

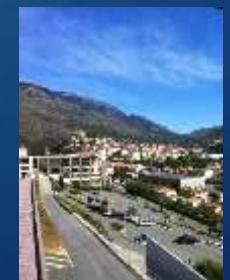
Cours de programmation web

PAUL-ANTOINE BISGAMBIGLIA

2021-2022 V1 2015-2016

[HTTP://PAUL-ANTOINE-BISGAMBIGLIA.UNIV-CORSE.FR/](http://PAUL-ANTOINE-BISGAMBIGLIA.UNIV-CORSE.FR/)

@PABISGAMBIGLIA



Licence



- ▶ Cours en licence libre
- ▶ Certaines images ont été récupérées sur google image avec le filtre « réutilisation utilisée sans but commercial »
- ▶ Certains exemples de code sont issus de sites web référencés en fin de chapitre via leurs urls

Objectifs

- ▶ Approfondir les notions de programmation web
- ▶ Apprendre à créer un site internet riche
- ▶ Technologies utilisées
 - ▶ HTML5
 - ▶ CSS3
 - ▶ JavaScript
 - ▶ PHP
 - ▶ MySQL
 - ▶ Des frameworks



Prérequis

Master 1

- ▶ Cours de L3
 - ▶ Algorithmique et programmation
 - ▶ Notion de réseau
 - ▶ Communication client serveur
 - ▶ Protocole HTTP
- ▶ Techo web : HTML/CSS/JS/PHP

Licence 3

- ▶ Notion d'algorithmique et de programmation
- ▶ Notion de réseau
 - ▶ Communication client serveur
 - ▶ Protocole HTTP

Programme

- ▶ Introduction et révisions
 - ▶ Le web, et ses techno
- ▶ HTML 5
 - ▶ Langage de balise <body> </body>
- ▶ CSS 3 avec bootstrap
 - ▶ Mise en forme
 - ▶ Ergonomie du web
- ▶ JavaScript avec jQuery
 - ▶ Dynamisme coté client
 - ▶ Angular.js
 - ▶ Node.js
- ▶ PHP
 - ▶ Dynamisme coté serveur
 - ▶ Les frameworks (Zend, Symfony, cakephp)
- ▶ Le +
 - ▶ les Design Patterns pour le web
 - ▶ Sécuriser son code
 - ▶ Django le framework web en python
 - ▶ Framework html 5 pour mobile (IONIC, Mobile Angular UI, Intel XDK, Appcelerator Titanium, Sencha Touch)
 - ▶ Apache Cordova, Meteor

Frameworks JS

Frameworks JS

[http://www.discoversdk.com/
blog/6-best-javascript-
frameworks-to-learn-in-2016/](http://www.discoversdk.com/blog/6-best-javascript-frameworks-to-learn-in-2016/)

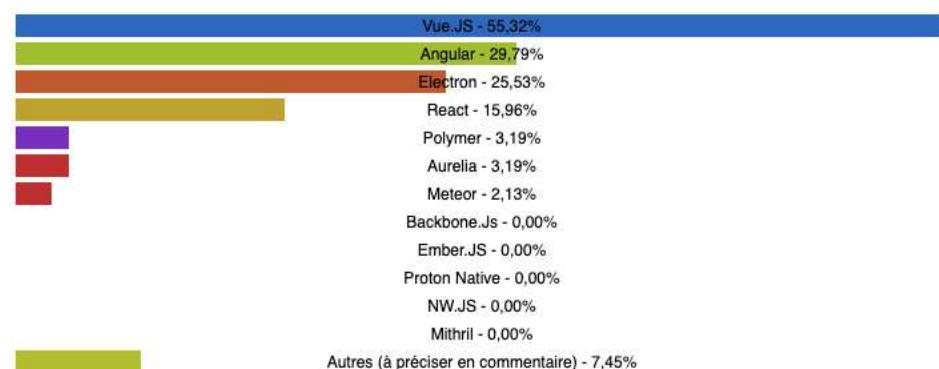
<https://colorlib.com/wp/javascript-frameworks/>

[http://noeticforce.com/best-
Javascript-frameworks-for-
single-page-modern-web-
applications](http://noeticforce.com/best-Javascript-frameworks-for-single-page-modern-web-applications)

<http://tutorialzine.com/2015/12/the-languages-and-frameworks-you-should-learn-in-2016/>

1. **AngularJS**
2. **React**
3. **Ember**
4. **Adonis**
5. **Vue.js**
6. **Backbone.js**
7. **Mobservable**
8. **Polymer 1.0**
9. **Ractive.js**
10. **Mercury**
11. **Aurelia.js**
12. **Meteor.js**

Quels sont les frameworks JavaScript que vous aimeriez apprendre en 2020 ?



Frameworks JS

[http://www.lafabriquedunet.fr
/blog/barometre-
technologies-web-tendances-
2015/](http://www.lafabriquedunet.fr/blog/barometre-technologies-web-tendances-2015/)

[https://www.classicinformatics
.com/top-javascript-
frameworks-2018/](https://www.classicinformatics.com/top-javascript-frameworks-2018/)

LIBRAIRIES JS

Nom	Tout le web	Top 10k	Top 100k	Top Million
jQuery	34,1%	12,4%	17,1%	20,1%
Google Libraries	6,8%	6,9%	7,8%	5,2%
html5shiv	6,2%	5,9%	6,4%	5,6%
jQuery UI	4,9%	5,1%	6,0%	6,7%
Facebook for Websites	4,6%	7,3%	7,3%	7,0%
Modernizr	3,6%	5,5%	5,4%	3,9%
jQuery Form	3,5%	1,6%	2,0%	2,7%
Facebook SDK	3,3%	6,1%	6,0%	5,7%
jQuery Easing	3,2%	1,5%	2,0%	2,5%
Fancybox	3,0%	2,0%	2,3%	3,0%
jQuery Cycle	2,3%	1,6%	2,1%	2,4%
Lightbox	2,1%	1,3%	1,7%	2,5%
FlexSlider	1,8%	1,2%	1,5%	1,6%
SuperFish	1,7%	0,8%	1,1%	1,6%

Source : [Builtwith](#)

FRAMEWORKS JS

Nom	Tout le web	Top 10k	Top 100k	Top Million
Angular JS	90,5%	47,3%	47,5%	47,5%
Backbone.js	6,6%	51,4%	50,8%	51,0%
Meteor	1,9%	0,1%	0,3%	0,3%
Ember	0,6%	1,0%	1,1%	1,0%
SailsJS	0,5%	0,3%	0,3%	0,2%

Source : [Builtwith](#)

API JS

Librairies JS

- ▶ La programmation JS avancée peut être très difficile et prendre du temps. Pour faire face à ces difficultés, il est possible d'utiliser des bibliothèques. Elles sont souvent appelées frameworks JavaScript.
- ▶ Liste des frameworks ici
<https://www.javascripting.com/>

Popular	New	Trending	Search
Angular.js	AngularJS lets you write client-side web applications as if you had a smarter browser. It let...	100	
React	React is a JavaScript library for building user interfaces.	99	
Moment	Parse, validate, manipulate, and display dates in javascript.	99	
Meteor	Meteor, an ultra-simple, database-everywhere, data-on-the-wire, pure-Javascript web fra...	99	
jQuery	jQuery JavaScript Library	97	
Foundation	The most advanced responsive front-end framework in the world. Quickly create prototyp...	97	
Ember	Ember.js is a JavaScript framework that does all of the heavy lifting that you'd normally hav...	97	
Backbone	Give your JS App some Backbone with Models, Views, Collections, and Events!	97	
Modernizr	Modernizr is a JavaScript library that detects HTML5 and CSS3 features in the user's brows...	96	
Selectz	Selectz is a jQuery based replacement for select boxes. It supports searching, remote dat...	96	
Underscore	JavaScript's utility belt	96	
Video.js	Universal video embed.	96	
Ionic	Advanced HTML5 Mobile App Framework. A beautiful front-end framework for developing...	95	

API JS

Librairies JS

- ▶ <https://rubygarage.org/blog/best-javascript-frameworks-for-front-end>

Name	Type	Shadow DOM EcmaScript 6+	Relative Popularity	Difficulty ofLearning
React	Library	Supported	****	****
Angular	Framework	Supported	***	****
Ember	Framework	Supported	*	****
Vue	Library	Supported	**	***
Backbone	Framework	Supported	*	***

jQuery

« THE WRITE LESS, DO MORE,
JAVASCRIPT LIBRARY. »

« SIMPLIFIEZ VOS
DÉVELOPPEMENTS
JAVASCRIPT AVEC JQUERY »

jQuery

jQuery est une bibliothèque JavaScript libre et multi-plateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web. La première version est lancée en janvier 2006 par John Resig.

La bibliothèque contient notamment les fonctionnalités suivantes :

- parcours et modification du DOM et des sélecteurs CSS 1 à 3 ;
- événements ;
- effets visuels et animations ;
- manipulations de CSS
- ajax ;
- plugins ;
- utilitaires

jQuery

- ▶ jQuery est une bibliothèque JavaScript.
- ▶ jQuery simplifie grandement la programmation JavaScript.
- ▶ jQuery est facile à apprendre.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>
</body>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.m
in.js"></script>
<script>
$(document).ready(function(){
    $("p").click(function(){
        $(this).hide();
    });
});
</script>
</html>
```

jQuery

But

- ▶ Le but de jQuery est de rendre plus facile l'utilisation du JS dans votre site web.
- ▶ jQuery simplifie le code JavaScript, comme les appels AJAX et la manipulation du DOM.
- ▶ La bibliothèque jQuery possède les fonctions suivantes: Manipulation du HTML/DOM, Manipulation CSS, Gestion d'événement, Effets et animations, AJAX, Utilitaires et plugin

- ▶ Utiliser jQuery :
 - ▶ Télécharger les fichiers

```
<head>
<script src="jquery-1.11.3.min.js"></script>
</head>
```

- ▶ Ou les inclure en ligne

```
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery
/1.11.3/jquery.min.js"></script>
<script
src="http://ajax.aspnetcdn.com/ajax/jQuery/jqu
ery-1.11.3.min.js"></script>
</head>
```

jQuery

Utilisation

- ▶ **`$(selector).action()`**
 - ▶ `$` utilisation de jQuery
 - ▶ `Selector` recherche ou requête sur un élément HTML
 - ▶ `action()` application d'une fonction

- ▶ `$(this).hide()`
 - ▶ cache l'élément courant
- ▶ `("p").hide()`
 - ▶ cache tous les éléments <p>
- ▶ `(".test").hide()`
 - ▶ cache tous les éléments de classe class="test".
- ▶ `("#test").hide()`
 - ▶ cache les éléments avec un id="test".

jQuery

Selecteurs

	Tous les éléments
<code>\$("")</code>	L'élément courant
<code>\$("p.intro")</code>	Balise <code><p></code> de classe intro
<code>\$("p:first")</code>	Le 1 ^{er} élément <code><p></code>
<code>\$("ul li:first")</code>	Le 1 ^{er} élément <code></code> du 1er <code></code>
<code>\$("ul li:first-child")</code>	Le 1 ^{er} élément <code></code> de chaque <code></code>
<code>\$("[href]")</code>	Tous les éléments avec un attribut href

```

<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $(".test").hide();
  });
});
</script>
</head>
<body>
<h2 class="test">This is a heading</h2>
<p class="test">This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>

```

jQuery

Evènements

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

jQuery

Evènements

- ▶ ready() : exécution après le chargement du document
- ▶ click() : au clique de souris
- ▶ dblclick()
- ▶ mouseenter() : la souris point sur l'élément
- ▶ mouseleave(): la souris quitte l'élément

```
$(document).ready(function(){  
    // jQuery methods go here...  
});  
  
$("p").click(function(){  
    // action goes here...  
});
```

jQuery

Evènements

- ▶ mousedown() : clique de souris sur l'élément
- ▶ mouseup() : relâche le bouton de la souris sur l'élément
- ▶ hover() : la souris passe sur un élément
- ▶ focus(), blur(), on()

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/
jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#p1").mousedown(function(){
    alert("Mouse down over p1!");
  });
});
</script>
</head>
<body>
<p id="p1">This is a paragraph.</p>
</body>
</html>
```

jQuery

Fonctions

- ▶ **\$(selector).hide(speed,callback);**
- ▶ **\$(selector).show(speed,callback);**
 - ▶ Speed = "slow", "fast", or milliseconds
 - ▶ callback = fonction à exécuter après
- ▶ **toggle()** affiche un élément caché, cache un élément affiché

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.m
in.js"></script>
<script>
$(document).ready(function(){
  $("#hide").click(function(){
    $("p").hide();
  });
  $("#show").click(function(){
    $("p").show();
  });
});
</script>
</head>
<body>
<p>If you click on the "Hide" button, I will disappear.</p>
<button id="hide">Hide</button>
<button id="show">Show</button>
</body>
</html>
```

jQuery

Fonctions de fondu

- ▶ fadeIn()
- ▶ fadeOut()
- ▶ fadeToggle()
- ▶ fadeTo()

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

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#div1").fadeToggle();
        $("#div2").fadeToggle("slow");
        $("#div3").fadeToggle(3000);
    });
});
</script>
</head>
<body>


Demonstrate fadeToggle() with different speed parameters.


<button>Click to fade in/out boxes</button><br><br>
<div id="div1" style="width:80px;height:80px;background-color:red;"></div>
<br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div>
<br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>
</body>
</html>
```

jQuery

Fonctions de mouvement

- ▶ `slideDown()`
- ▶ `slideUp()`
- ▶ `slideToggle()`

- ▶ `$(selector).slideDown(speed,callback);`

```
<!DOCTYPE html> <html> <head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#flip").click(function(){
        $("#panel").slideToggle("slow");
    });
});
</script>
<style>
#panel, #flip {
    padding: 5px;
    text-align: center;
    background-color: #e5eecc;
    border: solid 1px #c3c3c3;
}
#panel {
    padding: 50px;
    display: none;
}
</style>
</head>
<body>
<div id="flip">Click to slide the panel down or up</div>
<div id="panel">Hello world!</div>
</body>
</html>
```

jQuery

Fonctions d'animation

- ▶ `$(selector).animate({params},speed,callback);`

```
<!DOCTYPE html><html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("div").animate({
            left: '250px',
            opacity: '0.5',
            height: '150px',
            width: '150px'
        });
    });
}</script>
</head>
<body>
<button>Start Animation</button>
<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>
<div
style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>
</body>
</html>
```

jQuery

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.1
1.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        var div = $("div");
        div.animate({height: '300px', opacity: '0.4'},
"slow");
        div.animate({width: '300px', opacity: '0.8'},
"slow");
        div.animate({height: '100px', opacity: '0.4'},
"slow");
        div.animate({width: '100px', opacity: '0.8'},
"slow");
    });
});
```

</script>

</head>

<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>

<div style="background:blue;height:100px;width:100px;position:absolute;"></div>

</body>

</html>

jQuery

Fonctions d'animation

- ▶ `$(selector).animate({params},speed,callback);`

```
<!DOCTYPE html><html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("div").animate({
            left: '250px',
            opacity: '0.5',
            height: '150px',
            width: '150px'
        });
    });
}</script>
</head>
<body>
<button>Start Animation</button>
<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>
<div
style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>
</body>
</html>
```

jQuery

Fonctions d'animation

- ▶ `$(selector).stop(stopAll,goToEnd);`

```
<!DOCTYPE html><html><head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideDown(5000);
  });
  $("#stop").click(function(){
    $("#panel").stop();
  });
});
</script>
<style>
#panel, #flip {
  padding: 5px;
  font-size: 18px;
  text-align: center;
  background-color: #555;
  color: white;
  border: solid 1px #666;
  border-radius: 3px;
}
#panel {
  padding: 50px;
  display: none;
}
</style>
</head>
<body>
<button id="stop">Stop sliding</button>
<div id="flip">Click to slide down panel</div>
<div id="panel">Hello world!</div>
</body></html>
```

jQuery

Callback

- ▶ Le JS est exécuté ligne par ligne. Dans certain cas la ligne de code suivante peut être exécuté même si l'effet n'est pas fini. Cela peut créer des erreurs. Pour éviter cela, vous pouvez créer une fonction de rappel (callback). Les fonctions de callback sont exécutées après.

```
<!DOCTYPE html><html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/
1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").hide(1000);
        alert("The paragraph is now hidden");
    });
});
</script>
</head>
<body>
<button>Hide</button>
<p>This is a paragraph with little content.</p>
</body></html>
```

jQuery

Callback

- ▶ **`$(selector).hide(speed,callback);`**

```
<!DOCTYPE html><html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/
jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").hide("slow", function(){
            alert("The paragraph is now hidden");
        });
    });
}</script>
</head>
<body>
<button>Hide</button>
<p>This is a paragraph with little content.</p>
</body></html>
```

jQuery

CSS et liste de méthodes

```
$("#p1").css("color", "red")
        .slideUp(2000)
        .slideDown(2000);
```

```
<!DOCTYPE html><html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/
1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#p1").css("color",
"red").slideUp(2000).slideDown(2000);
    });
});
</script>
</head>
<body>
<p id="p1">jQuery is fun!!</p>
<button>Click me</button>
</body></html>
```

jQuery

Manipulation du CSS

- ▶ **addClass()** Ajoute une ou plusieurs classes aux éléments sélectionnés
- ▶ **removeClass()** Supprime une ou plusieurs classes aux éléments sélectionnés
- ▶ **toggleClass()** Bascule entre l'ajout / suppression de classes aux éléments sélectionnés
- ▶ **css()** Définit ou retourne un attribut de style

```
<!DOCTYPE html><html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.m
in.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("h1, h2, p").toggleClass("blue");
    });
});
</script>
<style>
.blue {
    color: blue;
}
</style></head><body>
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Toggle class</button>
</body></html>
```

jQuery

Manipulation du DOM

En lecture

- ▶ **text()** retourne le texte des éléments sélectionnés
- ▶ **html()** retourne le contenu des éléments sélectionnés (y compris les balises HTML)
- ▶ **val()** retourne la valeur du champs de formulaire
- ▶ **attr()** retourne l'attribut de l'élément

```
<!DOCTYPE html><html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#btn1").click(function(){
        alert("Text: " + $("#test").text());
    });
    $("#btn2").click(function(){
        alert("HTML: " + $("#test").html());
    });
});
</script>
</head>
<body>
<p id="test">This is some <b>bold</b> text in a
paragraph.</p>
<button id="btn1">Show Text</button>
<button id="btn2">Show HTML</button>
</body></html>
```

jQuery

Manipulation du DOM

En écriture

- ▶ **text("val")**
- ▶ **html("val")**
- ▶ **val("val")**
- ▶ **attr("nom", "val")**

```
<!DOCTYPE html><html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#btn1").click(function(){
        $("#test1").text("Hello world!");
    });
    $("#btn2").click(function(){
        $("#test2").html("<b>Hello world!</b>");
    });
    $("#btn3").click(function(){
        $("#test3").val("Dolly Duck");
    });
});
</script>
</head>
<body>
<p id="test1">This is a paragraph.</p>
<p id="test2">This is another paragraph.</p>
<p>Input field: <input type="text" id="test3" value="Mickey Mouse"></p>
<button id="btn1">Set Text</button>
<button id="btn2">Set HTML</button>
<button id="btn3">Set Value</button>
</body></html>
```

jQuery

Manipulation du DOM

Ajout d'éléments

- ▶ **append()** Inserts du contenu à la fin des éléments sélectionnés
- ▶ **prepend()** Inserts un contenu au début des éléments sélectionnés
- ▶ **after()** - insère un contenu après les éléments sélectionnés
- ▶ **before()** - Inserts du contenu avant les éléments sélectionnés

```
<!DOCTYPE html><html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("p").append("<b>Appended text</b>.");
  });

  $("#btn2").click(function(){
    $("ol").append("<li>Appended item</li>");
  });
});
</script>
</head>
<body>


This is a paragraph.



This is another paragraph.


<ol>
<li>List item 1</li>
<li>List item 2</li>
<li>List item 3</li>
</ol>
<button id="btn1">Append text</button>
<button id="btn2">Append list items</button>
</body></html>
```

jQuery

Manipulation du DOM

Suppression d'éléments

- ▶ **remove()** Supprime l'élément sélectionné (et ses éléments enfants)
- ▶ **empty ()** supprime les éléments enfants de l'élément sélectionné

```
$(“p”).remove(“.test”);
```

```
<!DOCTYPE html><html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#div1").remove();
    });
});
</script>
</head>
<body>
<div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow;">
This is some text in the div.
<p>This is a paragraph in the div.</p>
<p>This is another paragraph in the div.</p>
</div><br>
<button>Remove div element</button>
</body></html>
```

jQuery

Plugin

- ▶ [ui](#) (542)
- ▶ [jquery](#) (482)
- ▶ [form](#) (285)
- ▶ [animation](#) (273)
- ▶ [input](#) (252)
- ▶ [image](#) (210)
- ▶ [responsive](#) (184)
- ▶ [slider](#) (172)
- ▶ [ajax](#) (154)
- ▶ [scroll](#) (140)

jQuery UI <http://jqueryui.com/>

- ▶ jQuery UI est un ensemble d'éléments d'interactions pour la conception d'interface utilisateur : effets, widgets et thèmes.
- ▶ Que vous construisez des applications Web hautement interactives ou que vous ayez juste besoin d'ajouter un sélecteur de date à un contrôle de formulaire, jQuery UI peut vous faciliter la tâche.

jQuery

Plugin

- ▶ [ui](#) (542)
- ▶ [jquery](#) (482)
- ▶ [form](#) (285)
- ▶ [animation](#) (273)
- ▶ [input](#) (252)
- ▶ [image](#) (210)
- ▶ [responsive](#) (184)
- ▶ [slider](#) (172)
- ▶ [ajax](#) (154)
- ▶ [scroll](#) (140)

<http://jquerymobile.com/>

- ▶ jQuery Mobile est un système d'interface utilisateur basé sur HTML5 conçu pour rendre les sites Web réactifs (adaptable ou responsive) et les applications accessibles sur tous les périphériques : smartphone, tablette et PC de bureau.

Framework

TOP 2017 :

1. ANGULAR-JS
2. REACTJS
3. VUE.JS
4. EMBER.JS
5. MTERO.JS

<https://www.inteam.fr/blog/protege-top-5-frameworks-javascript-2017/>

Node

Nodejs

- ▶ Node.js est une plateforme logicielle libre et événementielle en JavaScript orientée vers les applications réseau qui doivent pouvoir monter en charge.
- ▶ Elle utilise la machine virtuelle v8 et implémente sous licence MIT les spécifications CommonJS.

- ▶ Node.js contient un serveur HTTP, ce qui permet de faire tourner un serveur web sans avoir besoin d'un logiciel externe comme Apache ou Lighttpd, et ainsi de mieux contrôler le fonctionnement de son serveur web.
- ▶ Il est utilisé par Groupon, SAP, LinkedIn, Microsoft, Yahoo!, Walmart, Rakuten et PayPal

Angular 1

<https://fr.wikipedia.org/wiki/AngularJS>

- ▶ AngularJS est un framework JavaScript libre et open-source développé par Google.
- ▶ AngularJS embarque un sous-ensemble de la bibliothèque open source jQuery appelé jQLite, mais peut aussi utiliser jQuery

AngularJS est fondé sur l'extension du langage HTML. Le code HTML étendu représente alors la partie « vue » du patron de conception MVC.

AngularJS utilise une boucle de dirty-checking (qui consiste à surveiller et à détecter des modifications sur un objet JavaScript) pour réaliser un data-binding bidirectionnel permettant la synchronisation automatique des modèles et des vues.

Angular 2

<http://blog.xebia.fr/2015/12/14/angular-2-presentation/>

- ▶ Objectifs :
 - ▶ Augmenter les performances
 - ▶ Améliorer la productivité
 - ▶ S'adapter au mobile
 - ▶ Embrasser les nouveaux standards du Web

src :

[https://www.synbioz.com/blog/angularjs
2 0 changements notables](https://www.synbioz.com/blog/angularjs-2-0-changements-notables)

- ▶ Première chose à noter, Google et Microsoft collaborent pour écrire cette version Angular 2.0 avec le langage TypeScript.
- ▶ Le MVW pattern est remplacé par le concept des components, pour lesquels les contrôleurs seront maintenant réservés

Angular 4

<https://angular.io/>

- ▶ Angular 4 est sorti en mars 2017
- ▶ applications plus petites et plus rapides
- ▶ Progressive Web Apps
- ▶ Build native mobile apps
- ▶ Create desktop-installed apps
- ▶ enrichit la syntaxe de la directive *ngIf

Installation

`npm install -g @angular/cli`

A lire

<https://blog.ippon.fr/2017/04/10/angular-4-quelques-nouveautés/>

Angular 5 / 6 / 7

<https://angular.io/>

- ▶ [Angular 4 est disponible](#) depuis le printemps 2017,
- ▶ [Angular 5 depuis le 1er novembre 2017](#),
- ▶ [Angular 6 depuis le 3 mai 2018](#) et
- ▶ [Angular 7 depuis le 18 octobre 2018](#).

<https://formationjavascript.com/angular-4-angular-5-angular-6-angular-7/>

Rétro-compatibilité Angular 4, 5, 6 & 7

Google demande désormais de parler d'**Angular tout court**, et d'AngularJS pour la première version.

<https://angularjs.org/>

Bootstrap et JS

JS + CSS

- ▶ Il existe de multiples possibilités de combinaison permettant d'appliquer des styles particuliers, de faire des menues dynamique, un carousel, etc.

```
<body>
<div class="container">
  <h2>Dynamic Tabs</h2>
  <ul class="nav nav-tabs">
    <li class="active"><a data-toggle="tab" href="#home">Home</a></li>
    <li><a data-toggle="tab" href="#menu1">Menu 1</a></li>
    <li><a data-toggle="tab" href="#menu2">Menu 2</a></li>
    <li><a data-toggle="tab" href="#menu3">Menu 3</a></li>
  </ul>

  <div class="tab-content">
    <div id="home" class="tab-pane fade in active">
      <h3>HOME</h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>
    </div>
    <div id="menu1" class="tab-pane fade">
      <h3>Menu 1</h3>
      <p>Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
    </div>
    <div id="menu2" class="tab-pane fade">
      <h3>Menu 2</h3>
      <p>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam.</p>
    </div>
    <div id="menu3" class="tab-pane fade">
      <h3>Menu 3</h3>
      <p>Eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.</p>
    </div>
  </div>
</div>
</body>
```

React

- ▶ développée par Facebook depuis 2013 en opensource depuis 2 ans
- ▶ Utilisé par Facebook, Instagram WhatsApp, Netflix, Yahoo, Airbnb, WordPress

Son objectif est de faciliter la création d'application web monopage, via des composants dépendant d'un état (State) et générant une page (ou portion) HTML à chaque changement d'état, page considérée comme la vue dans le modèle MVC.

Se démarque de ses concurrents par sa flexibilité et ses performances, en travaillant avec un DOM virtuel.

[https://fr.wikipedia.org/wiki/React_\(JavaScript\)](https://fr.wikipedia.org/wiki/React_(JavaScript))

VUE.JS

<https://www.inteam.fr/blog/protege-top-5-frameworks-javascript-2017/>

- ▶ Vue 2.0 a été introduit en 2016 et il a pris le meilleur de Ember, React et Angular, mettant tout cela dans un paquet pratique. Il est plus rapide et plus fin, comparé à React et Angular 2.0.

Vue.JS est une librairie qui permet de créer des interfaces web interactives en se reposant sur le principe des composants webs. Il propose un système de data-binding réactif capable de détecter automatiquement les changements d'états du composant.

<https://www.grafikart.fr/formations/vuejs>

Comment choisir

<https://www.codeur.com/blog/choisir-framework-javascript/>

Angular

- ▶ performants et robustes
- ▶ grande communauté
- ▶ modèles complets
- ▶ **apprentissage** long et ardu
- ▶ **trop complexe**
(nombreuses solutions offertes)

VueJS

- ▶ Simplicité
- ▶ documentation officielle claire et complète
- ▶ HTML et CSS compatible
- ▶ **manque d'historique** (il est assez récent)
- ▶ **communauté restreinte**

ReactJS

- ▶ ne gère que l'interface d'une application web
- ▶ progression permanente
- ▶ grande communauté
- ▶ la création est simplifiée grâce à sa flexibilité et son API qui permettent une utilisation facile des composants.
- ▶ DOM virtuel, permettant de gagner en performance et en rapidité.
- ▶ **modèles des composants sont en JSX et pas en HTML**
- ▶ **manque de documentation**

Programmation par composants

- Un composant est un élément de base pouvant être réutilisé, assemblé, combiné, étendu dans le but de faciliter un développement.
 - « l'**approche composant** doit permettre de trouver une structure globale, manipulable par des gens dont le métier est de composer des structures plutôt que d'inventer des éléments permettant de construire des structures. »
 - « La **programmation orientée composant** (POC) consiste à utiliser une approche modulaire de l'architecture d'un projet informatique, ce qui permet d'assurer au logiciel une meilleure lisibilité et une meilleure maintenance. » src : https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_composant

Data binding

- ▶ Il s'agit d'un moyen de lier la partie vue à la partie logique. En d'autres termes, grâce à cela, les éléments de votre code HTML seront liés à votre contrôleur JavaScript.
- ▶ Par exemple en ajoutant des directives comme attribut de balise (pour angular c'est ng-xxx)
- ▶ Ce mécanisme permet de mettre à jour des composants du DOM sans le modifier (gain de rapidité et de maintenabilité du code)
- ▶ A lire :
 - ▶ <http://sylvainpv.developpez.com/publications/data-binding/#LI>

Programme

- ▶ Introduction et révisions
 - ▶ Le web, et ses techno
- ▶ HTML 5
 - ▶ Langage de balise <body> </body>
- ▶ CSS 3 avec bootstrap
 - ▶ Mise en forme
 - ▶ Ergonomie du web
- ▶ JavaScript avec jQuery
 - ▶ Dynamisme coté client
 - ▶ Angular.js
 - ▶ Node.js
- ▶ PHP
 - ▶ Dynamisme coté serveur
 - ▶ Les frameworks (Zend, Symfony, cakephp, Phalcon)
- ▶ Le +
 - ▶ Les Design Patterns pour le web
 - ▶ Sécuriser son code
 - ▶ Django le framework web en python
 - ▶ Framework html 5 pour mobile (IONIC, Mobile Angular UI, Intel XDK, Appcelerator Titanium, Sencha Touch)