

JQuery

felipe@IFT3225 Hiver 2020



Your donations help fund the continued development and growth of jQuery.

[SUPPORT THE PROJECT](#)

[Download](#) [API Documentation](#) [Blog](#) [Plugins](#) [Browser Support](#)

Search



Lightweight Footprint

Only 30kB minified and gzipped. Can also be included as an AMD module



CSS3 Compliant

Supports CSS3 selectors to find elements as well as in style property manipulation



Cross-Browser

Chrome, Edge, Firefox, IE, Safari, Android, OS, and more



Download jQuery v3.4.1

The 1.x and 2.x branches no longer receive patches.

[View Source on GitHub](#) →

[How jQuery Works](#) →

What is jQuery?

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

Resources

- [jQuery Core API Documentation](#)
- [jQuery Learning Center](#)
- [jQuery Blog](#)
- [Contribute to jQuery](#)

Bonjour le monde

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
<title>Bonjour</title>
```

```
<script src="https://ajax.aspnetcdn.com/ajax/jquery/jquery-3.4.1.min.js"></script>
```

```
<script>
```

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

```
    document.write("bonjour");
```

```
  });
```

```
</script>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

quand la DOM est prête à être scriptée

\$(sélection) ou **jQuery(sélection)**

retourne un **jQuery Object**

Ex #1

```
<script type="text/javascript">

    $(document).ready(

        function(){

            $("a").click(
                function(event){
                    var link = $(this).attr("href");
                    alert("want to go to " + link + " ?");
                    event.preventDefault();
                });

        });

</script>
</head>

<body>
    <a href="http://jquery.com/">jQuery</a>
    <a href="http://www.iro.umontreal.ca/">DIRO</a>
</body>
```

équiv. de **onclick**
équiv. de **attributes**

jQuery()

```
$( "<p id='test'>My <em>new</em> text</p>" ).appendTo( "body" );
```

équivalent de **innerHTML**

Ex #2

```
<script>
  $(document).ready(function(){
    $("label + input").css("color", "blue").val("Labeled!")
  });
</script>
```

Fini les boucles !

Observez le chaînage

```
</head>
```

```
<body>
  <form>
    <label>Name:</label>
    <input name="name" />
    <input name="name" />
    <fieldset>
      <label>Newsletter:</label>
      <input name="newsletter" />
    </fieldset>
  </form>
  <input name="none" />
</body>
```

chaque fonction retourne un **jQuery Object**

Ex #4

```
<script>
  $(document).ready(function(){
    $("button[disabled]").next().text("this button is
disabled");
  });
</script>

</head>
<body>
  <div><button disabled="disabled">First</button> -
<span></span> <span> test</span></div>
  <div><button>Second</button> - <span></span></div>
  <div><button disabled="disabled">Third</button> -
<span></span></div>
</body>
```

Ex #5

```
<script>
```

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

jQuery Object

```
    jQuery.fn.showTags = function (n) {  
        var tags = this.map(function () {  
            return this.tagName;  
        })
```

Extension de jQuery
(javascript after all)

```
        .get().join(", ");
```

```
        $("b:eq(" + n + ")").text(tags);
```

```
        return this;
```

ce qui permet le chaînage

```
    };
```

```
    $("p").showTags(0)
```

```
        .find("span")
```

```
        .showTags(1)
```

```
        .css("background", "yellow")
```

```
        .end()
```

reset la dernière recherche

```
        .showTags(2)
```

```
        .css("font-style", "italic");
```

```
});
```

```
</script>
```


Quiz? (Ex #5)

`showTags (n)` récupère le nom des éléments de la sélection et ajoute ces noms dans le $n-1$ è `` du document.

```
$( "p" ).showTags ( 0 )  
    .find ( "span" )  
    .showTags ( 1 )  
    .css ( "background", "yellow" )  
    .end ( )  
    .showTags ( 2 )  
    .css ( "font-style", "italic" );
```

sélection: éléments `<p>`
sélection: `` dans ces `<p>`

sélection: éléments `<p>`

Quiz ! (Ex #5)

```
<body>
```

```
<p>
```

```
Hi there <span>how</span> are you <span>doing</span>?
```

```
</p>
```

```
<p>
```

```
This <span>span</span> is one of  
several <span>spans</span> in this  
<span>sentence</span>.
```

```
</p>
```

```
<div>
```

```
Tags in jQuery object initially: <b></b>
```

```
</div>
```

```
<div>
```

```
Tags in jQuery object after find: <b></b>
```

```
</div>
```

```
<div>
```

```
Tags in jQuery object after end: <b></b>
```

```
</div>
```

```
</body>
```

Hi there **how** are you **doing**?

This **span** is one of several **spans** in this **sentence**.

Tags in jQuery object initially: **P, P**

Tags in jQuery object after find: **SPAN, SPAN, SPAN, SPAN, SPAN**

Tags in jQuery object after end: **P, P**

.after()

- Ajoute un document fragment (ou l'équivalent) après tous les noeuds de la sélection

```
<div class="container">  
  <h2>Greetings</h2>  
  <div class="inner">Hello</div>  
  <div class="inner">Goodbye</div>  
</div>
```

```
$( ".inner" ).after( "<p>Test</p>" );
```



```
<div class="container">  
  <h2>Greetings</h2>  
  <div class="inner">Hello</div>  
  <p>Test</p>  
  <div class="inner">Goodbye</div>  
  <p>Test</p>  
</div>
```

- Si le noeud à ajouter est déjà dans la DOM, alors, si un seul élément dans la sélection, c'est juste un déplacement, sinon c'est un clonage (sauf le dernier élément) ...

```
<div class="container">  
  <h2>Greetings</h2>  
  <div class="inner">Hello</div>  
  <div class="inner">Goodbye</div>  
</div>
```

```
$( ".container" ).after( $( "h2" ) );
```



```
<div class="container">  
  <div class="inner">Hello</div>  
  <div class="inner">Goodbye</div>  
</div>  
<h2>Greetings</h2>
```

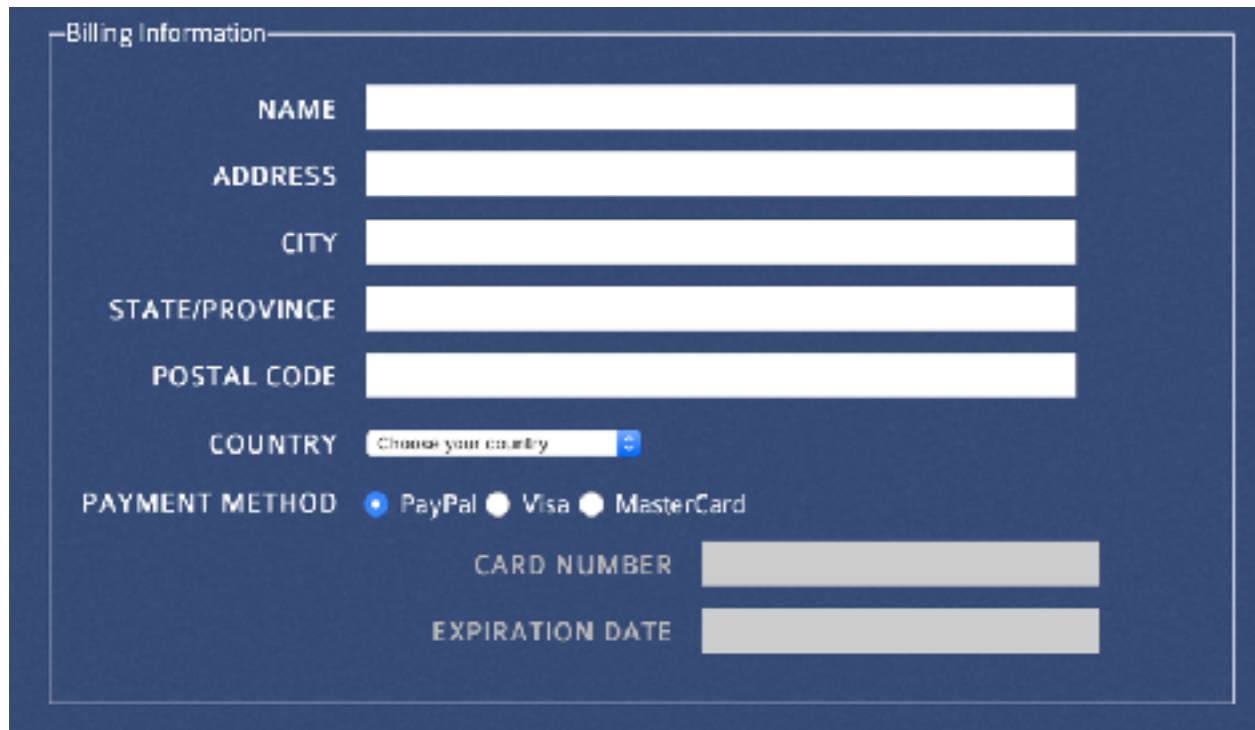
- Aussi possible de lui passer un fonctionneur (qui identifie un élément)

Pleins d'autres facilitateurs

```
<script>
$(document).ready(function() {
  $('.answer').hide();
  $('.main h2').toggle(
    function() {
      $(this).next('.answer').slideDown();
      $(this).addClass('close');
    },
    function() {
      $(this).next('.answer').fadeOut();
      $(this).removeClass('close');
    }
  ); // end toggle
}); // end ready
</script>
```

```
<h2>I've heard that JavaScript is the long-lost fountain of youth.
Is this true?</h2>
<div class="answer">
<p>Why, yes it is! Studies prove that learning JavaScript
freshens the mind and extends life span by several hundred
years. (Note: some scientists disagree with these claims.)
</p>
</div>
<h2>Can JavaScript really solve all of my problems?</h2>
```

Un beau formulaire



```
// pour éviter un clic  
$(':text:first').focus();
```

```
$('#paypal').click(function() { //disable credit card info  
    $('#creditCard input').attr('disabled', true).css('backgroundColor', '#CCC');  
    $('#creditCard label').css('color', '#BBB');  
}); // end click()
```

```
$('#visa, #mastercard').click(function() { //enable credit card info  
    $('#creditCard input').attr('disabled', false).css('backgroundColor', '');  
    $('#creditCard label').css('color', '');  
    $('#cardNumber').focus();  
}); // end click()
```

Un beau formulaire

Shipping Information

Same as billing

NAME

ADDRESS

CITY

STATE/PROVINCE

POSTAL CODE

COUNTRY

```
//hide shipping info
```

```
$('#hideShip').click(function() {  
    //check to make sure field is checked  
    if ($(this).attr('checked')) {  
        $('#shipping').slideUp('fast');  
    } else {  
        $('#shipping').slideDown('fast');  
    }  
}); // end click()
```

Validation d'un formulaire



Form validation with jQuery

Pas le plus moderne.
Fonctionnel. Léger

[HOME](#) [CONTRIBUTE](#) [DOCUMENTATION](#) [REFERENCE DOCUMENTATION](#)

jQuery Validation Plugin

This jQuery plugin makes simple clientside form validation easy, whilst still offering plenty of customization options. It makes a good choice if you're building something new from scratch, but also when you're trying to integrate something into an existing application with lots of existing markup. The plugin comes bundled with a useful set of validation methods, including URL and email validation, while providing an API to write your own methods. All bundled methods come with default error messages in english and translations into 37 other languages.

For a quick overview of what this plugin provides, check out this [Webucator](#) video (they also provide [jQuery trainings](#)):

RECENT POSTS

[Validator.destroy\(\)](#)

[step method](#)

[normalizer](#)

[jQuery.validator.methods](#)

[require_from_group method](#)

jQuery Validation

```
<script>
$(document).ready(function() {
  $('#signup').validate{
```

```
  rules: {
    email: {
      required: true,
      email: true
    }, ....
  },
```

```
    <div>
      <label for="email" class="label">E-mail Address</label>
      <input name="email" type="text" id="email">
    </div>
```

doit être de type email

```
  messages: {
    email: {
      required: "Please supply an e-mail address.",
      email: "This is not a valid email address."
    }, ....
  },
```

Les messages à afficher en cas d'erreur

```
  errorPlacement: function(error, element) {
    if ( element.is(":radio") || element.is(":checkbox")) {
      error.appendTo( element.parent());
    } else {
      error.insertAfter(element);
    }
  }
}
```

Si le placement par défaut n'est pas le bon

```
}); // end validate
```

Voir [ici](#)

Note

HTML5 offre déjà certains contrôles

```
<html>
<head>
  <meta charset="utf-8">
  <title>Favorite fruit with required and pattern attributes</title>
  <style>
    input:invalid {
      border: 2px dashed red;
    }

    input:valid {
      border: 2px solid black;
    }
  </style>
</head>

<body>
  <form>
    <label for="choose">Would you prefer a banana or a cherry?</label>
    <input id="choose" name="i_like" required pattern="[Bb]anana|[Cc]herry">
    <button>Submit</button>
  </form>
</body>
</html>
```

Sans javascript !

On peut cependant par javascript changer les messages

Bootstrap form validation

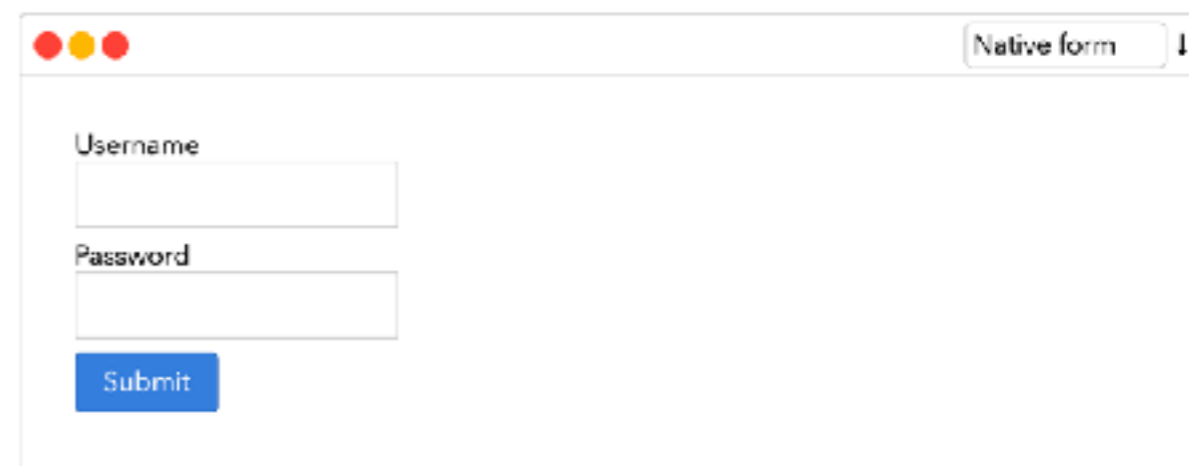
```
<script src="https://cdnjs.cloudflare.com/ajax/libs/es6-shim/0.35.3/es6-shim.min.js"></script>
<script src="/vendors/formvalidation/dist/js/FormValidation.min.js"></script>

<script>
document.addEventListener('DOMContentLoaded', function(e) {
  FormValidation.formValidation(
    document.getElementById('loginForm'),
    {
      fields: {
        username: {
          validators: {
            notEmpty: {
              message: 'The username is required'
            },
            stringLength: {
              min: 6,
              max: 30,
              message: 'The username must be more than 6 and less than 30 characters'
            },
            regexp: {
              regexp: /^[a-zA-Z0-9_]+$/,
              message: 'The username can only consist of alphabetical, number and underscore'
            }
          }
        },
        password: {
          validators: {
            notEmpty: {
              message: 'The password is required'
            },
            stringLength: {
              min: 8,
              message: 'The password must have at least 8 characters'
            }
          }
        }
      }
    }
  );
});
```

```
<form id="loginForm" method="POST">
  <div>
    <label>Username</label>
    <input type="text" name="username" />
  </div>

  <div>
    <label>Password</label>
    <input type="password" name="password" />
  </div>

  <button type="submit">Submit</button>
</form>
```



A screenshot of a browser window titled "Native form". The window contains a simple HTML form with two input fields. The first field is labeled "Username" and is a text input. The second field is labeled "Password" and is a password input. Below the fields is a blue "Submit" button. The browser's address bar is not visible, and the window has standard macOS-style window controls (red, yellow, green buttons) in the top-left corner.