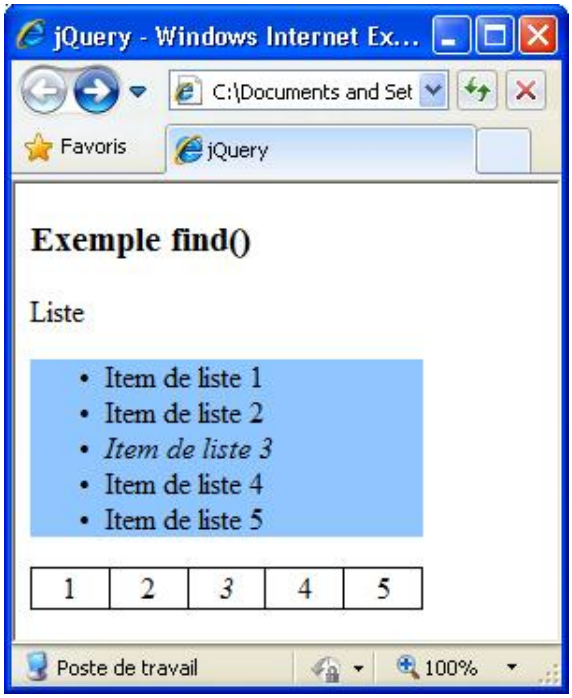
```
$("div").find("p")
```

Cette méthode renvoie un objet jQuery.

Cette méthode `find()` ne cherche pas l'élément spécifié mais bien les descendants de celui-ci.

Exemple

Trouvez la balise non ordonnée `<ul>` qui est un descendant de la division identifiée par `exemple`.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#exemple").find("ul").addClass("arriereplan");
});
</script>
<style type="text/css">
table { border: 1px solid black;
        border-collapse:collapse;}
td { border: 1px solid black;
     text-align: center;}
.contenu { width: 210px;}
.bordure { border: 1px solid black;}
.arriereplan { background-color: #9cf;}
</style>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
</head>
<body>
<div id="exemple">
<h3>Exemple find()</h3>
<div class="contenu">
<p>Liste</p>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li id="select_li"><i>Item de liste 3</i></li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
<table width="210">
<tr>
<td>1</td><td>2</td><td
id="select_table"><i>3</i></td><td>4</td><td>5</td>
</tr>
</table>
</div>
</div>
</body>
</html>
```

```
$("#exemple").find("ul").addClass("arriereplan");
```

En partant de la division `id="exemple"`, trouvez parmi les descendants, une liste non ordonnée (`<ul>`). Celle-ci est alors dotée d'un arrière-plan de couleur.

D'autres manipulations sont possibles :

```
$("#exemple").find("p").addClass("arriereplan");
```

ou

```
$("ul").find("#select_li").addClass("arriereplan");
```


© ENI Editions - All rigths reserved - Jonifar lina

Ajouter des éléments à la sélection

add(sélecteur ou élément(s) ou Html)

Ajoute des éléments, fournis en argument, à l'ensemble des éléments sélectionnés dans la recherche.

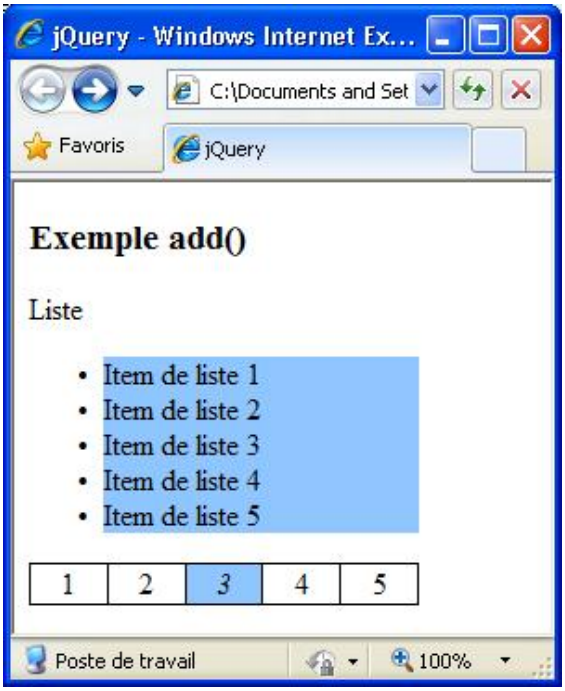
Les paramètres peuvent être :

- un sélecteur jQuery : `$("p").add("span") ;`
- un ou des éléments : `$("p").add(document.getElementById("a")) ;`
- du code Html : `$("p").add("<span>Again</span>").`

Cette méthode renvoie un objet jQuery.

Exemple

Après avoir retenu les éléments de la liste, ajoutons à la sélection, la cellule du tableau identifiée par l'identifiant `select_table`.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("ul").children().add("#select_table").addClass("arriereplan");
});
</script>
<style type="text/css">
table { border: 1px solid black;
        border-collapse:collapse;}
td { border: 1px solid black;
     text-align: center;}
```

© ENI Editions - All rigths reserved - Jonifar lina

```
.contenu { width: 210px;}
.bordure { border: 1px solid black;}
.arriereplan { background-color: #9cf;}
</style>
</head>
<body>
<div id="exemple">
<h3>Exemple add()</h3>
<div class="contenu">
<p>Liste</p>
<ul>
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li>Item de liste 3</li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
<table width="210">
<tr>
<td>1</td><td>2</td><td
id="select_table"><i>3</i></td><td>4</td><td>5</td>
</tr>
</table>
</div>
</div>
</body>
</html>
```

Le script jQuery :

```
$(document).ready(function(){
$("ul").children().add("#select_table").addClass("arriereplan");
});
```

Par `$("ul").children()`, les éléments de la liste sont sélectionnés. Le code y ajoute la cellule du tableau avec `add("#select_table")`. Tous ces éléments sont mis en valeur par un arrière-plan de couleur (`addClass("arriereplan")`).

© ENI Editions - All rigths reserved - Jonifar lina

Une loupe pour agrandir les vignettes

Faisons apparaître au passage du curseur sur une vignette, une loupe pour suggérer au visiteur d'agrandir celle-ci.

Les différentes images sont disponibles dans l'espace de téléchargement réservé à cet ouvrage.

Le fichier de départ est :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
#galerie { width: 100%;
           overflow: hidden;}
#galerie a { position:relative;
             float:left;
             margin:5px;}
#galerie a span { background-image:url(loupe.png);
                  background-repeat:no-repeat;
                  width:30px;
                  height:30px;
                  display:none;
                  position:absolute;
                  left:15px;
                  top:15px;}
#galerie img { border: solid 1px black;
               padding:5px;}
</style>
</head>
<body>
<div id="galerie">
<a href="tornade.jpg">
<img src="tornade_min.jpg" alt="" />
</a>
<a href="tornade1.jpg">
<img src="tornade1_min.jpg" alt="" />
</a>
</div>
</body>
</html>
```

Le script jQuery fait apparaître une loupe, avec un effet d'affichage progressif, au survol de la vignette.

© ENI Editions - All rigths reserved - Jonifar lina

L'astuce consiste à ajouter par jQuery, une balise qui contient la loupe et de la faire apparaître au survol du curseur de la souris.

```
<script type="text/javascript">
$(document).ready(function(){
$("#galerie a").append("<span></span>");
$("#galerie a").hover(function(){
$(this).children("span").fadeIn(600);
},
function(){
$(this).children("span").fadeOut(200);
});
});
</script>
```

Soit,

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

Démarrage de jQuery.

```
$("#galerie a").append("<span></span>");
```

Une balise est ajoutée par la méthode `append()` à la balise <a> de la division galerie.

```
$("#galerie a").hover(function(){
$(this).children("span").fadeIn(600);
},
```

Au survol (`hover`) du lien <a>, le script sélectionne parmi les enfants de celui-ci ceux qui sont une balise (`children("span")`) et on y applique un effet d'apparition progressive (`fadeIn(600)`).

```
function(){
$(this).children("span").fadeOut(200);
});
```

Lorsque le curseur quitte le lien, cette balise disparaît.

```
});
```

Fin du script jQuery.

Au final :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
```


© ENI Editions - All rigths reserved - Jonifar lina

```
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#galerie a").append("<span></span>");
$("#galerie a").hover(function(){
$(this).children("span").fadeIn(600);
},
function(){
$(this).children("span").fadeOut(200);
});
});
</script>
<style type="text/css">
#galerie { width: 100%;
           overflow: hidden;}
#galerie a { position:relative;
             float:left;
             margin:5px;}
#galerie a span { background-image:url(loupe.png);
                  background-repeat:no-repeat;
                  width:30px;
                  height:30px;
                  display:none;
                  position:absolute;
                  left:15px;
                  top:15px;}
#galerie img { border: solid 1px black;
               padding:5px;}
</style>
</head>
<body>
<div id="galerie">
<a href="tornade.jpg">
<img src="tornade_min.jpg" alt="" />
</a>
<a href="tornade1.jpg">
<img src="tornade1_min.jpg" alt="" />
</a>
</div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Introduction

L'intégration du DOM a profondément modifié l'écriture du JavaScript par son traçage des éléments. Mais sa véritable révolution est, sans conteste, la possibilité de modifier et d'ajouter à la volée des éléments dans la page Html.

© ENI Editions - All rigths reserved - Jonifar lina

Modifier le contenu

text()

Récupère le contenu texte de l'élément concerné.

Cette méthode fonctionne pour les documents Html, Xhtml et XML.

`$("div").text()` : récupère au format texte le contenu de la balise `<div>`.

Cette méthode renvoie une chaîne de caractères (String).

text(valeur)

Assigne un nouveau contenu texte (voir l'argument valeur) aux éléments concernés.

`$("div").text("les nouveaux éléments de texte")` : insère au format texte, le contenu `"les nouveaux éléments de texte"` dans la balise `<div>`.

Cette méthode renvoie un objet jQuery.

html()

Récupère au format Html, le contenu de l'élément concerné.

Cette méthode ne fonctionne pas pour les documents XML (à l'exception des documents Xhtml).

`$("div").html()` : récupère au format Html, le contenu de la balise `<div>`.

Cette méthode renvoie une chaîne de caractères (String).

Les balises sont retournées en majuscules sous Internet Explorer 8 et en minuscules sous Firefox 3 et Safari 4.

---

html(valeur)

Assigne un nouveau contenu Html (voir l'argument valeur) aux éléments concernés.

Cette propriété n'est pas disponible pour les documents XML mais elle l'est bien pour les documents Xhtml.

`$("div").html("<b>nouveau contenu</b>")` : insère comme du Html, les éléments fournis en argument dans la balise `<div>`.

Cette méthode renvoie un objet jQuery.

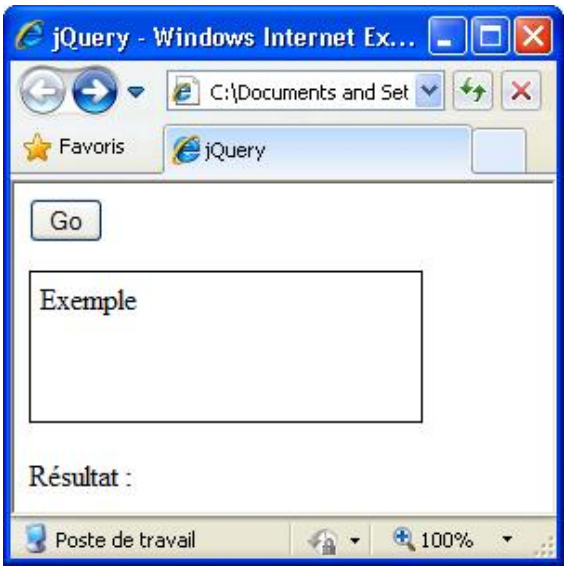
Avec les méthodes `text(valeur)` et `html(valeur)`, le nouveau contenu écrase le contenu précédent.

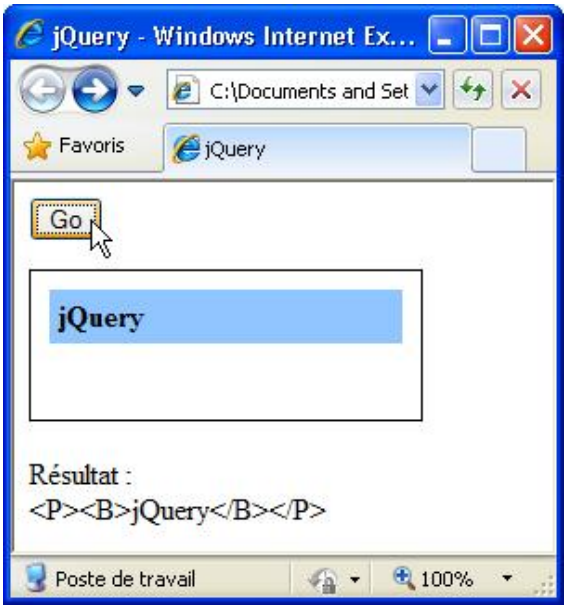
Les méthodes `html()` de JQuery utilisent la propriété JavaScript innerHTML. Pour rappel, cette propriété, à l'origine propriétaire Internet Explorer, a depuis été adoptée par les autres navigateurs mais son interprétation pose parfois des problèmes de compatibilité.

Exemple

Soit un élément boîte. Au clic sur le bouton, un nouveau contenu est injecté dans celui-ci. Cette opération réalisée, le nouveau contenu est affiché par la méthode `text()`.

© ENI Editions - All rigths reserved - Jonifar lina





Remarquez que le contenu précédent est détruit.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
#box { border: 1px solid black;
       width: 200px;
       height: 70px;
       margin-top: 15px;
       padding: 5px;}
p { background-color: #9cf;
    margin: 5px;
    padding: 5px;}
</style>
</head>
<body>
<input id="bouton" type="button" value="Go" />
<div id="box">Exemple</div><br />
Résultat :
<div id="resultat"></div>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
</body>
</html>
```

Le script JQuery :

```
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#bouton").click(function(){
$("#box").html("<p><b>jQuery</b></p>")
});
});
//]]>
</script>
```

Les explications :

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

Chargement du DOM.

```
$("#bouton").click(function(){
$("#box").html("<p><b>jQuery</b></p>")
```

Au clic sur le bouton, `<p><b>jQuery</b></p>` est inséré dans la boîte `box` comme du Html.

```
var contenu = $("#box").text();
$("#resultat").text(contenu);
```

Le nouveau contenu de la boîte `box` est récupéré par la méthode `text()` dans la variable `contenu`. Celui-ci est alors affiché dans la boîte `resultat`.

```
});
});
```

Fin de script.

Le fichier final devient :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#bouton").click(function(){
$("#box").html("<p><b>jQuery</b></p>");
var contenu = $("#box").text();
$("#resultat").text(contenu);
});
});
//]]>
</script>
<style type="text/css">
#box { border: 1px solid black;
       width: 200px;
       height: 70px;
       margin-top: 15px;
       padding: 5px;}
p { background-color: #9cf;
     margin: 5px;
     padding: 5px;}
```

© ENI Editions - All rigths reserved - Jonifar lina

```
</style>
</head>
<body>
<input id="bouton" type="button" value="Go" />
<div id="box">Exemple</div><br />
Résultat :
<div id="resultat"></div>
</body>
</html>
```


© ENI Editions - All rigths reserved - Jonifar lina

Insérer à l'intérieur

1. Première méthode

append(contenu)

Ajoute du contenu à la fin mais à l'intérieur de l'élément spécifié.

Le contenu peut être une chaîne de caractères, du Html ou un objet jQuery.

`$("p").append("<b>Hello</b>")` : insère à la fin du paragraphe, les éléments fournis en arguments.

Cette méthode renvoie un objet jQuery.

Les experts du JavaScript et de l'écriture du DOM auront reconnu la méthode `appendChild()`.

prepend(contenu)

Ajoute du contenu au début mais à l'intérieur de l'élément spécifié.

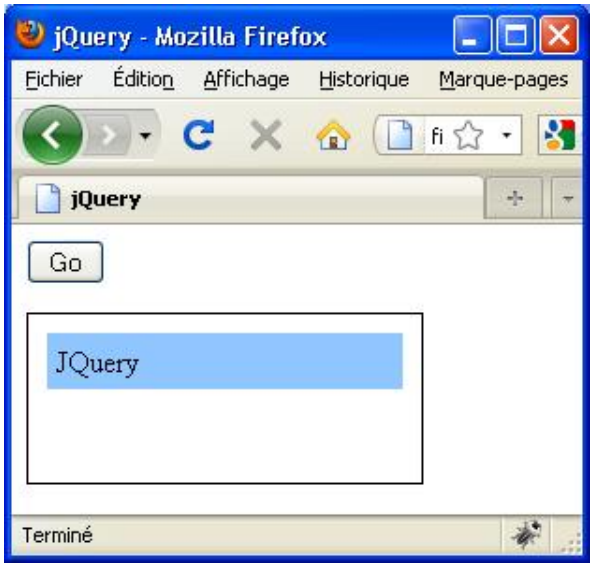
Le contenu peut être une chaîne de caractères, du Html ou un objet jQuery.

`$("p").prepend("<b>Hello</b>")` : insère au début du paragraphe, les éléments fournis en arguments.

Cette méthode renvoie un objet jQuery.

Exemple

Ajoutons du contenu au début et à la fin dans l'élément boîte du point précédent.

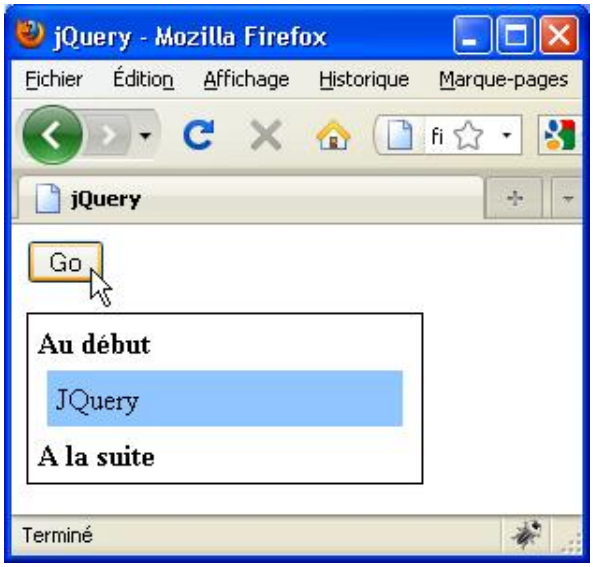



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
#box { border: 1px solid black;
       width: 200px;
       height: 80px;
       margin-top: 15px;
       padding: 5px;}
 p { background-color: #9cf;
     margin: 5px;
     padding: 5px;}
```

© ENI Editions - All rigths reserved - Jonifar lina

```
</style>
</head>
<body>
<input id="bouton" type="button" value="Go" />
<div id="box">
<p>JQuery</p>
</div>
</body>
</html>
```

Passons au script jQuery :




```
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#bouton").click(function(){
$("#box").prepend("<div><b>Au début</b></div>");
$("#box").append("<div><b>A la suite</b></div>");
});
});
//]]>
</script>
```

Explications :

```
$(document).ready(function(){
$("#bouton").click(function(){
```

Après le chargement du DOM et au clic sur le bouton.

```
$("#box").prepend("<div><b>Au début</b></div>");
```

Le script ajoute avec la méthode `prepend()`, le contenu repris en argument au début de l'élément boîte `box`.

```
$("#box").append("<div><b>A la suite</b></div>");
```

Le script ajoute avec la méthode `append()` du contenu repris en argument, à la fin de l'élément boîte `box`.

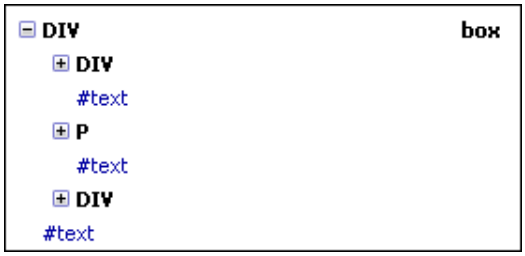
```
});
});
```

Fin de script.

Il faut constater que selon les captures d'écran, les éléments insérés le sont bien à l'intérieur de l'élément boîte.

Ceci est confirmé par exemple en utilisant l'extension DOM Inspector de Firefox qui permet de visualiser l'arbre DOM des éléments.

© ENI Editions - All rigths reserved - Jonifar lina



Le fichier complet devient :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#bouton").click(function(){
$("#box").prepend("<div><b>Au début</b></div>");
$("#box").append("<div><b>A la suite</b></div>");
});
});
//]]>
</script>
<style type="text/css">
#box { border: 1px solid black;
       width: 200px;
       height: 80px;
       margin-top: 15px;
       padding: 5px;}
p { background-color: #9cf;
     margin: 5px;
     padding: 5px;}
</style>
</head>
<body>
<input id="bouton" type="button" value="Go" />
<div id="box">
<p>JQuery</p>
</div>
</body>
</html>
```

2. Seconde méthode

Les méthodes `appendTo()` et `prependTo()` effectuent les mêmes tâches que `append()` et `prepend()`. La seule différence provient du placement dans le code, du contenu et de la cible. Avec `append()` ou `prepend()`, le sélecteur précédant la méthode est le conteneur dans lequel est inséré le contenu. Avec `appendTo()` ou `prependTo()`, le contenu précède la méthode et est alors inséré dans la conteneur cible. L'exemple qui suit illustrera utilement tout ceci.

appendTo()

Ajoute des éléments spécifiés par le sélecteur `A` à la fin d'autres spécifiés par `B` selon l'expression `$(A).appendTo(B)`.

`$("p").appendTo("#box")` : ajoute le contenu des éléments `<p>`, à la division portant l'identifiant `"box"` et à la fin de celle-ci.

Cette méthode renvoie un objet jQuery.

prependTo()

Ajoute des éléments spécifiés par le sélecteur `A` au début d'autres spécifiés par `B` selon l'expression `$(A).appendTo`

© ENI Editions - All rigths reserved - Jonifar lina

(B).

`$("p").prependTo("#box")` : ajoute le contenu des éléments `<p>`, à la division portant l'identifiant `"box"` et au début de celle-ci.

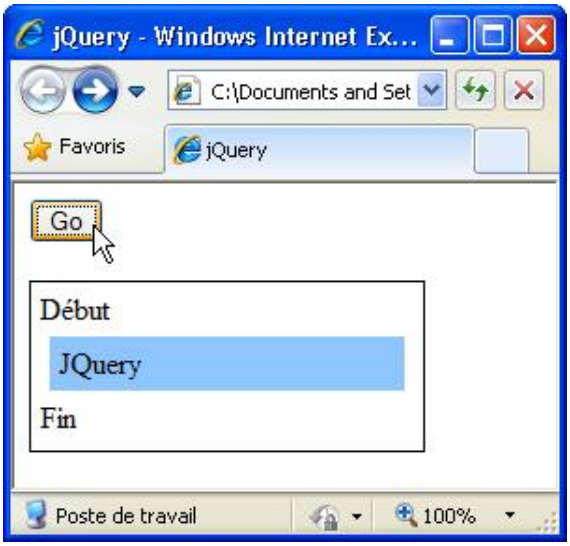
Cette méthode renvoie un objet jQuery.

Exemple

Reprenons les mêmes opérations de la section Insérer à l'intérieur - Première méthode.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
input { margin-bottom: 10px;}
#box { border: 1px solid black;
       width: 200px;
       height: 80px;
       margin-top: 10px;
       padding: 5px;}
p { background-color: #9cf;
    margin: 5px;
    padding: 5px;}
</style>
</head>
<body>
<input id="bouton" type="button" value="Go" />
<div id="debut">Début</div>
<div id="fin">Fin</div>
<div id="box">
<p>JQuery</p>
</div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina



Le script devient :

```
<script type="text/javascript">
$(document).ready(function(){
$("#bouton").click(function(){
$("#debut").prependTo("#box");
$("#fin").appendTo("#box");
});
});
</script>
```

Détaillons celui-ci.

```
$(document).ready(function(){
$("#bouton").click(function(){
```

Au chargement du DOM et au clic du bouton.

```
$("#debut").prependTo("#box");
```

Ajoute les éléments spécifiés par le sélecteur `#debut` au début des éléments spécifiés par le sélecteur `#box`. Soit exprimé autrement, ajoute la division `<div id="debut">Début</div>` au début des éléments de l'élément boîte `box`.

```
$("#fin").appendTo("#box");
```

Ajoute les éléments spécifiés par le sélecteur `#fin` à la fin des éléments spécifiés par `#box`. Soit ajoute la division `<div id="debut">Fin</div>` à la fin des éléments de l'élément boîte `box`.

```
});
});
```

Fin de script.

Le Html complet :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#bouton").click(function(){
$("#debut").prependTo("#box");
```

© ENI Editions - All rigths reserved - Jonifar lina

```
$("#fin").appendTo("#box");
});
});
</script>
<style type="text/css">
input { margin-bottom: 10px;}
#box { border: 1px solid black;
       width: 200px;
       height: 80px;
       margin-top: 10px;
       padding: 5px;}
p { background-color: #9cf;
    margin: 5px;
    padding: 5px;}
</style>
</head>
<body>
<input id="bouton" type="button" value="Go" />
<div id="debut">Début</div>
<div id="fin">Fin</div>
<div id="box">
<p>JQuery</p>
</div>
</body>
</html>
```


© ENI Editions - All rigths reserved - Jonifar lina

Insérer à l'extérieur

after(contenu)

Ajoute du contenu spécifié en argument après l'élément de la sélection.

Le contenu peut être une chaîne de caractères, du Html ou un objet jQuery.

`$("p").after("<b>Hello</b>")` : ajoute après la balise de paragraphe, le contenu fourni en argument.

Cette méthode renvoie un objet jQuery.

before(contenu)

Ajoute du contenu spécifié en argument avant chaque élément de la sélection.

Le contenu peut être une chaîne de caractères, du Html ou un objet jQuery.

`$("p").before("<b>Hello</b>")` : ajoute avant la balise de paragraphe, le contenu fourni en argument.

Cette méthode renvoie un objet jQuery.

Les experts du JavaScript classique et de la notation du DOM auront reconnu la méthode `insertBefore()`.

Exemple

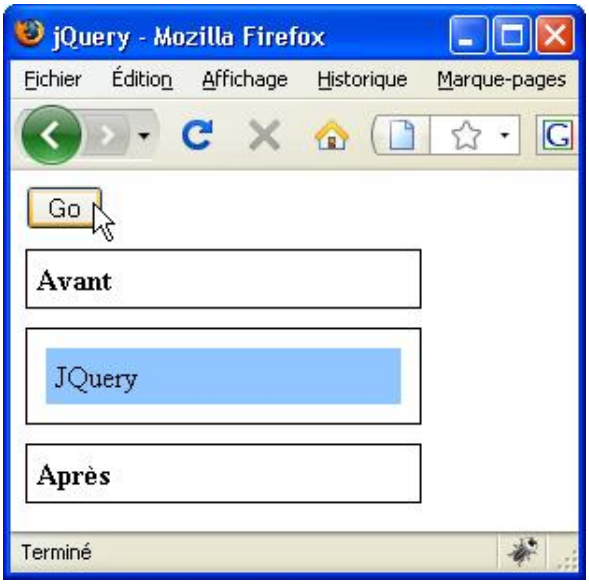
Ajoutons du contenu avant et après l'élément boîte du point précédent.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
input{margin-bottom: 10px;}
div {border: 1px solid black;width: 200px;padding: 5px;}
#box { border: 1px solid black;
       width: 200px;
       height: 40px;
       margin-top: 10px;
       margin-bottom: 10px;
       padding: 5px;}
p { background-color: #9cf;
     margin: 5px;
     padding: 5px;}
</style>
</head>
<body>
<input id="bouton" type="button" value="Go" />
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<div id="box">
<p>JQuery</p>
</div>
</body>
</html>
```

Le script jQuery :




```
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#bouton").click(function(){
$("#box").before("<div><b>Avant</b></div>");
$("#box").after("<div><b>Après</b></div>");
});
});
//]]>
</script>
```

```
$(document).ready(function(){
$("#bouton").click(function(){
```

Après le chargement du DOM et au clic sur le bouton.

```
$("#box").before("<div><b>Avant</b></div>");
```

Le script ajoute avec la méthode `before()`, une nouvelle division avant l'élément boîte `box`.

```
$("#box").after("<div><b>Après</b></div>");
```

Le script ajoute avec la méthode `after()`, une nouvelle division après l'élément boîte `box`.

```
});
});
```

Fin de script.

Il faut constater que selon les captures d'écran, les éléments insérés le sont bien à l'extérieur de l'élément boîte initial (`box`).

Le fichier complet devient :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#bouton").click(function(){
$("#box").before("<div><b>Avant</b></div>");
$("#box").after("<div><b>Après</b></div>");
});
});
//]]>
</script>
<style type="text/css">
input{margin-bottom: 10px;}
div {border: 1px solid black;width: 200px;padding: 5px;}
#box { border: 1px solid black;
       width: 200px;
       height: 40px;
       margin-top: 10px;
       margin-bottom: 10px;
       padding: 5px;}
p { background-color: #9cf;
     margin: 5px;
     padding: 5px;}
</style>
</head>
<body>
<input id="bouton" type="button" value="Go" />
<div id="box">
<p>JQuery</p>
</div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Entourer un élément

wrap(html ou élément)

Entoure chaque élément sélectionné avec l'élément fourni en argument. Cette procédure est très utile pour injecter une structure de code additionnelle dans un document sans modifier la sémantique originelle du document.

```
$("p").wrap("<div class='wrap'></div>");
```

Cette méthode renvoie un objet jQuery.

wrapAll(html ou élément)

Entoure tous les éléments de la sélection par un seul élément. Différent de la fonction `wrap()` qui entoure chaque élément de la sélection par un nouvel élément (voir exemples ci-après).

```
$("p").wrapAll("<div></div>");
```

Cette méthode renvoie un objet jQuery.

wrapInner(html ou élément)

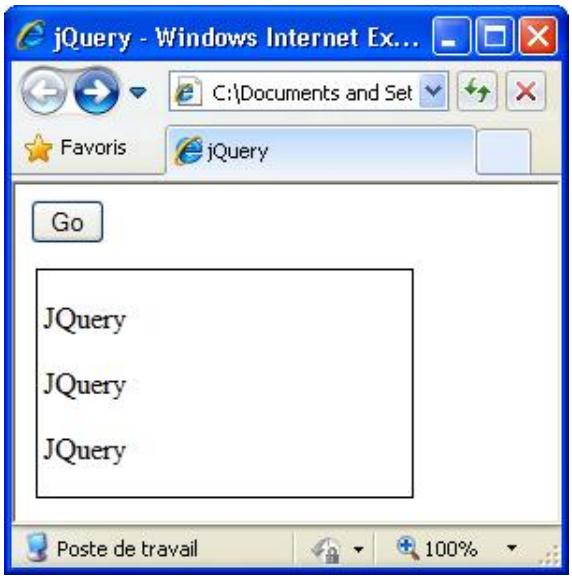
Entoure les enfants d'un élément (les nœuds textes inclus) par un autre élément.

```
$("p").wrapInner("<b></b>");
```

Cette méthode renvoie un objet jQuery.

Exemple

Appliquons ces méthodes à un exemple similaire aux précédents.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
input{margin-bottom: 10px;}
.wrap { border: 2px dashed black;
        margin: 3px;
        background-color: #9cf; }
#box { width: 200px;
```

© ENI Editions - All rigths reserved - Jonifar lina

```
        background-color: white;
        border: 1px solid black;}
p { padding-left: 3px;}
</style>
</head>
<body>
<input id="bouton" type="button" value="Go" />
<div id="box">
<p>JQuery</p>
<p>JQuery</p>
<p>JQuery</p>
</div>
</body>
</html>
```

Le script jQuery avec `wrap()` :

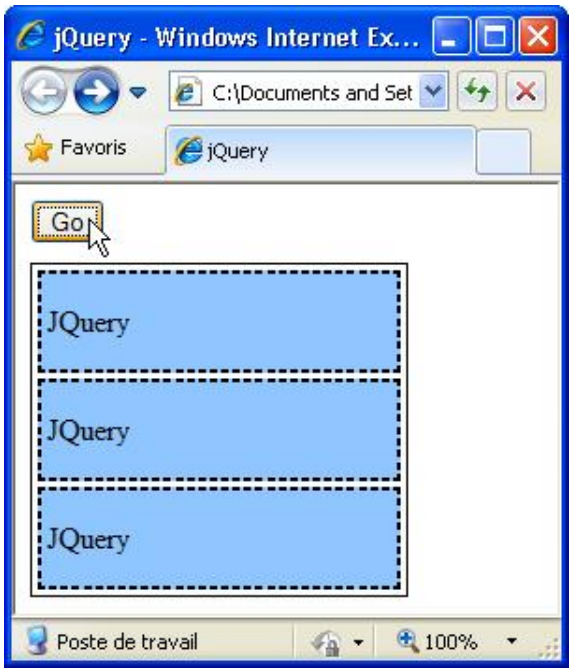
```
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#bouton").click(function(){
$("p").wrap("<div class='wrap'></div>");
});
});
//]]>
</script>
```

```
$(document).ready(function(){
$("#bouton").click(function(){
```

Au chargement du DOM et au clic sur le bouton.

```
$("p").wrap("<div></div>");
```

La méthode `wrap()` entoure chaque occurrence de paragraphe `<p>` avec la division qualifiée par la classe `wrap` soit avec un arrière-plan de couleur et une bordure en pointillés.



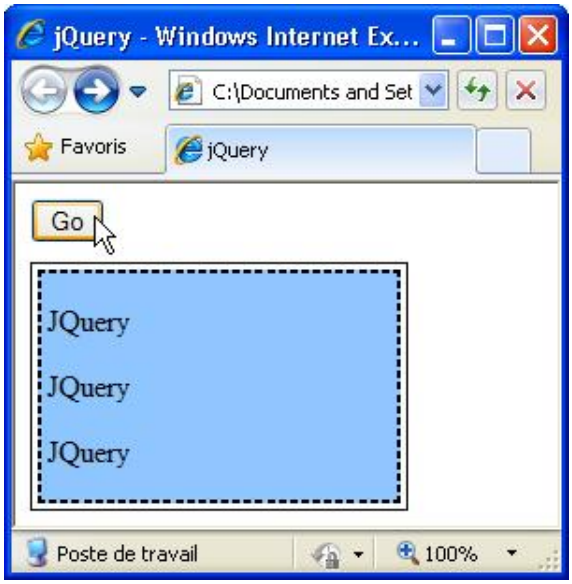
Par contre avec la méthode `wrapAll()`, le script devient :

```
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#bouton").click(function(){
```


© ENI Editions - All rigths reserved - Jonifar lina

```
$("p").wrapAll("<div class='wrap'></div>");
});
});
//]]>
</script>
```

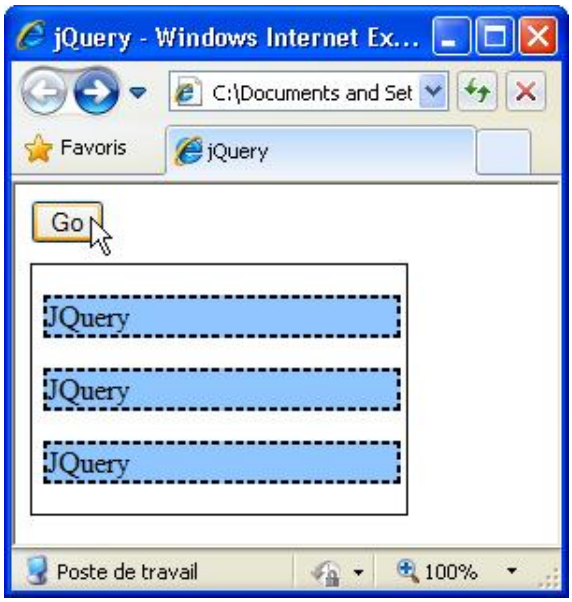
La méthode `wrapAll()` entoure toutes les occurrences de paragraphe <p> avec la division qualifiée par la classe `wrap` soit avec un arrière-plan de couleur et une bordure en pointillés.



Enfin avec la méthode `wrapInner()`, le script serait :

```
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#bouton").click(function(){
$("p").wrapInner("<div class='wrap'></div>");
});
});
//]]>
</script>
```

La méthode `wrapInner()` entoure tous les enfants des paragraphes <p> (soit ici le nœud texte) avec la division qualifiée par la classe `wrap` soit avec un arrière-plan de couleur et une bordure en pointillés.



Le fichier complet avec `wrap()` est :

© ENI Editions - All rigths reserved - Jonifar lina

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#bouton").click(function(){
$("p").wrap("<div class='wrap'></div>");
});
});
//]]>
</script>
<style type="text/css">
input{margin-bottom: 10px;}
.wrap { border: 2px dashed black;
        margin: 3px;
        background-color: #9cf; }
#box { width: 200px;
       background-color: white;
       border: 1px solid black;}
p { padding-left: 3px;}
</style>
</head>
<body>
<input id="bouton" type="button" value="Go" />
<div id="box">
<p>JQuery</p>
<p>JQuery</p>
<p>JQuery</p>
</div>
</body>
</html>
```


© ENI Editions - All rigths reserved - Jonifar lina

Remplacer un élément

replaceWith()

Remplace l'élément courant par le contenu spécifié, sous forme de code Html ou d'objet DOM. La fonction retourne l'élément JQuery qui vient d'être remplacé et supprimé du DOM.

```
$("#div1").replaceWith("<div>Nouvelle division</div>");
```

Cette méthode renvoie un objet jQuery.

Commentaires

La méthode `html()` remplace le contenu de l'élément tandis que `replaceWith()` remplace l'élément lui-même.

Exemple

Modifions un bouton de soumission de formulaire après un clic sur celui-ci. Les images de l'exemple sont disponibles dans l'espace de téléchargement réservé à cet ouvrage.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
button { cursor: pointer;}
img { float: left;
      width: 30px;
      height: 30px;}
span { float: left;
```

© ENI Editions - All rigths reserved - Jonifar lina

```
       margin-top: 7px;
       font: 1.2em arial,sans-serif}
p { margin-left: 35px;}
</style>
</head>
<body>
mail : <input type="text" size="30" /><br />
<p><button id="yes"><img src="yes.gif" alt="" />
<span>Soumettre</span> </button></p>
</body>
</html>
```

Le script jQuery :

```
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("button").click(function(){
$("img").replaceWith("<img src='no.gif' alt='' />");
$("span").css("text-decoration","line-through");
});
});
//]]>
</script>
```

Détails du script :

```
$(document).ready(function(){
$("button").click(function(){
```

Au chargement du DOM et au clic du bouton.

```
$("img").replaceWith("<img src='no.gif' alt='' />");
```

Remplace l'image initiale par celle fournie en argument de la méthode `replaceWith()`.

```
$("span").css("text-decoration","line-through");
```

Modifie le style de la balise `<span>` en barrant le texte.

```
});
});
```

Fin de script.

Le document complet se présente ainsi :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("button").click(function(){
$("img").replaceWith("<img src='no.gif' alt='' />");
$("span").css("text-decoration","line-through");
});
});
//]]>
</script>
<style type="text/css">
button { cursor: pointer;}
img { float: left;
```


© ENI Editions - All rigths reserved - Jonifar lina

```
      width: 30px;
      height: 30px;}
span { float: left;
       margin-top: 7px;
       font: 1.2em arial,sans-serif}
p { margin-left: 35px;}
</style>
</head>
<body>
mail : <input type="text" size="30" /><br />
<p><button id="yes"><img src="yes.gif" alt="" />
<span>Soumettre</span> </button></p>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Enlever un élément

1. Supprimer un élément

remove()

Supprime de l'arbre du DOM, tous les éléments répondant aux critères de sélection. Cependant cette méthode ne supprime pas les éléments de l'objet jQuery, ce qui permet une utilisation ultérieure de ces éléments même si ceux-ci ne figurent plus dans le document.

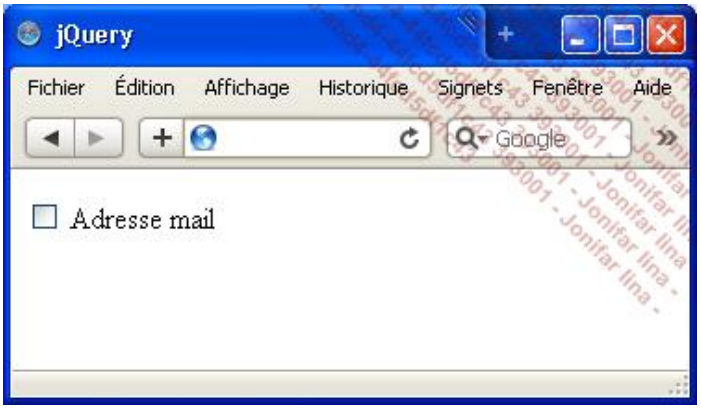
`$("p").remove()` : supprime le paragraphe.

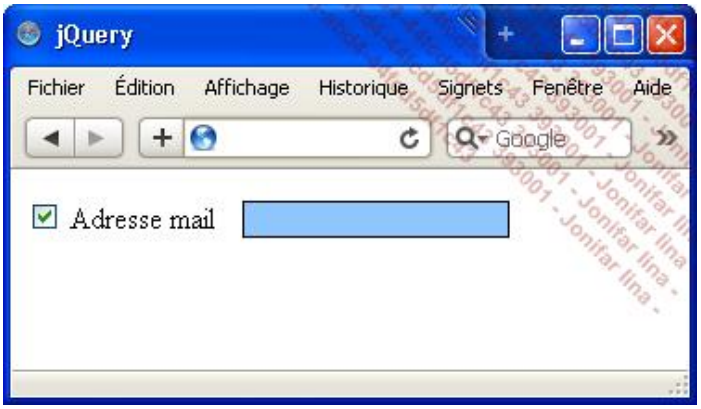
Cette méthode renvoie un objet jQuery.

Exemple

Prenons un bouton de type case à cocher (checkbox). Lorsque celui-ci est sélectionné, un formulaire de ligne de texte est affiché. Lorsque celui-ci est désélectionné, la ligne de texte disparaît.

Affichons l'exemple dans Opera.






```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
#plus { border: 1px solid black;
        background-color: #9cf;
        margin-left: 15px;}
</style>
</head>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<body>
<p id="mail"><input id="clic" type="checkbox" /> Adresse mail</p>
</body>
</html>
```

Le script jQuery.

```
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#clic").click(function(){
if ($(this).is(":checked")){
$("#mail").append("<input id=’plus’ type=’text’ />");
}
else {
$("#plus").remove();
}
});
});
//]]>
</script>
```

Détaillons les éléments du script.

```
$(document).ready(function(){
$("#clic").click(function(){
```

Au chargement du DOM et à la sélection de la case à cocher.

```
if ($(this).is(":checked")){
```

Si la case est cochée.

```
$("#mail").append("<input id=’plus’ type=’text’ />");
}
```

La ligne de texte est ajoutée dans le paragraphe.

```
else {
```

Si la case est décochée.

```
$("#plus").remove();
```

La ligne de texte est enlevée.

```
});
});
```

Fin du script.

<u>Commentaire :</u>

Le même effet peut être obtenu avec les méthodes `show()` et `hide()` (voir le chapitre Les effets - Afficher et cacher)

```
<script type="text/javascript">
$(document).ready(function(){
$("#plus").css("display","none");
$("#clic").click(function(){
if ($(this).is(":checked")){
$("#plus").show("fast");
}
else {
$("#plus").hide("fast");
}
});
});
</script>
```


© ENI Editions - All rigths reserved - Jonifar lina

Le fichier complet :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#clic").click(function(){
if ($(this).is(":checked")){
$("#mail").append("<input id=’plus’ type=’text’ />");
}
else {
$("#plus").remove();
}
});
});
//]]>
</script>
<style type="text/css">
#plus { border: 1px solid black;
        background-color: #9cf;
        margin-left: 15px;}
</style>
</head>
<body>
<p id="mail"><input id="clic" type="checkbox" /> Adresse mail</p>
</body>
</html>
```

2. Vider un élément

empty()

Supprime tous les nœuds enfants de l’élément sélectionné.

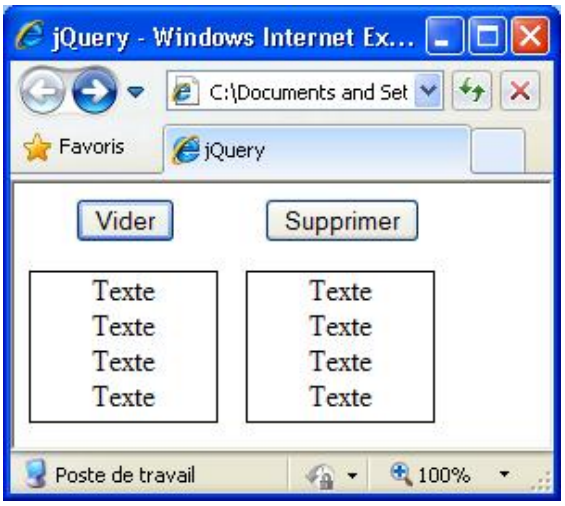
`$("p").empty()` : supprime le contenu du paragraphe.

Cette méthode renvoie un objet jQuery.

Exemple

Illustrons par un exemple, la différence entre les méthodes `empty()` et `remove()`.

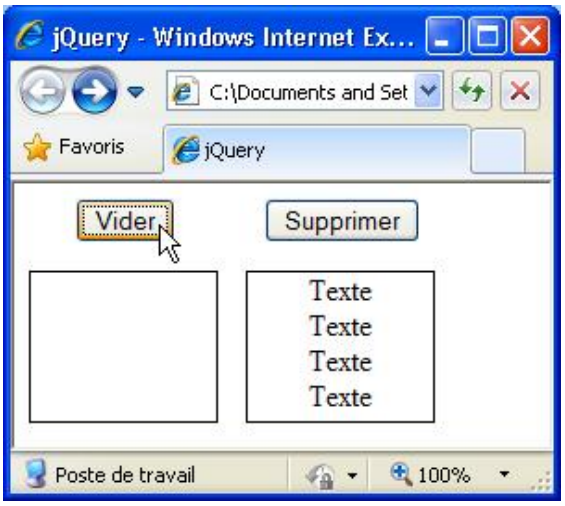
© ENI Editions - All rigths reserved - Jonifar lina

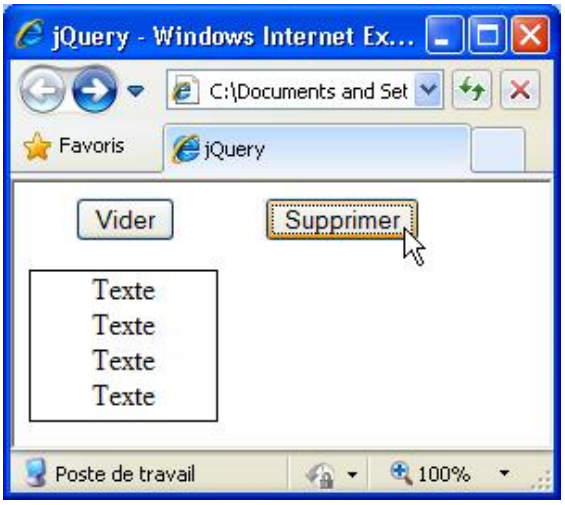



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
#bouton1 { margin-left: 25px;
           display: block;
           float: left;}
#bouton2 { margin-left: 206px;
           display: block;}
#div1 { border: 1px solid black;
        margin-top: 15px;
        width: 100px;
        height: 80px;
        text-align: center;
        float: left;}
#div2 { border: 1px solid black;
        margin-top: 15px;
        margin-left: 15px;
        width: 100px;
        height: 80px;
        text-align: center;
        float: left;}
</style>
</head>
<body>
<div><button id="bouton1">Vider</button> <button
id="bouton2">Supprimer</button></div>
<div id="div1">
Texte<br />Texte<br />Texte<br />Texte
</div>
<div id="div2">
Texte<br />Texte<br />Texte<br />Texte
</div>
</body>
</html>
```

Passons au script jQuery.

© ENI Editions - All rigths reserved - Jonifar lina





Au clic sur le bouton **Vider**, par la méthode `empty()`, le contenu (soit les nœuds enfants) est supprimé mais la division, illustrée par la bordure, est toujours en place.

Au clic sur le bouton **Supprimer**, par la méthode `remove()`, c'est toute la division qui disparaît.

Le script jQuery devient :

```
<script type="text/javascript">
$(document).ready(function(){
$("#bouton1").click(function () {
$("#div1").empty();
});
$("#bouton2").click(function () {
$("#div2").remove();
});
});
</script>
```

Détaillons celui-ci.

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

Chargement du DOM.

```
$("#bouton1").click(function () {
$("#div1").empty();
});
```

Au clic du bouton 1, le contenu de la division 1 est vidé.

```
$("#bouton2").click(function () {
```

© ENI Editions - All rigths reserved - Jonifar lina

```
$("#div2").remove();
});
```

Au clic du bouton 2, la division 2 est supprimée.

```
});
```

Fin du ready.

Le fichier Html complet devient :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#bouton1").click(function () {
$("#div1").empty();
});
$("#bouton2").click(function () {
$("#div2").remove();
});
});
</script>
<style type="text/css">
#bouton1 { margin-left: 25px;
           display: block;
           float: left;}
#bouton2 { margin-left: 206px;
           display: block;}
#div1 { border: 1px solid black;
        margin-top: 15px;
        width: 100px;
        height: 80px;
        text-align: center;
        float: left;}
#div2 { border: 1px solid black;
        width: 100px;
        height: 80px;
        margin-top: 15px;
        margin-left: 15px;
        text-align: center;
        float: left;}
</style>
</head>
<body>
<div><button id="bouton1">Vider</button> <button
id="bouton2">Supprimer</button></div>
<div id="div1">
Texte<br />Texte<br />Texte<br />Texte
</div>
<div id="div2">
Texte<br />Texte<br />Texte<br />Texte
</div>
</body>
</html>
```


© ENI Editions - All rigths reserved - Jonifar lina

Copier un élément

clone()

Copie (clone) les éléments DOM trouvés et les sélectionne. Cette fonction est utile pour créer des copies d'éléments et les déplacer vers un autre endroit spécifié du DOM.

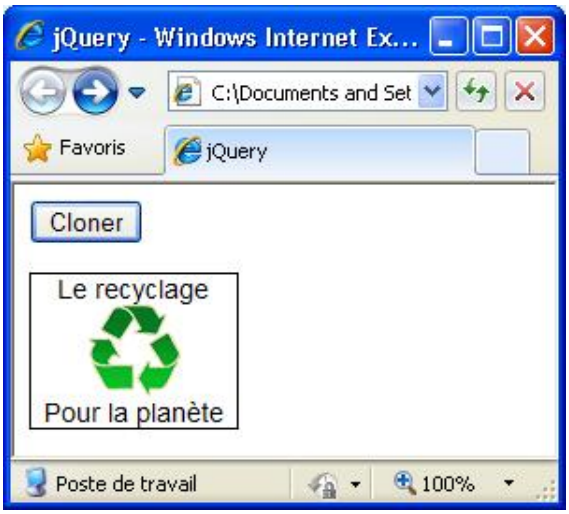
`$("p").clone()` : copie le paragraphe.

Paramètres (optionnel) : indiquez `true` si vous souhaitez cloner les gestionnaires d'événements associés à la sélection.

Cette méthode renvoie un objet jQuery.

Exemple

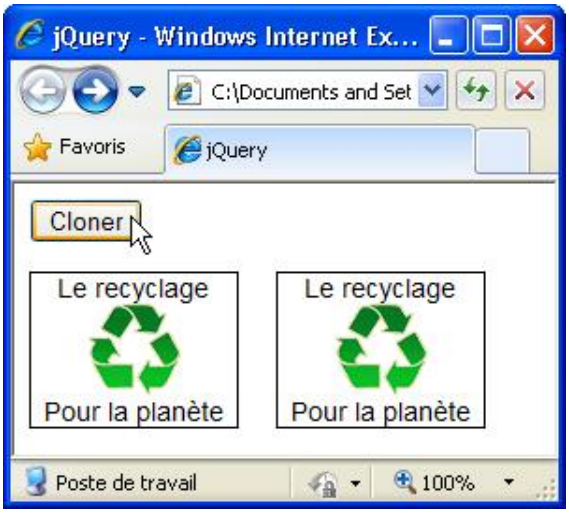
Clonons une division et son contenu (`<div id="div1">` pour les déplacer à un autre endroit de la page (`<div id="div2">`).




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
button {margin-bottom : 15px;}
#div1 { border: 1px solid black;
        width: 110px;
        font: 0.9em arial, sans-serif;
        float: left;
        text-align: center;}
#div2 { margin-left: 20px;
        float: left;}
</style>
</head>
<body>
<button id="bouton">Cloner</button><br />
<div id="div1">
Le recyclage<br />
<img src="recyclage.gif" alt="" /><br />
Pour la planète
</div>
<div id="div2"></div>
</body>
</html>
```

Le résultat :

© ENI Editions - All rigths reserved - Jonifar lina



Le code jQuery :

```
<script type="text/javascript">
$(document).ready(function(){
$("#div2").hide();
$("button").click(function () {
$("#div1").clone().prependTo("#div2");
$("#div2").show();
});
});
</script>
```

`$(document).ready(function(){$("#div2").hide();`

Au chargement du script, la division 2 est cachée.

`$("button").click(function () {`

Au clic du bouton.

`$("#div1").clone().prependTo("#div2");`

La division 1 est recopiée (`clone()`) et insérée (`prependTo()`) dans la seconde division.

`$("#div2").show();`

La division 2 est affichée.

```
});
});
```

Fin du script.

Au final :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#div2").hide();
$("button").click(function () {
```


© ENI Editions - All rigths reserved - Jonifar lina

```
$("#div1").clone().prependTo("#div2");
$("#div2").show();
});
});
</script>
<style type="text/css">
button { margin-bottom : 15px;}
#div1 { border: 1px solid black;
        width: 110px;
        font: 0.9em arial, sans-serif;
        float: left;
        text-align: center;}
#div2 { margin-left: 20px;
        float: left;}
</style>
</head>
<body>
<button id="bouton">Cloner</button><br />
<div id="div1">
Le recyclage<br />
<img src="recyclage.gif" alt="" /><br />
Pour la planète
</div>
<div id="div2"></div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Quelques applications

1. Ajouter un pied de page et des liens de retour

Nous allons ajouter avec quelques lignes de jQuery, un pied de page et des liens de retour vers le haut de la page.

Soit le fichier Html :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
a { color: black;}
#foot { border-top: 1px solid black;
        margin-top: 15px;}
</style>
</head>
<body>
<h2>Titre de la page</h2>
<div id="contenu">
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Praesent sodales, nisi eget aliquet interdum, urna risus sodales
urna, at auctor nisl tellus rutrum nibh. Pellentesque lorem diam,
tincidunt eget tincidunt non, rutrum non felis. In hac habitasse
platea dictumst. Ut nibh leo, placerat a tristique consequat,
vestibulum at tortor. In nec tristique turpis.</p>
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Praesent sodales, nisi eget aliquet interdum, urna risus sodales
urna, at auctor nisl tellus rutrum nibh. Pellentesque lorem diam,
tincidunt eget tincidunt non, rutrum non felis. In hac habitasse
platea dictumst. Ut nibh leo, placerat a tristique consequat,
vestibulum at tortor. In nec tristique turpis.</p>
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Praesent sodales, nisi eget aliquet interdum, urna risus sodales
urna, at auctor nisl tellus rutrum nibh. Pellentesque lorem diam,
tincidunt eget tincidunt non, rutrum non felis. In hac habitasse
platea dictumst. Ut nibh leo, placerat a tristique consequat,
vestibulum at tortor. In nec tristique turpis.</p>
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Praesent sodales, nisi eget aliquet interdum, urna risus sodales
urna, at auctor nisl tellus rutrum nibh. Pellentesque lorem diam,
tincidunt eget tincidunt non, rutrum non felis. In hac habitasse
platea dictumst. Ut nibh leo, placerat a tristique consequat,
vestibulum at tortor. In nec tristique turpis.</p>
</div>
</body>
</html>
```

À ce stade, les pieds de pages et les liens de retour n’apparaissent pas dans le fichier Xhtml.

© ENI Editions - All rigths reserved - Jonifar lina

Ajoutons le code jQuery.

```
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$('<a id="top" name="top"></a>').prependTo('body');
$("p").after("<a href='#top'>Haut de page</a>");
$("#contenu").after("<div id='foot'><i>Notes de pied de
page</i></div>");
});
//]]>
</script>
```

Expliquons ce script.

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

Initialisation du DOM dans jQuery.

`$('<a id="top" name="top"></a>').prependTo('body');`

Plaçons d'abord l'ancre (`<a id="top" name="top"></a>`) au début de la page. La méthode `prepend()` appliquée à la balise `<body>` permet de placer l'ancre juste après celle-ci.

`$("p").after("<a href='#top'>Haut de page</a>");`

Cette ligne de code placera le lien de retour vers le haut de la page (`<a href='#top'>Haut de page</a>`) après chaque occurrence de la balise `<p> ... </p>`.

`$("#contenu").after("<div id='foot'><i>Notes de pied de page</i></div>");`

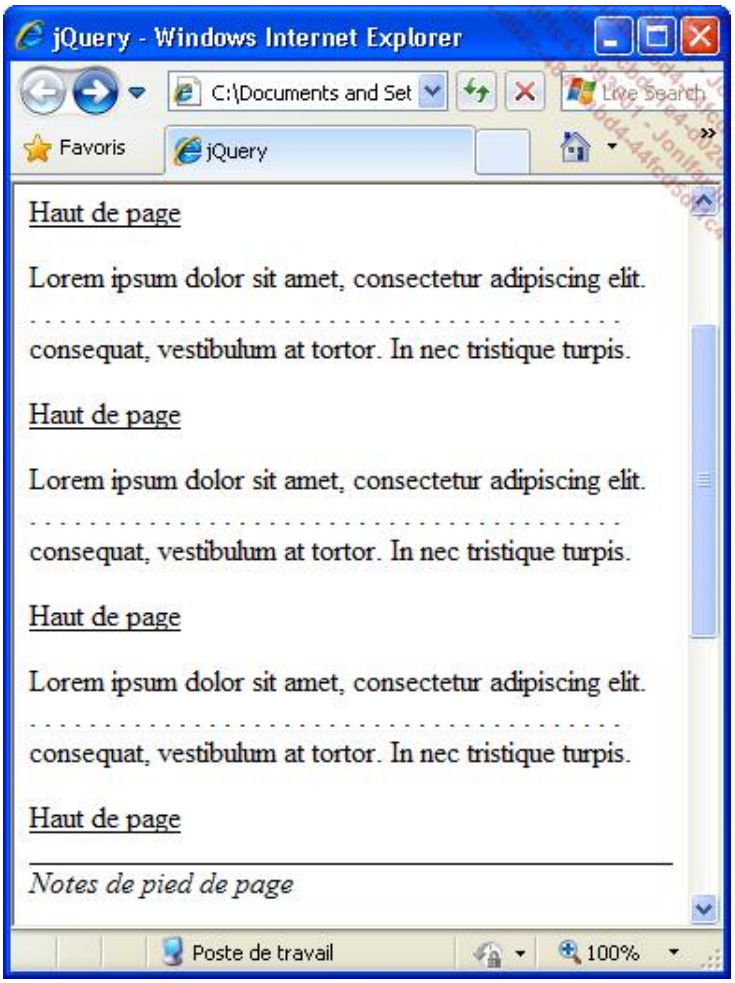
Pour le pied de page (`<div id='foot'><i>Notes de pied de page</i></div>`), il sera inséré après la division contenu qui contient les différents paragraphes.

© ENI Editions - All rigths reserved - Jonifar lina

```
});
```

Fin du script.

Cet exemple illustre bien, à notre avis, la concision du code jQuery même pour réaliser des opérations complexes.



Le fichier complet avec le script jQuery :

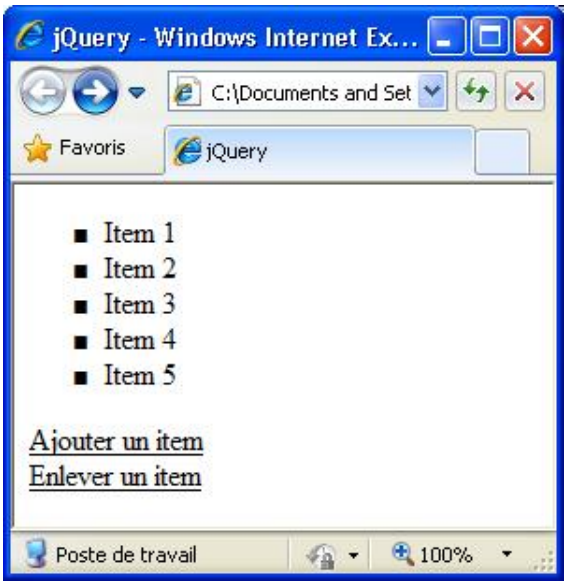
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$('<a id="top" name="top"></a>').prependTo('body');
$("p").after("<a href='#top'>Haut de page</a>");
$("#contenu").after("<div id='foot'><i>Notes de pied de
page</i></div>");
});
//]]>
</script>
<style type="text/css">
a { color: black;}
#foot { border-top: 1px solid black;
        margin-top: 15px;}
</style>
</head>
<body>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<h2>Titre de la page</h2>
<div id="contenu">
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Praesent sodales, nisi eget aliquet interdum, urna risus sodales
urna, at auctor nisl tellus rutrum nibh. Pellentesque lorem diam,
tincidunt eget tincidunt non, rutrum non felis. In hac habitasse
platea dictumst. Ut nibh leo, placerat a tristique consequat,
vestibulum at tortor. In nec tristique turpis.</p>
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Praesent sodales, nisi eget aliquet interdum, urna risus sodales
urna, at auctor nisl tellus rutrum nibh. Pellentesque lorem diam,
tincidunt eget tincidunt non, rutrum non felis. In hac habitasse
platea dictumst. Ut nibh leo, placerat a tristique consequat,
vestibulum at tortor. In nec tristique turpis.</p>
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Praesent sodales, nisi eget aliquet interdum, urna risus sodales
urna, at auctor nisl tellus rutrum nibh. Pellentesque lorem diam,
tincidunt eget tincidunt non, rutrum non felis. In hac habitasse
platea dictumst. Ut nibh leo, placerat a tristique consequat,
vestibulum at tortor. In nec tristique turpis.</p>
<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Praesent sodales, nisi eget aliquet interdum, urna risus sodales
urna, at auctor nisl tellus rutrum nibh. Pellentesque lorem diam,
tincidunt eget tincidunt non, rutrum non felis. In hac habitasse
platea dictumst. Ut nibh leo, placerat a tristique consequat,
vestibulum at tortor. In nec tristique turpis.</p>
</div>
</body>
</html>
```

2. Ajouter et enlever des éléments d'une liste

Soit une liste non ordonnée. Au clic sur un lien, ajoutons un élément de liste. Au clic sur un autre lien, supprimons un élément de liste.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
ul { list-style-type: square;}
```


© ENI Editions - All rigths reserved - Jonifar lina

```
a { color: black;}
</style>
</head>
<body>
<ul>
<li>Item 1</li>
<li>Item 2</li>
<li>Item 3</li>
<li>Item 4</li>
<li>Item 5</li>
</ul>
<a href="#" id="add">Ajouter un item</a><br />
<a href="#" id="remove">Enlever un item</a>
</body>
</html>
```

Le code jQuery :

```
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
var i = $('li').size() + 1;
$('a#add').click(function() {
$('<li>Item ' + i + '</li>').appendTo('ul');
i++;
});
$('a#remove').click(function() {
$('li:last').remove();
i--;
});
});
//]]>
</script>
```

Explications :

Le script s'articule autour des deux opérations suivantes.

```
$('a#add').click(function() {
$('<li>Item ' + i + '</li>').appendTo('ul');
```

Au clic sur le lien **Ajouter un item**, on ajoute comme dernier élément, un élément `<li> ... </li>` à la liste non ordonnée.

```
$('a#remove').click(function() {
$('li:last').remove();
```

Au clic sur le lien **Ajouter**, on supprime le dernier élément `<li> ... </li>` (`li:last`).

Il faut cependant ajouter un compteur pour incrémenter ou décrémenter le numéro associé à l'item.

```
var i = $('li').size() + 1;
$('a#add').click(function() {
$('<li>Item ' + i + '</li>').appendTo('ul');
i++;
});
```

On ajoute ainsi un compteur (`var i`). Celui-ci démarre au nombre d'éléments de liste (`size()`). On veillera à ajouter une unité car, comme souvent, JavaScript démarre ses comptages à 0. Une fois l'élément de liste ajouté, la variable `i` est incrémentée d'une unité (`i++`).

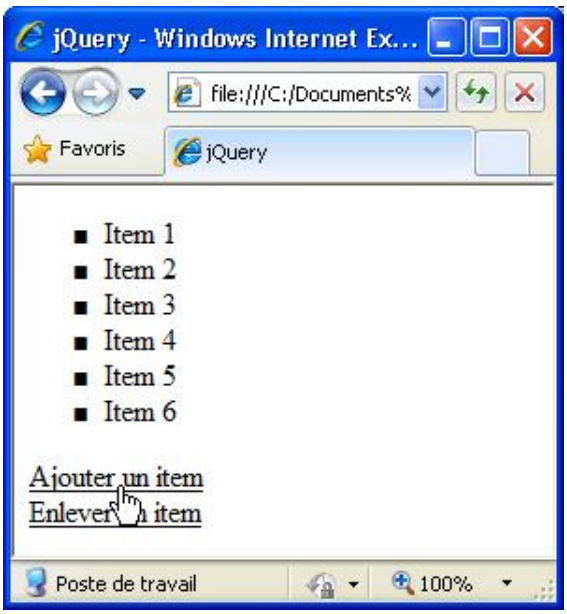
```
$('a#remove').click(function() {
$('li:last').remove();
i--;
});
```

De façon similaire, à chaque fois qu'un élément de liste est supprimé, le compteur `i` est décrémenté d'une unité.

© ENI Editions - All rigths reserved - Jonifar lina

```
});
```

Fin de script.



Le fichier final :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
//<![CDATA[
var i = $('li').size() + 1;
$('a#add').click(function() {
$('<li>Item ' + i + '</li>').appendTo('ul');
i++;
});
$('a#remove').click(function() {
$('li:last').remove();
i--;
});
});
//]]>
</script>
<style type="text/css">
ul {list-style-type: square;}
a {color: black;}
</style>
</head>
<body>
<ul>
<li>Item 1</li>
<li>Item 2</li>
<li>Item 3</li>
<li>Item 4</li>
<li>Item 5</li>
</ul>
<a href="#" id="add">Ajouter un item</a><br />
<a href="#" id="remove">Enlever un item</a>
</body>
</html>
```


© ENI Editions - All rigths reserved - Jonifar lina

3. Ajouter une icône aux liens externes

La petite icône qui suit les liens externes est très à la mode sur les sites Web 2.0. En ce qui nous concerne, jQuery permet d'ajouter facilement ces petites images.

L'icône est disponible dans l'espace de téléchargement.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
//<![CDATA[
$(document).ready(function() {
$("a").filter(function() {
return this.hostname !== location.hostname;
})
.after('<img src="external.png" class="external">');
});
//]]>
</script>
<style type="text/css">
a { color: black;
    text-decoration: none;}
.external { border: none;
            margin-left: 10px;}
</style>
</head>
<body>
<a href="http://www.editions-eni.fr/" title="Lien externe"
>Editions Eni</a><br />
<a href="#">Lien interne</a><br />
<a href="#">Lien interne</a><br />
</body>
</html>
```

Pour identifier les liens externes, nous testons le nom de domaine de la page (`location.hostname`) par rapport au nom de domaine (`this.hostname`) des liens. Nous nous assurons que les deux ne correspondent pas (`this.hostname && location.hostname !== this.hostname`).

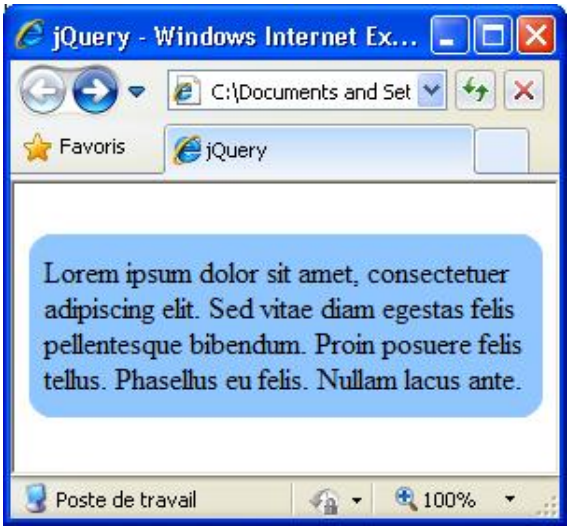
```
.after('<img src="external.png" class="external">');
```

Ce filtre appliqué, le script insère après le lien (`after`) l'icône de lien externe (`<img src="external.png" class="external">`).

© ENI Editions - All rigths reserved - Jonifar lina

4. Ajouter des bords arrondis

Mis à part les éléments ajoutés par les images, le design des pages Html est essentiellement linéaire avec des lignes verticales et horizontales. C'est le cas, par exemple, des tableaux, des divisions et jadis des cadres. La présence de lignes courbes ou de coins arrondis est une tendance qui apporte non seulement une certaine originalité mais aussi une touche de "douceur".



Avant de se pencher sur le code, nous avons besoin des images pour l'arrondi des bords. À cet effet, nous utilisons un générateur de bords arrondi soit par exemple, celui proposé par le site www.roundedcorner.com.

Nous obtenons alors les images rounded_tl.png (tl pour top left), rounded_tr.png (tr pour top right), rounded_bl.png (bl pour bottom left) et rounded_br (br pour bottom right).

Ces images sont disponibles dans l'espace de téléchargement consacré à cet ouvrage.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="fr-FR">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
//<![CDATA]
$(document).ready(function(){
$('div.rounded').wrap('<div class="cadre"></div>');
$('div.cadre').prepend('<div class="cadre_tl"></div>');
$('div.cadre').prepend('<div class="cadre_tr"></div>');
$('div.cadre').append('<div class="cadre_bl"></div>');
$('div.cadre').append('<div class="cadre_br"></div>');
});
//]]>
</script>
<style type="text/css">
#contenu { background-color: #9cf;
           padding: 0 8px;}
p { margin: 0; }
.cadre { background-color: #9cf;
         width: 275px;}
.cadre_tl, .cadre_tr,
.cadre_bl, .cadre_br{
width: 100%;
height: 11px;
font-size: 1px;
}
.cadre_tl{
background: url('rounded_tl.png') no-repeat top left;
}
```

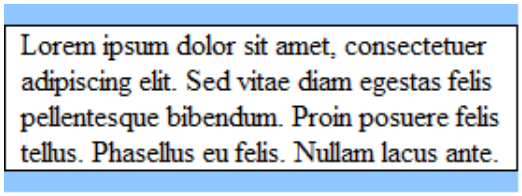

© ENI Editions - All rigths reserved - Jonifar lina

```
.cadre_tr{
background: url('rounded_tr.png') no-repeat top right;
float: right;
}
.cadre_bl{
background: url('rounded_bl.png') no-repeat bottom left;
float: right;
}
.cadre_br{
background: url('rounded_br.png') no-repeat bottom right;
}
</style>
</head>
<body>
<br />
<div id="contenu" class="rounded">
<p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed
vitae diam egestas felis pellentesque bibendum. Proin posuere
felis tellus. Phasellus eu felis. Nullam lacus ante.</p>
</div>
</body>
</html>
```

Penchons nous sur le code jQuery.

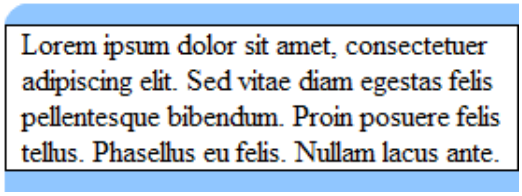
```
$("div.contenu").wrap("<div class="cadre"></div>");
```

Commençons par ajouter au-dessus et en-dessous (méthode `wrap()`) de la division dont la classe est `contenu`, la division dont la classe est `cadre`.



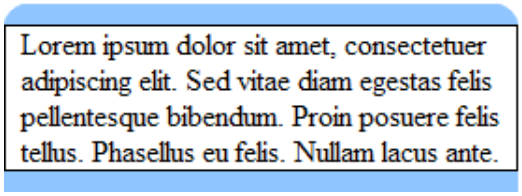

```
$("div.cadre").prepend("<div class='cadre_tl'></div>");
```

À l'intérieur de la division `cadre`, ajoutons en premier élément (méthode `prepend()`), l'image rounded_tl.png contenue en arrière-plan dans la division `cadre_tl`.




```
$("div.cadre").prepend("<div class='cadre_tr'></div>");
```

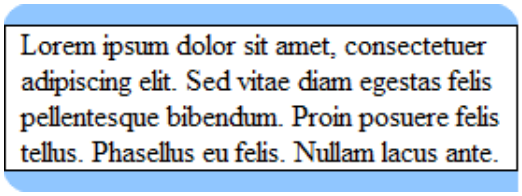
Procédons de même (méthode `prepend()`) pour l'image rounded_tr.png contenue en arrière-plan dans la division `cadre_tr`.




```
$("div.cadre").append("<div class='cadre_bl'></div>");
```

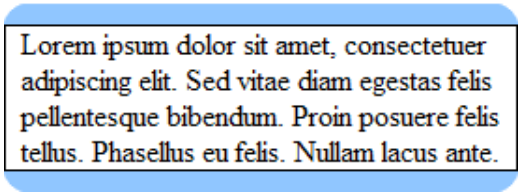
Ajoutons maintenant, toujours à l'intérieur de la division cadre mais en dernière position (méthode `append()`), l'image

rounded_bl.png contenue en arrière-plan dans la division `cadre_bl`.




```
$("div.cadre").append("<div class='cadre_br'></div>");
```

Enfin, procédons de même (méthode `append()`), pour l'image rounded_br.png contenue en arrière-plan dans la division `cadre_br`.



Il est possible d'écrire le code de façon plus concise. Ainsi, les lignes avec `prepend()` et `append()` deviendraient :

```
$("div.cadre").prepend("<div class='cadre_hd'></div><div
class='cadre_hg'></div>");
$("div.cadre").append("<div class='cadre_bd'></div><div
class='cadre_bg'></div>");
```


© ENI Editions - All rigths reserved - Jonifar lina

Introduction

Traverser le DOM et le manipuler sont des notions que l'on retrouvait déjà dans le JavaScript traditionnel. La librairie jQuery rend ces démarches plus aisées mais surtout moins fastidieuses. En introduisant la notion de filtrage des éléments, jQuery toujours dans son souci d'atteindre plus rapidement les éléments, apporte un concept innovant qui permet de réduire les résultats de la recherche selon les critères retenus par le développeur.

© ENI Editions - All rigths reserved - Jonifar lina

Trouver un élément déterminé

La méthode `eq()` (eq pour equal) permet de pointer la recherche directement sur un élément spécifique.

eq(index)

Réduit le résultat de la recherche à un élément dont la position est fournie en argument (index).

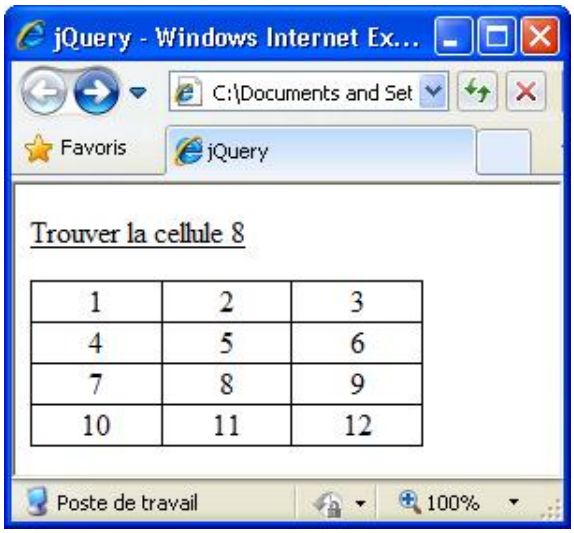
- index (entier) : détermine la position de l'élément. L'intervalle des positions commence à 0 et se termine à la taille de l'index - 1.

`$("p").eq(1)` : sélectionne le second paragraphe.

La méthode renvoie un objet jQuery.

Exemple

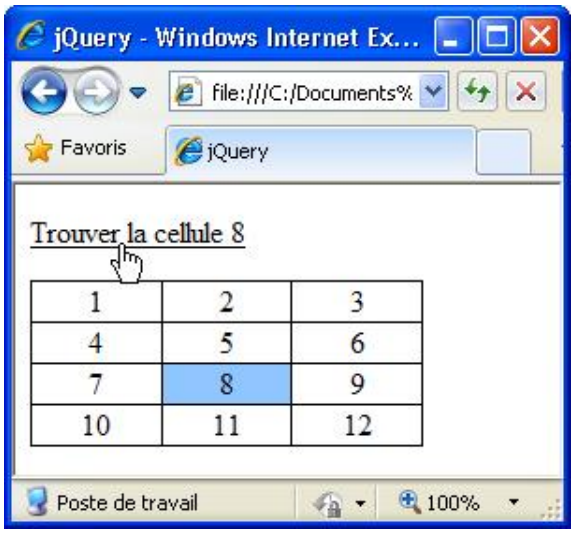
Soit un tableau de 4 lignes et trois colonnes. Au clic sur le lien, mettre un arrière-plan de couleur à la cellule 8.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
a { color: black;}
.bleu { background-color: #9cf;}
table { width: 210px;
        border-collapse: collapse;
        border: 1px solid black;}
td { text-align: center;
     border: 1px solid black;}
</style>
</head>
<body>
<p><a href="#">Trouver la cellule 8</a></p>
<table>
<tr><td>1</td><td>2</td><td>3</td></tr>
<tr><td>4</td><td>5</td><td>6</td></tr>
<tr><td>7</td><td>8</td><td>9</td></tr>
<tr><td>10</td><td>11</td><td>12</td></tr>
</table>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Le script jQuery.




```
<script type="text/javascript">
$(document).ready(function(){
$("a").click(function(){
$("td").eq(7).addClass("bleu");
});
});
</script>
```

Procédons aux explications.

```
$(document).ready(function(){
$("a").click(function(){
```

Au chargement du DOM et au clic du lien.

```
$("td").eq(7).addClass("bleu");
```

Les cellules du tableau sont chargées dans un objet jQuery (`$("td")`). Parmi celles-ci, la cellule dont la position d'index est égale à 7 est retenue (`eq(7)`). La classe bleu lui est alors appliquée (`addClass("bleu")`).

Les positions d'index commençant à 0 (comme c'est le cas avec le JavaScript classique), la position 7 correspond bien à la cellule 8.

```
});
});
```

Fin de script.

Le fichier final devient :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("a").click(function(){
$("td").eq(7).addClass("bleu");
});
});
</script>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<style type="text/css">
a { color: black;}
.bleu { background-color: #9cf;}
table { width: 210px;
        border-collapse: collapse;
        border: 1px solid black;}
td { text-align: center;
     border: 1px solid black;}
</style>
</head>
<body>
<p><a href="#">Trouver la cellule 8</a></p>
<table>
<tr><td>1</td><td>2</td><td>3</td></tr>
<tr><td>4</td><td>5</td><td>6</td></tr>
<tr><td>7</td><td>8</td><td>9</td></tr>
<tr><td>10</td><td>11</td><td>12</td></tr>
</table>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Trouver une séquence d'éléments

slice(position de départ [,position de fin])

Extrait une séquence parmi les éléments de la recherche.

- Position de départ (entier) : indique la position du premier élément de la séquence. Cet entier peut être négatif. Dans ce cas, la sélection débute à partir de la fin de la sélection initiale.

- Position de fin (entier) (facultatif) : indique la position (non comprise, strictement inférieur) du dernier élément de la séquence.

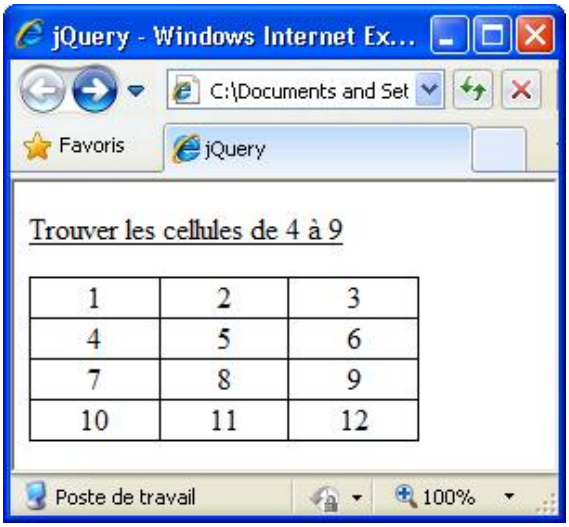
L'index des positions commence à 0.

```
$("div").slice(4, 6).css("background", "yellow");
```

La méthode renvoie un objet jQuery.

Exemple

Reprenons le tableau de 4 lignes et 3 colonnes. Remplissons d'un arrière-plan de couleur les cellules de 4 à 9 (soit la troisième et quatrième ligne).



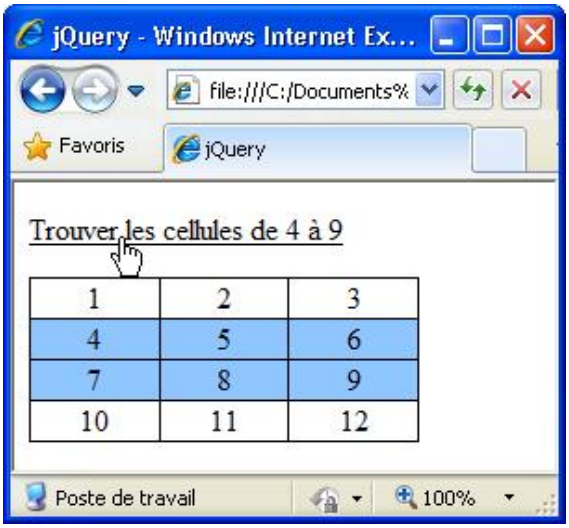

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
a { color: black;}
.bleu { background-color: #9cf;}
table { width: 210px;
        border-collapse: collapse;
        border: 1px solid black;}
td { text-align: center;
     border: 1px solid black;}
</style>
</head>
<body>
<p><a href="#">Trouver les cellules de 4 à 9</a></p>
<table>
<tr><td>1</td><td>2</td><td>3</td></tr>
<tr><td>4</td><td>5</td><td>6</td></tr>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<tr><td>7</td><td>8</td><td>9</td></tr>
<tr><td>10</td><td>11</td><td>12</td></tr>
</table>
</body>
</html>
```

Le script jQuery.

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("a").click(function(){
$("td").slice(3,9).addClass("bleu");
});
});
</script>
```



Explications pas à pas :

```
$(document).ready(function(){
$("a").click(function(){
```

Au chargement et au clic sur le lien.

```
$("td").slice(3,9).addClass("bleu");
```

Le script sélectionne les différentes cellules (`$("td")`). Il extrait ensuite la séquence allant de la position 3 (soit la cellule 4) jusqu'à la position 9 non comprise (soit jusqu'à la cellule 10 non comprise, donc la cellule 9).

```
});
});
```

Fin de script.

Le fichier final :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("a").click(function(){
$("td").slice(3,9).addClass("bleu");
```


© ENI Editions - All rigths reserved - Jonifar lina

```
});
});
</script>
<style type="text/css">
a { color: black;}
.bleu { background-color: #9cf;}
table { width: 210px;
        border-collapse: collapse;
        border: 1px solid black;}
td { text-align: center;
     border: 1px solid black;}
</style>
</head>
<body>
<p><a href="#">Trouver les cellules de 4 à 9</a></p>
<table>
<tr><td>1</td><td>2</td><td>3</td></tr>
<tr><td>4</td><td>5</td><td>6</td></tr>
<tr><td>7</td><td>8</td><td>9</td></tr>
<tr><td>10</td><td>11</td><td>12</td></tr>
</table>
</body>
</html>
```

Commentaires

```
$("td").slice(3).addClass("bleu");
```

Si la position de fin n'est pas indiquée, la séquence retenue commence à la position de départ jusqu'à la fin de la sélection.

1	2	3
4	5	6
7	8	9
10	11	12

```
$("td").slice(3,4).addClass("bleu");
```

Cette formulation ne reprend qu'un élément de la séquence.

1	2	3
4	5	6
7	8	9
10	11	12

```
$("td").slice(-2).addClass("bleu");
```

Une valeur négative indique que la sélection débute à la fin de la séquence. La valeur -2 signifie qu'elle reprend deux éléments en partant de la fin.

1	2	3
4	5	6
7	8	9
10	11	12

```
$("td").slice(2,-1).addClass("bleu");
```

Le code `slice(-2,1)` reprend une séquence allant du troisième à l'avant-dernier élément.

© ENI Editions - All rigths reserved - Jonifar lina

1	2	3
4	5	6
7	8	9
10	11	12

```
$("td").slice(0).addClass("bleu");
```

Tous les éléments sont ainsi repris dans la séquence.

1	2	3
4	5	6
7	8	9
10	11	12

© ENI Editions - All rigths reserved - Jonifar lina

Trouver un élément selon un critère

is(expression)

Indique si la sélection répond à un critère déterminé par l'expression. Renvoie `true` ou `false`.

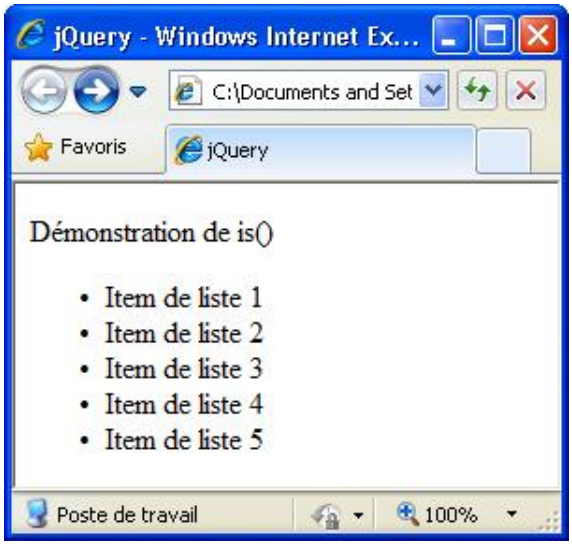
- expression (chaîne de caractères) : expression correspondant au critère à vérifier.

```
$(":checkbox").parent().is("form")
```

La méthode renvoie un booléen.

Exemple

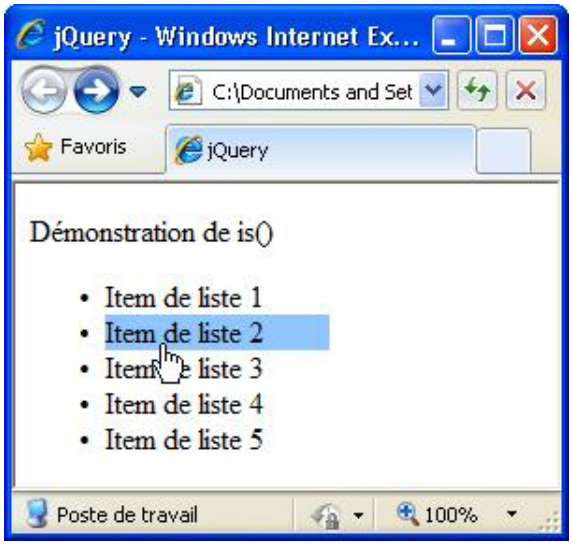
Soit une liste de 5 éléments. Au clic sur un élément de la liste, celui-ci sera orné d'un arrière-plan de couleur.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
.highlight { background-color: #9cf;}
a { color: black;}
li { width: 120px;
     cursor: pointer;}
</style>
</head>
<body>
<p>Démonstration de is()</p>
<ul id="exemple">
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li>Item de liste 3</li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
</body>
</html>
```

Le script jQuery.

© ENI Editions - All rigths reserved - Jonifar lina




```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("ul").click(function(event){
if ($(event.target).is("li")) {
$(event.target).addClass('highlight');
}
});
});
</script>
```

Détaillons celui-ci.

```
$(document).ready(function(){
$("ul").click(function(event){
```

Au chargement du DOM et au clic sur un élément de la liste non ordonnée.

```
if ($(event.target).is("li")) {
$(event.target).addClass('highlight');
}
```

Si le clic (`event.target`) porte sur un élément `<li>` de la liste, celui-ci sera mis en évidence par un arrière-plan de couleur (`addClass('highlight')`).

```
});
});
```

Fin de script.

Le fichier complet.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("ul").click(function(event){
if ($(event.target).is("li")) {
$(event.target).addClass('highlight');
}
});
```


© ENI Editions - All rigths reserved - Jonifar lina

```
});
</script>
<style type="text/css">
.highlight { background-color: #9cf;}
a { color: black;}
li { width: 120px;
     cursor: pointer;}
</style>
</head>
<body>
<p>Démonstration de is()</p>
<ul id="exemple">
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li>Item de liste 3</li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Supprimer un élément

La fonction `not()` est utilisée pour supprimer un élément d'un objet jQuery.

not(sélecteur ou expression)

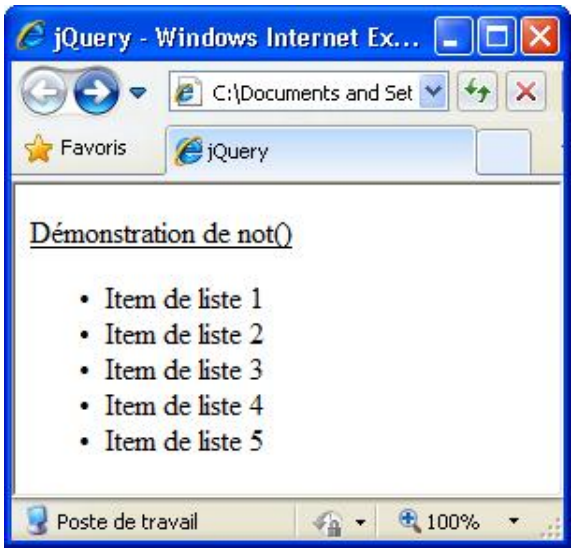
Supprime de la sélection, l'élément qui répond à l'expression spécifiée.

```
$("p").not("#selected")
```

Cette méthode renvoie un objet jQuery.

Exemple

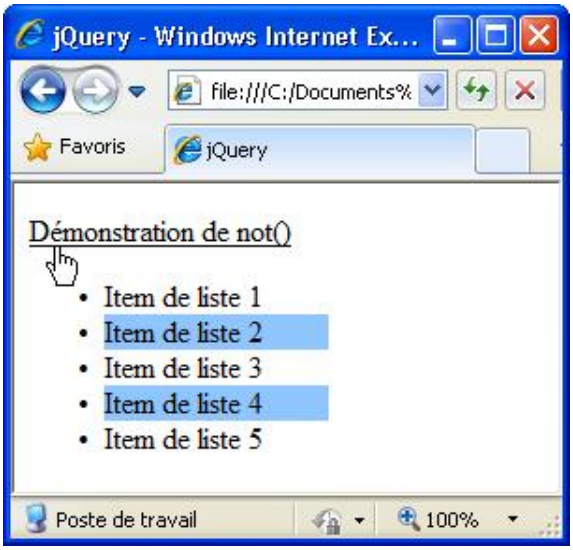
Soit une liste non ordonnée de 5 éléments.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
.highlight { background-color: #9cf;}
a { color: black;}
li { width: 120px;}
</style>
</head>
<body>
<p><a href="#">Démonstration de not()</a></p>
<ul id="exemple">
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li>Item de liste 3</li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
</body>
</html>
```

Le script jQuery va sélectionner les lignes paires en supprimant les lignes impaires.

© ENI Editions - All rigths reserved - Jonifar lina




```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("a").click(function(){
$('#exemple li').not(':even').addClass('highlight');
});
});
</script>
```

Détaillons ce script.

```
$(document).ready(function(){
$("a").click(function(){
```

Au chargement et au clic sur le lien.

```
$('#exemple li').not(':even').addClass('highlight');
```

Le script supprime de la sélection, les éléments dont l'index est un nombre pair (`not(':even')`) de la liste non ordonnée (`$('#exemple li')`). Les éléments restants sont dotés d'un arrière-plan de couleur, soit les éléments dont l'index est un nombre impair. Il est utile pour la bonne compréhension de se rappeler que l'index en JavaScript débute à 0.

```
});
});
```

Fin du script.

Le fichier final se présente comme suit :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("a").click(function(){
$('#exemple li').not(':even').addClass('highlight');
});
});
</script>
<style type="text/css">
.highlight { background-color: #9cf;}
a { color: black;}
```


© ENI Editions - All rigths reserved - Jonifar lina

```
li { width: 120px;}
</style>
</head>
<body>
<p><a href="#">Démonstration de not()</a></p>
<ul id="exemple">
<li>Item de liste 1</li>
<li>Item de liste 2</li>
<li>Item de liste 3</li>
<li>Item de liste 4</li>
<li>Item de liste 5</li>
</ul>
</body>
</html>
```

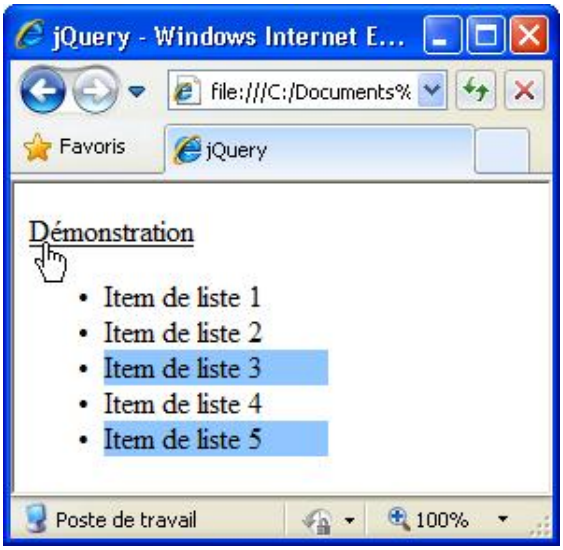
Commentaire

Il est possible de combiner les expressions. Par exemple :

```
$('#exemple li').not(':odd,:first').addClass('highlight');
```

Ainsi, avec le code précédent, le script supprime les éléments dont l'index est impair et le premier élément.

Le résultat est :



© ENI Editions - All rigths reserved - Jonifar lina

Former un tableau (Array) d'éléments

map(fonction de rappel)

Renvoie un tableau d'éléments (`Array`) résultant d'une action sur un ensemble d'éléments. Chaque ligne du tableau est le retour de la fonction appliquée à un élément.

fonction de rappel (callback) : fonction appliquée aux éléments ciblés.

```
map(function(){ return $(this).val();})
```

La méthode retourne un objet jQuery.

Exemple

Formons un tableau de type `Array` avec les valeurs des différentes lignes de texte d'un formulaire.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
p { font-weight: bold;
    border : 1px solid black;
    padding: 3px; }
</style>
</head>
<body>
<form action="">
Nom : <input type="text" name="nom" value="Igor" /><br />
Ville : <input type="text" name="ville" value="Lille" /><br />
Pays : <input type="text" name="pays" value="France" /><br />
Mail : <input type="text" name="mail" value="igor@gmail.com" />
</form>
<p><b>Les valeurs sont : </b></p>
</body>
</html>
```

Le script :

```
<script type="text/javascript">
```

© ENI Editions - All rigths reserved - Jonifar lina

```
$(document).ready(function(){
$("p").append($("input").map(function(){
return $(this).val();
})
.get().join(", ") );
});
</script>
```

Détaillons celui-ci :

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

Au chargement du DOM.

```
$("p").append($("input").map(function(){
return $(this).val();
})
.get().join(", ") );
```

Le script insère dans le paragraphe `<p>` (`$("p").append`) les valeurs retournées par la méthode `map()` soit la valeur (`val()`) des différentes balises `<input>` du formulaire.

La méthode jQuery `get()` permet d'accéder à tous les éléments du formulaires. Enfin, la méthode JavaScript classique `join()` convertit le tableau Array en une chaîne de caractères composée de tous les éléments séparés par une virgule.

```
});
```

Fin du script.

Au final :

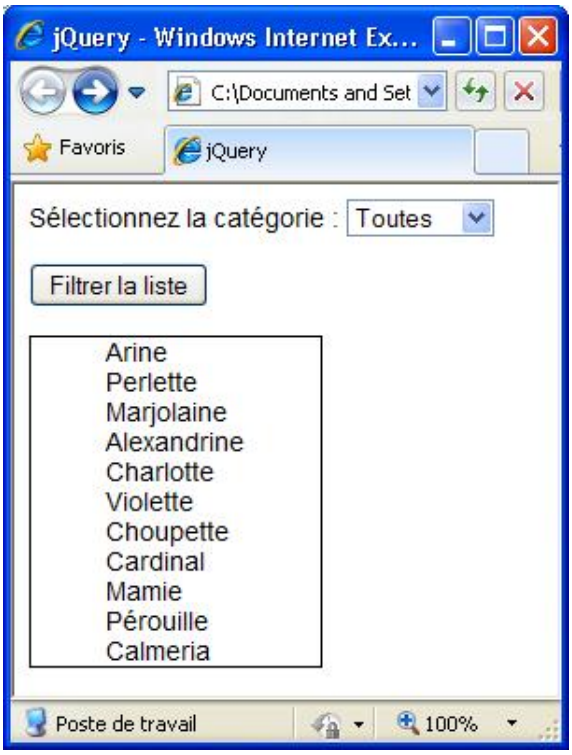
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("p").append( $("input").map(function(){
return $(this).val();
})
.get().join(", ") );
});
</script>
<style type="text/css">
p { font-weight: bold;
    border : 1px solid black;
    padding: 3px; }
</style>
</head>
<body>
<form action="">
Nom : <input type="text" name="nom" value="Igor" /><br />
Ville : <input type="text" name="ville" value="Lille" /><br />
Pays : <input type="text" name="pays" value="France" /><br />
Mail : <input type="text" name="mail" value="igor@gmail.com" />
</form>
<p><b>Les valeurs sont : </b></p>
</body>
</html>
```


© ENI Editions - All rigths reserved - Jonifar lina

Applications

1. Filtrer une liste

Soit une liste de variétés de fruits. Au choix d'un élément de formulaire de type `<select>`, le script ne retiendra que les variétés d'un fruit spécifique (pomme, poire, raisin ou fraise).



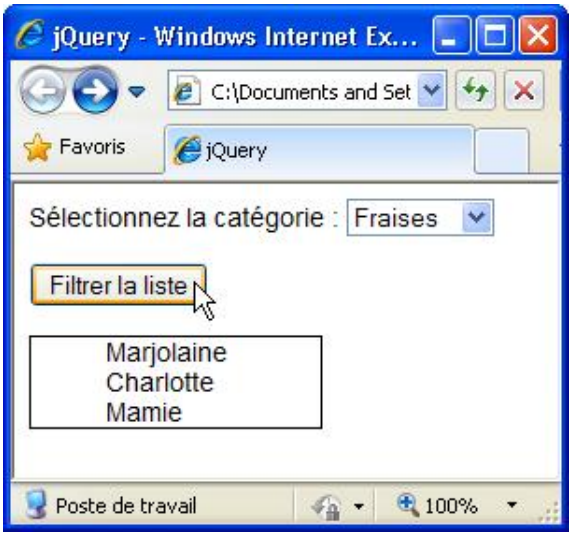
Le fichier de départ apparaît comme suit.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
body { font-family: Arial;
       font-size: 0.9em;
ul { list-style-type: none;
     width: 8em;
     border: 1px solid black;}
li { pagdding : 5px;}
</style>
</head>
<body>
Sélectionnez la catégorie : <select id="categorie">
<option selected="selected" value="toutes">Toutes</option>
<option value="pomme">Pommes</option>
<option value="poire">Poires</option>
<option value="raisin">Raisins</option>
<option value="fraise">Fraises</option>
</select>
<p><input id="bouton" type="button" value="Filtrer la liste" /></p>
<ul>
<li class="item pomme">Arine</li>
<li class="item raisin">Perlette</li>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<li class="item fraise">Marjolaine</li>
<li class="item poire">Alexandrine</li>
<li class="item fraise">Charlotte</li>
<li class="item pomme">Violette</li>
<li class="item pomme">Choupette</li>
<li class="item raisin">Cardinal</li>
<li class="item fraise">Mamie</li>
<li class="item poire">Pérouille</li>
<li class="item raisin">Calmeria</li>
</ul>
</body>
</html>
```

Le script jQuery.




```
<script type="text/javascript">
$(document).ready(function(){
$("#bouton").click(function(){
var selection = $("#categorie").val();
if (selection == "all"){
$("li").filter(".item").show();
}
else {
$("li").filter(".item").hide();
$("li").filter("."+selection).show();
}
});
});
</script>
```

Détaillons celui-ci.

```
$(document).ready(function(){
$("#bouton").click(function(){
```

Au chargement et au clic du bouton **Filtrer la liste**.

```
var selection = $("#categorie").val();
```

Le script charge dans la variable `selection`, la valeur retenue par l'utilisateur des choix proposés dans la liste de sélection.

```
if (selection == "all"){
$("li").filter(".item").show();
}
```

Si le choix `Toutes` a été retenu, tous les éléments de la liste non ordonnée sont affichés selon un filtre portant sur la classe `item`.

© ENI Editions - All rigths reserved - Jonifar lina

```
else {
$("li").filter(".item").hide();
$("li").filter("."+selection).show();
}
```

Si une autre valeur a été reprise dans la liste de sélection, la liste complète des items est cachée dans un premier temps. Ensuite, les éléments de la liste non ordonnée sont filtrés selon le critère stocké dans la variable `selection`. Ceux-ci sont alors affichés.

```
});
});
```

Fin de script.

Au final.

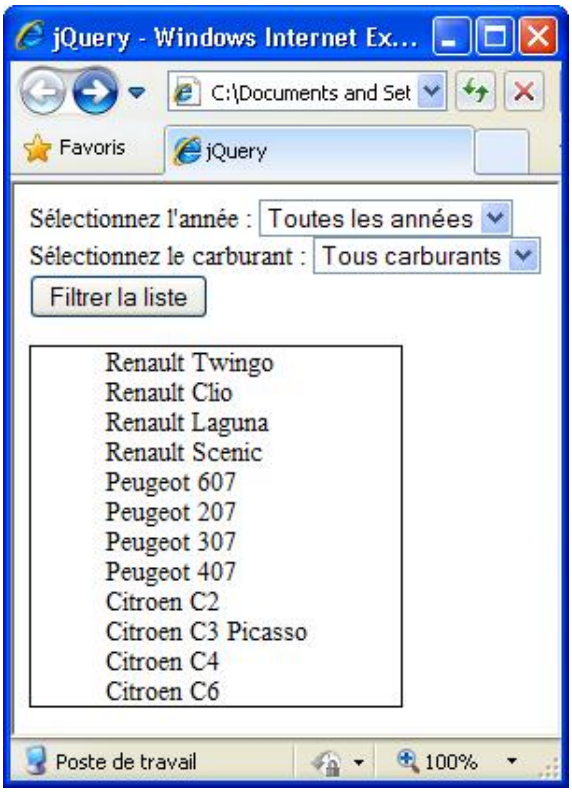
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#bouton").click(function(){
var selection = $("#categorie").val()
if (selection == "all"){
$("li").filter(".item").show();
}
else {
$("li").filter(".item").hide();
$("li").filter("."+selection).show();
}
});
});
</script>
<style type="text/css">
body { font-family: Arial;
       font-size: 0.9em;}
ul { list-style-type: none;
     width: 8em;
     border: 1px solid black;}
li { pagdding : 5px;}
</style>
</head>
<body>
Sélectionnez la catégorie : <select id="categorie">
<option selected="selected" value="toutes">Toutes</option>
<option value="pomme">Pommes</option>
<option value="poire">Poires</option>
<option value="raisin">Raisins</option>
<option value="fraise">Fraises</option>
</select>
<p><input id="bouton" type="button" value="Filtrer la liste"
/></p>
<ul>
<li class="item pomme">Arine</li>
<li class="item raisin">Perlette</li>
<li class="item fraise">Marjolaine</li>
<li class="item poire">Alexandrine</li>
<li class="item fraise">Charlotte</li>
<li class="item pomme">Violette</li>
<li class="item pomme">Choupette</li>
<li class="item raisin">Cardinal</li>
<li class="item fraise">Mamie</li>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<li class="item poire">Pérouille</li>
<li class="item raisin">Calmeria</li>
</ul>
</body>
</html>
```

2. Filtrer une liste selon deux critères

Application semblable à la précédente mais avec cette fois, deux critères de sélection. Soit une liste de voitures. Une sélection portera sur l'année (2007, 2008, 2009) et une autre sur le type de carburant (essence ou diesel).

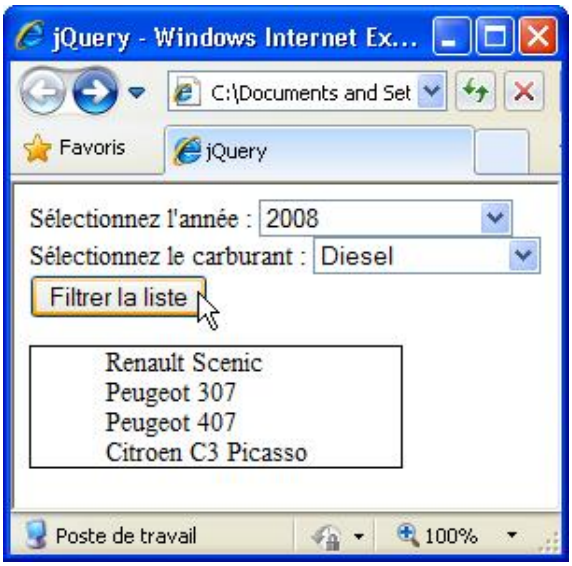



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
body { font-family: Arial sans-serif;
       font-size: 0.9em;}
ul { list-style-type: none;
     width: 11em;
     border: 1px solid black;}
li { pagdding : 5px;}
</style>
</head>
<body>
Sélectionnez l'année : <select id="annee">
<option selected="selected" value="toutes">Toutes les années</option>
<option value="2009">2009</option>
<option value="2008">2008</option>
<option value="2007">2007</option></select>
<br />
Sélectionnez le carburant : <select id="carburant">
<option selected="selected" value="tous" >Tous carburants</option>
<option value="essence">Essence</option>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<option value="diesel">Diesel</option>
</select>
<br />
<input id="bouton" type="button" value="Filtrer la liste" />
<ul>
<li class="item 2009 essence">Renault Twingo</li>
<li class="item 2008 essence">Renault Clio</li>
<li class="item 2007 diesel">Renault Laguna</li>
<li class="item 2008 diesel">Renault Scenic </li>
<li class="item 2007 diesel">Peugeot 607</li>
<li class="item 2009 essence">Peugeot 207</li>
<li class="item 2008 diesel">Peugeot 307</li>
<li class="item 2008 diesel">Peugeot 407</li>
<li class="item 2007 essence">Citroen C2</li>
<li class="item 2008 diesel">Citroen C3 Picasso</li>
<li class="item 2007 diesel">Citroen C4</li>
<li class="item 2009 diesel">Citroen C6 </li>
</ul>
</body>
</html>
```

Le script jQuery.




```
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#bouton").click(function(){
var annee = $("#annee").val();
var carburant = $("#carburant").val();
var selecteur_annee = '';
var selecteur_carburant = '';
if (annee == "toutes" && carburant == "tous"){
$(".item").show();
}
else {
if (carburant != "tous"){
selecteur_carburant = '.' + carburant
}
if (annee != "toutes")
{
selecteur_annee = '.' + annee
}
$(".item").hide();
$("li"). filter(selecteur_carburant + selecteur_annee).show();
}
});
});
//]]>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
</script>
```

Comme le script comporte le signe &, ce dernier risque d'être interprété comme un début de caractère spécial par le Xhtml. Il est ainsi prudent de faire appel à une section CDATA pour éviter que le navigateur n'analyse ces lignes de code comme du Xhtml.

Détaillons ce script.

```
$(document).ready(function(){
$("#bouton").click(function(){
```

Au chargement et au clic du bouton.

```
var annee = $("#annee").val();
var carburant = $("#carburant").val();
var selecteur_annee = '';
var selecteur_carburant = '';
```

Nous créons une série de variables. La variable `annee` reprend le choix de la première liste de sélection. La variable `carburant` celui de la seconde liste de sélection. Les variables `selecteur_annee` et `selecteur_carburant` contiendront (comme leur nom l'indique) les sélecteurs de classe du filtrage.

```
if (annee == "toutes" && carburant == "tous"){
$(".item").show();
}
```

Si la liste de sélection des années est positionnée sur `Toutes` et celle des carburants sur `Tous`, tous les éléments de la liste non ordonnée (`.item`) sont affichés.

```
else {
if (carburant != "tous"){
selecteur_carburant = '.' + carburant
}
```

Sinon, le script teste si la liste de sélection relative au carburant n'est plus positionnée sur `Tous`. Dans ce cas la variable `selecteur_carburant` contiendra un point (pour la classe) et le type de carburant retenu.

```
if (annee != "toutes"){
selecteur_annee = '.' + annee
}
```

De façon analogue, si la liste de sélection relative aux années n'est plus positionnée sur `Toutes`, la variable `selecteur_annee` contiendra la classe avec l'année retenue.

```
$(".item").hide();
$(selecteur_carburant + selecteur_annee).show();
```

La liste complète des items de la liste non ordonnée est cachée. Et la liste filtrée selon le carburant et l'année retenus est affichée.

```
}
```

Fin du else.

```
});
});
```

Fin du script jQuery.

Le fichier final.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
```

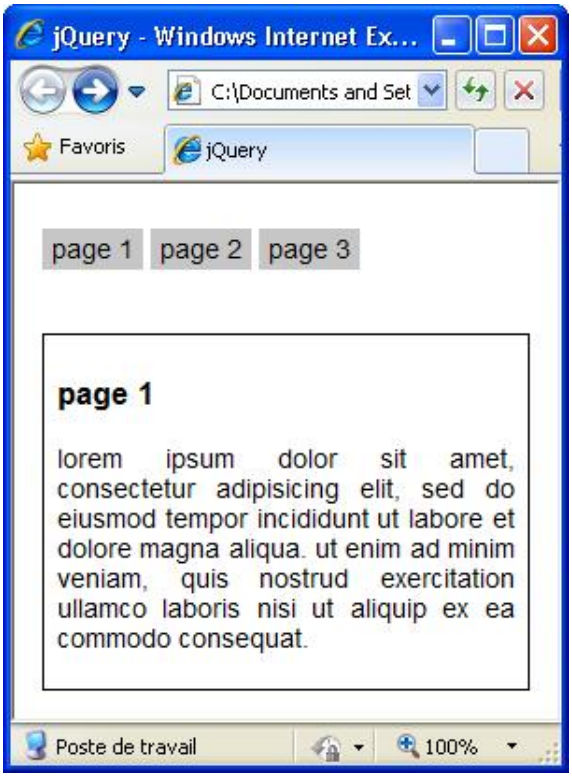

© ENI Editions - All rigths reserved - Jonifar lina

```
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#bouton").click(function(){
var annee = $("#annee").val();
var carburant = $("#carburant").val();
var selecteur_annee = ’’;
var selecteur_carburant = ’’;
if (annee == "toutes" && carburant == "tous"){
$(".item").show();
}
else {
if (carburant != "tous"){
selecteur_carburant = ’.’ + carburant
}
if (annee != "toutes")
{
selecteur_annee = ’.’ + annee
}
$(".item").hide();
$("li"). filter(selecteur_carburant + selecteur_annee).show();
}
});
});
//]]>
</script>
<style type="text/css">
body { font-family: Arial sans-serif;
       font-size: 0.9em;}
ul { list-style-type: none;
     width: 11em;
     border: 1px solid black;}
li { pagdding : 5px;}
</style>
</head>
<body>
Sélectionnez l’année : <select id="annee">
<option selected="selected" value="toutes">Toutes les
années</option>
<option value="2009">2009</option>
<option value="2008">2008</option>
<option value="2007">2007</option></select>
<br />
Sélectionnez le carburant : <select id="carburant">
<option selected="selected" value="tous" >Tous carburants</option>
<option value="essence">Essence</option>
<option value="diesel">Diesel</option>
</select>
<br />
<input id="bouton" type="button" value="Filtrer la liste" />
<ul>
<li class="item 2009 essence">Renault Twingo</li>
<li class="item 2008 essence">Renault Clio</li>
<li class="item 2007 diesel">Renault Laguna</li>
<li class="item 2008 diesel">Renault Scenic </li>
<li class="item 2007 diesel">Peugeot 607</li>
<li class="item 2009 essence">Peugeot 207</li>
<li class="item 2008 diesel">Peugeot 307</li>
<li class="item 2008 diesel">Peugeot 407</li>
<li class="item 2007 essence">Citroen C2</li>
<li class="item 2008 diesel">Citroen C3 Picasso</li>
<li class="item 2007 diesel">Citroen C4</li>
<li class="item 2009 diesel">Citroen C6 </li>
</ul>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

3. Une navigation par onglets

Cette application est fréquemment rencontrée dans le Web 2.0. Selon l'onglet cliqué, un contenu différent s'affiche dans la page ou plus précisément dans une division de celle-ci.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
body { padding: 0px;
       margin: 15px;
       font: 90% arial, sans-serif;}
ul.navigation { padding: 0px;
                list-style-type: none;
                line-height: 40px;
                overflow: hidden;}
ul.navigation li { display: inline;}
ul.navigation li a { padding-top: 3px;
                     padding-bottom: 3px;
                     background-color: #ccc;
                     padding-left: 5px;
                     padding-right: 5px;
                     color: #000;
                     text-decoration: none;
                     line-height: 27px;
                     overflow: hidden;}
ul.navigation li a.selected { background-color: #9cf;
                              border: 2px solid black;
                              color: #000;
                              padding-top: 7px;}
ul.navigation li a:hover { background-color: #9cf;
                           color: #000;
                           padding-top: 7px;
```

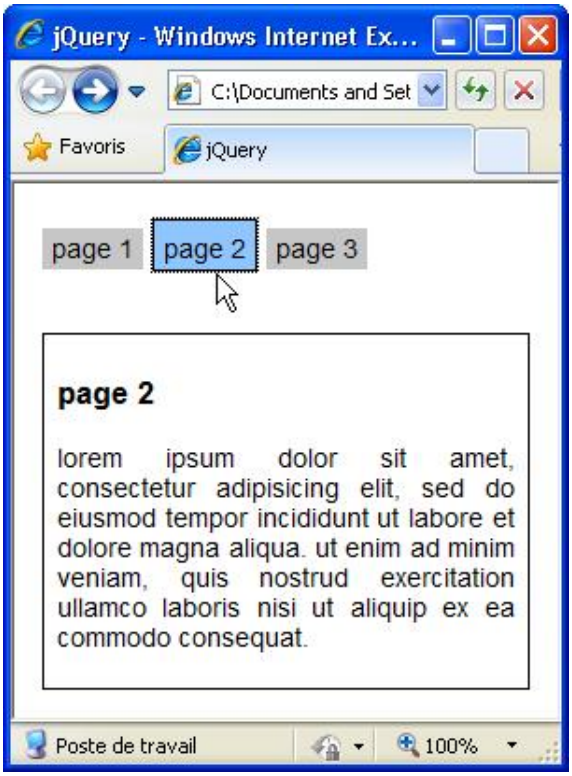

© ENI Editions - All rigths reserved - Jonifar lina

```
                            border: 1px solid black;}
div.menu > div { padding: 5px;
                 margin-top: 3px;}
#un { margin-top: 25px;
      text-align: justify;
      border: 1px solid black;
      padding-left: 7px;
      padding-right: 7px;}
#deux { margin-top: 25px;
        text-align: justify;
        border: 1px solid black;
        padding-left: 7px;
        padding-right: 7px;}
#trois { margin-top: 25px;
         text-align: justify;
         border: 1px solid black;
         padding-left: 7px;
         padding-right: 7px;}
</style>
</head>
<body>
<div class="menu">
<ul class="navigation">
<li><a href="tabs.htm#un">page 1</a></li>
<li><a href="tabs.htm#deux">page 2</a></li>
<li><a href="tabs.htm#trois">page 3</a></li></ul>
<div id="un">
<h3>page 1</h3>
<p>lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
do eiusmod
tempor incididunt ut labore et dolore magna aliqua. ut enim ad
minim veniam, quis nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat.</p>
</div>
<div id="deux">
<h3>page 2</h3>
<p>lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
do eiusmod
tempor incididunt ut labore et dolore magna aliqua. ut enim ad
minim veniam, quis nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo
consequat.</p>
</div>
<div id="trois">
<h3>page 3</h3>
<p>lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
do eiusmod
tempor incididunt ut labore et dolore magna aliqua. ut enim ad
minim veniam, quis nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat.</p>
</div>
</div>
</body>
</html>
```

Le script jQuery.

Au clic sur un onglet, la page correspondante est affichée.

© ENI Editions - All rigths reserved - Jonifar lina




```
<script type="text/javascript">
$(document).ready(function(){
var boites_page = $('div.menu > div');
boites_page.hide()
.filter(':first').show();
$('div.menu ul.navigation a').click(function () {
boites_page.hide();
boites_page.filter(this.hash).show();
$('div.menu ul.navigation a').removeClass('selected');
$(this).addClass('selected');
return false;
});
});
</script>
```

Étudions ce script en détail.

```
$(document).ready(function(){
var boites_page = $('div.menu > div');
```

Au chargement de la page, les éléments `<div>`, enfants de `<div id="menu">` sont chargés dans la variable `boite_page`.

```
boites_page.hide()
.filter(':first').show();
```

Les différentes pages sont, dans un premier temps, cachées pour réinitialiser le processus. Cependant, la première page (`filter(':first')`) est affichée par défaut.

```
$('div.menu ul.navigation a').click(function () {
boites_page.hide();
boites_page.filter(this.hash).show();
```

Au clic sur un onglet, le contenu précédent est caché (`hide()`) et la méthode de filtrage (`filter(this.hash)`) est appliquée pour retrouver la page concernée. Celle-ci est alors affichée (`show()`). La propriété JavaScript (classique) `window.location.hash` qui permet de récupérer l'ancre d'une URL est utilisée ici.

```
$('div.menu ul.navigation a').removeClass('selected');
$(this).addClass('selected');
```


© ENI Editions - All rigths reserved - Jonifar lina

```
return false;
```

La classe `selected` est appliquée à l'onglet sélectionné par l'utilisateur.

```
});
});
```

Fin de script.

Au final :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
body { padding: 0px;
       margin: 15px;
       font: 90% arial, sans-serif;}
ul.navigation { padding: 0px;
                list-style-type: none;
                line-height: 40px;
                overflow: hidden;}
ul.navigation li { display: inline;}
ul.navigation li a { padding-top: 3px;
                     padding-bottom: 3px;
                     background-color: #ccc;
                     padding-left: 5px;
                     padding-right: 5px;
                     color: #000;
                     text-decoration: none;
                     line-height: 27px;
                     overflow: hidden;}
ul.navigation li a.selected { background-color: #9cf;
                              border: 2px solid black;
                              color: #000;
                              padding-top: 7px;}
ul.navigation li a:hover { background-color: #9cf;
                           color: #000;
                           padding-top: 7px;
                           border: 1px solid black;}


div.menu > div { padding: 5px;
                 margin-top: 3px;}
#un { margin-top: 25px;
      text-align: justify;
      border: 1px solid black;
      padding-left: 7px;
      padding-right: 7px;}
#deux { margin-top: 25px;
        text-align: justify;
        border: 1px solid black;
        padding-left: 7px;
        padding-right: 7px;}
#trois { margin-top: 25px;
         text-align: justify;
         border: 1px solid black;
         padding-left: 7px;
         padding-right: 7px;}
</style>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
var boites_page = $('div.menu > div');
```

© ENI Editions - All rigths reserved - Jonifar lina

```
boites_page.hide()
.filter(':first').show();
$('div.menu ul.navigation a').click(function () {
boites_page.hide();
boites_page.filter(this.hash).show();
$('div.menu ul.navigation a').removeClass('selected');
$(this).addClass('selected');
return false;
});
});
</script>
</head>
<body>
<div class="menu">
<ul class="navigation">
<li><a href="tabs.htm#un">page 1</a></li>
<li><a href="tabs.htm#deux">page 2</a></li>
<li><a href="tabs.htm#trois">page 3</a></li></ul>
<div id="un">
<h3>page 1</h3>
<p>lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
do eiusmod
tempor incididunt ut labore et dolore magna aliqua. ut enim ad
minim veniam, quis nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat.</p>
</div>
<div id="deux">
<h3>page 2</h3>
<p>lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
do eiusmod
tempor incididunt ut labore et dolore magna aliqua. ut enim ad
minim veniam, quis nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat.</p>
</div>
<div id="trois">
<h3>page 3</h3>
<p>lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
do eiusmod
tempor incididunt ut labore et dolore magna aliqua. ut enim ad
minim veniam, quis nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat.</p>
</div>
</div>
</body>
</html>
```


© ENI Editions - All rigths reserved - Jonifar lina

Introduction

En quelques années, AJAX (*Asynchronous JavaScript and XML*) s'est fait une place incontournable dans la conception des pages Web. Son apparition correspond au concept du Web 2.0 dont il a en quelque sorte été l'instigateur par son approche innovante et les techniques nouvelles qu'il a apportées.

La librairie jQuery se devait de traiter l'AJAX. Nous retiendrons en préliminaire à l'étude de ce chapitre, l'extrême concision du code et la facilité de mise en œuvre de jQuery en la matière.

Toute notre étude s'est déroulée jusqu'à maintenant côté client. Ce qui était somme toute assez pratique car un éditeur de texte et un navigateur étaient les seuls outils nécessaires. Avec ce chapitre sur AJAX, l'installation d'un serveur Web local, aussi appelé serveur Web personnel, se révèle utile, voire indispensable, pour tester le code, sans passer par la procédure contraignante de téléchargement FTP. Comme serveur Web local, Microsoft IIS ou EasyPHP sont des solutions fiables et aisées à mettre en place.

© ENI Editions - All rigths reserved - Jonifar lina

Les requêtes AJAX raccourcies

1. Charger un fichier

La méthode `load()` permet de charger d'une façon extrêmement simple, un fichier selon le procédé mis en place par Ajax.

Nous détaillerons plus loin, la méthode `ajax()` (voir la section La requête AJAX complète du présent chapitre) qui permet de réaliser la même opération d'une façon plus sophistiquée et plus détaillée. Pour le développeur débutant ou pour des applications simples, la méthode `load()` suffit amplement et fait, de par sa facilité, le bonheur des concepteurs.

load(url[, données,][, fonction])

Charge le code Html (ou Xhtml) à partir d'un fichier donné et place celui-ci dans l'élément sélectionné.

- url : une chaîne de caractères contenant l'URL du fichier Html à charger.
- données (optionnel) : liste de paires de la forme clé/valeur qui seront envoyées en tant que données au serveur.
- fonction (optionnel) : la fonction qui doit être exécutée en cas de réussite de la requête. Par défaut, les données sont chargées dans l'élément sélectionné.

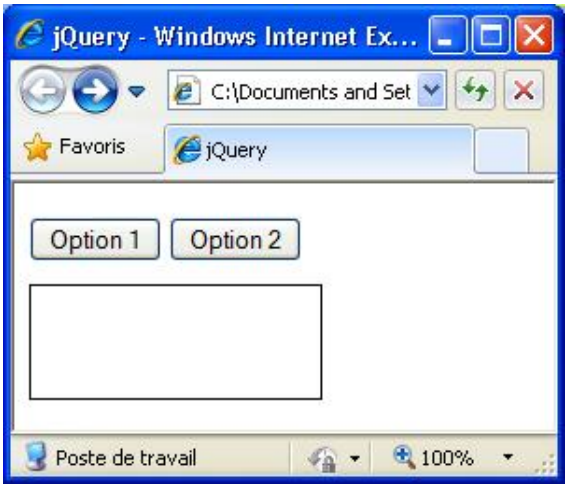
`$("div").load("test.htm")` : charge les données du fichier test.htm et place celles-ci dans la division `<div>`.

Cette méthode renvoie un objet jQuery.

La méthode GET est retenue par défaut sauf si des données sont fournies en argument.

Exemple

Par deux boutons, charger dans une même division des contenus différents.



Le fichier Html de départ se présente ainsi :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
```

© ENI Editions - All rigths reserved - Jonifar lina

```
button { margin-top: 10px;}
div { width: 145px;
      height: 50px;
      border: 1px solid black;
      margin-top: 12px;
      padding: 5px;
      text-align: center;}
</style>
</head>
<body>
<button id="bouton1">Option 1</button>
<button id="bouton2">Option 2</button>
<div>
</div>
</body>
</html>
```

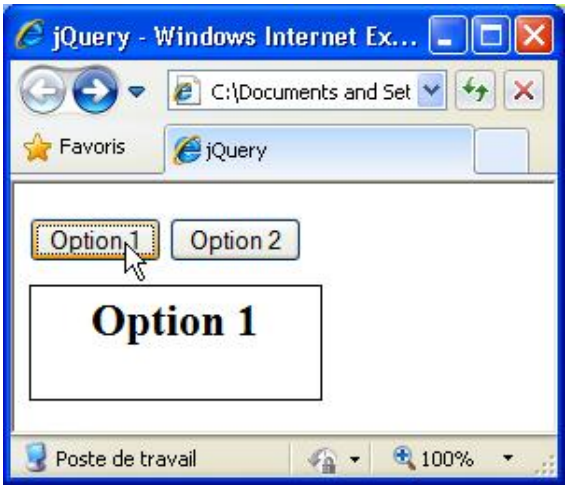
En outre, prévoyons également sur le serveur, le fichier option1.htm qui comporte simplement le code Html suivant :

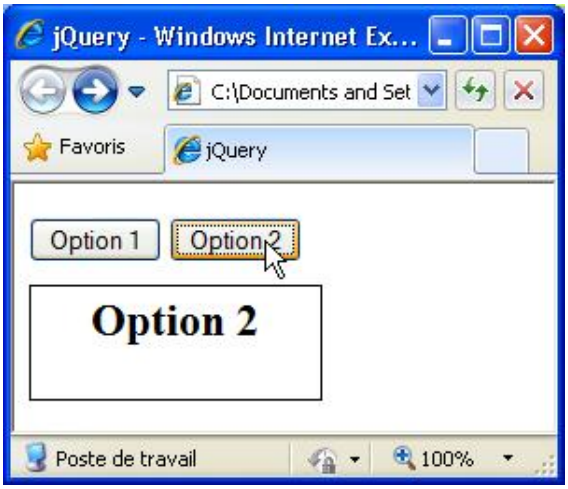
```
<b><font size="5">Option 1</font></b>
```

Et le fichier option2.htm :

```
<b><font size="5">Option 2</font></b>
```

Le script jQuery doit, au clic du bouton 1, charger dans la division `<div>`, le fichier option1.htm et au clic du bouton 2 charger le fichier option2.htm.





Le script jQuery :

```
<script type="text/javascript">
$(document).ready(function(){
$("#bouton1").click(function() {
```


© ENI Editions - All rigths reserved - Jonifar lina

```
$("div").load("option1.htm");
});
$("#bouton2").click(function() {
$("div").load("option2.htm");
});
});
</script>
```

Détaillons celui-ci.

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

Initialisation de jQuery au chargement du DOM.

```
$("#bouton1").click(function() {
$("div").load("option1.htm");
});
```

Au clic du bouton 1 (`$("#bouton1").click()`), le fichier option1.htm est chargé (`load("option1.htm")`) dans la division `<div>`.

```
$("#bouton2").click(function() {
$("div").load("option2.htm");
});
```

Au clic du bouton 2 (`$("#bouton2").click()`), le fichier option2.htm est chargé (`load("option2.htm")`) dans la division `<div>`.

```
});
```

Fin du script.

Le fichier final devient :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#bouton1").click(function() {
$("div").load("option1.htm");
});
$("#bouton2").click(function() {
$("div").load("option2.htm");
});
});
</script>
<style type="text/css">
button { margin-top: 10px;}
div { width: 145px;
      height: 50px;
      border: 1px solid black;
      margin-top: 12px;
      padding: 5px;
      text-align: center;}
</style>
</head>
<body>
<button id="bouton1">Option 1</button>
<button id="bouton2">Option 2</button>
<div>
</div>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
</body>
</html>
```

2. Charger qu'en cas de modification

La méthode `loadIfModified()`, équivalente à la méthode `load()` étudiée au point précédent, ne charge le fichier demandé seulement s'il a été modifié depuis la dernière requête.

loadIfModified(url[, données,][, fonction])

Charge le code Html à partir d'un fichier donné et place celui-ci dans l'élément sélectionné s'il a été modifié depuis la dernière requête.

- url : une chaîne de caractères contenant l'URL du fichier Html à charger.
- données (optionnel) : liste de paires de la forme clé/valeur qui seront envoyées en tant que données au serveur.
- fonction (optionnel) : la fonction qui doit être exécutée en cas de réussite de la requête. Par défaut, les données sont chargées dans l'élément sélectionné.

`$("div").loadIfModified("test.htm")` : charge les données du fichier test.htm et ne place celles-ci dans la division `<div>` que si le fichier a subi des changements depuis la dernière requête.

Cette méthode renvoie un objet jQuery.

3. Charger selon la méthode GET ou POST

Autres façons raccourcies offertes par jQuery pour effectuer des requêtes AJAX sont les méthodes `$.get()` et `$.post()`.

$.get(url[, données][, fonction][, type])

Charge un fichier du serveur selon une requête HTTP GET.

- url : une chaîne de caractères contenant l'URL du fichier à charger.
- données (optionnel) : liste de paires de la forme clé/valeur qui seront envoyées en tant que données au serveur.
- fonction (optionnel) : la fonction qui doit être exécutée en cas de réussite de la requête.
- type (optionnel) : chaîne de caractères qui spécifie le type de données transmises à la fonction : "xml", "html", "script", "json", "jsonp" ou "text".

```
$.get("test.html", function(data){
$("#resultat").html(data);
});
```

Cette méthode renvoie un objet XMLHttpRequest.

$.post(url[, données][, fonction][, type])

Charge un fichier du serveur selon une requête HTTP POST.

- url : une chaîne de caractères contenant l'URL du fichier à charger.
- données (optionnel) : liste de paires de la forme clé/valeur qui seront envoyées en tant que données au serveur.
- fonction (optionnel) : la fonction qui doit être exécutée en cas de réussite de la requête.

© ENI Editions - All rigths reserved - Jonifar lina

- type (optionnel) : chaîne de caractères qui spécifie le type de données transmises à la fonction : "xml", "html", "script", "json", "jsonp" ou "text".

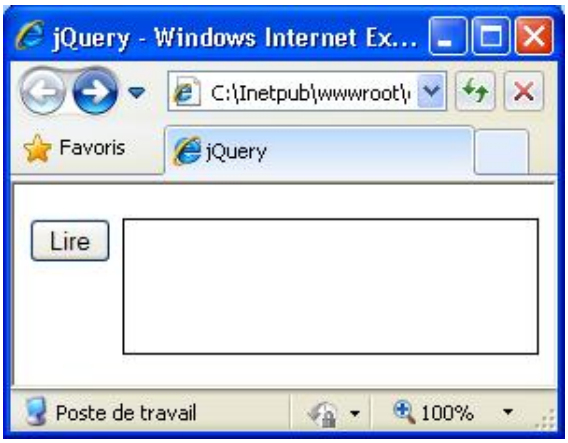
```
$.post("test.html", function(data){
$("#resultat").html(data);
});
```

Cette méthode renvoie un objet XMLHttpRequest.

Au contraire de `load()`, il faut spécifier ici une fonction pour traiter les données retournées par le serveur.

Exemple

Au clic d'un bouton, afficher dans une division, le contenu d'un article. La méthode `$.get()` est utilisée.



Le fichier de départ peut se présenter comme suit.

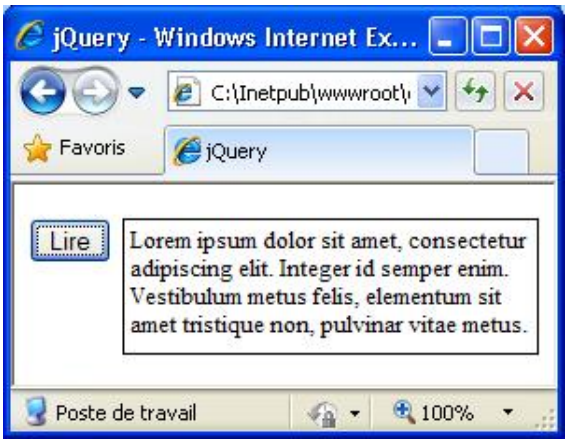
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
button { margin-top: 10px;}
#gauche { width: 50px;
          float: left;}
#droit { width: 215px;
         height: 65px;
         border: 1px solid black;
         margin-top: 10px;
         padding: 3px;
         font-size: 0.8em;
         float: left;}
</style>
</head>
<body>
<div id="gauche">
<button id="bouton">Lire</button>
</div>
<div id="droit">
</div>
</body>
</html>
```

Le fichier à charger du serveur (lorem.htm) comporte le contenu suivant :

© ENI Editions - All rigths reserved - Jonifar lina

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer
id semper enim. Vestibulum metus felis, elementum sit amet
tristique non, pulvinar vitae metus.
```

Le script jQuery utilise `$.get()` pour charger le fichier depuis le serveur et l'affiche dans la division droite entourée d'une bordure.




```
<script type="text/javascript">
$(document).ready(function(){
$("#bouton").click(function() {
$.get("lorem.htm", function(contenu){
$("#droit").append(contenu);
});
});
});
</script>
```

Explication du script.

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

Initialisation du DOM et de jQuery.

```
$("#bouton").click(function() {
```

Au clic du bouton.

```
$.get("lorem.htm", function(contenu){
$("#droit").append(contenu);
});
```

La requête est réalisée par `$.get()` qui demande le fichier lorem.htm. Ensuite, une fonction est élaborée pour ajouter (`append()`) le contenu du fichier dans la division identifiée par `droit`.

```
});
});
```

Fin de script.

Au final, le code de la page est :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
```


© ENI Editions - All rigths reserved - Jonifar lina

```
$(document).ready(function(){
$("#bouton").click(function() {
$.get("lorem.htm", function(contenu){
$("#droit").append(contenu);
});
});
});
</script>
<style type="text/css">
button { margin-top: 10px;}
#gauche { width: 50px;
          float: left;}
#droit { width: 215px;
         height: 65px;
         border: 1px solid black;
         margin-top: 10px;
         padding: 3px;
         font-size: 0.8em;
         float: left;}
</style>
</head>
<body>
<div id="gauche">
<button id="bouton">Lire</button>
</div>
<div id="droit">
</div>
</body>
</html>
```

Commentaire

En parallèle à `loadIfModified( url[, données,][, fonction])`, la méthode `$.getIfModified( url[, données][, fonction][, type])` existe également. Son fonctionnement est identique.

Par contre, l'option `IfModified` n'existe pas avec `$.post()` car avec POST, les pages ne sont jamais mises en cache.

4. Charger un script

$.getScript(url[, fonction])

Charge un script JavaScript du serveur en utilisant la méthode HTTP GET et exécute celui-ci.

- url : une chaîne de caractères qui indique l'adresse du script à charger.
- fonction (optionnel) : une fonction à exécuter si la requête est réussie. Cette fonction n'est généralement pas nécessaire car le script s'exécute automatiquement.

```
$.getScript("test.js");
```

Cette méthode renvoie un objet XMLHttpRequest.

Exemple

Déclencher un message d'alerte JavaScript chargé à partir du serveur.

© ENI Editions - All rigths reserved - Jonifar lina

Le fichier de départ tout simple.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
button { margin-top: 10px;}
</style>
</head>
<body>
<button id="bouton">Script</button>
</body>
</html>
```

Le message d'alerte est inclus dans le fichier alerte.js.

```
alert("Je viens du serveur");
```

Le script jQuery va, au clic sur le bouton, rapatrier le fichier alerte.js et l'exécuter.

```
<script type="text/javascript">
$(document).ready(function(){
$("#bouton").click(function() {
$.getScript("alert.js");
});
});
</script>
```

Au final :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#bouton").click(function() {
$.getScript("alert.js");
});
});
```


© ENI Editions - All rigths reserved - Jonifar lina

```
</script>
<style type="text/css">
button { margin-top: 10px;}
</style>
</head>
<body>
<button id="bouton">Script</button>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

La requête AJAX complète

Cette méthode permet d'effectuer une requête AJAX en maîtrisant, grâce aux nombreuses options disponibles, les différents paramètres et étapes de celle-ci.

ajax(options)

Réalise une requête HTTP asynchrone (AJAX).

```
$.ajax({
url: "test.htm",
success: function(data ) {
$("#resultat").html(data);
$().log("Terminé");
},
});
```

La méthode renvoie un objet XMLHttpRequest.

Passons en revue les nombreuses options disponibles.

- url (obligatoire): une chaîne de caractères contenant l'adresse de la requête.
- type (optionnel) : une chaîne de caractères qui définit la méthode HTTP à utiliser pour la requête (GET ou POST). La valeur par défaut est GET. D'autres méthodes d'envoi HTTP peuvent être utilisées, comme PUT ou DELETE, mais celles-ci ne sont pas supportées par tous les navigateurs.
- dataType (optionnel) : une chaîne de caractères qui spécifie le format des données qui seront renvoyées par le serveur (xml, html, json ou script). Si rien n'est spécifié, jQuery utilisera le type MIME pour déterminer le format adéquat soit `responseXML` ou `ResponseText`. Les types disponibles sont :
  - "xml" : retourne un document XML qui pourra être traité par jQuery.
  - "html" : retourne du code Html au format texte.
  - "script" : évalue la réponse en JavaScript et retourne cette dernière au format texte.
  - "json" : évalue la réponse en JSON et retourne un objet JavaScript.
- ifModified (optionnel) : une valeur booléenne qui indique que le serveur doit vérifier si les données retournées sont différentes de la dernière requête avant de renvoyer le fichier avec succès. Par défaut, cette option vaut `false`.
- timeout (optionnel) : nombre de millisecondes après lequel la requête est considérée comme non réussie.
- global (optionnel) : une valeur booléenne qui permet le déclenchement du gestionnaire d'évènements global d'AJAX. Par défaut, la valeur est `true`. Avec une valeur `false`, les déclenchements d'évènements de type `ajaxStart()` ou `ajaxStop()` sont ignorés.
- beforeSend (optionnel) : une fonction qui doit être exécutée avant l'envoi de la requête. Ceci permet de modifier l'objet `XMLHttpRequest` avant qu'il soit envoyé pour spécifier, par exemple, des en-têtes HTTP personnalisées.
- error (optionnel) : une fonction qui doit être appelée en cas d'échec de la requête. La fonction dispose de trois arguments : l'objet `XMLHttpRequest`, une chaîne de caractères décrivant le type d'erreur rencontré et un objet d'exception, dans le cas où ce dernier a été généré.
- success (optionnel) : fonction à appeler si la requête s'exécute avec succès. Un seul argument est passé en paramètre soit les données retournées par le serveur.
- complete (optionnel) : fonction à exécuter lorsque la requête se termine. La fonction dispose de deux

© ENI Editions - All rigths reserved - Jonifar lina

arguments : l'objet `XMLHttpRequest` et une chaîne de caractères décrivant le type de succès de la requête.

- data (optionnel) : données à envoyer au serveur. L'objet doit être formé de paires de la forme clé/valeur. Les données sont converties en chaîne de caractères (si elles ne le sont pas déjà). Voir l'option processData ci-après pour empêcher ce processus automatique.

- processData (optionnel) : valeur booléenne qui indique si les données de l'option data doivent être converties en chaîne de caractères. La valeur par défaut est `true`. Pour empêcher la conversion, passez cette option à `false`.

- contentType (optionnel) : chaîne de caractères contenant le MIME des données lorsque des données sont envoyées au serveur. Par défaut, le MIME application/x-www-form-urlencoded est retenu.

- async (optionnel) : une valeur booléenne qui indique si la requête doit s'effectuer de façon asynchrone ou synchrone. La valeur par défaut pour une requête AJAX est bien entendu `true`.

D'autres options sont encore disponibles mais d'un usage moins fréquent. Énumérons simplement :

- cache (optionnel) : une valeur booléenne qui, lorsqu'elle est mise sur `false`, empêche la page chargée d'être mise dans le cache du navigateur.

- password (optionnel) : dans le cas où l'accès HTTP de la requête nécessite un mot de passe.

- username (optionnel) : dans le cas où l'accès HTTP de la requête nécessite un nom d'utilisateur (username).

- scriptCharset (optionnel) : force les requêtes du type script à être interprétées avec un charset particulier.

- xhr (optionnel) permet de créer l'`ActiveXObject` (Internet Explorer) ou le `XMLHttpRequest` (pour les autres).

Commentaires

La forme la plus simple de cette fonction `$.ajax()` doit au moins spécifier l'URL des données à charger.

```
$.ajax({
url: "test.htm",
});
```

Nous allons voir au point ci-après que ce seul paramètre peut, à son tour, devenir optionnel avec la méthode `ajaxSetup()`.

Il faut souligner que la méthode `ajax()` charge les contenus de l'URL spécifiée mais (au contraire de `load()` par exemple) ne fait rien avec ceux-ci. Pour que ces contenus apparaissent dans la page, il faut préciser les opérations à effectuer en utilisant les fonctions associées aux options success ou complete.

```
$.ajax({
url: "test.htm",
success: function(data ) {
$("div").html(data);
},
});
```

Ainsi, les données chargées par la requête à l'adresse test.htm sont en cas de succès, affichées comme du Html dans la division `<div>`.

Pour des opérations aussi simples que celle de l'exemple, il est bien plus aisé d'utiliser `load()` ou `$.get()`.

Exemple

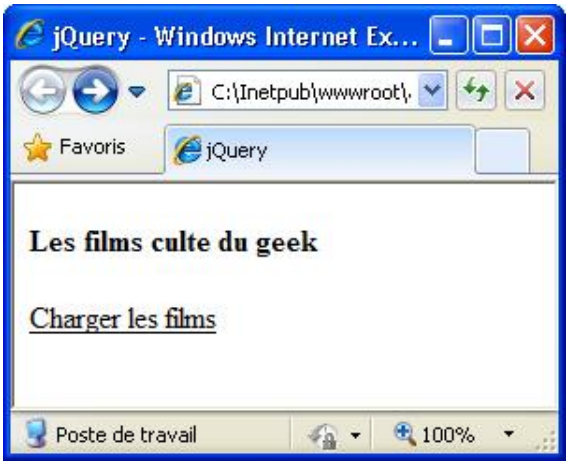
Au clic d'un lien, faisons apparaître un contenu à partir du serveur.

Le fichier de départ se présente comme suit :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
a { color: black;}
li { list-style-type: none;}
</style>
</head>
<body>
<p>Les films culte du geek</p>
<a href="#">Charger les films</a>
<div id="contenu">
</div>
</body>
</html>
```



Le fichier à charger (films.htm) comporte la liste suivante :

```
<ul>
<li>Retour vers le futur</li>
<li>Matrix</li>
<li>Tron</li>
<li>Star Wars</li>
<li>Star Trek</li>
<li>Le seigneur des anneaux</li>
</ul>
```

Le script jQuery doit lancer une requête AJAX vers le fichier films.htm et en afficher le contenu dans la division `contenu` de la page.

© ENI Editions - All rigths reserved - Jonifar lina

```
<script type="text/javascript">
$(document).ready(function(){
$('a').click(function() {
$('a').hide();
$("#contenu").empty();
$.ajax({
url: 'films.htm',
async: true,
type: 'GET',
global: false,
cache: false,
success: function(html){
$("#contenu").append(html)
}
});
});
});
</script>
```

Détaillons ce script.

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

Chargement du DOM.

```
$('a').click(function() {
```

Au clic sur le lien <a>.

```
$('a').hide();
```

Le lien et son contenu sont cachés.

```
$("#contenu").empty();
```

Le contenu de la division `contenu` est vidé.

```
$.ajax({
url: 'films.htm',
async: true,
type: 'GET',
global: false,
cache: false,
success: function(html){
$("#contenu").append(html)
```


© ENI Editions - All rigths reserved - Jonifar lina

```
}
});
```

La requête AJAX de jQuery charge le fichier situé à l'URL films.htm. Le processus est effectué sous un mode asynchrone. La méthode HTTP retenue est GET. Le gestionnaire d'événements global d'AJAX est désactivé. Le fichier rapatrié du serveur n'est pas mis en cache. Enfin, si la requête s'est déroulée avec succès, les données du fichier chargé par celle-ci sont ajoutées à la division contenu.

```
});
});
```

Fin du script.

Le fichier final devient :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$('a').click(function() {
$('a').hide();
$("#contenu").empty();
$.ajax({
url: 'films.htm',
async: true,
type: 'GET',
global: false,
cache: false,
success: function(html){
$("#contenu").append(html)
}
});
});
});
</script>
<style type="text/css">
a { color: black;}
li { list-style-type: none;}
</style>
</head>
<body>
<p>Les films culte du geek</p>
<a href="#">Charger les films</a>
<div id="contenu">
</div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Définir une requête par défaut

Comme suggéré au point précédent, il est possible de définir l'URL par défaut des requêtes AJAX de la page (pour autant qu'elle soit identique pour toutes les requêtes AJAX de la page).

ajaxSetup(paramètres)

Définit les paramètres globaux pour toutes les requêtes AJAX de la page.

```
$.ajaxSetup( {
url: test.htm",
global: false,
type: "POST"
});
```

Toutes les requêtes AJAX de la page seront paramétrées avec l'URL indiquée, le gestionnaire d'événements AJAX désactivé et les envois seront effectués par la méthode POST au lieu de GET.

Ainsi, l'URL par défaut peut être définie par le code :

```
$.ajaxSetup({
url: "test.htm",
});
```

Chaque fois qu'une requête doit être réalisée, l'URL sera automatiquement utilisée. Ainsi, il suffit alors d'écrire :

```
$.ajax({});
```

© ENI Editions - All rigths reserved - Jonifar lina

Les événements associés à la requête

1. ajaxSend()

ajaxSend(fonction)

Assigne une fonction qui sera exécutée avant l'envoi de la requête AJAX.

```
$("#loading").ajaxSend(function(){
$(this).show();
});
```

Cette méthode renvoie un objet jQuery.

Exemple

Afficher un message avant que la requête ne débute.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#message").hide();
$("#resultat").hide();
$('#message').ajaxSend(function() {
$(this).append('La requête AJAX va débuter !<br>').show();
});
$('a').click(function() {
$('#resultat').load('a.html');
});
});
//]]>
</script>
<style type="text/css">
a { color: black;}
#message { width: 200px;
           border: 1px solid black;
           margin-top: 15px;
           background-color: #9cf;
           padding-left: 5px;}
</style>
</head>
<body>
<h3>ajaxSend()</h3>
<a href="#">Démarrer AJAX</a>
<div id="message">
</div>
<div id="resultat">
</div>
</body>
</html>
```

Passons en revue les différentes lignes du code jQuery.

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

Chargement de jQuery.

```
$("#message").hide();
```

La division `message` qui contiendra le message associé à l'événement `ajaxSend()` est cachée au chargement de la page.

```
$("#resultat").hide();
```

La division `resultat` qui contiendra le contenu du fichier chargé est cachée car ce contenu n'a pas d'importance dans cet exemple.

```
$('#message').ajaxSend(function() {
$(this).append('La requête AJAX débute !<br>').show();
});
```

Lors de l'événement `ajaxSend()`, la phrase `"La requête AJAX va débuter !"` est ajoutée à la division `message` et celle-ci est alors rendue visible.

```
$('a').click(function() {
$('#resultat').load('a.html');
});
```

Au clic du lien, le fichier a.html est chargé et inséré dans la division `resultat`. Pour rappel, cette division a été cachée en début de script.

```
});
```

Fin de jQuery.

© ENI Editions - All rigths reserved - Jonifar lina

2. ajaxStart()

ajaxStart(fonction)

Assigne une fonction à exécuter lorsqu'une requête AJAX débute.

```
$("#loading").ajaxStart(function(){
$(this).show();
});
```

Cette méthode renvoie un objet jQuery.

Exemple

Appliqué à l'exemple précédent, le script avec `ajaxStart()` devient :

```
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#message").hide();
$("#resultat").hide();
$('#message').ajaxStart(function() {
$(this).append('La requête AJAX débute !<br>').show();
});
$('a').click(function() {
$('#resultat').load('a.html');
});
});
//]]>
</script>
```

3. ajaxStop()

ajaxStop(fonction)

Assigne une fonction qui sera exécutée à chaque fois qu'une requête se termine.

Cette méthode renvoie un objet jQuery.

Exemple

Appliqué à l'exemple précédent, le script avec `ajaxStop()` devient :

```
<script type="text/javascript">
//<![CDATA[
```

© ENI Editions - All rigths reserved - Jonifar lina

```
$(document).ready(function(){
$("#message").hide();
$("#resultat").hide();
$('#message').ajaxStop(function() {
$(this).append('La requête AJAX se termine !<br>').show();
});
$('a').click(function() {
$('#resultat').load('a.html');
});
});
//]]>
</script>
```

4. ajaxSuccess()

ajaxSuccess(fonction)

Assigne une fonction qui sera exécutée à chaque fois qu'une requête AJAX se sera terminée avec succès.

Cette méthode renvoie un objet jQuery.

Exemple

Appliqué à l'exemple précédent, le script avec `ajaxSuccess()` devient :

```
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#message").hide();
$("#resultat").hide();
$('#message').ajaxSuccess(function() {
$(this).append('La requête AJAX est un succès !<br>').show();
});
$('a').click(function() {
$('#resultat').load('a.html');
});
});
//]]>
</script>
```


© ENI Editions - All rigths reserved - Jonifar lina

5. ajaxComplete()

ajaxComplete(fonction)

Assigne une fonction qui sera exécutée lorsque le processus total de la requête AJAX sera terminé.

Cette méthode renvoie un objet jQuery.

Exemple

Appliqué à l'exemple précédent, le script avec `ajaxComplete()` devient :

```
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#message").hide();
$("#resultat").hide();
$('#message').ajaxComplete(function() {
$(this).append('La requête AJAX est terminée !<br>').show();
});
$('a').click(function() {
$('#resultat').load('a.html');
});
});
//]]>
</script>
```

© ENI Editions - All rigths reserved - Jonifar lina

6. ajaxError()

ajaxError(fonction)

Assigne une fonction qui sera exécutée si la requête AJAX échoue.

Cette méthode renvoie un objet jQuery.

Exemple

Reprenons notre fichier exemple mais (volontairement) avec une URL qui n'existe pas sur le serveur soit x.html.

Cette absence de fichier empêche que la requête AJAX soit menée à bien et son échec entraîne, bien entendu, une erreur.

Le script, appliqué à l'exemple, avec `ajaxError()` devient :

```
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("#message").hide();
$("#resultat").hide();
$('#message').ajaxError(function() {
$(this).append('Erreur de la requête !<br>').show();
});
$('a').click(function() {
$('#resultat').load('x.html');
});
});
//]]>
</script>
```


© ENI Editions - All rigths reserved - Jonifar lina

Sérialiser les données

Cette méthode transforme les données des champs de formulaire en une chaîne de caractères reprenant celles-ci.

Ce procédé est fort utile pour envoyer ces données au serveur par une requête AJAX sous un format compatible avec la plupart des langages de programmation côté serveur.

Pour le bon fonctionnement de la méthode `serialize()`, les champs de formulaire doivent posséder un attribut name.

serialize()

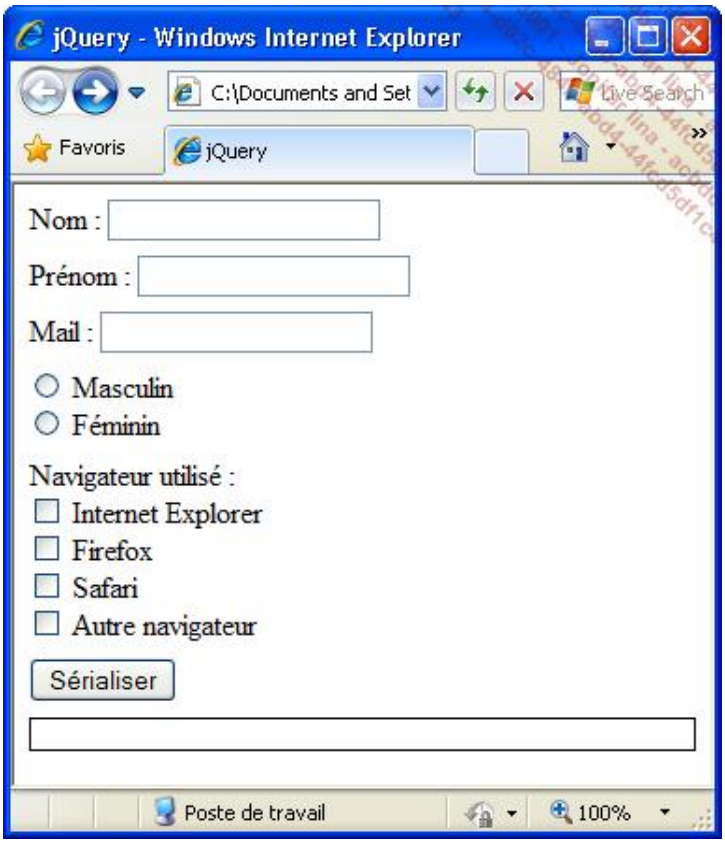
Transforme les données des champs de formulaire en une chaîne de caractères.

```
$("form").serialize();
```

Cette méthode renvoie une chaine de caractères (String).

Exemple

Soit un formulaire de départ.

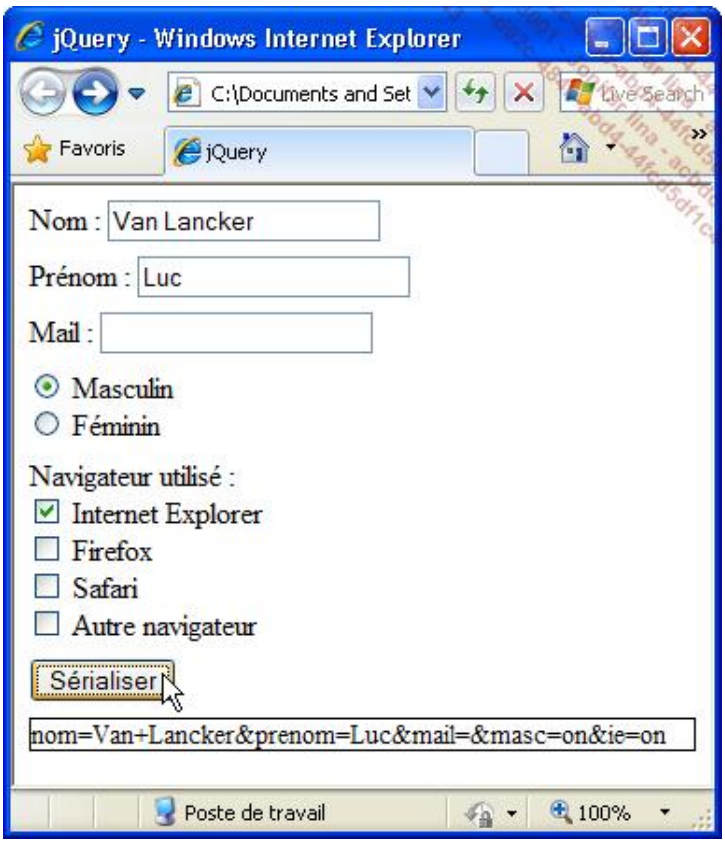



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
div { margin-bottom: 8px;}
p { margin-top: 8px;
    font-size:14px;
    border: 1px solid black;}
</style>
</head>
<body>
<form>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<div>Nom : <input type="text" name="nom" id="nom" /></div>
<div>Prénom : <input type="text" name="prenom" id="prenom"
/></div>
<div>Mail : <input type="text" name="mail" id="mail" /></div>
<div>
<input type="radio" name="masc" /> Masculin<br />
<input type="radio" name="fem" /> Féminin<br />
</div>
<div>
Navigateur utilisé :<br />
<input type="checkbox" name="ie" /> Internet Explorer<br />
<input type="checkbox" name="ff" /> Firefox<br />
<input type="checkbox" name="saf" /> Safari<br />
<input type="checkbox" name="autre_nav" /> Autre navigateur
</div>
</form>
<button>Sérialiser</button>
<p id="resultat"> </p>
</body>
</html>
```

Au clic du bouton, les données encodées par l'utilisateur seront sérialisées.



Le script suivant a été utilisé.

```
<script type="text/javascript">
$(document).ready(function(){
$('button').click(function() {
var str = $("form").serialize();
$("#resultat").text(str);
});
});
</script>
```

Soit,

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


© ENI Editions - All rigths reserved - Jonifar lina

Initialisation de jQuery.

```
$(’button’).click(function() {
```

Au clic du bouton.

```
var str = $("form").serialize();
```

Le formulaire (`$("form")`) est sérialisé (`serialize()`) et stocké dans la variable `str`.

```
$("#resultat").text(str);
```

Le contenu de la variable `str` est ajouté comme du texte (`text(str)`) dans le paragraphe identifié par `resultat`.

```
});
});
```

Fin du script.

Le fichier final devient :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$(’button’).click(function() {
var str = $("form").serialize();
$("#resultat").text(str);
});
});
</script>
<style type="text/css">
div { margin-bottom: 8px;}
p { margin-top: 8px;
    font-size:14px;
    border: 1px solid black;}
</style>
</head>
<body>
<form action="">
<div>Nom : <input type="text" name="nom" id="nom" /></div>
<div>Prénom : <input type="text" name="prenom" id="prenom"
/></div>
<div>Mail : <input type="text" name="mail" id="mail" /></div>
<div>
<input type="radio" name="masc" /> Masculin<br />
<input type="radio" name="fem" /> Féminin<br />
</div>
<div>
Navigateur utilisé :<br />
<input type="checkbox" name="ie" /> Internet Explorer<br />
<input type="checkbox" name="ff" /> Firefox<br />
<input type="checkbox" name="saf" /> Safari<br />
<input type="checkbox" name="autre_nav" /> Autre navigateur
</div>
</form>
<button>Sérialiser</button>
<p id="resultat"> </p>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

© ENI Editions - All rigths reserved - Jonifar lina

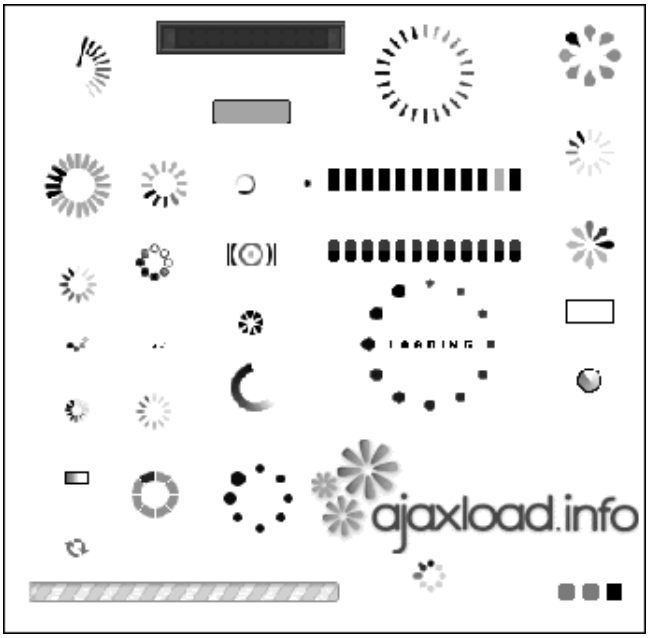
Applications

1. Une icône de chargement

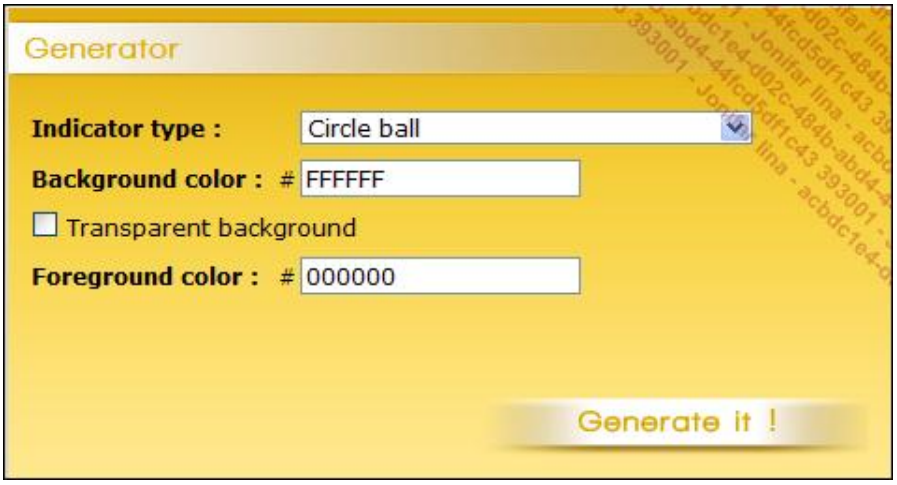
Les requêtes AJAX peuvent parfois entraîner une relative lenteur au chargement du fichier externe en fonction de la taille du fichier, de l'encombrement du serveur et/ou de la qualité de la connexion.

Ainsi, pour éviter que l'utilisateur ne soit désappointé par ces moments de latence, le concepteur peut prévoir une petite icône indiquant que le chargement est en cours. Cette icône est en fait une image gif animée qui apparaît à l'initialisation de la requête et qui disparaît lorsque la requête a abouti avec succès.

Le site ajaxload (http://www.ajaxload.info/) propose une série impressionnante de ces icônes animées de téléchargement.



Son interface permet de choisir le type d'animation (Indicator type), la couleur de fond (Background color) et la couleur de l'animation (Foreground color). Le bouton **Generate it !** crée l'image et en donne un aperçu. Le bouton **Download it !**, vous permet de télécharger l'icône animée. Rien de plus simple. Parfait pour ne pas perdre de temps à créer soi-même ces images d'attente. Bien entendu, c'est entièrement gratuit.



Exemple

Soit le fichier de départ qui comporte simplement un bouton et une division destinée à recevoir le fichier une fois chargé.

© ENI Editions - All rigths reserved - Jonifar lina

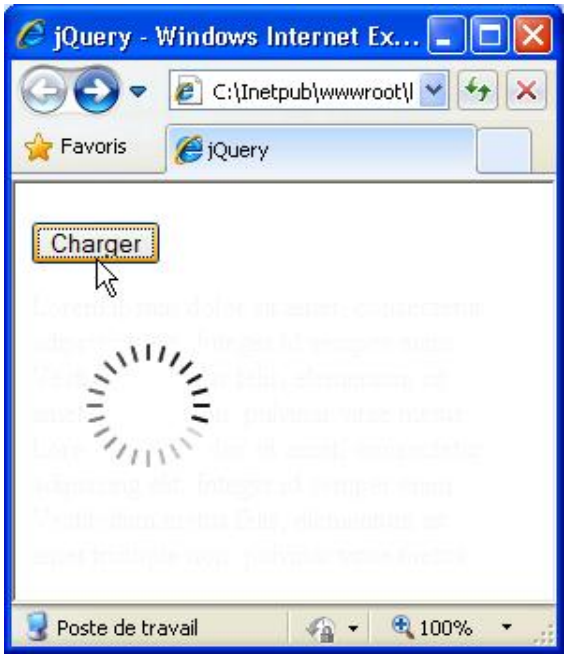
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
button { margin: 12px 0 12px 0}
#contenu { width: 250px;}
</style>
</head>
<body>
<button id="bouton">Charger</button>
<div id="contenu"></div>
</body>
</html>
```

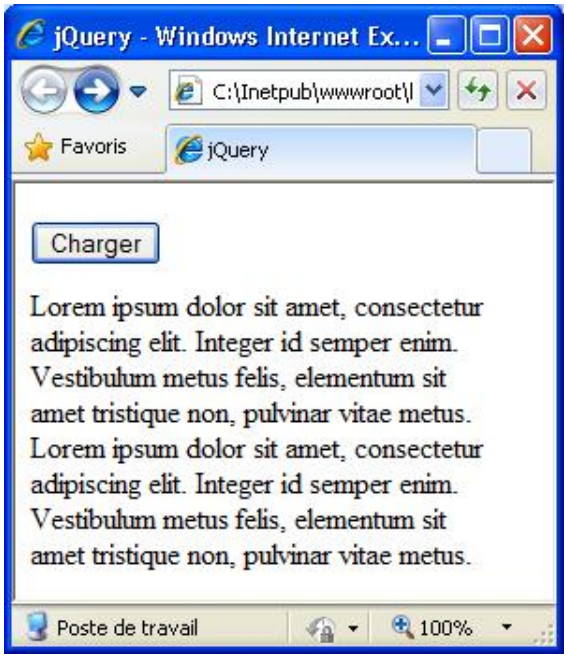
Le fichier à charger depuis le serveur s'appelle ici lorembis.htm. Lors de la phase de développement et de test en interne, l'apparition de l'icône de téléchargement risque d'être assez furtive. Un fichier assez long est plutôt conseillé pour apercevoir (ou plutôt entrapercevoir) celle-ci.

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer
id semper enim. Vestibulum metus felis, elementum sit amet
tristique non, pulvinar vitae metus.Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Integer id semper enim. Vestibulum
metus felis, elementum sit amet tristique non, pulvinar vitae
metus.
...
```

Le script jQuery doit prendre en charge non seulement la requête AJAX mais aussi l'apparition et la disparition de l'image de chargement.

© ENI Editions - All rigths reserved - Jonifar lina





L'icône est disponible dans l'espace de téléchargement.

```
<script type="text/javascript">
$(document).ready(function() {
$('#bouton').click(function() {
$('#contenu').hide().load('lorembis.htm', function() {
$(this).fadeIn(4000);
});
return false;
});
$('<div id="loading"><img src="ajax-loader.gif" /></div>')
.insertBefore('#contenu')
.ajaxStart(function() {
$(this).show();
})
.ajaxStop(function() {
$(this).hide();
});
});
</script>
```

© ENI Editions - All rigths reserved - Jonifar lina

Ce script nécessite quelques explications.

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

Initialisation de jQuery

```
$('#bouton').click(function() {
```

Au clic du bouton.

```
$('#contenu').hide().load('lorembis.htm', function() {
$(this).fadeIn(4000);
});
```

Une requête AJAX est effectuée vers le fichier lorembis.htm et son contenu est inséré dans la division `contenu`. Un effet apparition progressive a été ajouté soit `fadeIn(4000)`.

```
return false;
});
```

Fin de la partie de chargement.

```
$('<div id="loading"><img src="ajax-loader.gif" /></div>')
.insertBefore('#contenu')
```

L'image ajax-loader.gif, incluse dans la division `loading` est insérée (`insertBefore`) avant la division `contenu`.

```
.ajaxStart(function() {
$(this).show();
})
```

Lorsque la requête AJAX démarre (`ajaxStart`), cette image (`this`) est rendue visible (`show()`).

```
.ajaxStop(function() {
$(this).hide();
});
```

Lorsque la requête AJAX se termine (`ajaxStop`), cette image (`this`) est alors cachée (`hide()`).

```
});
```

Fin du script jQuery.

Au final, voici la page définitive.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function() {
$('#bouton').click(function() {
$('#contenu').hide().load('lorembis.htm', function() {
$(this).fadeIn(4000);
});
return false;
});
$('<div id="loading"><img src="ajax-loader.gif" /></div>')
.insertBefore('#contenu')
.ajaxStart(function() {
$(this).show();
```

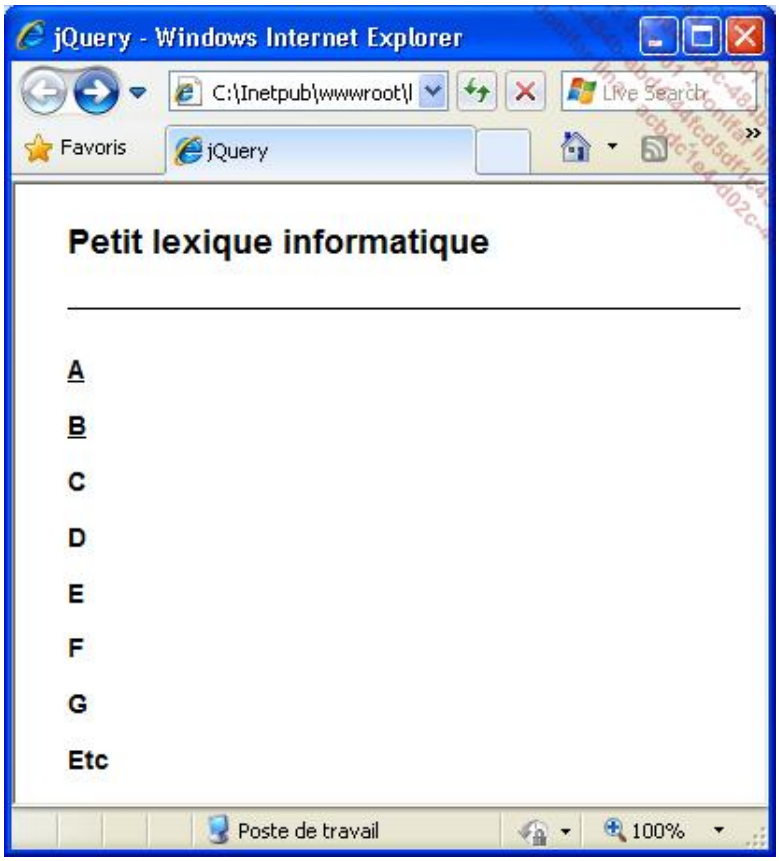

© ENI Editions - All rigths reserved - Jonifar lina

```
})
.ajaxStop(function() {
$(this).hide();
});
});
</script>
<style type="text/css">
button { margin: 12px 0 12px 0}
#contenu { width: 250px;}
#loading { margin: 30px 0 0 30px;
           position: absolute;
           display: none;}
</style>
</head>
<body>
<button id="bouton">Charger</button>
<div id="contenu"></div>
</body>
</html>
```

2. Un lexique en AJAX

Élaborons un petit lexique informatique dont les définitions sont chargées par des requêtes AJAX dans la page principale.

Cette page principale se présente comme suit.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
```

© ENI Editions - All rigths reserved - Jonifar lina

```
<style type="text/css">
body { font: 62.5% Arial, Verdana, sans-serif;}
a { color: black}
#container { font-size: 1.2em;
             margin: 10px 20px;
             width: 360px;}
#header { margin-top: 20px;
          margin-bottom: 10px;
          padding-bottom: 10px;
          border-bottom: 1px solid black;}
#alphabet { float: left;
            width: 30px;
            padding-right: 10px;
            margin-right: 10px; }
#lexique { float: left;
           width: 300px;}
</style>
</head>
<body>
<div id="container">
<div id="header">
<h2>Petit lexique informatique</h2>
</div>
<div id="alphabet">
<div id="lettre_a">
<h3><a href="#">A</a></h3>
</div>
<div id="lettre_b">
<h3><a href="#">B</a></h3>
</div>
<div><h3>C</h3></div>
<div><h3>D</h3></div>
<div><h3>E</h3></div>
<div><h3>F</h3></div>
<div><h3>G</h3></div>
<div><h3>Etc</h3></div>
</div>
<div id="lexique">
</div>
</div>
</body>
</html>
```

Le fichier des définitions de la lettre A (lettre_a.htm) est le suivant :

```
<h3>ActiveX</h3>
<div>
Technologie Microsoft. La technologie ActiveX permet la création
de composants logiciels qui peuvent être inclus dans des pages
Html.
</div>
<h3>AVI</h3>
<div>
Audio Video Interleave. Format de fichier Microsoft pour les
données audio et vidéo.
</div>
<h3>AZERTY</h3>
<div>
Nom du clavier français qui vient de l’ordre des six premières
lettres de la première rangée.
</div>
```

Le fichier des définitions de la lettre B (lettre_b.htm) est le suivant :

```
<h3>Backup</h3>
<div>
Version anglaise de sauvegarde, ou de secours.
</div>
<h3>Big Blue</h3>
```

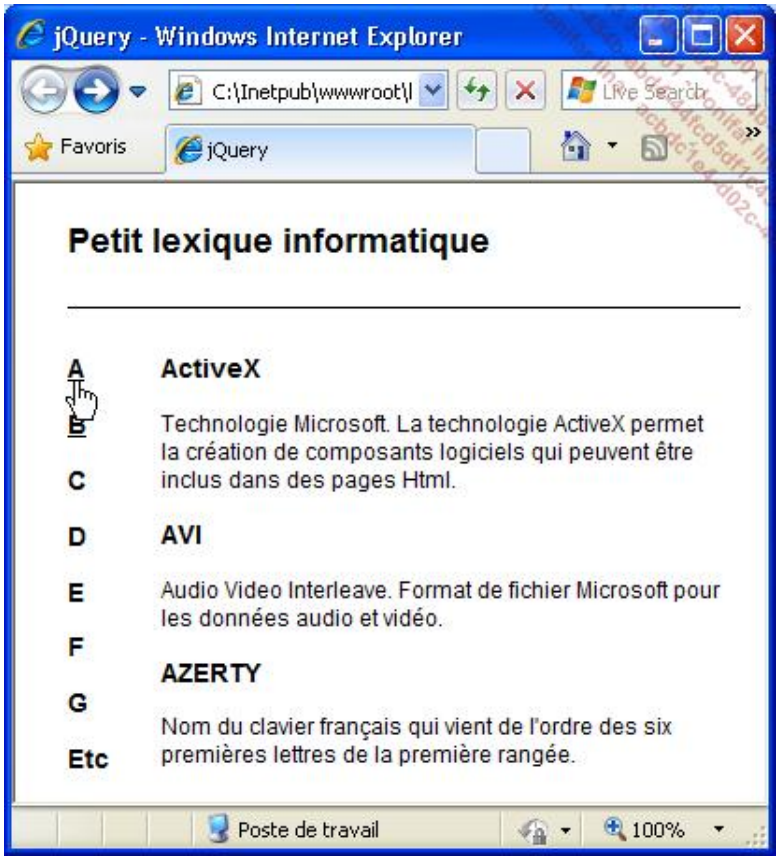

© ENI Editions - All rigths reserved - Jonifar lina

```
<div>
Big Blue est le surnom d'IBM, par référence à Big Brother, qui est
le personnage inquiétant d'Orwell dans « 1984 ».
</div>
<h3>Bureautique</h3>
<div>
Mot apparu en 1976. Application de l'informatique au travail
de bureau (tertiaire). Couramment, il s'agit du trio tableur -
traitement de texte - base de données.
</div>
```

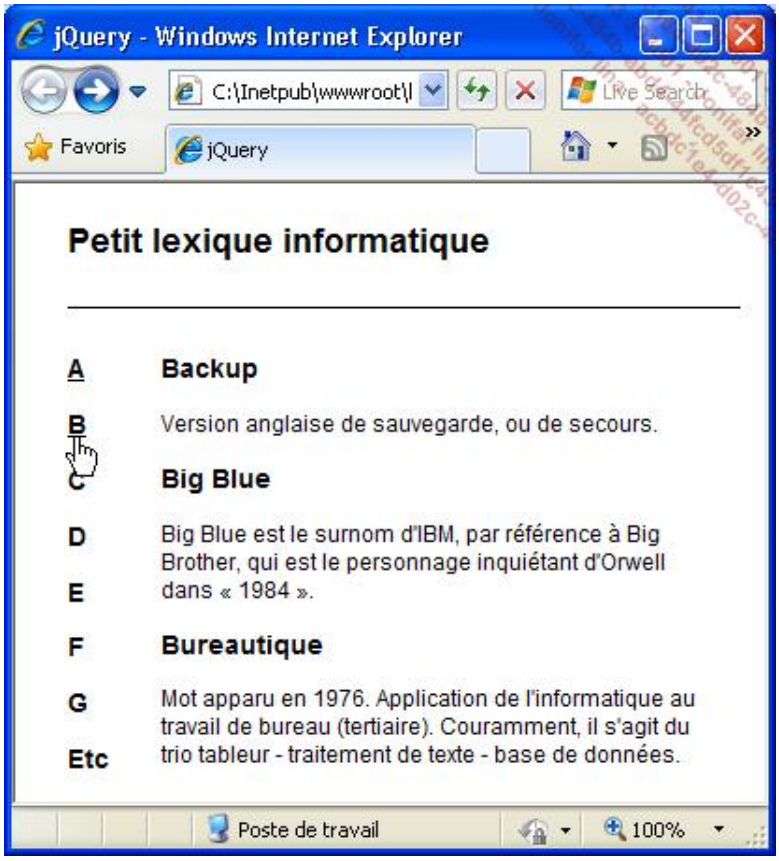
Ces fichiers sont disponibles dans l'espace de téléchargement réservé à cet ouvrage.

Lorsque les fichiers comportent des accents, il est recommandé de les enregistrer au format UTF-8 ou iso-8859-1.

Au clic sur la lettre A ou B, une requête AJAX est diligentée afin de charger les définitions correspondantes (lettre_a.htm ou lettre_b.htm) dans la division `lexique`.



© ENI Editions - All rigths reserved - Jonifar lina



Le script jQuery prend la forme :

```
<script type="text/javascript">
$(document).ready(function() {
$('#lettre_a').click(function() {
$('#lexique').hide().load('lettre_a.htm', function() {
$(this).slideDown(1000);
});
return false;
});
$('#lettre_b').click(function() {
$('#lexique').hide().load('lettre_b.htm', function() {
$(this).slideDown(1000);
});
return false;
});
});
</script>
```

À ce stade de cet ouvrage, les explications peuvent être plus succinctes.

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

Initialisation de jQuery.

```
$('#lettre_a').click(function() {
```

Au clic de la lettre A.

```
$('#lexique').hide().load('lettre_a.htm', function() {
$(this).slideDown(1000);
});
```

Une requête AJAX en jQuery (`load()`) est effectuée vers le fichier lettre_a.htm et son contenu inséré dans la division `lexique`. Un effet visuel a été ajouté au script. Dans un premier temps, le contenu de `lexique` est effacé (`hide()`) et le nouveau contenu apparaît avec un effet de glissement vers le bas (`slideDown(1000)`).

© ENI Editions - All rigths reserved - Jonifar lina

La page finale devient :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function() {
$('#lettre_a').click(function() {
$('#lexique').hide().load('lettre_a.htm', function() {
$(this).slideDown(1000);
});
return false;
});
$('#lettre_b').click(function() {
$('#lexique').hide().load('lettre_b.htm', function() {
$(this).slideDown(1000);
});
return false;
});
});
</script>
<style type="text/css">
body { font: 62.5% Arial, Verdana, sans-serif;}
a { color: black}
#container { font-size: 1.2em;
             margin: 10px 20px;
             width: 360px;}
#header { margin-top: 20px;
          margin-bottom: 10px;
          padding-bottom: 10px;
          border-bottom: 1px solid black;}
#alphabet { float: left;
            width: 30px;
            padding-right: 10px;
            margin-right: 10px; }
#lexique { float: left;
           width: 300px;}
</style>
</head>
<body>
<div id="container">
<div id="header">
<h2>Petit lexique informatique</h2>
</div>
<div id="alphabet">
<div id="lettre_a">
<h3><a href="#">A</a></h3>
</div>
<div id="lettre_b">
<h3><a href="#">B</a></h3>
</div>
<div><h3>C</h3></div>
<div><h3>D</h3></div>
<div><h3>E</h3></div>
<div><h3>F</h3></div>
<div><h3>G</h3></div>
<div><h3>Etc</h3></div>
</div>
<div id="lexique">
</div>
</div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

© ENI Editions - All rigths reserved - Jonifar lina

Introduction

Dès son origine, jQuery se voulait un framework JavaScript complet. Il comporte ainsi une série de fonctions, dites, utilitaires. Il n'est pas possible, dans le cadre de cet ouvrage de les passer toutes en revue. Celles-ci peuvent être consultées dans la documentation de jQuery à l'adresse : http://docs.jquery.com/Utilities

Nous souhaitons cependant en épingler quelques-unes.

© ENI Editions - All rigths reserved - Jonifar lina

Détecter le navigateur

Les différences entre le fonctionnement des navigateurs sont une triste réalité de la conception des sites Web. Il est souvent nécessaire, surtout dans le cadre d'applications pointues, de prévoir des lignes de codes spécifiques à l'un ou l'autre navigateur (Internet Explorer, malgré ses progrès, reste visé). Ainsi, la détection du navigateur reste un outil essentiel du développeur Web.

1. Par le nom du navigateur

jQuery.browser()

Détecte les navigateurs Safari, Opera, Msie et Mozilla.

```
jQuery.browser, function()
```

Commentaire

Au demeurant fort pratique, la méthode `jQuery.browser()` est dépréciée (deprecated) en jQuery 1.3 et ne se retrouvera plus dans les versions futures. Il faut lui préférer la méthode `jQuery.support()` abordée au point suivant de ce chapitre.

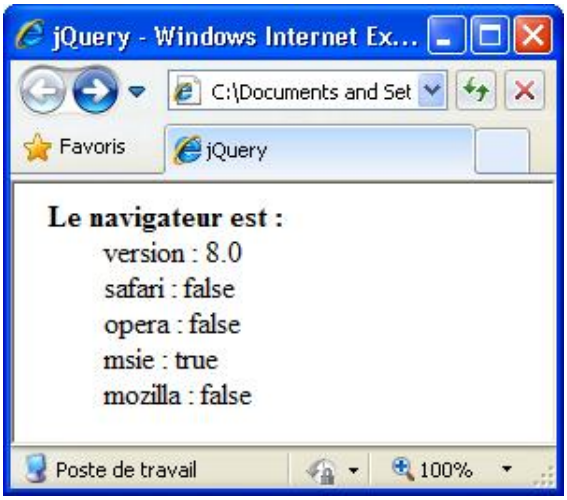
C'est en fait une interprétation du classique navigator.userAgent de JavaScript.

Exemple

Explorons divers navigateurs avec la méthode `jQuery.browser()`.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$.each(jQuery.browser, function(i, val) {
$("<div>" + i + " : <span>" + val + "</span>")
.appendTo(document.body);
});
});
//]]>
</script>
<style type="text/css">
p { font-weight: bold;
    margin: 3px 0 0 10px; }
div { margin-left:40px; }
</style>
</head>
<body>
<p>Le navigateur est :</p>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina



2. Par les capacités du navigateur

Avec la relance des navigateurs et les nouvelles versions qui se succèdent à un rythme assez élevé, il devient un véritable casse-tête de savoir si la version x du navigateur y gère telle propriété. Avec la méthode `jQuery.support()`, la démarche est plus pragmatique. En effet, elle détecte si une propriété spécifique est reconnue ou non par le navigateur de l'utilisateur final. Par exemple, détecter si le (célèbre) modèle de boîte du W3C est appliqué ou ignoré.

jQuery.support

Ajouté en jQuery 1.3. Teste la présence ou l'absence d'une série de propriétés (voir documentation officielle).

`jQuery.support.boxModel` : teste si le modèle de boîte est appliqué.

Exemple

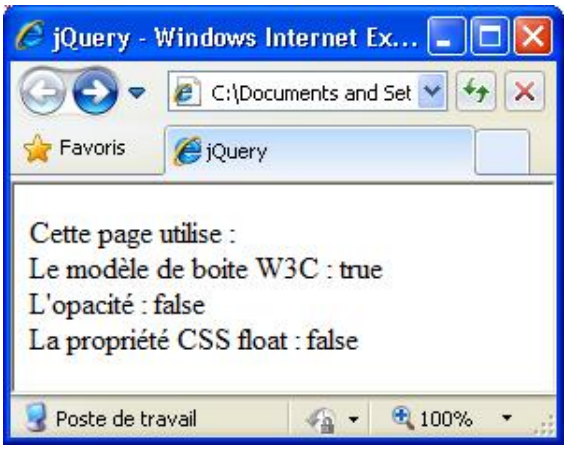
Testons la prise en charge du modèle de boîte, de l'opacité et de la propriété CSS float du navigateur.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
//<![CDATA[
$(document).ready(function(){
$("p").html("Cette page utilise : <br>Le modèle de boite W3C :
<span>" +
```

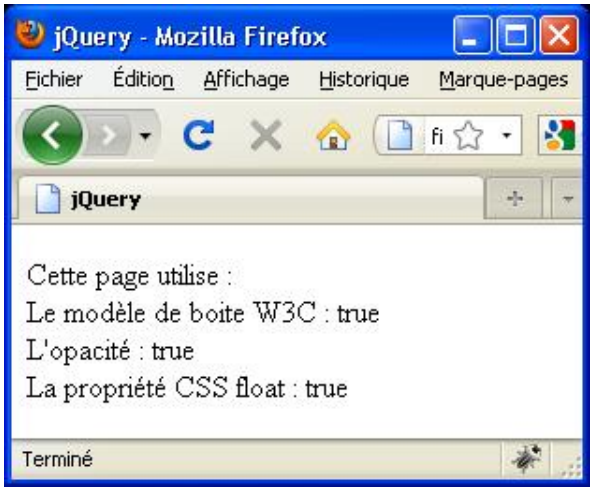

© ENI Editions - All rigths reserved - Jonifar lina

```
jQuery.support.boxModel +
"</span><br>L'opacité : <span>" +
jQuery.support.opacity +
"</span><br>La propriété CSS float : <span>" +
jQuery.support.cssFloat +
"</span>");
});
//]]>
</script>
</head>
<body>
<p></p>
</body>
</html>
```

Internet Explorer 8 reconnaît le modèle de boîte du W3C (grâce à la présence du doctype), ne gère pas la propriété de style opacity (utilise les filtres alpha pour l'opacité) et ne prend pas en compte style.cssFloat (utilise styleFloat).



Firefox 3.5 réussit quant à lui les différents tests.



Éviter les conflits

Les frameworks JavaScript tels que jQuery, Mootools ou Prototype sont fréquemment utilisés pour le développement des applications récentes. Leur cohabitation pose souvent des problèmes car le signe dollar $ est utilisé par chacun d'eux. Pour rappel, jQuery utilise le $ comme alias de "jQuery".

La méthode `jQuery.noConflict()` permet d'éviter les conflits possibles avec les autres frameworks. Ainsi l'appel à $ dans le code du script, ne sera plus considéré comme du jQuery et sera réservé aux autres librairies. Le nommage initial jQuery sera repris pour le code jQuery.

Pour plus de détails, voir : http://docs.jquery.com/Using_jQuery_with_Other_Libraries

Exemple

Soit deux divisions. Le contenu de l'une est géré par jQuery et l'autre par Prototype.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery et Protoype</title>
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript" src="jquery.js"></script>
<style type="text/css">
div { width: 160px;
      height: 30px;
      border: 1px solid black;
      margin-bottom: 15px;
      padding-left: 5px;}
.jquery { background-color: #9cf;
          margin-top: 15px;}
.prototype { background-color: white;
             font-style: italic; }
</style>
</head>
<body>
<div id="jquery">Produit en jQuery</div>
<div id="prototype">Produit par Prototype</div>
<script type="text/javascript">
// Partie jQuery
jQuery.noConflict();
jQuery('#jquery').addClass('jquery');
// Partie Prototype
$('prototype').addClassName('prototype');
</script>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Explications du script.

```
// Partie jQuery
jQuery.noConflict();
```

La méthode `jQuery.noConflict()` indique à jQuery de ne plus prendre en compte les variables `$`.

```
jQuery('#jquery').addClass('jquery');
```

Le signe `$` est remplacé par jQuery.

```
// Partie Prototype
$('prototype').addClassName('prototype');
```

Instruction gérée par Prototype. Le signe `$` n'est plus considéré comme du jQuery grâce à l'instruction `jQuery.noConflict()`.

© ENI Editions - All rigths reserved - Jonifar lina

Itérations en jQuery

La boucle for en JavaScript classique n'est jamais anodine à programmer (par exemple, `for (i=1; i<6; i++)`). Il faut préciser le nom de la variable du compteur ainsi que sa valeur initiale, la condition qui fixe la limite de la boucle et enfin une instruction qui incrémente (ou décrémente) le compteur.

Pour ce faire, jQuery propose la méthode `each()`.

each(fonction)

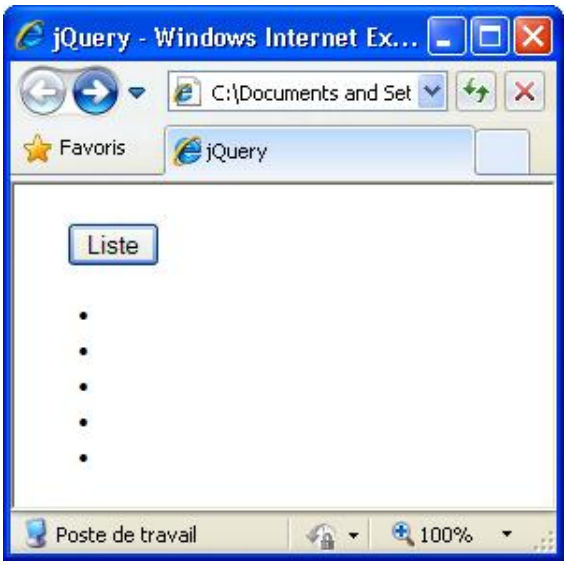
Exécute la fonction passée en paramètre à chaque occurrence de l'objet sélectionné.

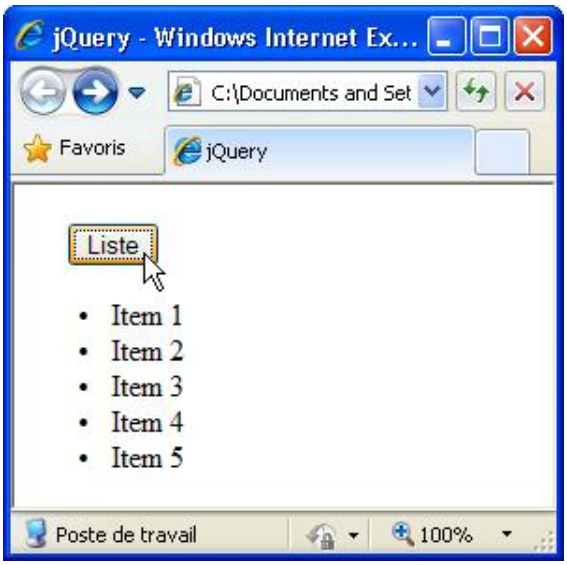
La fonction doit disposer elle-même d'un argument (un entier) qui représentera la position de l'élément en cours de traitement.

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

Exemple

Au clic d'un bouton, le script remplit les éléments d'une liste numérotée.






```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
```

© ENI Editions - All rigths reserved - Jonifar lina

```
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(  function() {
$("button").click(function() {
$("#liste").find("li").each( function(i) {
$(this).html( $(this).html() + " Item " + (i+1) );
});
});
});
</script>
<style type="text/css">
.bouton { margin-top: 12px;
          margin-left: 20px;}
</style>
</head>
<body>
<button class="bouton">Liste</button>
<ul id="liste">
<li></li>
<li></li>
<li></li>
<li></li>
<li></li>
</ul>
</body>
</html>
```

Détaillons le script.

```
$(document).ready( function() {
$("button").click(function() {
```

Au démarrage du DOM et au clic du bouton.

```
$("#liste").find("li").each( function(i) {
$(this).html( $(this).html() + " Item " + (i+1) );
});
```

Le script boucle sur chaque élément (each) de la liste (`$("#liste").find("li")`). Remarquons l'argument de la fonction associée à `each() (function(i))`. À chaque élément `<li>` trouvé, la fonction ajoute la mention "Item" et la valeur de `i` augmentée de 1.

```
});
});
```

Fin de script.

© ENI Editions - All rigths reserved - Jonifar lina

Introduction

Même si jQuery n’apporte rien de spécifiquement nouveau en la matière, nous consacrons un chapitre particulier aux formulaires car ils représentent toujours un développement particulier dans l’élaboration des pages Web.

© ENI Editions - All rigths reserved - Jonifar lina

Les sélecteurs de formulaires

:input

Sélectionne tous les éléments de formulaires (input, textarea, select et boutons).

:text

Sélectionne tous les éléments de formulaires de type ligne de texte.

:password

Sélectionne tous les éléments de formulaires de type mot de passe.

:radio

Sélectionne tous les éléments de formulaires de type boutons radio.

:checkbox

Sélectionne tous les éléments de formulaires de type boutons de cases à cocher.

:submit

Sélectionne tous les éléments de formulaires de type submit.

:image

Sélectionne tous les éléments de formulaires de type image.

:reset

Sélectionne tous les éléments de formulaires de type reset.

:button

Sélectionne toutes les balises `<button>` et tous les éléments de formulaires de type button.

:file

Sélectionne tous les éléments de formulaires de type fichier.

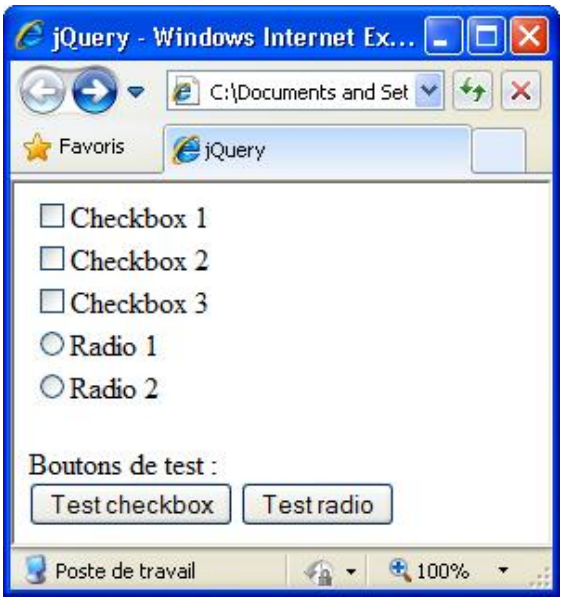
:hidden

Sélectionne tous les éléments de formulaires de type hidden.

Exemple

Soit des cases à cocher et des boutons radio. Au clic d'un bouton, retrouvons les premières et au clic d'un autre bouton, ces derniers.

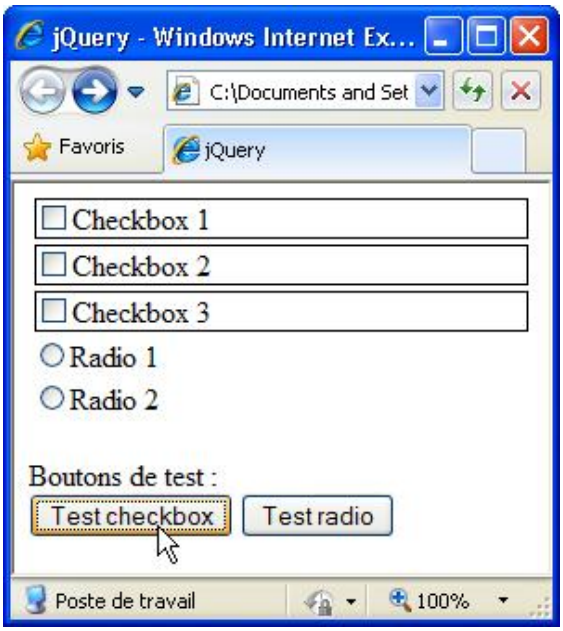
© ENI Editions - All rigths reserved - Jonifar lina

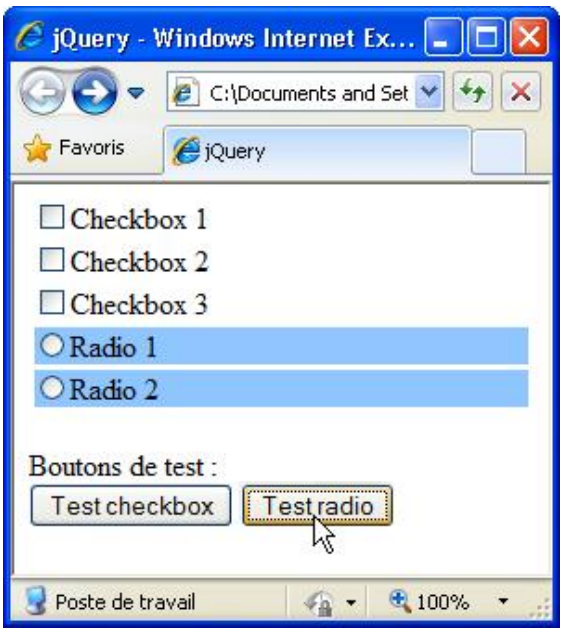


Le fichier de départ :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
p { margin: 3px;}
</style>
</head>
<body>
<form action="">
<p><input type="checkbox" />Checkbox 1</p>
<p><input type="checkbox" />Checkbox 2</p>
<p><input type="checkbox" />Checkbox 3</p>
<p><input type="radio" />Radio 1<br /></p>
<p><input type="radio" />Radio 2</p>
<br />
Boutons de test :<br />
<button id="test1">Test checkbox</button>
<button id="test2">Test radio</button>
</form>
</body>
</html>
```


© ENI Editions - All rigths reserved - Jonifar lina





Le script jQuery :

```
<script type="text/javascript">
$(document).ready(function(){
$("#test1").toggle(
function () {
$("form :checkbox").parent().css({"border": "1px solid black"});
},
function () {
$("form :checkbox").parent().css({"border": "none"});
});
$("#test2").toggle(
function () {
$("form :radio").parent().css({"background": "#9cf"});
},
function () {
$("form :radio").parent().css({"background": "none"});
});
});
</script>
```

Détaillons le script.

© ENI Editions - All rigths reserved - Jonifar lina

```
$(document).ready(function(){
$("#test1").toggle(
```

Après chargement du DOM, le script introduit un effet de permutation (`toggle`) sur le premier bouton (`#test1`).

```
function () {
$("form :checkbox").parent().css({"border": "1px solid black"});
},
```

Au premier clic, le script retrouve les cases à cocher (`form :checkbox`). Puis remonte au parent direct (`parent`) soit les balises `<span>` qui entourent les champs de formulaires. Enfin, modifie la propriété de style (`css`) consistant à mettre une bordure.

```
function () {
$("form :checkbox").parent().css({"border": "none"});
});
```

Au second clic, le script remet le tout dans son état initial.

```
$("#test2").toggle(
function () {
$("form :radio").parent().css({"background": "#9cf"});
},
function () {
$("form :radio").parent().css({"background": "none"});
});
```

Le même procédé est appliqué pour les boutons radio (`form :radio`).

```
});
```

Fin du ready.

Le fichier final :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#test1").toggle(
function () {
$("form :checkbox").parent().css({"border": "1px solid black"});
},
function () {
$("form :checkbox").parent().css({"border": "none"});
});
$("#test2").toggle(
function () {
$("form :radio").parent().css({"background": "#9cf"});
},
function () {
$("form :radio").parent().css({"background": "none"});
});
});
</script>
<style type="text/css">
p { margin: 3px;}
</style>
</head>
<body>
<form action="">
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<p><input type="checkbox" />Checkbox 1</p>
<p><input type="checkbox" />Checkbox 2</p>
<p><input type="checkbox" />Checkbox 3</p>
<p><input type="radio" />Radio 1<br /></p>
<p><input type="radio" />Radio 2</p>
<br />
Boutons de test :<br />
<button id="test1">Test checkbox</button>
<button id="test2">Test radio</button>
</form>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Les filtres de sélection

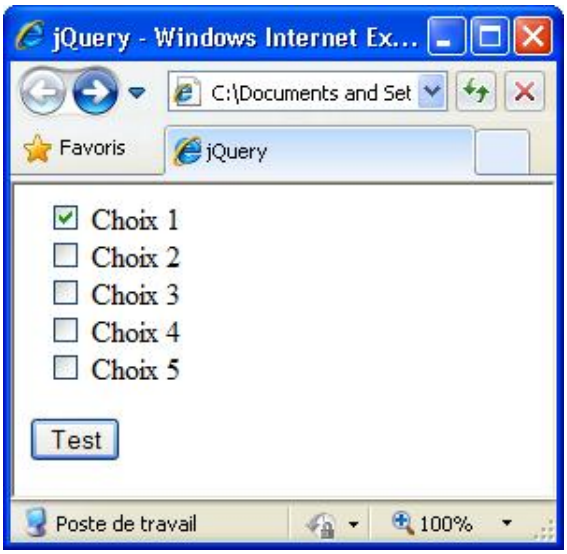
1. Les éléments cochés

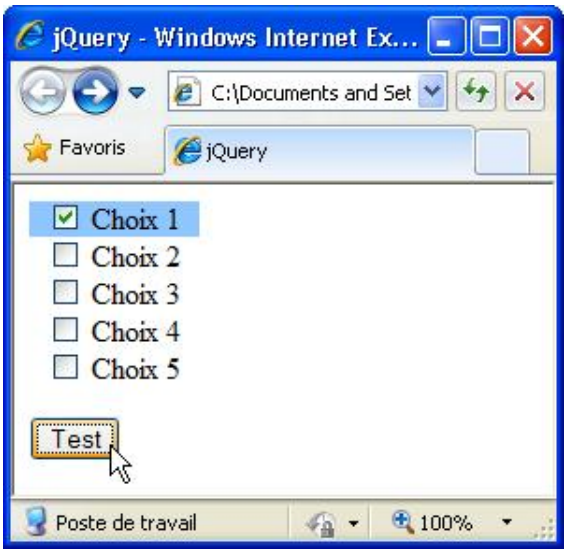
:checked

Reprend tous les éléments de bouton radio ou de cases à cocher sélectionnés.

Exemple

Retrouvons au clic du bouton, les éléments sélectionnés des cases à cocher.





Le fichier de départ.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
span { padding:  0 10px 0 10px;}
```

© ENI Editions - All rigths reserved - Jonifar lina

```
</style>
</head>
<body>
<form action="">
<span><input type="checkbox" name="box" checked="checked" /> Choix
1</span><br />
<span><input type="checkbox" name="box" /> Choix 2</span><br />
<span><input type="checkbox" name="box" /> Choix 3</span><br />
<span><input type="checkbox" name="box" /> Choix 4</span><br />
<span><input type="checkbox" name="box" /> Choix 5</span><br />
</form>
<p><button>Test</button></p>
</body>
</html>
```

Le script jQuery.

```
<script type="text/javascript">
$(document).ready(function(){
$(":button").toggle( function () {
$("input:checked").parent().css({"background": "#9cf"});
},
function () {
$("input:checked").parent().css({"background": "none"});
});
});
</script>
```

Le script reprend la démarche du précédent. Il ne nécessite donc pas de plus amples explications. Notons simplement `input:checked` qui permet de ne reprendre que les champs de formulaires qui sont cochés.

Au final :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$(":button").toggle( function () {
$("input:checked").parent().css({"background": "#9cf"});
},
function () {
$("input:checked").parent().css({"background": "none"});
});
});
</script>
<style type="text/css">
span { padding:  0 10px 0 10px;}
</style>
</head>
<body>
<form action="">
<span><input type="checkbox" name="box" checked="checked" /> Choix
1</span><br />
<span><input type="checkbox" name="box" /> Choix 2</span><br />
<span><input type="checkbox" name="box" /> Choix 3</span><br />
<span><input type="checkbox" name="box" /> Choix 4</span><br />
<span><input type="checkbox" name="box" /> Choix 5</span><br />
</form>
<p><button>Test</button></p>
</body>
</html>
```

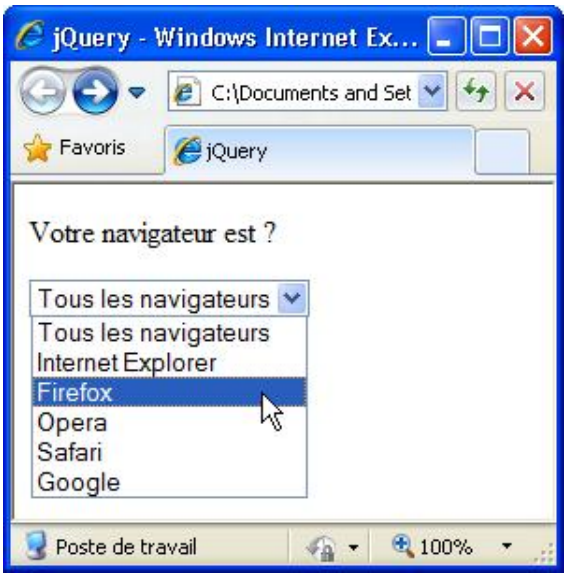

© ENI Editions - All rigths reserved - Jonifar lina

2. Les éléments sélectionnés

:selected

Reprend tous les éléments d'une liste de choix qui sont sélectionnés.

Exemple



Le fichier html initial :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
</head>
<body>
<p>Votre navigateur est ?</p>
<form action="">
<select id="navigateurs">
<option selected="selected" value="0">Tous les
```

© ENI Editions - All rigths reserved - Jonifar lina

```
navigateurs</option>
<option value="1">Internet Explorer</option>
<option value="2">Firefox</option>
<option value="3">Opera</option>
<option value="4">Safari</option>
<option value="5">Google</option>
</select>
</form>
<br />
<div id="resultat"></div>
</body>
</html>
```

Le script se présente ainsi.

```
<script type="text/javascript">
$(document).ready(function(){
$("#navigateurs").change(onSelectChange);
});
function onSelectChange(){
var selected = $("#navigateurs option:selected");
var output = "";
if(selected.val() != 0){
output = "Votre sélection : " + selected.text();
}
$("#resultat").html(output);
}
</script>
```

Explorons-le.

```
$(document).ready(function(){
$("#navigateurs").change(onSelectChange);
});
```

Lorsqu'un changement est effectué dans la liste de choix (`change`), la fonction `onSelectChange` est appelée.

```
function onSelectChange(){
var selected = $("#navigateurs option:selected");
var output = "";
```

La fonction `onSelectChange` définit d'abord la variable `selected` comme étant le choix retenu par l'utilisateur dans la liste (`$("#navigateurs option:selected")`). La variable `output` qui contiendra l'énoncé du choix de l'utilisateur est initialisée.

```
if(selected.val() != 0){
output = "Votre sélection : " + selected.text();
}
```

Si l'attribut valeur du choix effectué est différent de 0 (`(selected.val() != 0)`), la variable `output` est élaborée compte tenu du choix effectué.

```
$("#resultat").html(output);
}
```

La valeur de la variable `output` est affichée dans la division identifiée par resultat.

```
}
```

Fin du script.

Le fichier complet :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
```


© ENI Editions - All rigths reserved - Jonifar lina

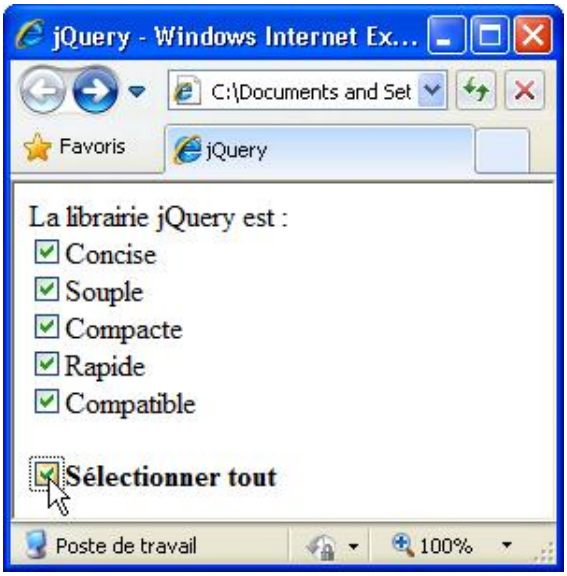
```
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#navigateurs").change(onSelectChange);
});
function onSelectChange(){
var selected = $("#navigateurs option:selected");
var output = "";
if(selected.val() != 0){
output = "Votre sélection : " + selected.text();
}
$("#resultat").html(output);
}
</script>
</head>
<body>
<p>Votre navigateur est ?</p>
<form action="">
<select id="navigateurs">
<option selected="selected" value="0">Tous les
navigateurs</option>
<option value="1">Internet Explorer</option>
<option value="2">Firefox</option>
<option value="3">Opera</option>
<option value="4">Safari</option>
<option value="5">Google</option>
</select>
</form>
<br />
<div id="resultat"></div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina

Applications

1. Sélectionner toutes les cases à cocher

Très en vogue sur les sites récents, ce script offre la possibilité à l'utilisateur d'activer toutes les cases à cocher en une seule action.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#tous").click(function(){
var checked_status = this.checked;
$("input[name=case]").each(function(){
this.checked = checked_status;
```

© ENI Editions - All rigths reserved - Jonifar lina

```
});
});
});
</script>
</head>
<body>
La librairie jQuery est :<br />
<input type="checkbox" name="case" />Concise<br />
<input type="checkbox" name="case" />Souple<br />
<input type="checkbox" name="case" />Compacte<br />
<input type="checkbox" name="case" />Rapide<br />
<input type="checkbox" name="case" />Compatible<br />
<br />
<input type="checkbox" id="tous" /><b>Sélectionner tout</b>
</body>
</html>
```

Le script jQuery :

```
<script type="text/javascript">
$(document).ready(function(){
$("#tous").click(function(){
var checked_status = this.checked;
$("input[name=case]").each(function(){
this.checked = checked_status;
});
});
});
</script>
```

Explications :

```
$(document).ready(function(){
$("#tous").click(function(){
```

Initialisation de jQuery et au clic de la case à cocher permettant de sélectionner tous les éléments.

```
var checked_status = this.checked;
```

La variable `checked_status` note que la case est cochée.

```
$("input[name=case]").each(function(){
this.checked = checked_status;
});
```

La méthode `each()` passe en revue toutes les cases à cocher (`input[name=case]`) et coche celles-ci.

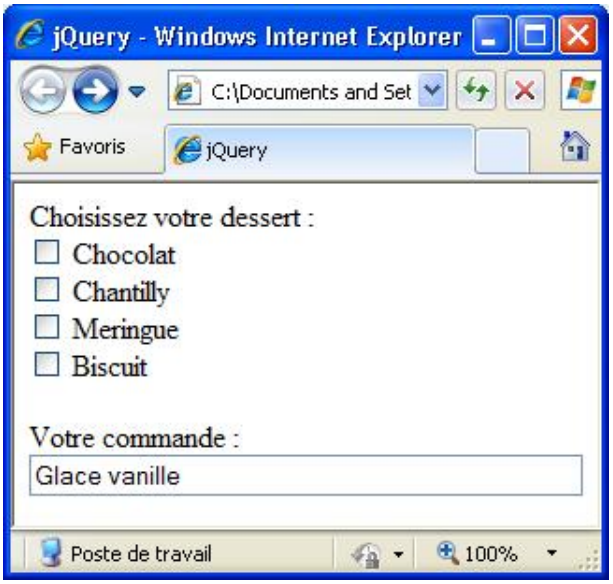
```
});
});
```

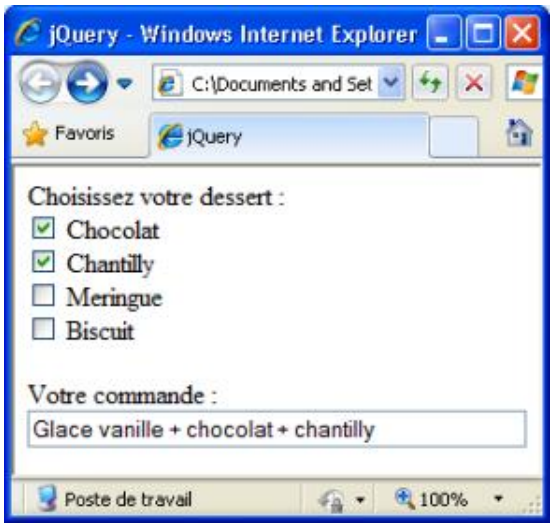
Fin de script.

2. Confirmer une commande

Reprenons dans une ligne de texte, les éléments cochés d'un formulaire.

© ENI Editions - All rigths reserved - Jonifar lina






```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
//<![CDATA[
$(document).ready(function() {
$('input:checkbox[name=dessert]').click(function(){
var check = this.checked;
var val = $(this).val();
var keys = $("input[name='commande']").val().split(', ');
var newKeys = new Array();
var inArray = false;
$.each(keys, function(i){
var key=this;
if (key == val) {
if (check){
newKeys.push(key);
}
inArray = true;
}
else {
```

© ENI Editions - All rigths reserved - Jonifar lina

```
newKeys.push(key);
}
});
if (!inArray && check) newKeys.push(val);
$("input[name='commande']").val(newKeys.join(' + '));
});
});
//]]>
</script>
</head>
<body>
Choisissez votre dessert :
<form action="">
<input value="chocolat" type="checkbox" name="dessert" />
Chocolat<br />
<input value="chantilly" type="checkbox" name="dessert" />
Chantilly<br />
<input value="meringue" type="checkbox" name="dessert" />
Meringue<br />
<input value="biscuit" type="checkbox" name="dessert" />
Biscuit<br />
<br />
Votre commande : <br />
<input type="text" name="commande" size="45" value="Glace vanille" />
</form>
</body>
</html>
```

Explications du script

```
$(document).ready(function() {
$('input:checkbox[name=dessert]').click(function(){
```

Initialisation de jQuery et au clic d'une case à cocher.

```
var check = this.checked;
var val = $(this).val();
```

La variable `check` note que la case est cochée. La variable `val` récupère la valeur correspondante de la case.

```
var keys = $("input[name='commande']").val().split(', ');
```

Préparation du contenu de la ligne de texte qui confirme la commande en reprenant la valeur initiale de l'attribut value.

```
var newKeys = new Array();
var inArray = false;
```

Création d'un tableau (Array) appelé `newKeys` destiné à contenir les intitulés des cases cochées.

```
$.each(keys, function(i){
var key=this;
if (key == val) {
if (check){
newKeys.push(key);
}
inArray = true;
}
else {
newKeys.push(key);
}
});
```

Par la méthode JavaScript classique `Array.push()`, les intitulés des cases cochées sont ajoutés à la fin du tableau `newkeys`.

```
if (!inArray && check) newKeys.push(val);
```


© ENI Editions - All rigths reserved - Jonifar lina

```
$("input[name='commande']").val(newKeys.join(' + '));
```

La confirmation de la commande s'affiche dans la ligne de texte par la modification de son attribut value.

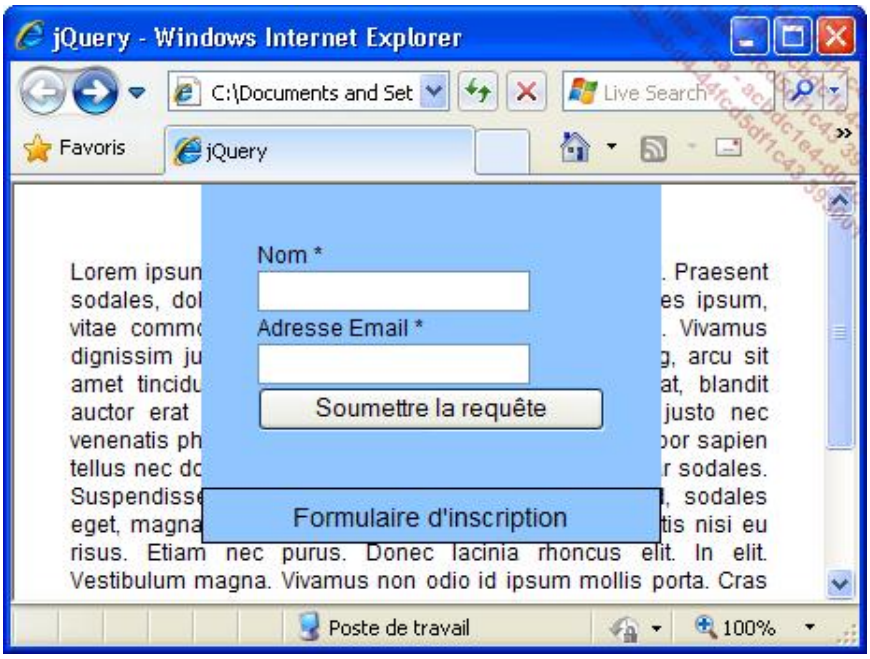
```
});
});
```

Fin de script.

3. Un formulaire d'inscription original

Les formulaires de contact ou d'inscription sont une partie indispensable de tout site Web. Ils sont souvent implémentés dans une page séparée et font rarement preuve de créativité. Cet exemple illustre comment créer avec jQuery, un formulaire disponible dès la page d'accueil.

Au clic de la division **Formulaire d'inscription**, le formulaire de contact apparaît en glissant verticalement vers le bas.




```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<style type="text/css">
```

© ENI Editions - All rigths reserved - Jonifar lina

```
body{ margin: 0px;
      font-family: Arial, Sans-Serif;
      font-size: 0.75em;}
.box { margin: 0px auto;
       background-color: #ffffff;
       text-align: justify;
       position: relative;}
.content { padding: 20px 30px;}
#container { position: absolute;
             left: 100px;
             float: right;}
#contactForm { height: 117px;
               width: 245px;
               background-color: #9cf;
               display: none;}
#contactForm fieldset { padding: 30px;
                        border: none; }
#contactForm label { display: block;
                     color: black;}
#contactForm input[type=text] { display: block;
                                border: solid 1px #4d3a24;
                                margin-bottom: 10px;
                                height: 24px;}
#contactForm input[type=submit] { background-color: #4d3a24;
                                  border: solid 1px #23150c;
                                  color: #fecd28;
                                  padding: 5px;}
#contact { height: 30px;
           width: 246px;
           background-color: #9cf;
           border: 1px solid black;
           display: block;
           cursor: pointer;
           font-size: 14px;
           padding-top: 7px;
           text-align: center;}
</style>
<script src="jquery.js" type="text/javascript"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#contact").click(function(){
if ($("#contactForm").is(":hidden")){
$("#contactForm").slideDown("slow");
}
else{
$("#contactForm").slideUp("slow");
}
});
});
function closeForm(){
$("#contactForm").slideUp("slow");
}
</script>
</head>
<body>
<div class="box">
<div id="container">
<div id="contactForm">
<fieldset>
<label for="Name">Nom *</label>
<input id="name" type="text" />
<label for="Email">Adresse Email *</label>
<input id="Email" type="text" />
<input id="envoi" type="submit" name="submit"
onclick="closeForm()" />
</fieldset>
</div>
<div id="contact">Formulaire d’inscription</div>
</div>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<div class="content">
<p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
Praesent sodales, dolor ut tempor sollicitudin, nulla lacus
sodales ipsum, vitae commodo dui nisi et sapien. Phasellus ac
nisl. Vivamus dignissim justo iaculis mauris. Suspendisse
adipiscing, arcu sit amet tincidunt pretium, lorem mauris
ullamcorper erat, blandit auctor erat massa sit amet neque.
...
</p>
</div>
</div>
</body>
</html>
```

Le script jQuery est (à ce stade de votre étude) relativement simple.

```
$(document).ready(function(){
$("#contact").click(function(){
```

Au chargement du DOM et au clic de la division identifiée par `contact`.

```
if ($("#contactForm").is(":hidden")){
$("#contactForm").slideDown("slow");}
```

Si le formulaire de contact (`contactForm`) est caché (`is(":hidden")`), celui-ci apparaît en glissant lentement vers le bas (`slideDown("slow")`).

```
else{
$("#contactForm").slideUp("slow");
}
```

Sinon, il remonte (`slideUp("slow")`) pour trouver sa position initiale.

```
});
});
```

Fin du ready.

```
function closeForm(){
$("#contactForm").slideUp("slow");
}
```

Reste à définir la fonction (`closeForm()`) associée au bouton de soumission. Celle-ci fait remonter le formulaire (`slideUp("slow")`) à sa position initiale.

© ENI Editions - All rigths reserved - Jonifar lina

Introduction

Les plug-ins jQuery sont des scripts dédiés à des tâches spécifiques comme, par exemple, le tri d'un tableau, l'implémentation d'un carrousel d'images ou la validation de formulaires.

Ces plug-ins, initiés par la communauté jQuery, sont nombreux, généralement d'excellentes qualités et souvent librement disponibles sur la toile. Un aperçu de leur variété peut être effectué à l'adresse : http://plugins.jquery.com/ ou http://www.julien-verkest.fr/22/11/2007/240-plugins-jquery/

Ces plug-ins génèrent un gain de temps précieux en évitant de devoir réécrire un code qui a déjà été traité par ailleurs. Ils permettent parfois aussi, avouons-le, de se lancer dans des opérations qui dépassent le niveau de programmation du concepteur moyen en jQuery.

© ENI Editions - All rigths reserved - Jonifar lina

Concevoir un plug-in jQuery

1. Aspects théoriques

L'écriture d'un plug-in jQuery passe par des étapes bien précises et certaines règles bien précises sont à respecter.

Le plug-in jQuery prend la forme d'un fichier JavaScript externe (extension .js) qui se place immédiatement après la balise d'appel de jQuery (voir la section Utiliser un plug-in jQuery du présent chapitre) soit :

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

Nommer le plug-in

Un plug-in doit toujours se nommer sous la forme jquery.nom_du_plug-in.js. Il sera ainsi immédiatement identifiable.

Isoler le code

Dans ce fichier js nouvellement créé, il faut englober le code du plug-in dans une fonction anonyme. De cette manière, toutes les variables du plug-in ne rentreront pas en conflit avec les autres scripts de la page.

```
(function () {
// code jQuery
}) ()
```

Profitons pour passer la variable jQuery par son alias `$` à cette fonction, ce qui permettra d'utiliser la variable `$` à l'intérieur de celle-ci.

```
(function ($) {
// code jQuery
}) ()
```

Ajouter la nouvelle méthode à jQuery

Nous sommes presque prêts à écrire notre plug-in. Il faut encore ajouter cette nouvelle méthode aux objets jQuery par l'instruction `$.fn.nom_du_plug-in`. Dans cette fonction le mot-clé `this` représentera l'objet jQuery sélectionné par l'utilisateur du plug-in.

```
(function($) {
$.fn.nom_du_plugin = function () {
// code jQuery
};
}) (jQuery)
```

La documentation de jQuery concernant la conception de plug-ins insiste sur le fait que chaque méthode ou fonction doit bien se terminer par le point-virgule. Sinon, le code risque de ne pas fonctionner après compression du plug-in.

Les règles essentielles de l'élaboration d'un plug-in sont ainsi passées en revue. Cependant, il reste des points à ne pas perdre de vue :

- Lorsque plusieurs éléments sont ou risquent d'être sélectionnés, la méthode `each()` est recommandée pour boucler sur les éléments sélectionnés.
- Il est utile de passer des paramètres à la méthode du plug-in pour faciliter la personnalisation de celui-ci par l'utilisateur.
- Sauf exception, le plug-in doit retourner l'objet traité (return this).

Le fin du fin, quoique souvent pas indispensable, est de compresser le code pour réduire le temps de téléchargement du plug-in. Il existe une petite application en ligne http://dean.edwards.name/packer/ qui réalise ceci à merveille. Effectuez un copier/coller de votre code dans le champ de formulaire, cochez **Base62 encode** et/ou **Shrink variables** et cliquez sur le bouton **Pack**. Et le résultat s'affiche.

Le code obtenu commence comme celui-ci :

© ENI Editions - All rigths reserved - Jonifar lina

```
eval(function(p,a,c,k,e,r){e=function(c){return(c
```

2. Une application pratique

Le plug-in de notre exemple ajoute une infobulle explicative sur des liens.

La première étape consiste à nommer le plug-in. Appelons-le, suivant les conventions de jQuery, `jquery.mon_infobulle.js`.

Après avoir ouvert un éditeur de texte, l'encodage de celui-ci peut débuter.

Commençons par déterminer l'environnement jQuery.

```
(function($){
jQuery.fn.mon_infobulle = function(options) {
// code du plug-in
};
})(jQuery)
```

Incluons maintenant le code du plug-in proprement dit.

```
(function($){
jQuery.fn.mon_infobulle = function(options) {
var element = document.createElement("div");
$(element).addClass(options.infobullecss).hide();
document.body.appendChild(element);
return this.each(function() {
$(this).hover(function() {
$(element).show();
$(element).html($(this).attr("rel"));
$(this).mousemove(function(e) {
$(element).css({ "position": "absolute",
                 "top": e.pageY + options.offsetY,
                 "left": e.pageX + options.offsetX
});
});
}, function() {
$(element).hide()
});
});
};
})(jQuery)
```

Le script crée une division `<div>` (celle de l'infobulle), à laquelle il adjoint la classe de style `infobullecss`. Cette division est ajoutée au corps (`body`) du document. Au survol du lien par le curseur de la souris, cette division est affichée. Le contenu de celle-ci reprend les éléments de l'attribut `rel` du lien. Il est prudent de prévoir un léger décalage horizontal et vertical de l'infobulle par rapport au lien (left et top).

Trois éléments doivent donc être paramétrés :

© ENI Editions - All rigths reserved - Jonifar lina

- La classe qui prend en charge l'aspect de l'infobulle (`options.infobullecss`).
- Le décalage vertical de l'infobulle par rapport au lien (`options.offsetY`).
- Le décalage horizontal de l'infobulle par rapport au lien (`options.offsetX`).

Et voilà notre plug-in prêt à l'emploi !

© ENI Editions - All rigths reserved - Jonifar lina

Utiliser un plug-in jQuery

L'utilisation d'un plug-in est des plus simples. Il suffit de l'appeler (par son nom) dans le code de la page.

Exemple

Exploitons notre plug-in dans une page avec plusieurs liens.

Partons du document Html initial.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
</head>
<style type=text/css>
.infobulle { width: 150px;
             padding: 3px;
             background-color: #9CF;
             border: black 1px solid;
             color: black;}
a { color: black}
</style>
<body>
<p><a href="#" rel="Explication du lien 1" class="pop">Lien
1</a></p>
<p><a href="#" rel="Explication du lien 2" class="pop">Lien
2</a></p>
<p><a href="#" rel="Explication du lien 3" class="pop">Lien
3</a></p>
</body>
</html>
```

Remarquons :

Que l'attribut `rel` des balises de lien `<a>` contient déjà le texte de l'infobulle.

Que l'aspect de l'infobulle est déjà prévu par la classe `.infobulle`.

Passons au script :

```
<script type="text/javascript">
$(document).ready(function(){
var options = {
offsetX: 30,
offsetY: 5,
infobullecss: "infobulle"
};
$("a.pop").mon_infobulle(options);
});
</script>
```

Quelques explications...

```
var options = {
offsetX: 30,
offsetY: 5,
infobullecss: "infobulle"
};
```

Détermine les paramètres passés au plug-in, soit le décalage vertical, horizontal et le nom de la classe de style de la fenêtre infobulle.

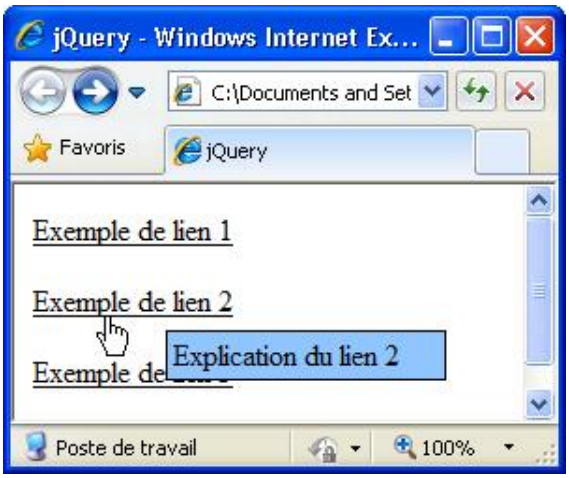
© ENI Editions - All rigths reserved - Jonifar lina

```
$("a.pop").mon_infobulle(options);
```

Pour les liens dotés de la classe pop, le plug-in mon_infobulle est appelé.

Le fichier final complet devient :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript"
src="jquery.mon_infobulle.js"></script>
<script type="text/javascript">
$(document).ready(function(){
var options = {
offsetX: 30,
offsetY: 5,
infobullecss: "infobulle"
};
$("a.info").mon_infobulle(options);
});
</script>
<style type="text/css">
.infobulle { width: 150px;
             padding: 3px;
             background-color: #9CF;
             border: black 1px solid;
             color: black;}
a { color: black}
</style>
</head>
<body>
<p><a href="#" rel="Explication du lien 1" class="info">Lien
1</a></p>
<p><a href="#" rel="Explication du lien 2" class="info">Lien
2</a></p>
<p><a href="#" rel="Explication du lien 3" class="info">Lien
3</a></p>
</body>
</html>
```



© ENI Editions - All rigths reserved - Jonifar lina

Quelques plug-ins

Nous avons retenu quelques exemples de plug-ins. Pour pleinement apprécier le côté spectaculaire de ceux-ci, nous vous invitons plus que jamais, à consulter l'espace de téléchargement dédié à ce livre.

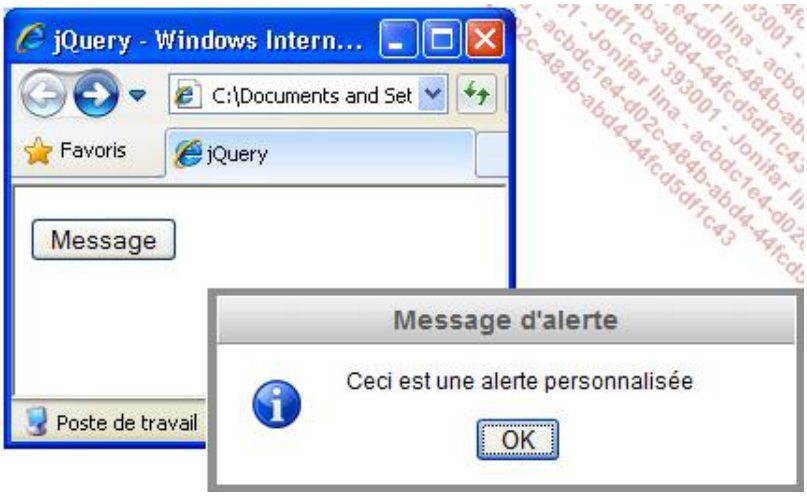
1. De nouveaux messages d'alerte

Il faut bien admettre que l'aspect des boîtes de dialogue n'a pas beaucoup évolué depuis le début du Html et que celui-ci est devenu quelque peu désuet. Les concepteurs peuvent souhaiter en changer l'apparence. C'est ce que propose le plug-in jquery.alerts.js disponible à l'adresse : http://abeautifulsite.net/notebook/87

Après avoir lu la documentation (en anglais), il faut télécharger les fichiers **jquery.alerts.js**, **jquery.alerts.css** ainsi que le dossier **images**. Tout ceci doit alors être inclus dans le dossier contenant la page Html.

Le document Html devient alors :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="jquery.alerts.js"></script>
<link rel="stylesheet" type="text/css" href="jquery.alerts.css" />
<script type="text/javascript">
$(document).ready(function(){
$("#bouton_alerte").click( function() {
jAlert('Ceci est une alerte personnalisée', 'Message d\'alerte');
});
});
</script>
</head>
<body>
<p><input type="button" id="bouton_alerte" value="Message" /></p>
</body>
</html>
```



Quelques commentaires concernant le plug-in et le script.

L'appel du plug-in s'effectue par :

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

Il ne faut pas oublier de relier la page à la feuille de style externe :

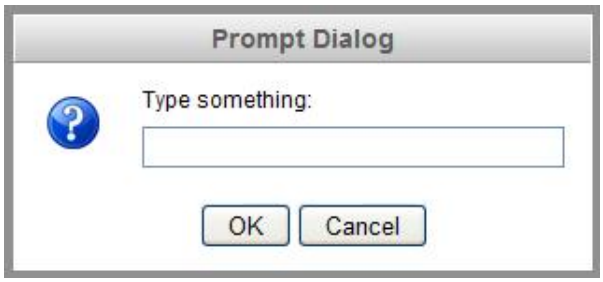
© ENI Editions - All rigths reserved - Jonifar lina

```
<link rel="stylesheet" type="text/css" href="jquery.alerts.css">
```

Le plug-in s'initialise simplement par `jAlert` avec les paramètres souhaités.

```
jAlert('Ceci est une alerte personnalisée', 'Message d\'alerte');
```

Le même plug-in permet de modifier les boîtes de confirmation et d'invite.

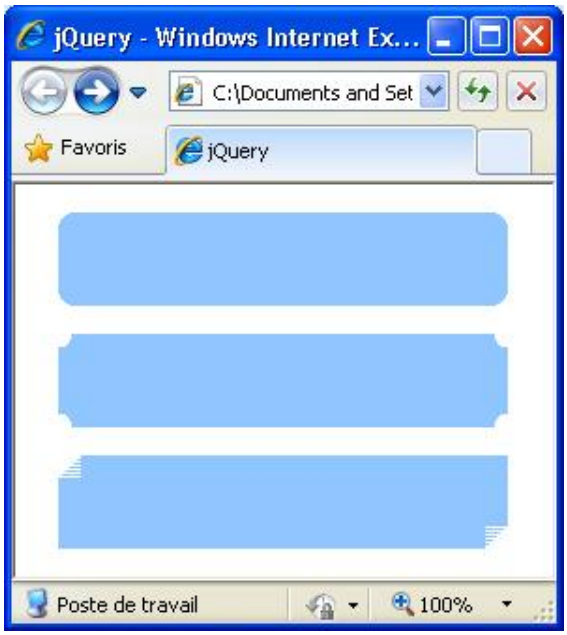


2. Des bords arrondis et variés

L'ajout d'une bordure à une division `<div>` entraîne des bords rectangulaires. Le plug-in `jquery.corner.js` permet de varier aisément leur présentation.

Le plug-in peut être téléchargé à l'adresse http://plugins.jquery.com/project/corners et la documentation est consultable à la page http://www.malsup.com/jquery/corner/.

Exemple




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="jquery.corner.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$('#div1').corner();
$('#div2').corner("bite 7px");
$('#div3').corner("dog tl br 12px");
});
</script>
<style type="text/css">
div.demo { margin:15px;
            width: 220px;
            height: 50px;
            background: #9cf;}
</style>
</head>
<body>
<div id="div1" class="demo"></div>
<div id="div2" class="demo"></div>
<div id="div3" class="demo"></div>
</body>
</html>
```

Notons :

L'appel du plug-in par `<script type="text/javascript" src="jquery.corner.js"> </script>`.

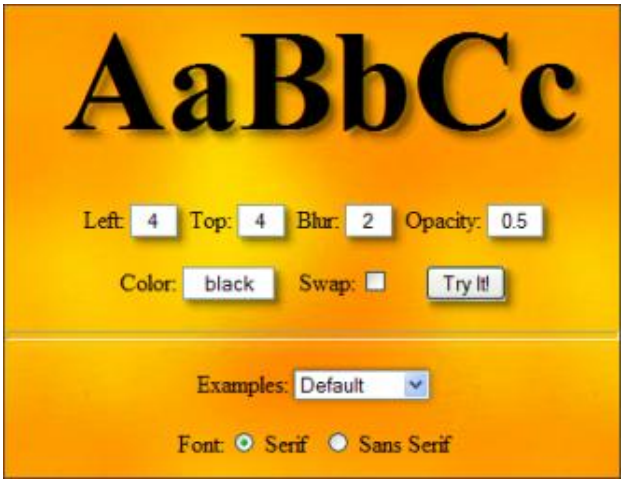
L'implémentation des bords se réalise, après avoir consulté la documentation, par `$('#div1').corner()`, `$('#div2').corner("bite 7px")` et `$('#div3').corner("dog tl br 12px")`.

3. Ajouter une ombre

L'ombrage de texte ou d'autres éléments fait partie intégrante des recommandations CSS3. Cet effet de style est attendu avec impatience par les designers de sites Web pour son apport nouveau dans la présentation des pages.

Cette recommandation commence à être reprise par les navigateurs de la dernière génération mais de Firefox à Opera en passant par Internet Explorer, chacun y va de sa méthode personnelle. Voilà une belle source d'incompatibilité en perspective. Le plug-in jQuery vous propose, d'ores et déjà, une solution directement applicable et surtout compatible pour tous les navigateurs récents.

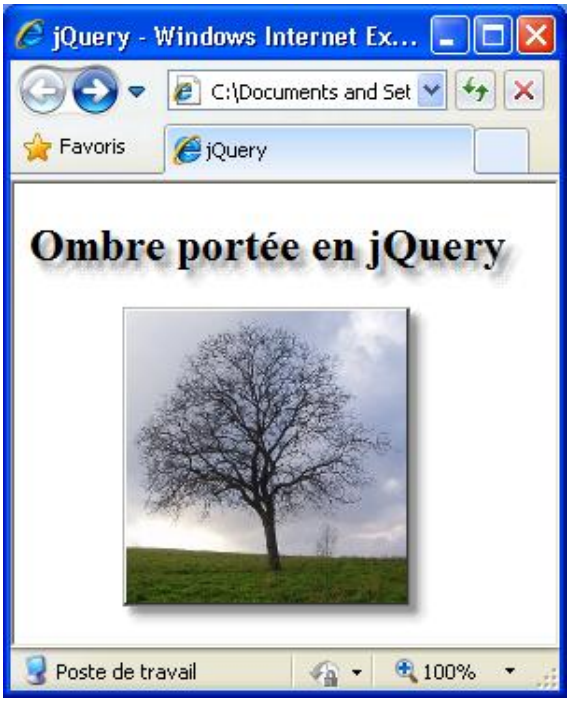
Le script est téléchargeable à l'adresse http://eyebulb.com/dropshadow/. Le site propose un assistant en ligne pour vous permettre de configurer et visualiser une série de paramètres comme le déport vertical et horizontal de l'ombre, sa dispersion et son opacité.



Le plug-in n'est pas très facile à configurer car certains aspects tiennent quelque peu du domaine des trucs et astuces (la balise `<span>` pour le texte, l'image en arrière-plan d'une division).

© ENI Editions - All rigths reserved - Jonifar lina

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"> </script>
<script type="text/javascript" src="jquery.dropshadow.js">
</script>
<script type="text/javascript">
$(document).ready(function(){
$("h2 span").dropShadow();
$("#test").dropShadow({left: 4, top: 4,  blur: 2, opacity: 0.5,
color: "#000"});
});
</script>
<style type="text/css">
body{ font-family: serif;}
#test { border: white 2px outset;
        width: 150px;
        height: 156px;
        background: url(arbre.jpg) no-repeat;
        margin-left : 50px;}
</style>
</head>
<body>
<h2><span>Ombre</span> <span>portée</span> <span>en</span>
<span>jQuery</span></h2>
<div id="test"></div>
</body>
</html>
```



4. Un glisser/déposer (drag and drop)

Très en vogue sur le Web 2.0, le glisser/déposer ou drag and drop est souvent assez complexe à mettre en place en JavaScript classique. Le plug-in easydrag réalise, comme son nom le suggère, cette opération très facilement.

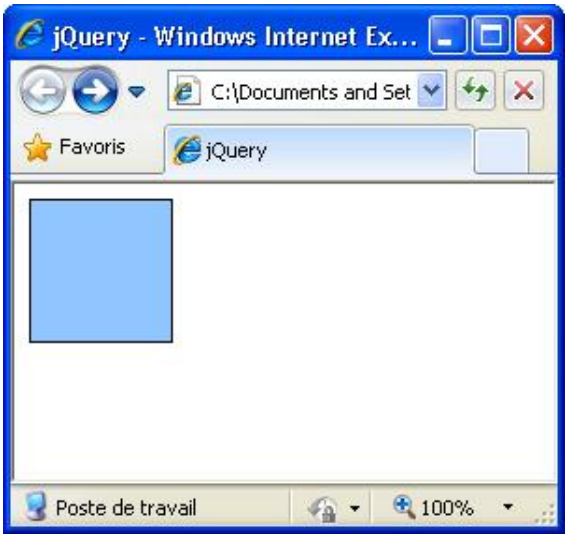
La documentation et le fichier du plug-in **jquery.easydrag.js** sont disponibles à l'adresse : http://fromvega.com/wordpress/2007/07/14/easydrag-jquery-plugin/

© ENI Editions - All rigths reserved - Jonifar lina

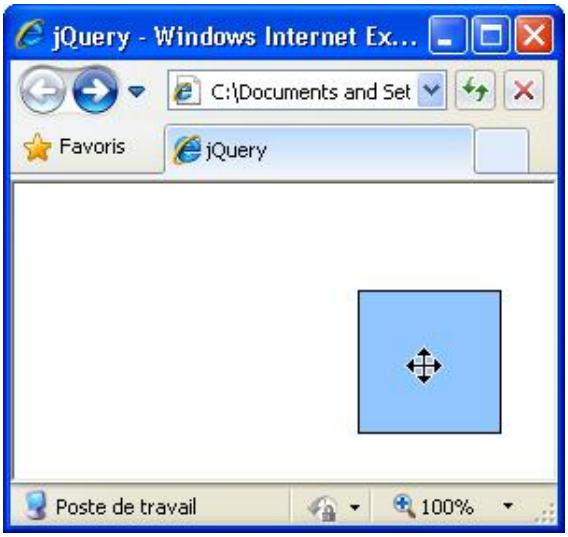
Une fois le fichier **jquery.easydrag.js** inclus dans le même dossier que la page Html, cette fonction de glisser/déposer est disponible.

Exemple

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="jquery.easydrag.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#box").easydrag();
});
</script>
<style type="text/css">
div { position: relative;
      width: 75px;
      height: 75px;
      border: 1px solid black;
      background-color: #9cf;}
</style>
</head>
<body>
<div id="box"></div>
</body>
</html>
```



© ENI Editions - All rigths reserved - Jonifar lina



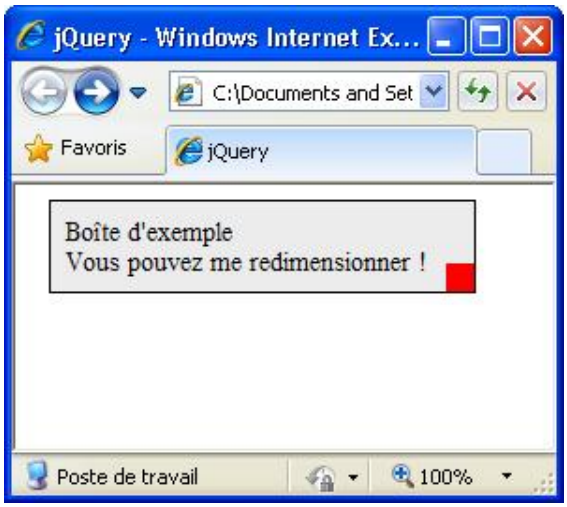
5. Redimensionner les divisions

Le redimensionnement des boîtes (divisions) rendu possible par ce plug-in est un plus assez spectaculaire et original.

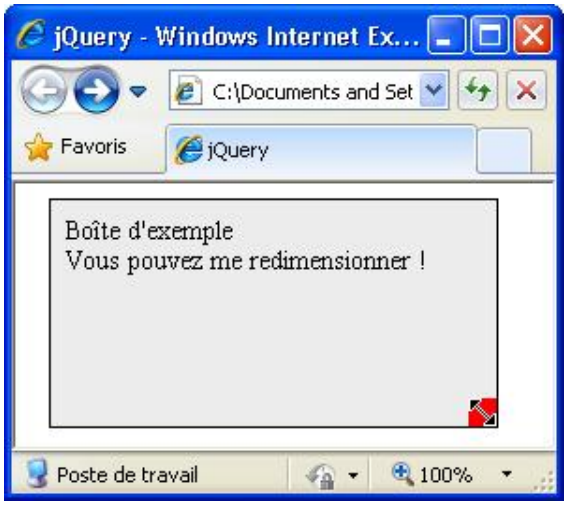
Les fichiers **jqDnR.js** et **dimensions.js** nécessaires au plug-in sont téléchargeables à l'adresse http://dev.iceburg.net/jquery/jqDnR/

Ce script permet en outre, comme le plug-in précédent de déplacer les éléments (drag/drop).

Exemple



© ENI Editions - All rigths reserved - Jonifar lina




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="jqDnR.js"></script>
<script type="text/javascript" src="dimensions.js"></script>
<script type="text/javascript">
$(document).ready(function() {
$('#ex1').jqResize('.jqResize');
});
</script>
<style type="text/css">
.jqHandle { background: red;
            height:15px;}
.jqDrag { width: 100%;
          cursor: move;}
.jqResize { width: 15px;
            position: absolute;
            bottom: 0;
            right: 0;
            cursor: se-resize;}
.jqDnR { z-index: 3;
         position: relative;
         width: 210px;
         font-size: 0.87em;
         color: black;
         margin: 5px 10px 10px 10px;
         padding: 8px;
         background-color: #EEE;
         border: 1px solid black;
}
</style>
</head>
<body>
<div id="ex1" class="jqDnR">
Boîte d'exemple<br />
Vous pouvez me redimensionner !
<div class="jqHandle jqResize"></div>
</div>
</body>
</html>
```

6. Un carrousel d'images

© ENI Editions - All rigths reserved - Jonifar lina

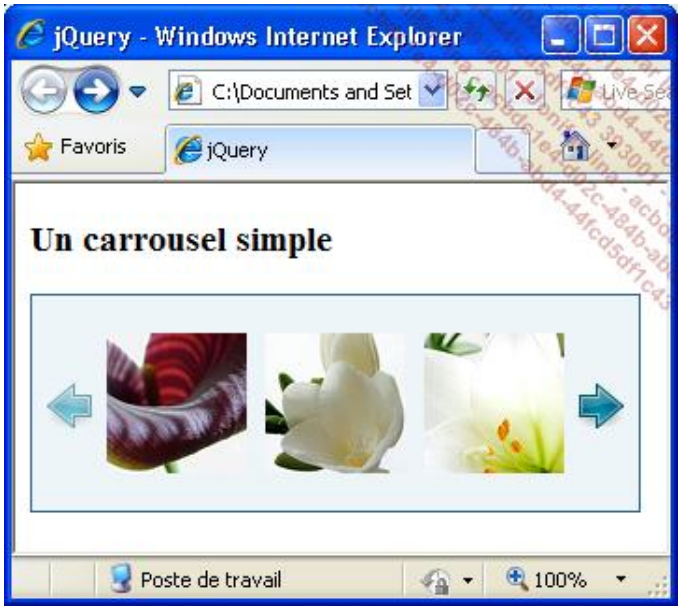
Les carrousels d'images sont très appréciés des designers car ils offrent la possibilité à l'utilisateur de visionner de multiples images, tout en leur consacrant une place réduite sur la page.

Le plug-in jCarousel est l'un des plus réputés et est disponible à l'adresse : http://sorgalla.com/projects/jcarousel/

Il permet de contrôler des images disposées horizontalement ou verticalement. Celles-ci peuvent être du contenu Html classique ou chargées dynamiquement par de l'AJAX. Elles peuvent défiler avec ou sans animation.

Exemple

Les images de l'exemple sont disponibles dans l'espace de téléchargement consacré à cet ouvrage.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="jquery.jcarousel.js"></script>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<link rel="stylesheet" type="text/css" href="jquery.jcarousel.css" />
<link rel="stylesheet" type="text/css" href="skins/tango/skin.css" />
<script type="text/javascript">
jQuery(document).ready(function() {
jQuery('#mycarousel').jcarousel();
});
</script>
</head>
<body>
<div id="wrap">
<h3>Un carrousel simple</h3>
<ul id="mycarousel" class="jcarousel-skin-tango">
<li><img src="a.jpg" width="75" height="75" alt="" /></li>
<li><img src="b.jpg" width="75" height="75" alt="" /></li>
<li><img src="c.jpg" width="75" height="75" alt="" /></li>
<li><img src="d.jpg" width="75" height="75" alt="" /></li>
<li><img src="e.jpg" width="75" height="75" alt="" /></li>
<li><img src="f.jpg" width="75" height="75" alt="" /></li>
<li><img src="g.jpg" width="75" height="75" alt="" /></li>
<li><img src="h.jpg" width="75" height="75" alt="" /></li>
<li><img src="i.jpg" width="75" height="75" alt="" /></li>
<li><img src="j.jpg" width="75" height="75" alt="" /></li>
</ul>
</div>
</body>
</html>
```

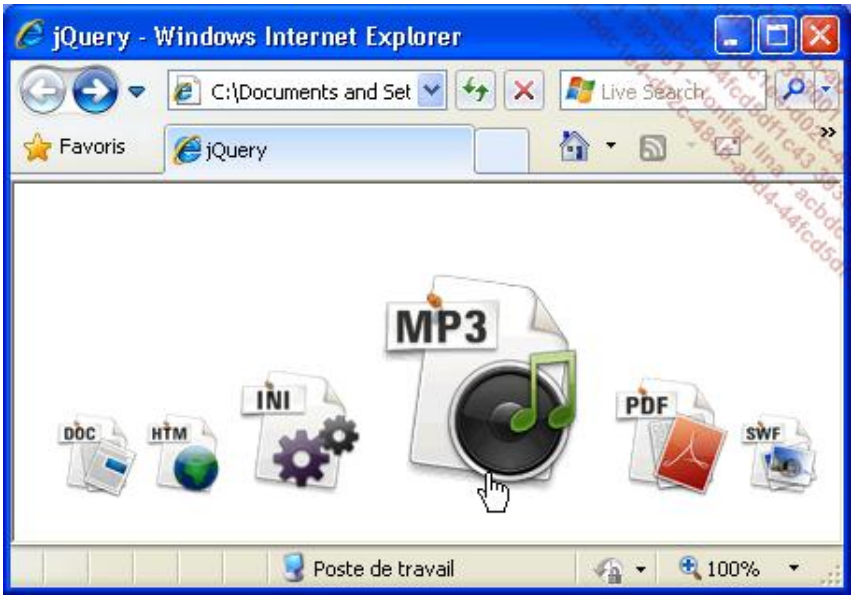
7. Un menu façon Mac

Le plug-in jqDock (http://www.wizzud.com/jqDock/) transforme une série d'images en un menu qui n'est pas sans rappeler la fonctionnalité Dock de l'environnement graphique du Mac OS X.

La documentation (en anglais) est particulièrement soignée et détaillée.

Exemple

Les petites icônes utilisées dans l'exemple sont disponibles dans l'espace de téléchargement.




```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="jquery.jqDock.min.js"></script>
<link rel="stylesheet" type="text/css" href="style.css" />
<script type="text/javascript">
//<![CDATA[
jQuery(function($){
$('div.demo>img').each(function(i){
$(this).one('click', function(){ location.href =
'index.php?dt='+(i%5); return false; });
});
$('div.demo').each(function(i){
var opts = { align: [ 'bottom', 'right', 'top' , 'middle', 'left',
'center' ][i] || 'bottom'
, size:         [  48     ,  48    ,  48   ,  48     ,  36   ,  60
][i] || 36       //default
, distance:     [  60     ,  60    ,  60   ,  60     ,  48   ,  80
][i] || 54       //default
, coefficient : [  1.5    ,  1.5   ,  1.5  ,  1      ,  1.5  ,
1.5     ][i] || 1.5      //default
, labels:       [  true   ,  'mc'  ,  true ,  'br'   ,  true ,
false   ][i] || false    //default
, duration:     500 //default
};
$(this).jqDock(opts);
});
$('div.demo a>img').not($('#menu1 a>img')).bind('click',
function(){
var Href = $(this).parent().get(0).href;
if(Href && !/^javascript:/i.test(Href)){
location.href = Href;
}else{
$(this).parent().trigger('click');
}
return false;
});
});
//]]>
</script>
</head>
```


© ENI Editions - All rigths reserved - Jonifar lina

```
<body>
<div id="menu6" class="demo">
<a href="#"><img alt="" src="doc.png" /></a>
<a href="#"><img alt="" src="htm.png" /></a>
<a href="#"><img alt="" src="ini.png" /></a>
<a href="#"><img alt="" src="mp3.png" /></a>
<a href="#"><img alt="" src="pdf.png" /></a>
<a href="#"><img alt="" src="swf.png" /></a>
</div>
</body>
</html>
```

8. La validation de formulaires

Autre vedette des plug-ins, le " jQuery Validation" dont le téléchargement et la documentation sont disponibles à l'adresse : http://bassistance.de/jquery-plugins/jquery-plugin-validation/

Ce plug-in vérifie les formulaires en 19 points (adresse e-mail valide, mention obligatoire, format de date, données numériques, numéro de carte de crédit, etc.).

Pour une utilisation courante, il faut veiller à mettre les messages d'erreur (inclus dans le plug-in) en français.

Exemple

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="jquery.validate.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("#formulaire").validate();
});
</script>
<style type="text/css">
label { width: 5em;
        float: left; }
label.error { float: none;
              color: red;
              padding-left: .5em;
              vertical-align: top; }
```

© ENI Editions - All rigths reserved - Jonifar lina

```
p { clear: both; }
.submit { margin-left: 6em; }
</style>
</head>
<body>
<form id="formulaire" action="">
<p>
<label for="cnom">Nom * :</label>
<input type="text" id="cnom" name="nom" size="20" class="required"
minlength="2" />
</p>
<p>
<label for="cprenom">Prénom :</label>
<input type="text" id="cprenom" name="prenom" size="20"
minlength="2" />
</p>
<p><label for="cemail">Email * :</label>
<input type="text" id="cemail" name="email" size="20"
class="required email" />
</p>
<p>
<input class="submit" type="submit" value="Soumettre"/>
</p>
</form>
</body>
</html>
```

9. Des graphiques à partir d'un tableau

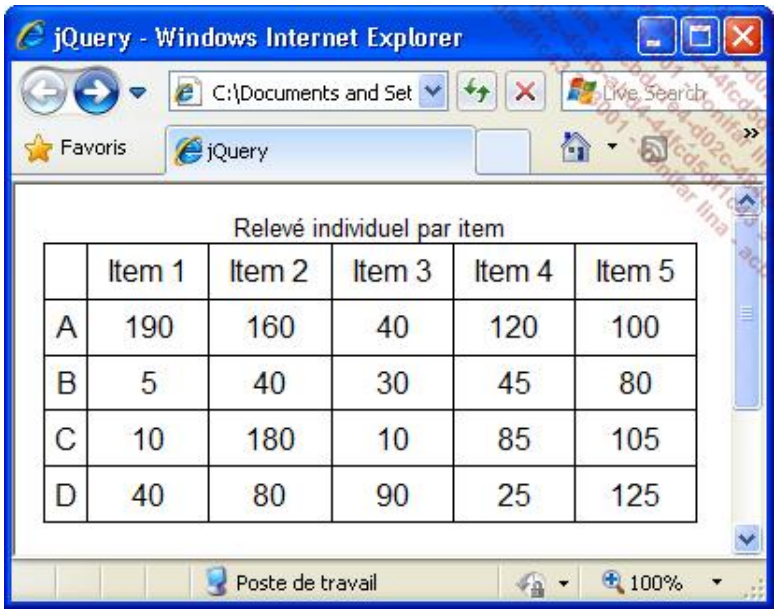
Il est indubitable qu'un graphique est souvent plus explicite qu'un tableau de données. Pour afficher un graphique dans les pages Html, il n'y avait comme alternative que la capture d'écran du tableau dans un tableur du style Excel.

Le plug-in visualize de jQuery permet de créer à la volée un graphique. Celui-ci s'affiche alors directement dans la page. Intéressant surtout lorsque le tableau de données connaît des mises à jour fréquentes. La documentation et le téléchargement sont disponibles à l'adresse : http://www.filamentgroup.com/lab/jquery_visualize_plugin_accessible_charts_ graphs_from_tables_html5_canvas/

Exemple

Après avoir inclus les différents éléments du plug-in (visualize.jQuery.js, visualize.jQuery.css et excanvas.compiled.js) dans le dossier de notre page, expérimentons ce plug-in pour le moins surprenant.

Soit le tableau de données suivant :



	Item 1	Item 2	Item 3	Item 4	Item 5
A	190	160	40	120	100
B	5	40	30	45	80
C	10	180	10	85	105
D	40	80	90	25	125

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
```

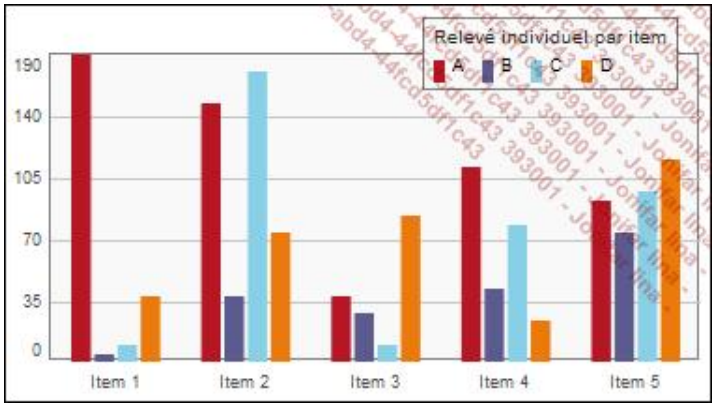

© ENI Editions - All rigths reserved - Jonifar lina

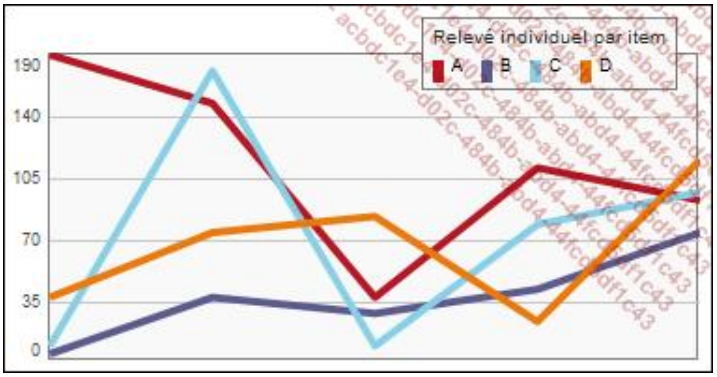
```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
<link type="text/css" rel="stylesheet"
href="visualize.jQuery.css"/>
<link type="text/css" rel="stylesheet"
href="demopage.css"/>
<script type="text/javascript" src="jquery.js"></script>
<!--[if IE]><script type="text/javascript"
src="excanvas.compiled.js"></script><![endif]-->
<script type="text/javascript" src="visualize.jQuery.js"></script>
<script type="text/javascript">
$(function(){
$('table').visualize({type: 'pie', pieMargin: 10, title: 'Relévé
par individu'});
$('table').visualize({type: 'line'});
$('table').visualize();
});
</script>
</head>
<body>
<table >
<caption>Relevé individuel par item</caption>
<thead>
<tr>
<td></td>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
</tr>
</thead>
<tbody>
<tr>
<th>A</th>
<td>190</td>
<td>160</td>
<td>40</td>
<td>120</td>
<td>100</td>
</tr>
<tr>
<th>B</th>
<td>5</td>
<td>40</td>
<td>30</td>
<td>45</td>
<td>80</td>
</tr>
<tr>
<th>C</th>
<td>10</td>
<td>180</td>
<td>10</td>
<td>85</td>
<td>105</td>
</tr>
<tr>
<th>D</th>
<td>40</td>
<td>80</td>
<td>90</td>
<td>25</td>
<td>125</td>
</tr>
</tbody>
</table>
```

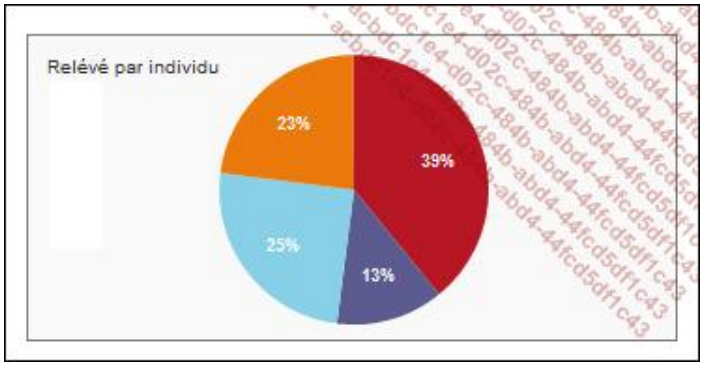
© ENI Editions - All rigths reserved - Jonifar lina

```
</body>
</html>
```

Le plug-in de l'exemple produit automatiquement 3 graphiques :







10. Trier un tableau de données

Autre merveille de programmation, le plug-in jQuery "tablesorter" qui permet de trier à la volée les données de n'importe quelle colonne d'un tableau par ordre ascendant ou descendant. Le script détecte automatiquement le type d'informations contenu dans la colonne. Il reconnait entre autres, les nombres, les dates, les adresses IP, etc.

La documentation (à lire soigneusement) et le téléchargement sont disponibles à l'adresse : http://tablesorter.com

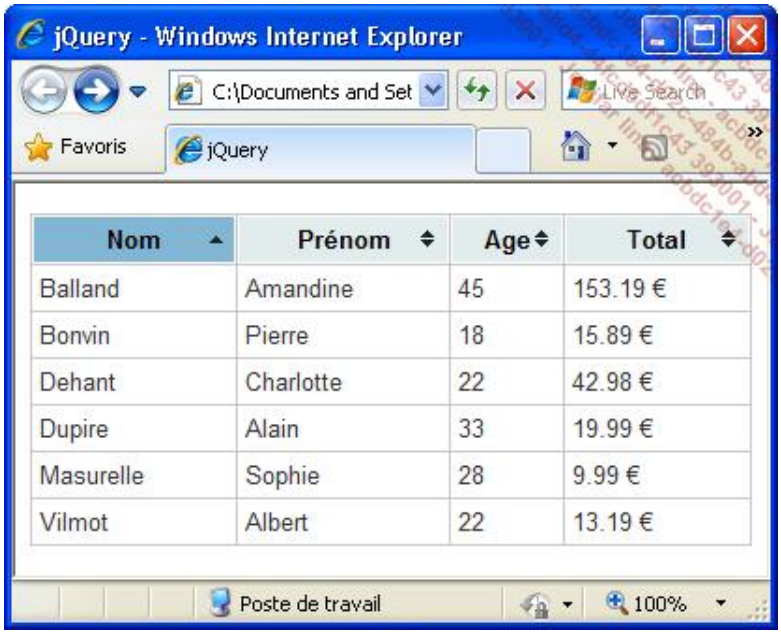
Exemple

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr"
lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<title>jQuery</title>
```


© ENI Editions - All rigths reserved - Jonifar lina

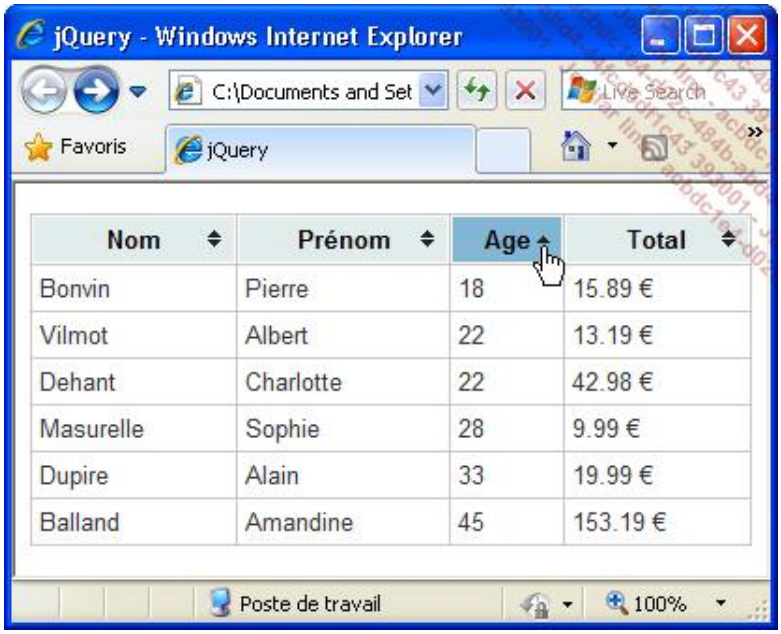
```
<link rel="stylesheet" href="blue/style.css" type="text/css" />
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="jquery.tablesorter.js"></script>
<script type="text/javascript">
$(document).ready(function() {
$("table").tablesorter({
sortList: [[0,0]]
});
});
</script>
</head>
<body>
<p></p>
<div id="demo">
<table cellspacing="1" class="tablesorter">
<thead>
<tr>
<th>Nom</th>
<th>Prénom</th>
<th>Age</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Masurelle</td>
<td>Sophie</td>
<td>28</td>
<td>9.99 €</td>
</tr>
<tr>
<td>Dupire</td>
<td>Alain</td>
<td>33</td>
<td>19.99 €</td>
</tr>
<tr>
<td>Bonvin</td>
<td>Pierre</td>
<td>18</td>
<td>15.89 €</td>
</tr>
<tr>
<td>Balland</td>
<td>Amandine</td>
<td>45</td>
<td>153.19 €</td>
</tr>
<tr>
<td>Vilmot</td>
<td>Albert</td>
<td>22</td>
<td>13.19 €</td>
</tr>
<tr>
<td>Dehant</td>
<td>Charlotte</td>
<td>22</td>
<td>42.98 €</td>
</tr>
</tbody>
</table>
</div>
</body>
</html>
```

© ENI Editions - All rigths reserved - Jonifar lina



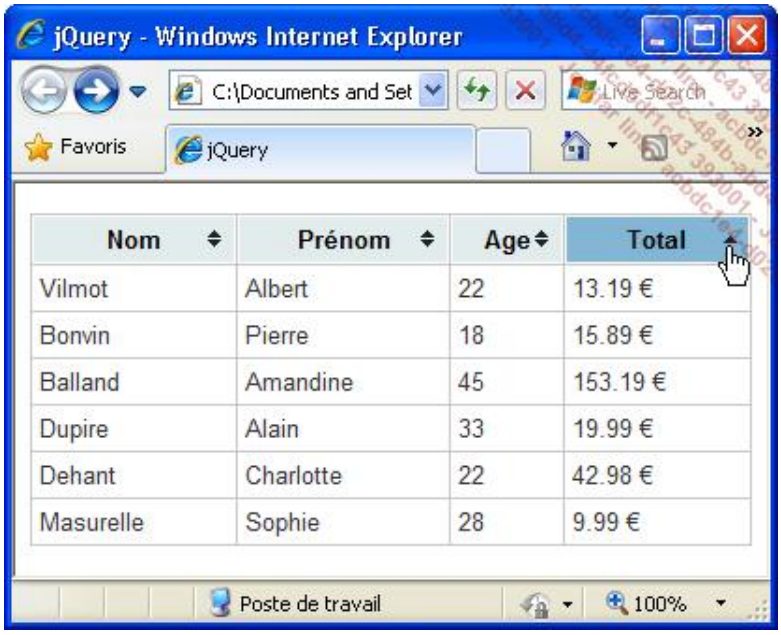
On remarquera que le plug-in tel qu'il est configuré a trié selon un ordre ascendant la première colonne (`sortList: [[0,0]]`).

Il est possible de trier selon l'âge :



ou d'effectuer un tri selon la colonne "Total".

© ENI Editions - All rigths reserved - Jonifar lina



© ENI Editions - All rigths reserved - Jonifar lina