

Parcourir et filtrer le DOM

En permettant au développeur d'accéder à chacun des éléments de la page, le DOM a relancé le JavaScript. Mais il faut admettre que le traçage de ces éléments n'est pas toujours très aisé à mettre en place. En outre, une simple modification de la page entraîne souvent une réécriture complète du code. Le framework jQuery y remédie par ses nombreux sélecteurs et des méthodes innovantes pour traverser et filtrer le DOM.

Parcourir le DOM

jQuery reprend et affine la navigation dans la structure du DOM par le système d'éléments parent, enfant, frère, etc. De multiples méthodes permettent de trouver les enfants (`children()`), le parent direct (`parent()`), tous les parents (`parents()`), tous les frères (`siblings()`), le frère précédent (`prev()`), les frères précédents (`prevAll()`), le frère suivant (`next()`), les frères suivants (`nextAll()`), le contenu d'un nœud (`content()`).

Le premier ou le dernier élément

<code>first()</code>	Retourne le premier élément de la sélection. <code>\$("li").first().css("background-color", "red");</code>
<code>last()</code>	Retourne le dernier élément de la sélection. <code>\$("p span").last().addClass("highlight");</code>

Filtrer des éléments

<code>filter(expression)</code>	Supprime de la sélection tous les éléments sélectionnés qui ne répondent pas à la recherche initiée selon le critère fourni en paramètre. <code>\$("div").filter(".selected");</code>
---------------------------------	--

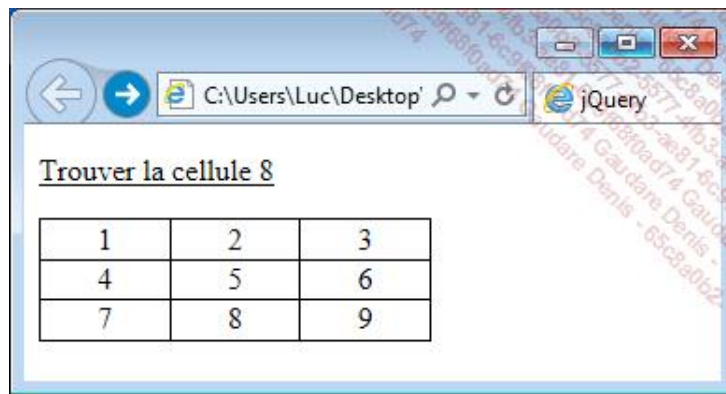
Trouver un élément déterminé

La méthode `eq()` (*eq* pour *equal*) permet de pointer la recherche directement sur un élément spécifique.

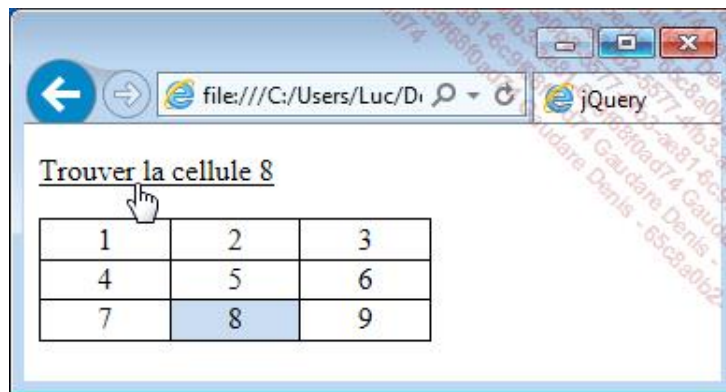
<code>eq(index)</code>	Détermine la position de l'élément par un entier fourni en argument. La numérotation débute à 0 comme toujours en JavaScript. Ainsi <code> \$("p").eq(1)</code> sélectionne le second paragraphe.
------------------------	---

Exemple

Soit un tableau à trois lignes et trois colonnes. Au clic sur un lien, mettre un arrière-plan de couleur à la cellule 8.



Au clic sur un lien :



```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>jQuery</title>
<meta charset="UTF-8">
<script src="http://code.jquery.com/jquery-1.11.0.min.js"></script>
<script>
$(document).ready(function(){
$("a").click(function(){
$("td").eq(7).addClass("bleu");
});
});
</script>
<style>
a { color: black;}
.bleu { background-color: rgb(195,215,235);}
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>
</table>
</body>
</html>
```

Trouver un élément selon un critère

`is(critère)` Indique par `true` ou `false` si la sélection correspond au critère à vérifier.

```
$( "div" ).is( ".bleu" );
```

Supprimer de la sélection selon un critère

`not(critère)` Supprime de la sélection les éléments correspondants au critère spécifié.

```
$( "div" ).not( ".vert" );
```