

jQuery

jQuery est une bibliothèque qui permet de simplifier la programmation JavaScript afin d'interagir plus efficacement avec l'utilisateur et d'animer plus facilement des contenus.

La bibliothèque jQuery possède également quelques extensions très intéressantes :

- **jQuery UI**, pour manipuler et enjoliver les interfaces utilisateur
- **jQuery mobile**, pour adapter aisément des contenus à des téléphones ou tablettes

Le code jQuery

Le jQuery étant du JavaScript, il se situe naturellement entre deux balises `<script>`.

Code jQuery	Explications
<code><script></code>	
<code>\$(function() {</code>	Quand la page Web est prête, faire ceci :
<code>\$(« button »).click(function() {</code>	Quand on clique sur le bouton faire ceci :
<code>\$(« h1 »).hide(« slow »);</code>	Faire disparaître lentement toutes les balises <code><h1></code> de la page
<code>\$(« h2 »).show(« fast »);</code>	Faire apparaître rapidement toutes les balises <code><h2></code> de la page
<code>\$(« img »).slideUp();</code>	Faire disparaître vers le haut toutes les balises <code></code>
<code>});</code>	Fin de la fonction <code>click()</code>
<code>});</code>	Fin de la fonction <code>document.ready()</code>
<code></script></code>	

Remarque : la première fonction `$(function()` n'est pas strictement obligatoire, mais elle permet d'attendre le chargement de toutes les balises avant d'agir dessus. Le risque est d'assigner un comportement à un élément qui n'est pas encore défini.

Comparatif méthode JavaScript traditionnelle et méthode jQuery

Dans le code suivant, nous écrivons le texte *Tomates* dans la balise HTML portant l'id *menu*. on voit que le jQuery simplifie le JavaScript :

JavaScript Traditionnel

Intérieur
de la fenêtre
du navigateur

↓
`document.getElementById("menu")`

• `innerHTML = "Tomates";`

↑
à l'intérieur
de la balise HTML

↑
Texte à écrire

ID
"menu"

jQuery

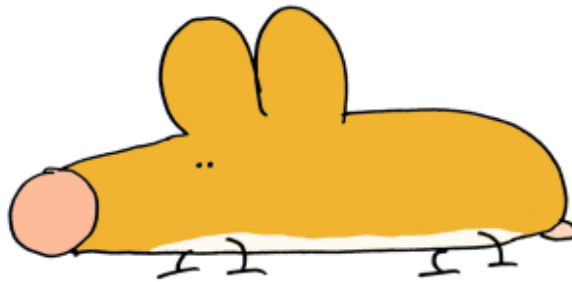
`$("#menu").html = "Tomates"`

↓
ID "menu"

↓
écrire
dans la balise
HTML ...

↓
... le texte
"Tomates"

ÇA SIMPLIFIE
QUAND MÊME !



MAIS SI ON VOULAIT TOUT SIMPLIFIER,
ON IRAIT DIRECT AU CIMETIÈRE... *

* Michel Audiard

La fonction jQuery et son raccourci : \$

Le signe dollar est l'abréviation de la fonction `jQuery()`. Ce raccourci évite d'écrire `jQuery()` à chaque appel de la fonction.

Le signe dollar avec ses parenthèses est l'abréviation de la **fonction** jQuery. Ce raccourci nous évite d'écrire « `jQuery()` » chaque fois que l'on veut appeler la fonction jQuery. On désigne également la fonction jQuery sous le nom de **jQuery wrapper**.

jQuery ()
Fonction jQuery dont le but est de récupérer les éléments entre parenthèses.

↓

→ \$ ()

Raccourci jQuery. On saisit un seul caractère au lieu d'en taper six.

Les sélecteurs jQuery identiques aux sélecteurs CSS

A l'intérieur de la fonction `$()`, nous pouvons mettre des sélecteurs identiques aux sélecteurs CSS :



Les sélecteurs jQuery sont strictement identiques aux **sélecteurs CSS**. Une bonne connaissance du langage CSS vous aidera toujours.

CSS	jQuery
<code>h2 {color: green; }</code>	<code>\$ (« h2 »).hide()</code>
<code>.vignette {border-style: solid; }</code>	<code>\$ (« .vignette »).slideUp()</code>
<code>#menu {color: red; }</code>	<code>\$ (« #menu »).fadeOut()</code>

- Les sélecteurs CSS sélectionnent des éléments pour leur ajouter une **mise en forme**
- Les sélecteurs jQuery sélectionnent des éléments pour leur ajouter une **mise en forme** ou un **comportement**

Quelques exemples jQuery

Les quatre boutons

En cliquant sur les boutons, vous allez découvrir certaines facettes de jQuery. Ici les animations, la modification des CSS, puis l'apparition et la disparition des balises.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>jQuery</title>
    <style>
      #change_moi {position: absolute; top: 100px; left: 400px; }
      #bouge_haut,#bouge_bas,#couleur,#disparais {padding: 5px;}
    </style>
    <script src="https://code.jquery.com/jquery-latest.js"></script>
  </head>
  <body>
    <button id="bouge_haut">Bouge haut</button>
    <button id="bouge_bas">Bouge bas</button>
    <button id="couleur">Change couleur</button>
    <button id="disparais">Disparais/Réapparais</button>

    <div id="change_moi">Zy va !</div>
  <script>
    $(function() { // Quand la page Web est prête, faire ceci :
      $("#bouge_haut").click( function() {
        $("#change_moi").animate({top:60},400);
      });// fin bouge_haut
      $("#bouge_bas").click( function() {
        $("#change_moi").animate({top:500},1500);
      });// fin bouge_bas
      $("#couleur").click( function() {
        $("#change_moi").css("color", "maroon");
      });// fin couleur
      $("#disparais").click( function() {
        $("#change_moi").toggle('slow');
      });// fin disparais
    });// fin page prête
  </script>
</body>
</html>
```

Évènement *click* et apparition en fondu

La méthode *slideToggle()* permet de faire apparaître l'élément s'il a disparu, et inversement, à la manière d'un interrupteur. La méthode *fadeIn()* fait apparaître le cadre contenant l'image en 1000 millisecondes, soit 1 seconde.

```
<!doctype html>
<html>
  <head>
    <style>
      #cadreimage {
        background: #D8B36E;
        padding: 20px;
```

```

        display: inline-block;
        border: 2px solid #aaa;
    }
</style>
<script src="https://code.jquery.com/jquery-latest.js"></script>
</head>
<body>
    <button>Montre</button>
    <div id="cadreimage">
        
    </div>
    <script>
    $(function() {
        $("button").click(function() {
            $("img").fadeIn(1000);
            $("#cadreimage").slideToggle("slow");
        });
    });
    </script>
</body>
</html>

```

Méthode *append()* et génération d'un nombre aléatoire

Le code ci-dessous calcule votre remise pour les soldes ☐ en générant un nombre aléatoire compris entre 5% et 9%.

- La méthode *append()* permet d'insérer du contenu à la fin d'une balise, en l'occurrence la balise `<div>` sous le bouton.
- La fonction JavaScript *Math.floor()* permet d'arrondir un nombre aléatoire généré par ce script.
- La fonction JavaScript *Math.random()* génère un nombre aléatoire compris entre 0 et 1. Par exemple : 0.478996
- La méthode *remove()* évite l'accumulation des balises `<p>`

```

<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
        <script src="https://code.jquery.com/jquery-latest.js"></script>
    </head>
    <body>
        <h1>Soldes !</h1>
        <button>Cliquez ici pour connaître votre remise</button>
        <div></div>
        <script>
        $(function() {
            $("button").click( function() {
                $("p").remove();
                var reduc = Math.floor((Math.random()*5) + 5);
                var reduc_mess = "<p>Votre réduction est de "+reduc+"%</p>";
                $("div").append(reduc_mess);
            });
        });
        </script>
    </body>
</html>

```

Validation de formulaire avec la méthode *submit()*

Si vous souhaitez vérifier une saisie de texte avant l'envoi d'un formulaire, jQuery permettra de lire les contenus des champs, puis de bloquer l'action du formulaire tant que les données ne seront pas conformes. Dans l'exemple ci-dessous, l'attribut *action* de la balise *<form>* ne sera pris en compte que si la méthode *submit()* renvoie la valeur *true*. La méthode *text()* permet d'écrire du texte dans la balise indiquée.

```
<!doctype html>
<html>

<head>
  <meta charset="utf-8">
  <style>
    p { margin:0; color: blue; }
    div, p { margin-left: 10px; }
    span { color: red; }
  </style>
  <script src="https://code.jquery.com/jquery-latest.js"></script>
</head>

<body>
  <p>Tapez 'correct' pour valider le formulaire</p>
  <form action="javascript:alert('Réussi !');">
    <div>
      <input type="text">
      <input type="submit">
    </div>
  </form>
  <span></span>
  <script>
    $("form").submit(function() {
      if ($("#input:first").val() == "correct") {
        $("#span").text("Valide !").show();
        return true;
      }
      $("#span").text("Non valide !").show().fadeOut(1000);
      return false;
    });
  </script>
</body>

</html>
```

Agrandissement d'une boîte

```
<!doctype html>
<html>

<head>
  <meta charset="utf-8">
  <style>
    #boite {
      background: black;
      color: snow;
      width: 100px;
```

```

        padding: 10px;
    }
</style>
<script src="https://code.jquery.com/jquery-latest.js"></script>
</head>

<body>
    <div id="boite">Cliquez sur moi pour m'agrandir</div>
    <script>
        $("#boite").one("click", function() {
            $(this).css("width", "+=200");
        });
    </script>
</body>

</html>

```

Lire du contenu HTML

La méthode *html()* permet de lire une portion de code HTML à la manière de *innerHTML* en JavaScript. Dans l'exemple ci-dessous, c'est le contenu du div conteneur qui est lu.

```

<!doctype html>
<html>
    <head>
        <script src="https://code.jquery.com/jquery-latest.js"></script>
    </head>
    <body>
        <div class="conteneur">
            <div class="boite">Demonstration Box</div>
        </div>
        <script>
            alert(
                $('div.conteneur').html()
            );
        </script>
    </body>
</html>

```

Écrire du contenu HTML

La méthode *html(string)* permet d'écrire du code HTML à la manière de *innerHTML*.

```

<!doctype html>
<html>
    <head>
        <style>
            .red { color:red; }
        </style>
        <script src="https://code.jquery.com/jquery-latest.js"></script>
    </head>
    <body>
        <span>Bonjour</span>
    </body>
</html>

```

```

        <div></div>
        <div></div>
        <div></div>
        <script>$("div").html("<span class='red'>Bonjour <b>encore</b></span>");</script>
    </body>
</html>

```

Manipulation des CSS

La méthode `css()` permet de manipuler les mises en forme. On pourra donc lire des propriétés CSS, ou modifier les styles que l'on souhaite. La propriété équivalente en JavaScript est *style*.

Lire la couleur de fond d'un div

```

<!doctype html>
<html>
    <head>
        <style>
            div { width:60px; height:60px; margin:5px; float:left; }
        </style>
        <script src="https://code.jquery.com/jquery-latest.js"></script>
    </head>
    <body>
        <span id="resultat"> </span>
        <div style="background-color:blue;"></div>
        <div style="background-color:rgb(15,99,30);"></div>
        <div style="background-color:#123456;"></div>
        <div style="background-color:#f11;"></div>
        <script>
            $("div").click(function () {
                var color = $(this).css("background-color");
                $("#resultat").html("Ce div a une couleur de fond : <span
style='color:"+color+";'>"+color+"</span>.");
            });
        </script>
    </body>
</html>

```

Changement de couleur CSS et événement *mouseover*

```

<!doctype html>
<html>
    <head>
        <style>
            p { color:blue; width:200px; font-size:14px; }
        </style>
        <script src="https://code.jquery.com/jquery-latest.js"></script>
    </head>
    <body>
        <p>Passez la souris dessus.</p>
        <p>Et voyez le changement de couleur.</p>
        <script>

```



```

        $("p").mouseover(function () {
            $(this).css("color","red");
        });
    </script>
</body>
</html>

```

Liste des propriétés CSS modifiables

Voici la liste complète des propriétés CSS que vous pouvez manipuler : www.w3schools.com/jsref/dom_obj_style.asp

Ajout d'une classe CSS

Dans cet exemple, jQuery permet d'ajouter la propriété et la valeur `class='bleu'` dans la dernière balise `<p>`. On peut alors régler une série de mises en forme dans le CSS de la classe.

```

<!doctype html>
<html>
  <head>
    <style>
      p { margin: 8px; font-size:16px; }
      .bleu { color:blue;background-color:red }
    </style>
    <script src="https://code.jquery.com/jquery-latest.js"></script>
  </head>
  <body>
    <p>Bonjour</p>
    <p>et</p>
    <p>au revoir.</p>
    <script>
      $("p:last").addClass("bleu");
    </script>
  </body>
</html>

```

Animations

jQuery nous offre la possibilité d'animer facilement des balises de votre choix avec la méthode `animate()`.

La différence entre les méthodes `css()` et `animate()` est que `animate()` va appliquer des changements de valeurs de propriété CSS graduellement, ce qui provoque un effet d'animation. On pourra donc jouer sur la durée de l'animation et les effets d'accélération du mouvement.

`animate(Propriétés [, Durée] [,Assouplissement] [, Fonction])`

- **Propriété** : Un ensemble de propriétés CSS pour mettre en forme l'animation.

- **Durée (option) :** Des vitesses prédéfinies (« slow » ou « fast ») ou le nombre de millisecondes pour lancer l'animation (par exemple 1000).
- **Assouplissement (option) :** L'effet d'accélération que vous souhaitez utiliser (« linear » ou « swing »).
- **Fonction (option) :** Une fonction à exécuter lorsque l'animation se termine.

Animation d'une boîte

```
<!doctype html>
<html>
  <head>
    <style>
      div {
        background-color:#bca;
        width:100px;
        border:1px solid green;
      }
    </style>
    <script src="https://code.jquery.com/jquery-latest.js"></script>
  </head>
  <body>
    <button id="go">>> Go</button>
    <div id="boite">Bonjour !</div>
    <script>
      /* on peut utiliser différentes unités de mesure dans une animation */
      $("#go").click(function() {
        $("#boite").animate({
          width: "70%",
          opacity: 0.4,
          marginLeft: "0.6in",
          fontSize: "3em",
          borderWidth: "10px"
        }, 1500 );
      });
    </script>
  </body>
</html>
```

Déplacement d'une boîte bleue

Dans cet exemple, la boîte bleue va se déplacer de plus ou moins 50 pixels selon le bouton cliqué. Les opérateurs += et -= permettent d'ajouter ou de retirer 50 pixels à la position actuelle de la boîte.

```
<!doctype html>
<html>
    <head>
        <style>
            div {
                position: absolute;
                background-color: #abc;
                left: 50px;
                width: 90px;
                height: 90px;
                margin: 5px;
            }
        </style>
    </head>
    <body>
```

```

</style>
<script src="https://code.jquery.com/jquery-latest.js"></script>
</head>
<body>
  <button id="left"><<</button>
  <button id="right">>></button>
  <div class="block"></div>
  <script>
    $( "#right" ).click(function() {
      $( ".block" ).animate({ "left": "+=50px" }, "slow" );
    });
    $( "#left" ).click(function(){
      $( ".block" ).animate({ "left": "-=50px" }, "slow" );
    });
  </script>
</body>
</html>

```

Tableau des méthodes jQuery

Un petit tableau pour se remémorer les méthodes jQuery décrites sur cette page. En combinant toutes ces capacités, vous pouvez déjà faire beaucoup de choses en jQuery.

Méthode	Description	Animation	Exemple
.hide(), .show(), .toggle()	Apparaître, disparaître, interrupteur	Taille et opacité	<code>\$(« p »).show(« slow »);</code>
.slideDown(), .slideUp(), .slideToggle()	Apparaître, disparaître, interrupteur	Glissement vers le bas ou vers le haut	<code>\$(« #truc »).slideUp(« fast »);</code>
.fadeIn(), .fadeOut(), .fadeToggle()	Apparaître, disparaître, interrupteur	Opacité	<code>\$(« img »).fadeIn(1000);</code>
.css()	Changement d'état des feuilles de style		<code>\$(« p.citation »).css(« fontSize », »3em »);</code>
.animate()	Changement d'état des feuilles de style	Durée et assouplissement	<code>\$(« #boite »).animate({ « width »: « 100px » }, « slow »);</code>
.html()	Lire ou écrire du code HTML		<code>\$(« div »).html(« Bonjour tout le monde »);</code>
.text()	Écrire du texte		<code>\$(« span »).text(« Au revoir »);</code>
.val()	Lire ou écrire la valeur d'un champ de formulaire.		<code>nom = \$(« #nom »).val();</code>
.append()	Insérer du code HTML à la fin du sélecteur		<code>\$(« div »).append(« <p>Coucou</p> »);</code>
.remove()	Supprime le sélecteur		<code>\$(« p »).remove();</code>
addClass()	Ajoute une classe		<code>\$(« p:last »).addClass(« bleu »);</code>