

# Syntaxe

Laurence  
X  NATIS



Compétence demandée :  
Comprendre la syntaxe de jQuery

1. jQuery
2. Manipulation du DOM

# 1. jQuery

On peut accéder aux différentes propriétés et méthodes de la librairie jQuery en utilisant **deux objets créés lors de l'inclusion** de la librairie : **jQuery()** ou **\$()**.

Ces deux objets ou propriétés sont ce que l'on appelle des **alias** : cela signifie qu'on peut utiliser l'une ou l'autre de façon complètement interchangeable. La plupart des développeurs utilisent plutôt la propriété **\$** puisqu'elle est plus rapide à écrire que jQuery.

Le **jQuery** est un langage de "Queries JavaScript" c'est-à-dire un **langage de requêtes**, et il sera utilisé pour effectuer des actions sur des éléments HTML. Pour effectuer ces actions, il faut sélectionner les éléments grâce à l'une des grandes forces du jQuery : **l'utilisation des sélecteurs CSS**.

Pour sélectionner un élément HTML en jQuery, il suffira de mentionner le sélecteur CSS associé à cet élément dans la méthode `$()` (ou `jquery()`).

Lorsqu'on utilise la syntaxe `$(selecteur)`, jQuery retourne un **objet jQuery** (ensemble de balises du document ou nœuds du DOM). On peut alors appliquer des méthodes ou des propriétés à l'objet retourné afin d'effectuer des actions ou d'obtenir des informations sur celui-ci.

On va alors utiliser une fonctionnalité du JavaScript qui s'appelle le **chaînage des méthodes** (qui signifie appliquer directement cette méthode au résultat de la précédente).



On utilise l'opérateur qui représente le « **point** ». La syntaxe est la suivante :

`$(selecteur).action()` : la première partie `$(selecteur)` renvoie un objet auquel on applique la méthode `action()`.

👉 Remarque: Il est possible d'appliquer plusieurs méthodes les unes à la suite des autres.

Ex : Plutôt que d'écrire les instructions comme ceci :

```
$('.box').addClass('is-collapsed');  
$('.box').slideUp();
```

Optimisez votre code comme cela :

```
$('.box').addClass('is-collapsed').slideUp();
```

## Sélecteurs de base

Equivalents des fonctions *getElementById* & *getElementsByTagName* de JavaScript

`$('*')` : sélectionne tous les éléments

`$(':header')` : sélectionne toutes les balises `<h>`

`$('div')` : sélectionne toutes les `<div>`

`$('a')` : sélectionne tous les éléments `<a>`

`$("p")` : sélectionne tous les éléments `<p>`

## Sélecteur par filtre de contenu

`$('div:contains('test'))` : sélectionne les `<div>` contenant le texte 'test'.

## Sélecteurs d'attribut

`$("[href]")` : sélectionne tous les éléments avec attribut href.

`$("[href='#']")` : sélectionne tous les éléments avec attribut href et la valeur associée (#).

`$("[href!='#']")` : sélectionne tous les éléments avec attribut href différent de la valeur « # ».

`$("[href$='.jpg']")` : sélectionne tous les éléments avec attribut href qui se termine par «.jpg ».

`$('a[href]')` : Les éléments <a> ayant l'attribut href.

`$('a[href="#"]')` : Les éléments <a> avec attribut href et la valeur associée (#).

`$('a[href!="#"]')` : Les éléments <a> n'ayant pas de valeur (#) dans l'attribut href.

`$('a[href^="#"]')` : Les éléments <a> ayant une valeur qui commence par la valeur#.

`$('a[href$="#"]')` : Les éléments <a> ayant une valeur qui se termine par #.

`$('a[href*="#"]')` : Les éléments <a> ayant une valeur qui contient #.

`$(this)` : correspond à l'objet en cours de manipulation. Il permet de retourner un objet jQuery à partir de l'objet JavaScript natif « this ». En d'autres termes, cela permet de passer du JavaScript au jQuery.

## Sélecteurs CSS

Permettent de modifier les propriétés CSS pour les éléments HTML

`$("ul li:first")` : Le premier élément `<li>` de chaque liste `<ul>`

`$("table button")` : Le(s) boutons inclus dans une table (tableau)

`$("table td")` : Tous les `td` inclus dans une table (tableau)

`$("td")` : Tous les `td` inclus dans un document global

`$("[href$='.jpg'] »)` : Tous les éléments avec un attribut `href` qui se termine par « `.jpg` ».

`$("div#intro .head")` : Tous les éléments de classe = « `head` » à l'intérieur d'un élément `<div>` portant l'id = « `intro` ».

`$("div #intro .head")` : Tous les éléments de classe = « `head` » ET d'id = « `intro` » à l'intérieur d'un élément `<div>`.

## Pour résumer ...

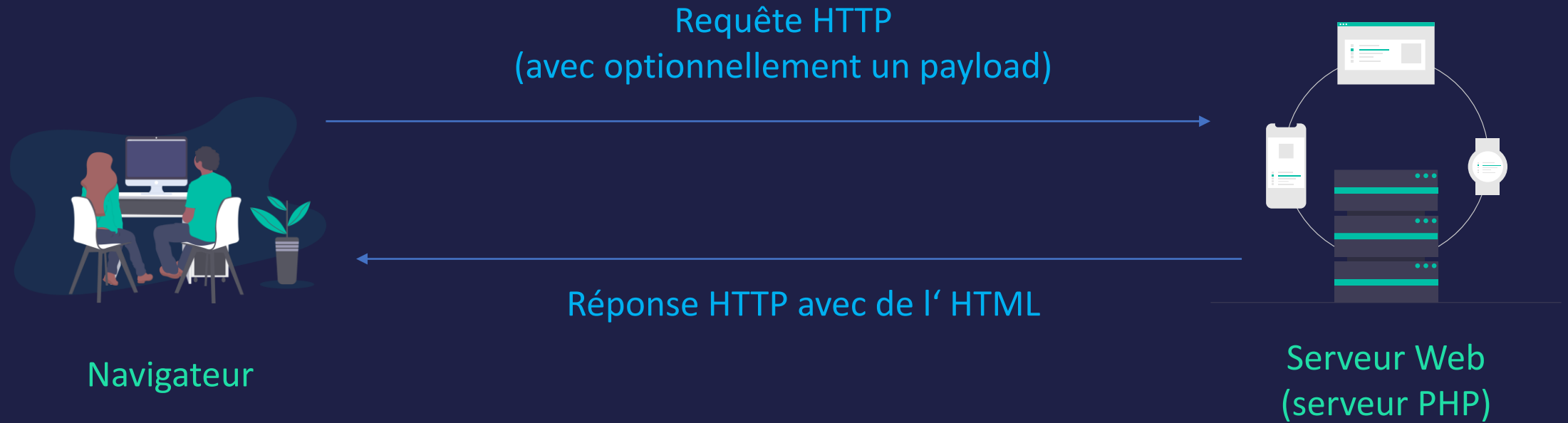
On cible un élément grâce à :

- ✓ Sa balise : `$("h4")`
- ✓ Son #id : `$("#mon_id")`
- ✓ Sa .class : `$(".ma_class")`
- ✓ Par lui-même : `$(this)`
- ✓ Par le type : `$("input:text")`
- ✓ Tous les types : `$(":radio")`
- ✓ Par l'attribut : `$("img[alt]")`
- ✓ Tous les attributs : `$("[alt]")`
- ✓ Par le nom de l'attribut ET sa valeur : `$("img[title='chaton']")`
- ✓ Sélection de plusieurs sélecteurs : `$("h1, p, #mon_id")`

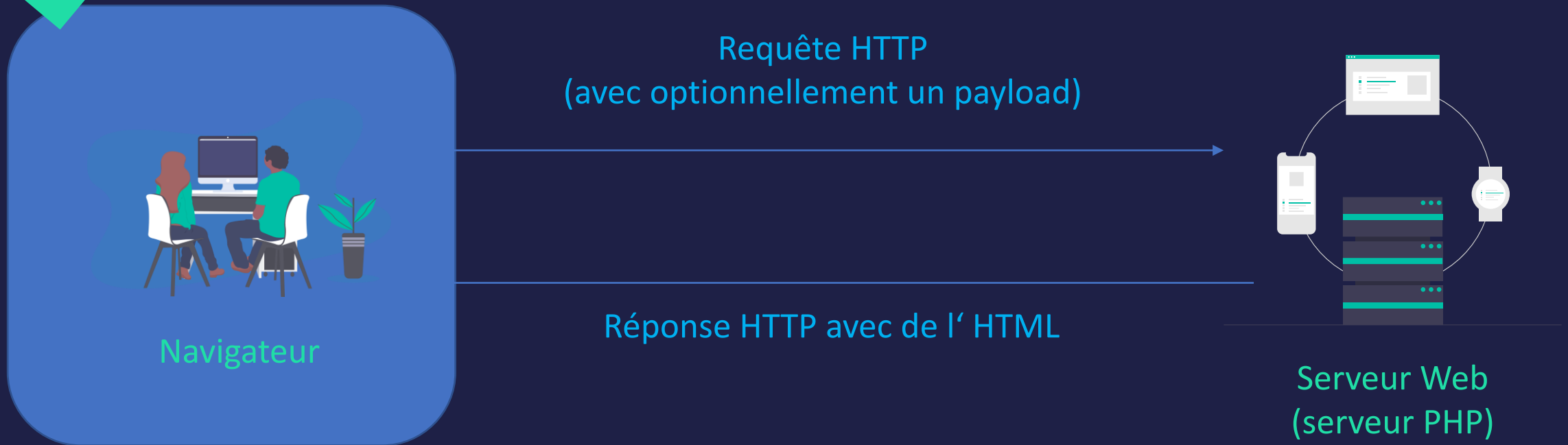
## 2. Manipulation du DOM

# Qu'est-ce que le DOM ?

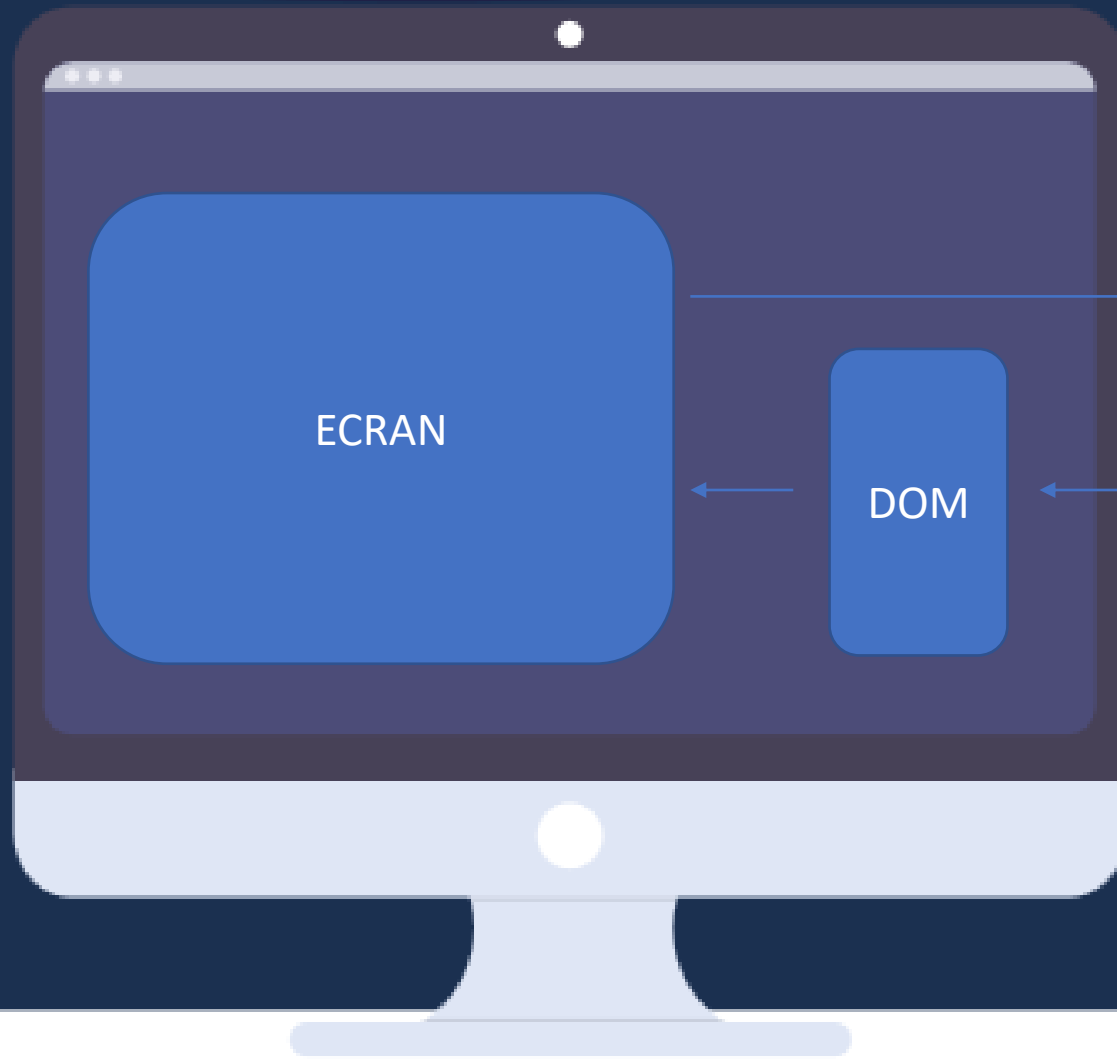








ZOOM A L'INTERIEUR  
DE CE SCHEMA



Requête HTTP  
(avec optionnellement un payload)

Réponse HTTP avec de l'HTML

Le jQuery étant une manipulation de la structure HTML, il faut s'assurer que le navigateur a fini de charger la page et sa structure avant d'effectuer toute opération en jQuery.

On utilise un événement mis à disposition par jQuery : l'événement « **ready** » qui **va permettre d'exécuter le code jQuery** dès que le document est **prêt à être manipulé**.

```
$(document).ready(function() {  
  // J'intégrerai du code jQuery ici  
});
```

Ici, on accroche l'événement `ready` au `document` et on passe une **fonction anonyme** (fonction sans nom) à cet événement.

L'événement `ready` se déclenche dès que le **document est prêt à être manipulé** et le code contenu dans la fonction anonyme s'exécute alors.

- **Getter** : afin d'obtenir la valeur
- **Setter** : pour définir la valeur

# Les méthodes de contenu

Contenu	Style	Animation
.text()	.css()	.hide()
.attr()	.height()	.show()
.removeAttr()	.width()	.toggle()
.html()	.addClass()	.fadeIn()
.val()	.removeClass()	.fadeOut()
	.hasClass()	.slideDown()
	.toggleClass()	.slideUp()
		.slideToggle()

## .text()

Permet de définir ou retourner le contenu texte du ou des élément(s) sélectionné(s).

Lorsque cette méthode est utilisée pour retourner une valeur, elle retourne le contenu texte combiné de tous les éléments sélectionnés (la balisage HTML est supprimé).

La syntaxe :

`$(selector).text()`

`$(selector).text(content)`



## .text() Exemple : Utilisation comme « Getter »

```
13 <body>
14   <h2>JQUERY</h2>
15   <!-- -->
16   <div>
17     <p id="p1">Lorem ipsum dolor</p>
18     <p id="p2">Sit amet consectetur</p>
19   </div>
```

```
<> cours.html  JS cours.js  X
JS cours.js > ...
1  $(document).ready(function(){
2
3    $('p').text();
4
5    alert($('p').text());
6
7  });|
```

Cette page indique  
Lorem ipsum dolorSit amet consectetur

OK

## .text() Exemple : Utilisation comme « Setter »

```
13 <body>
14   <h2>JQUERY</h2>
15   <!-- -->
16   <div>
17     <p id="p1">Lorem ipsum dolor</p>
18     <p id="p2">Sit amet consectetur</p>
19   </div>
```

```
JS cours.js  X
JS cours.js > ...
1 $(document).ready(function(){
2
3   $('#p2').text('<strong>Nouveau contenu</strong>');
4
5   alert($('#p2').text());
6
7 });
```

Cette page indique  
<strong>Nouveau contenu</strong>

OK

## .html()

Permet d'**obtenir le contenu HTML** (*innerHTML*) du premier élément d'une sélection si on l'utilise sans argument **ou définir le contenu HTML** de tous les éléments d'une sélection si on précise ce contenu en argument.

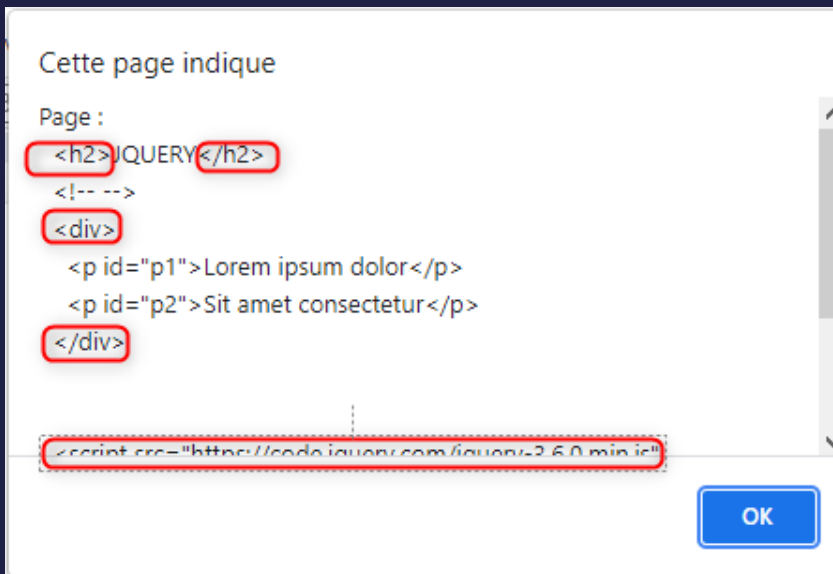
La syntaxe :

`$(selector).html()`

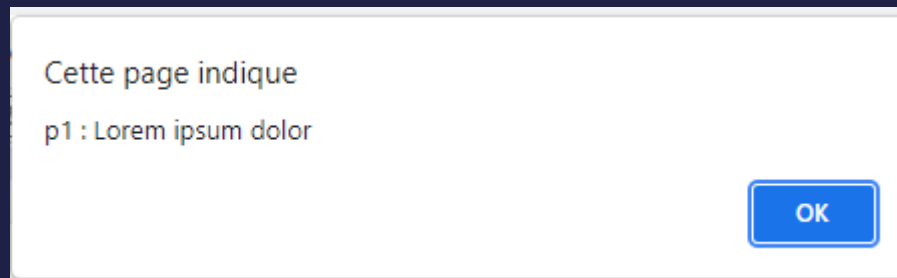
`$(selector).html(content)`

## .html() Exemple : Utilisation comme « Getter »

```
13 <body>
14   <h2>JQUERY</h2>
15   <!-- -->
16   <div>
17     <p id="p1">Lorem ipsum dolor</p>
18     <p id="p2">Sit amet consectetur</p>
19   </div>
```



```
<> cours.html  JS cours.js  X
JS cours.js > ...
1 $(document).ready(function(){
2
3   let page = $("body").html();
4   let p1 = $("#p1").html();
5
6   alert("Page : " + page);
7   alert("p1 : " + p1);
8
9 });
```



## .html() Exemple : Utilisation comme « Setter »

```
13 <body>
14   <h2>JQUERY</h2>
15   <!-- -->
16   <div>
17     <p id="p1">Lorem ipsum dolor</p>
18     <p id="p2">Sit amet consectetur</p>
19   </div>
```

Cette page indique

<strong>Nouveau contenu !</strong>

OK

```
<> cours.html JS cours.js X
JS cours.js > ready() callback
1 $(document).ready(function(){
2
3   $("#p2").html('<strong>Nouveau contenu !</strong>');
4
5   alert($("#p2").html());
6
7 });
```

## JQUERY

Lorem ipsum dolor

Nouveau contenu !

## .attr()

Permet de définir ou retourner la valeur d'un attributs du ou des élément(s) sélectionné(s).

La syntaxe :

```
$(selector).attr(attribute)
```

```
$(selector).attr({attribute:value, attribute:value,...})
```

## Exemple 1: Utilisation comme « Getter »

```
<body>
  <h2>JQUERY</h2>
  <!-- -->
  <div>
    <p id="p1">Lorem ipsum dolor</p>
    <p id="p2">Sit amet consectetur</p>
    
  </div>
  <button>Cliquez ici</button>
```

```
<> cours.html JS cours.js X
```

```
JS cours.js > ...  
1 $(document).ready(function(){  
2  
3     $("button").click(function() {  
4         alert("Image width: " +  
5             |   |   |    $("img").attr("width"));  
6         });  
7  
8     });|
```

## Exemple 1: Utilisation comme « Getter »

JQUERY

Lorem ipsum dolor

Sit amet consectetur



Cliquez ici



Taille affichée : 425 × 282 px  
Format affiché : 425:282  
Taille du fichier : 36.0 kB  
Source actuelle : [https://www.citationbonheur.fr/wp-content/uploads/2018/09/L\\_influence\\_du\\_paysage\\_sur\\_le\\_bonheur.jpg](https://www.citationbonheur.fr/wp-content/uploads/2018/09/L_influence_du_paysage_sur_le_bonheur.jpg)

Cette page indique

Image width: 425

OK



## Exemple 2 : Utilisation comme « Setter »

```
<body>
  <h2>JQUERY</h2>
  <!-- -->
  <div>
    <p id="p1">Lorem ipsum dolor</p>
    <p id="p2">Sit amet consectetur</p>
    
  </div>
  <button>Cliquez ici</button>
```

```
<> cours.html  JS cours.js  X
JS cours.js > ...
1  $(document).ready(function(){
2
3    $("button").click(function() {
4      $("img").attr("width", "600");
5      alert("Image width: " +
6        | | |  $("img").attr("width"));
7      });
8  });
```

## Exemple 2: Utilisation comme « Setter »

Cette page indique

Image width: 600

OK

### JQUERY

Lorem ipsum dolor

Sit amet consectetur



Cliquez ici

## Exemple 3:

Permet de définir ou retourner la valeur d'un attributs du ou des élément(s) sélectionné(s).

The screenshot shows a web browser window displaying a jQuery tutorial page titled "jQuery: Getting Started". The page has a red header with navigation links: "jQuery: Getting Started", "Hello World", "Example Page", and "Resources". Below the header, there are tabs for "Content", "HTML", "CSS", and "JavaScript". The main content area shows a form for a house listing. The form includes a "SHOWING DATE" field, a "NOTES" field, and an "AVAILABILITY" section with radio buttons for "Immediate" and "Delayed". A "Cancel" button and a "Save" button are at the bottom of the form. The house listing details include a photo of a house, the address "15509 Main Street" (highlighted in yellow), the location "Anytown, CO 80525", and a table of features: Bedrooms (6), Bathrooms (4.5), and Square Feet (4,500). Below the table, it says "In a quiet neighborhood next to excellent schools." and "SPECIAL FEATURES" with a list: "Crown molding throughout", "Granite counter tops", "New carpet and title", and "Shutter blinds on all windows".

The JavaScript console on the right shows the following code being executed:

```
> $('h2').attr('style', 'background-color:#ff3');  
< [  
  <h2 class="expanded" style="background-color:#ff3">15509 Main Street</h2>  
]  
>
```

## .removeAttr()

Permet de supprimer un attribut spécifié des éléments sélectionnés en utilisant la fonction « *removeAttribute()* » de JavaScript.

La syntaxe :

```
$(selector).removeAttr(attribute)
```

## `.val()`

Permet de retourner la valeur (attribut « *value* ») d'un ou des élément(s) sélectionné(s) notamment utilisé pour les inputs (Formulaires).

La syntaxe :

```
$(selector).val()
```

```
$(selector).val(value)
```

# Les méthodes de style

Contenu	Style	Animation
.text()	.css()	.hide()
.attr()	.height()	.show()
.removeAttr()	.width()	.toggle()
.html()	.addClass()	.fadeIn()
.val()	.removeClass()	.fadeOut()
	.hasClass()	.slideDown()
	.toggleClass()	.slideUp()
		.slideToggle()

## .css()

Permet d'obtenir une propriété de style de l'élément sélectionné.

La syntaxe :

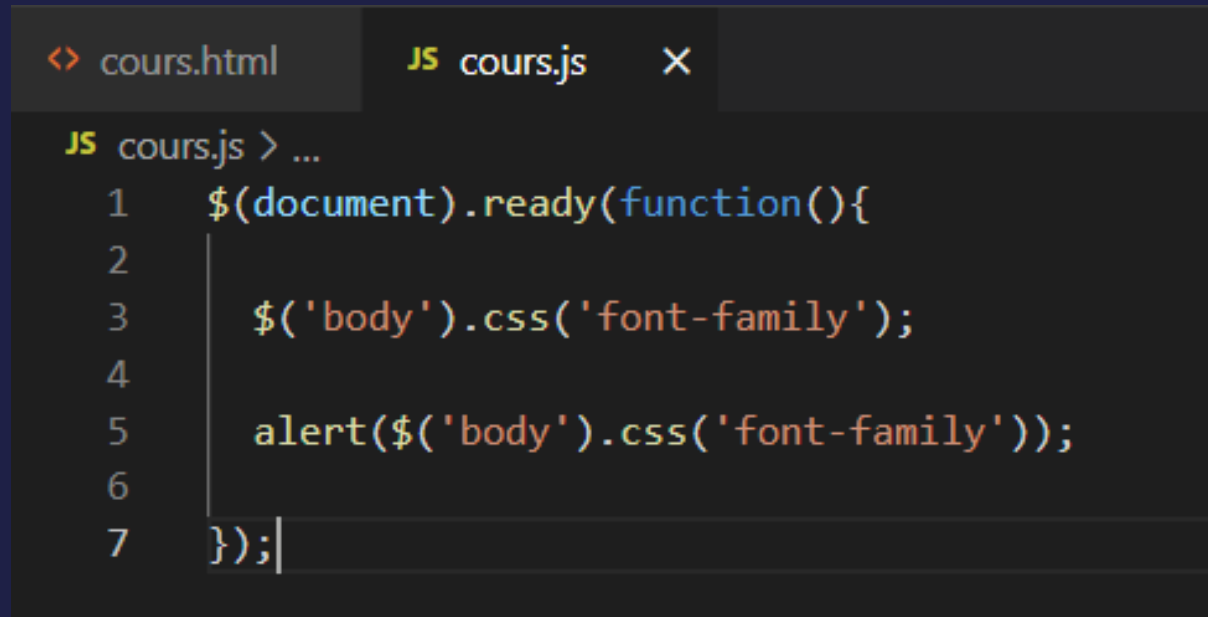
```
$(selector).css("property name")
```

```
$(selector).css("property name", "value")
```

```
$(selector).css("property name": "value", "property name":  
"value", ...)
```

## Exemple 1:

Pour obtenir la valeur liée à une propriété CSS pour un élément, on se contentera de préciser la propriété CSS dont on souhaite obtenir la valeur en argument de `css()`.

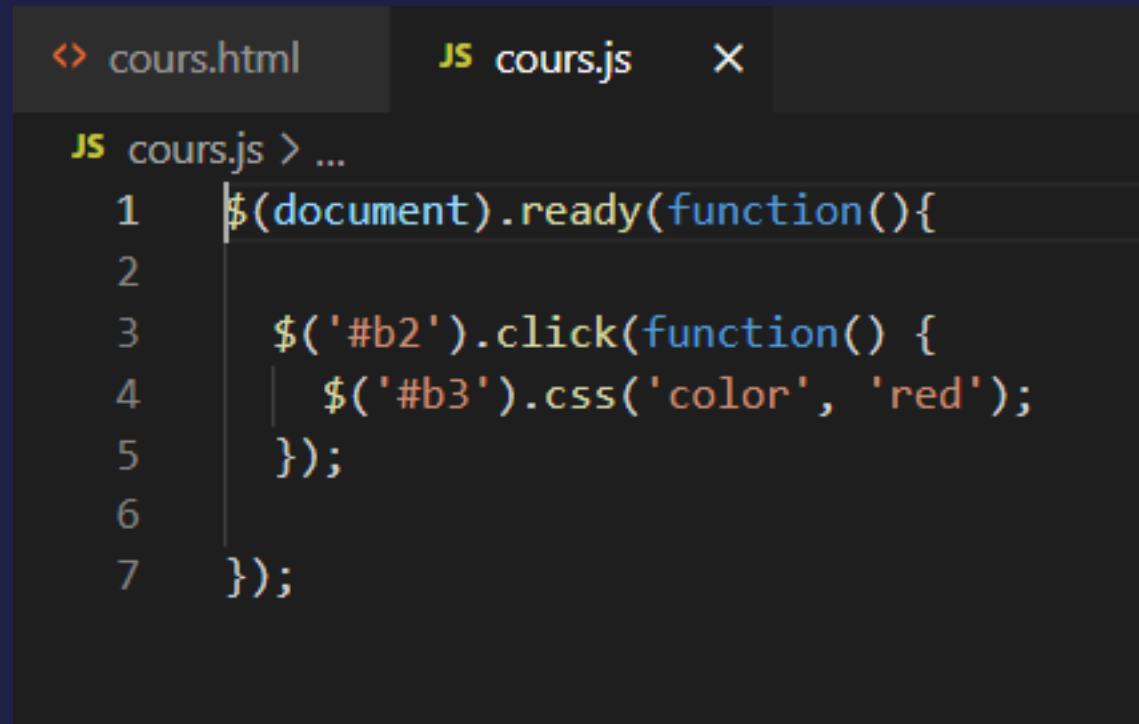


```
<> cours.html  JS cours.js  X
JS cours.js > ...
1  $(document).ready(function(){
2
3      $('body').css('font-family');
4
5      alert($('body').css('font-family'));
6
7  });|
```



## Exemple 2:

Pour définir une propriété et sa valeur pour une sélection d'éléments, on devra passer deux arguments à `css()` : la propriété CSS ET sa valeur.



```
<> cours.html    JS cours.js    X
JS cours.js > ...
1  $(document).ready(function(){
2
3      $('#b2').click(function() {
4          $('#b3').css('color', 'red');
5      });
6
7  });
```

### Exemple 3:

Pour définir de multiples propriétés CSS.

```
<> cours.html JS cours.js X
JS cours.js > ready() callback
1  $(document).ready(function(){
2
3      $("p").css({"background-color": "yellow", "font-size": "200%"});
4  });
```

## .height()

Permet de retourner la **hauteur actuelle** calculée d'un élément ou permet de définir/mettre à jour la hauteur de tout élément sélectionné

La syntaxe :

`$(selector).height()`

`$(selector).height(value)`

## .width()

Permet de retourner la **largeur actuelle** calculée d'un élément ou permet de définir/mettre à jour la hauteur de tout élément sélectionné

La syntaxe :

`$(selector).width()`

`$(selector).width(value)`

## `.addClass()`

Permet d'ajouter une ou plusieurs classes aux éléments HTML sélectionnés.

La syntaxe :

`$(selector).addClass(classname)`

## Exemple

Permet d'ajouter une ou plusieurs classes aux éléments HTML sélectionnés.

The screenshot displays a web browser window with a jQuery tutorial page. The page has a red header with navigation links: "jQuery: Getting Started", "Hello World", "Example Page", and "Resources". Below the header, there are tabs for "Content", "HTML", "CSS", and "JavaScript". The main content area shows a real estate form for "CODED homes". The form includes a photo of a house, a text input for "15509 Main Street" (highlighted in yellow), a text input for "Anytown, CO 80525", a table with property details (Bedrooms: 6, Bathrooms: 4.5, Square Feet: 4,500), a paragraph "In a quiet neighborhood next to excellent schools.", a "SPECIAL FEATURES" section with a bulleted list, a "SHOWING DATE" input, a checkbox for "Owner present?", a "NOTES" text area, and "AVAILABILITY" radio buttons for "Immediate" and "Delayed". At the bottom are "Cancel" and "Save" buttons.

The browser's developer console on the right shows the following JavaScript code:

```
<top frame> Preserve log  
> $('h2').addClass('highlight');  
< [   
    <h2 class="expanded highlight">15509  
      Main Street</h2>  
  ]  
> |
```

## `.removeClass()`

Permet de supprimer une ou plusieurs classes.

La syntaxe :

```
$(selector).removeClass(classname)
```

## Exemple 1

Permet de supprimer une ou plusieurs classes.

The screenshot displays a web browser window with a red header bar containing navigation links: "jQuery: Getting Started", "Hello World", "Example Page", and "Resources". Below the header, there are tabs for "Content", "HTML", "CSS", and "JavaScript". The main content area shows a real estate listing form for "CODED homes". The form includes a photo of a house, the address "15509 Main Street, Anytown, CO 80525", and a table of features:

Bedrooms	6
Bathrooms	4,5
Square Feet	4,500

Below the table, there is a description: "In a quiet neighborhood next to excellent schools." and a list of "SPECIAL FEATURES":

- Crown molding throughout
- Granite counter tops
- New carpet and tile
- Shutter blinds on all windows

The form also includes fields for "SHOWING DATE" (mm/dd/yyyy), "Owner present?" (checkbox), "NOTES" (text area), and "AVAILABILITY" (radio buttons for "Immediate" and "Delayed"). At the bottom, there are "Cancel" and "Save" buttons.

On the right side of the browser, the "Elements" panel is open, showing the DOM tree. The selected element is an 

## tag with the class "expanded". The code in the panel is:

```
> $('h2').addClass('highlight');  
< [   
    <h2 class="expanded highlight">15509  
    Main Street</h2>  
  ]  
> $('h2').removeClass('highlight');  
< [   
    <h2 class="expanded"> 15509 Main  
    Street</h2>  
  ]  
> |
```



## Exemple 2


Autre syntaxe pour ajouter/supprimer une classe.

The screenshot displays a web browser window with a real estate listing page on the left and the Chrome DevTools 'Elements' panel on the right. The listing page, titled '15509 Main Street', includes a photo of a house, a table of specifications (6 bedrooms, 4.5 bathrooms, 4,500 square feet), and a list of special features. The 'Elements' panel shows the DOM tree with the address '15509 Main Street' highlighted. The code in the panel demonstrates the use of jQuery's `addClass` and `removeClass` methods with a function argument to dynamically add or remove the 'bordered' class.

**jQuery: Getting Started** | **Hello World** | **Example Page** | **Resources**

Content | **HTML** | CSS | JavaScript

**CODED** themes



**15509 Main Street**  
Anytown, CO 80525

Bedrooms	6
Bathrooms	4,5
Square Feet	4,500

In a quiet neighborhood next to excellent schools.

**SPECIAL FEATURES**

- Crown molding throughout
- Granite counter tops
- New carpet and tile
- Shutter blinds on all windows

**SHOWING DATE**  
mm/dd/yyyy

☐ Owner present?

**NOTES**  
Notes

**AVAILABILITY:**  
☒ Immediate  
☐ Delayed

Cancel **Save**

**Elements** | Network | >> | >≡ | ⚙️ | 🖨️ | ✕

< top frame > | ▾ | ☐ Preserve log

```
> $('h2').addClass('highlight');  
< [  
  <h2 class="expanded highlight">15509  
    Main Street</h2>  
]  
> $('h2').removeClass('highlight');  
< [  
  <h2 class="expanded">15509 Main  
    Street</h2>  
]  
:  
> $('h2').addClass(function(){  
  return 'bordered';  
});  
< [  
  <h2 class="expanded bordered">15509 Main  
    Street</h2>  
]  
> $('h2').removeClass(function(){  
  return 'bordered';  
});  
< [  
  <h2 class="expanded">15509 Main  
    Street</h2>  
]  
>
```

## .hasClass()

La méthode **hasClass()** vérifie la présence d'une classe CSS sur un ou des élément(s) sélectionné(s). La méthode renvoie « *true* » si une classe est spécifiée pour le ou les élément(s) sélectionné(s), et « *false* » dans le cas contraire.

La syntaxe :

**\$(*selector*).hasClass(*classname*)**

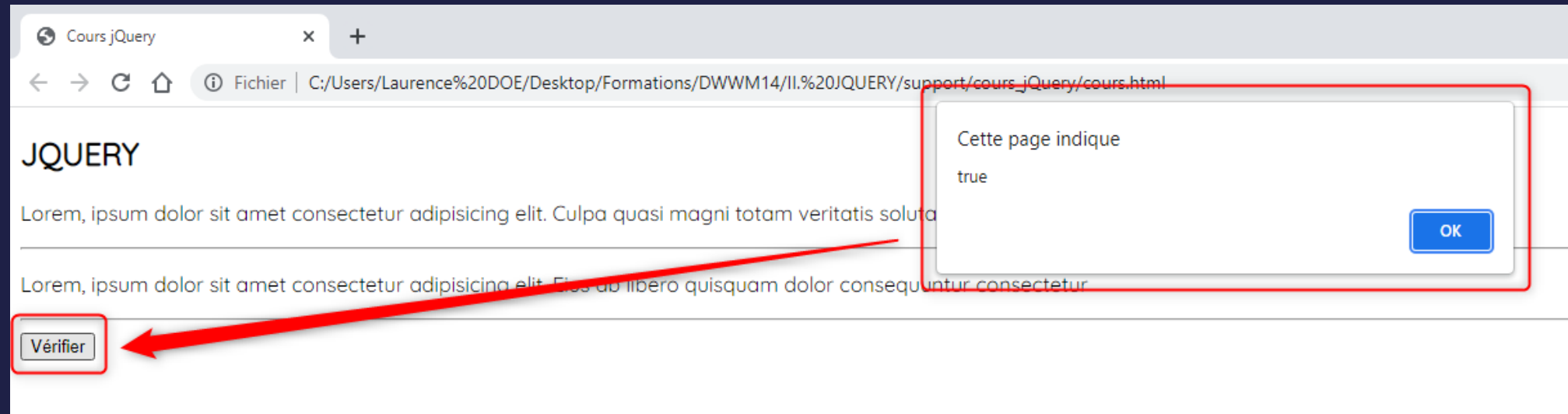
# Exemple

```
<? cours.html X JS cours.js # cours.css
cours.html > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3 > <head> ...
12 </head>
13 <body>
14 <h2>JQUERY</h2>
15 <!--Ecrire une fonction pour vérifier la présence ou non
16 de la classe intitulé "classe1" sur les éléments p -->
17 <div>
18 <p class="classe1">Lorem, ipsum dolor sit amet consectetur adipisicing elit.
19 Culpa quasi magni totam veritatis soluta, ea corporis</p>
20 <hr>
21 <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit.
22 Eius ab libero quisquam dolor consequuntur consectetur</p>
23 <hr>
24 <button>Vérifier</button>
25 </div>
26
27
28 <script src="https://code.jquery.com/jquery-3.6.0.min.js"
29 integrity="sha256-/xUj+30jU5yEx1q6GSYGSHk7tPXikynS7ogEvDej/m4="
30 crossorigin="anonymous"></script>
31 <script src="cours.js"></script>
32 </body>
33 </html>
```

```
<? cours.html JS cours.js X # cours.css
JS cours.js > ...
1 $(document).ready(function(){
2
3     $("button").click(function(){
4         alert($("p").hasClass("classe1"));
5     });
6
7 });
```

# Exemple

## Navigateur



## `.toggleClass()`

Permet de basculer entre l'ajout et la suppression d'une ou plusieurs classes pour le ou les élément(s) sélectionné(s).

- Ajoute la classe si elle n'est pas présente
- Supprime la classe si elle est présente
- Si plusieurs classes sont spécifiées, elles sont séparées par des virgules.

La syntaxe :

`$(selector).toggleClass(classname)`

## Exemple

```
13 <body>
14   <h2>JQUERY</h2>
15   <!-- -->
16   <div>
17     <p id="p1">Lorem ipsum dolor.</p>
18     <p id="p2">Sit amet consectetur.</p>
19   </div>
20   <button>Changer de classe</button>
21   <p><i>Note : Cliquer plusieurs fois pour l'effet de bascule</i></p>
```

<> cours.html JS cours.js X # cours.css

JS cours.js > ready() callback

```
1 $(document).ready(function(){
2   $("button").click(function(){
3     $("p").toggleClass("main");
4   });
5 });
```

```
14 .main
15 {
16   font-size:150%;
17   color: red;
18 }
```

## Exemple

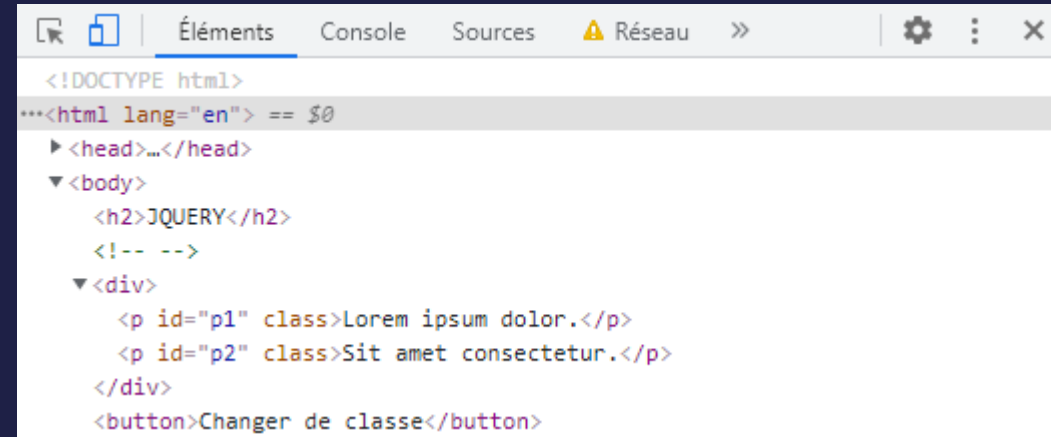
## JQUERY

Lorem ipsum dolor.

Sit amet consectetur.

Changer de classe

*Note : Cliquer plusieurs fois pour l'effet de bascule*



```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <h2>JQUERY</h2>
    <!-- -->
    <div>
      <p id="p1" class>Lorem ipsum dolor.</p>
      <p id="p2" class>Sit amet consectetur.</p>
    </div>
    <button>Changer de classe</button>
  </body>
</html>
```

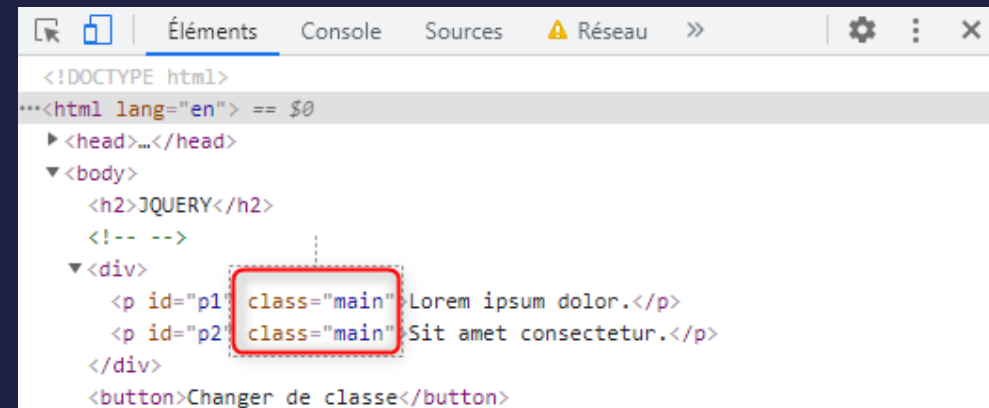
## JQUERY

Lorem ipsum dolor.

Sit amet consectetur.

Changer de classe

*Note : Cliquer plusieurs fois pour l'effet de bascule*



```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <h2>JQUERY</h2>
    <!-- -->
    <div>
      <p id="p1" class="main">Lorem ipsum dolor.</p>
      <p id="p2" class="main">Sit amet consectetur.</p>
    </div>
    <button>Changer de classe</button>
  </body>
</html>
```

# Les méthodes d'animation

Contenu	Style	Animation
.text()	.css()	.hide()
.attr()	.height()	.show()
.removeAttr()	.width()	.toggle()
.html()	.addClass()	.fadeIn()
.val()	.removeClass()	.fadeOut()
	.hasClass()	.slideDown()
	.toggleClass()	.slideUp()
		.slideToggle()



## `.hide()`

Permet de masquer/cacher un ou des élément(s) sélectionné(s).

La syntaxe :  
`$(selector).hide()`

## Exemple



15509 Main Street

Anytown, CO 80525

Bedrooms	6
Bathrooms	4.5
Square Feet	4,500

jQuery: Getting StartedHello WorldExample PageResources

ContentHTMLCSSJavaScript



SHOWING DATE

mm/dd/yyyy

☐ Owner present?

NOTES

Notes

AVAILABILITY:

☒ Immediate

☐ Delayed

CancelSave

Anytown, CO 80525

Bedrooms	6
Bathrooms	4.5
Square Feet	4,500

In a quiet neighborhood next to excellent schools.

SPECIAL FEATURES

- Crown molding throughout
- Granite counter tops
- New carpet and tile
- Shutter blinds on all windows

ElementsNetwork>>

☒ Preserve log

```
> $('h2').hide()
< [
  <h2 class="expanded" style="display: none;">15509 Main Street</h2>
]
```

## `.show()`



Permet d'afficher un ou des élément(s) qui a(ont) été masqué(s).

La syntaxe :

`$(selector).show()`

jQuery: Getting StartedHello WorldExample PageResources

ContentHTMLCSSJavaScript



SHOWING DATE

mm/dd/yyyy

☐ Owner present?

NOTES

Notes

AVAILABILITY:

☒ Immediate

☐ Delayed

CancelSave

15509 Main Street

Anytown, CO 80525

Bedrooms	6
Bathrooms	4.5
Square Feet	4,500

In a quiet neighborhood next to excellent schools.

SPECIAL FEATURES

- Crown molding throughout
- Granite counter tops
- New carpet and tile
- Shutter blinds on all windows

ElementsNetwork>>⌵⚙️🖨️

<top frame> ▾ ☐ Preserve log

> \$('h2').hide()

< [

<h2 class="expanded" style="display: none;">15509 Main Street</h2>

]

> \$('h2').show()

< [

<h2 class="expanded" style="display: block;">15509 Main Street</h2>

]

> |

## .toggle()

Permet de basculer entre les méthodes `hide()` et `show()`. Il gère les cycles de sélection et dé-sélection d'un élément par le clic souris. On peut dire qu'il bascule un élément du statut invisible à visible et inversement.

La syntaxe :



```
$(selector).toggle()
```

```
$(selector).toggle(speed, easing, callback)
```

# Exemple

jQuery: Getting StartedHello WorldExample PageResources

ContentHTMLCSSJavaScript



Anytown, CO 80525

Bedrooms	6
Bathrooms	4.5
Square Feet	4,500

In a quiet neighborhood next to excellent schools.

SPECIAL FEATURES

- Crown molding throughout
- Granite counter tops
- New carpet and tile
- Shutter blinds on all windows

SHOWING DATE

mm/dd/yyyy

☐ Owner present?

NOTES

Notes

AVAILABILITY:

☒ Immediate

☐ Delayed

Cancel Save

ElementsNetwork»

Preserve log

```
$( 'h2' ).toggle(function(){
  console.log('toggled');
})
[
  <h2 class="expanded" style="display:
    block;">15509 Main Street</h2>
]
toggledVM346:3
```

## Exemple

```
13 <body>
14   <h2>JQUERY</h2>
15   <!-- -->
16   <div>
17     <p id="p1">Lorem ipsum dolor.</p>
18     <p id="p2">Sit amet consectetur.</p>
19   </div>
20   <button>Cacher le titre</button>
```

```
<> cours.html JS cours.js X # cours.css
JS cours.js > ...
1 $(document).ready(function(){
2   $("button").click(function(){
3     $("h2").toggle();
4   });
5 });
```

### JQUERY

Lorem ipsum dolor.

Sit amet consectetur.

Cacher le titre

Lorem ipsum dolor.

Sit amet consectetur.

Cacher le titre

### JQUERY

Lorem ipsum dolor.

Sit amet consectetur.

Cacher le titre

## .fadeOut()

Permet de cacher les éléments ciblés en les rendant progressivement transparents (disparition avec effet de fondu).

La syntaxe :

`$(selector).fadeOut()`



`$(selector).fadeOut(speed, easing, callback)`



# Exemple 1

jQuery: Getting StartedHello WorldExample PageResources

ContentHTMLCSSJavaScript



Anytown, CO 80525

Bedrooms	6
Bathrooms	4,5
Square Feet	4,500

In a quiet neighborhood next to excellent schools.

SPECIAL FEATURES

- Crown molding throughout
- Granite counter tops
- New carpet and title
- Shutter blinds on all windows

SHOWING DATE

mm/dd/yyyy

☐ Owner present?

NOTES

Notes:

AVAILABILITY:

☒ Immediate

☐ Delayed

Cancel Save

ElementsNetwork»

<top frame> Preserve log

```
> $('h2').fadeOut();  
< [  
  <h2 class="expanded" style="opacity:  
    0.975528258147577;">15509 Main  
    Street</h2>  
> ]
```

## `.fadeIn()`

Affiche progressivement des éléments cachés en les opacifiant progressivement (apparition avec effet de fondu).

La syntaxe :


```
$(selector).fadeIn()
```

```
$(selector).fadeIn(speed, easing, callback)
```

## Exemple 2

jQuery: Getting StartedHello WorldExample PageResources

ContentHTMLCSSJavaScript



15509 Main Street  
Anytown, CO 80525

Bedrooms	6
Bathrooms	4,5
Square Feet	4,500

In a quiet neighborhood next to excellent schools.

**SPECIAL FEATURES**

- Crown molding throughout
- Granite counter tops
- New carpet and title
- Shutter blinds on all windows

SHOWING DATE

mm/dd/yyyy

☐ Owner present?

NOTES

Notes

AVAILABILITY:

☒ Immediate

☐ Delayed

Cancel Save

ElementsNetwork»



Filter icon <top frame> Preserve log

```
> $('h2').fadeOut();
< [
  <h2 class="expanded" style="opacity:
0.975528258147577;">15509 Main
Street</h2>
]
> $('h2').fadeIn();
< [
  <h2 class="expanded" style="display:
block; opacity:
0.0232729138404994;">15509 Main
Street</h2>
]
>
```

## Exemple 3

jQuery: Getting StartedHello WorldExample PageResources

ContentHTMLCSSJavaScript



Anytown, CO 80525

Bedrooms	6
Bathrooms	4.5
Square Feet	4,500

In a quiet neighborhood next to excellent schools.

SPECIAL FEATURES

- Crown molding throughout
- Granite counter tops
- New carpet and title
- Shutter blinds on all windows

SHOWING DATE

mm/dd/yyyy

☐ Owner present?

NOTES

Notes

AVAILABILITY:

☒ Immediate

☐ Delayed

Cancel Save

ElementsNetwork»



Preserve log

```
> $('h2').fadeOut();
< [
  <h2 class="expanded" style="opacity:
    0.975528258147577;">15509 Main
    Street</h2>
]
> $('h2').fadeIn();
< [
  <h2 class="expanded" style="display:
    block; opacity:
    0.0232729138404994;">15509 Main
    Street</h2>
]
> $('h2').fadeOut('fast');
< [
  <h2 class="expanded" style="display:
    block; opacity:
    0.938153340021932;">15509 Main
    Street</h2>
]
> |
```

## Exemple 4

jQuery: Getting StartedHello WorldExample PageResources

ContentHTMLCSSJavaScript



## 15509 Main Street

Anytown, CO 80525

Bedrooms	6
Bathrooms	4.5
Square Feet	4,500

In a quiet neighborhood next to excellent schools.

SHOWING DATE

☐ Owner present?

NOTES

AVAILABILITY:

☒ Immediate

☐ Delayed

CancelSave

SPECIAL FEATURES

- Crown molding throughout
- Granite counter tops
- New carpet and tile
- Shutter blinds on all windows

ElementsNetwork»

Preserve log

```
> $('h2').fadeIn(function(){
  $('#special-
  features').addClass('highlight');
});
< [
  <h2 class="expanded" style="display:
  none;">15509 Main Street</h2>
]
```

## `.slideDown()`

Permet de **faire apparaître** (ou déplier) un élément **par glissement vers le bas**.

La syntaxe :

```
$(selector).slideDown()
```

```
$(selector).slideDown(speed, easing, callback)
```

## Exemple

```
<> cours.html  JS cours.js  X
JS cours.js > ...
1  $(document).ready(function(){
2
3      //Déplie le titre en 2 secondes lors du clic sur #b2
4      $("#b2").click(function(){
5          $("h1").slideDown(2000);
6      });
7  });
```

## `.slideUp()`

Permet, à l'inverse de *slideDown* de **faire disparaître** (ou déplier) un élément par **glissement vers le haut**.

La syntaxe :

```
$(selector).slideUp()
```

```
$(selector).slideUp(speed, easing, callback)
```



## Exemple

<> cours.html

JS cours.js X

# cours.css

JS cours.js > ...

```
1 $(document).ready(function(){  
2   | $("button").click(function(){  
3   |   | $("p").slideUp();  
4   |   });  
5   });
```

## `.slideToggle()`

Permet de déplier progressivement un élément plié ou au contraire de replier progressivement un élément déplié. C'est **l'effet de bascule** entre les évènements `slideUp` et `slideDown`.

La syntaxe :

```
$(selector).slideToggle()
```

```
$(selector).slideToggle(speed, easing, callback)
```

## Exemple

&lt;&gt; cours.html

JS cours.js

X

# cours.css

JS cours.js &gt; ...

```
1 $(document).ready(function(){  
2   | $("button").click(function(){  
3   |   | $("p").slideToggle();  
4   |   });  
5   });
```

👉 Rappel : Ces méthodes acceptent jusqu'à **trois arguments** qui vont être de même type que les précédentes :

- **durée**
- **fonction d'accélération**
- **fonction de rappel** (exécutée une fois l'effet terminé).