

Jquery

Inicialmente, para usar o Jquery em sua máquina, é necessário fazer o download. Para isso, acesse o site <https://jquery.com/> e clique em download.



Fonte: <https://jquery.com.>

Uma nova tela abrirá; escolha a opção do primeiro link, Download the compressed, production jQuery 3.6.0.



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

SUPPORT THE PROJECT

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

Search

Downloading jQuery

Compressed and uncompressed copies of jQuery files are available. The uncompressed file is best used during development or debugging; the compressed file saves bandwidth and improves performance in production. You can also download a [source map file](#) for use when debugging with a compressed file. The map file is *not* required for users to run jQuery, it just improves the developer's debugger experience. As of jQuery 1.11.0/2.1.0 the `/// sourceMappingURL` comment is *not included* in the compressed file.

To locally download these files, right-click the link and select "Save as..." from the menu.

jQuery

For help when upgrading jQuery, please see the [upgrade guide](#) most relevant to your version. We also recommend using the [jQuery Migrate plugin](#).

[Download the compressed, production jQuery 3.6.0](#)

[Download the uncompressed, development jQuery 3.6.0](#)

[Download the map file for jQuery 3.6.0](#)

You can also use the slim build, which excludes the `ajax` and `effects` modules:

[Download the compressed, production jQuery 3.6.0 slim build](#)

Fonte: <https://jquery.com>.

Copie o código da página e cole-o em um arquivo vazio do bloco de notas.

Observação: Tenha cuidado com a opção “traduzir esta página” (para o português), pois pode alterar algumas funcionalidades do código.

[illegible]

Fonte: <https://jquery.com>.

Salve o arquivo com o nome `jquery-v3.6.0.min.js`. Caso você tenha baixado outra versão, lembre-se de alterar o número da versão, mas a extensão `.min.js` precisa ser mantida.

Nome:	<input type="text" value="jquery-v3.6.0.min.js"/>
Tipo:	<input type="text" value="JavaScript (*.js;*.es6;*.mjs;*.cjs;*.pac)"/>

Salvar arquivo

Chame o arquivo pela tag `<script>` em seu código HTML.

```
<script type="text/javascript" src="js/jquery-v3.6.0.min.js"></script>
<script type="text/javascript" src="js/script.js"></script>
```

Tag script

Sintaxe básica

O código JQuery tem início pelo símbolo “\$” (cifrão). Dessa forma, todo início de código será iniciado da seguinte maneira: `$(" ")`, ou seja, cifrao + parêntese + aspas + parêntese.

Sendo:

- \$ início da função JQuery;
- parênteses, dentro dos quais serão passados os parâmetros que serão alterados;
- por fim, as ações que irão ocorrer.

Nesse exemplo, temos o código escrito em HTML, no qual adicionamos uma *class* para a tag de paragrafo <p>.

```
<div class="section-banner-home">
  <section class="texto-exemplo">
    
    <h1>Venha conhecer nosso diferencial</h1>
    <p class="texto">Lorem, ipsum dolor sit amet consectetur adipisicing elit. Veritatis et adipisci
      dignissimos delectus quia consectetur saepe. Veritatis esse soluta impedit fuga eaque
      beatae nobis ea? Reiciendis exercitationem nisi necessitatibus eligendi.</p>
    <a class="btn-banner" href="#" target="_blank">Saiba mais</a>
  </section>
</div>
```

HTML

Imagine que você tem diversas *class* com o mesmo nome em seu código. Você terá a opção de selecionar todas (atalho no VSCode: ctrl + d) ou poderá alterar no seu código script, passando a seguinte linha:

```
$(".p").removeClass("texto").addClass("texto-paragrafo");
```

Jquery

Agora, siga o seguinte ao passo a passo:

1º. Iniciamos o código pelo \$ e passamos como primeiro parâmetro “p”, item que queremos alterar.

2º. Em seguida, informamos que queremos remover uma *class* e passamos seu nome (“texto”).

3º. Informamos, então, que queremos adicionar a *class* “texto-paragrafo”.

Dessa forma, serão removidas todas as *class* que tiverem o nome de “texto”, sendo adicionado “texto-paragrafo” em seu lugar.

Uma outra vantagem do uso do JQuery é o tamanho da função, quando comparado ao JavaScript, conforme mostra o exemplo a seguir:

```
//Javascript
function fadeIn(el){
    el.style.opacity = 0;

    var last = +new Date();
    var tick = function(){
        el.style.opacity = +el.style.opacity + (new Date() - last)/400;
        last = +new Date();

        if(+el.style.opacity < 1){
            (window.requestAnimationFrame && requestAnimationFrame(tick)) || setTimeout(tick,16);
        }
    };

    tick();
}
fadeIn(el);

//Jquery
$(el).fadeIn();
```

JavaScript – Fonte: Elaborado pela autora.

Ambos os códigos têm a mesma funcionalidade: alterar a opacidade de um elemento. No entanto, utilizando a biblioteca JQuery, o código fica simplificado.

Importante

No endereço <https://api.jquery.com/>, é possível encontrar vários recursos para a implementação de seu código.



Exemplo de função em formulário

HTML

```
<div class="container">
  <div class="row">
    <div class="col-md-8">
      <form class="form" id="formExample">
        <div class="form-group">
          <label class="col-md-4 control-label">Idade</label>
          <div class="col-md-6">
            <input type="number" class="form-control" id="idade" value="">
          </div>
        </div>
        <div class="form-group">
          <label class="col-md-4 control-label">Profissão</label>
          <div class="col-md-6">
            <input type="text" class="form-control" id="profissao">
            <div class="col-md-8">
              <button class="btn btn-primary" disabled>Login</button>
            </div>
          </div>
        </div>
      </form>
    </div>
  </div>
</div>
```

HTML

Observe que na tag <button>, há o atributo “disabled”. Iremos alterar seu estado com JQuery.

JS (JQUERY)

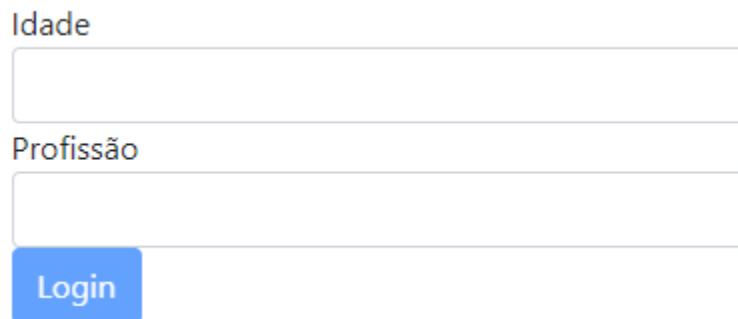
```
$('#formExample, idade, #profissao').change(function(){
  var idade = $('#idade').val();
  var profissao = $('#profissao').val();
  if(idade > 18 && profissao != null){
    $('.btn').removeAttr('disabled');
  }else{
    $('.btn').attr('disabled');
  }
});
```

Jquery

Na primeira linha, estamos chamando os ids e passando o método `.change`, que tem a função de manipular a alteração de um elemento. Nesse caso, queremos alterar o estado do botão que está desabilitado.

Para isso, criamos uma condição com *if*, que verificará se a idade é maior que 18 e se a profissão é diferente de nulo. Caso sejam, o atributo de “disabled” será removido; se não, permanecerá.

Observe as etapas de validações:



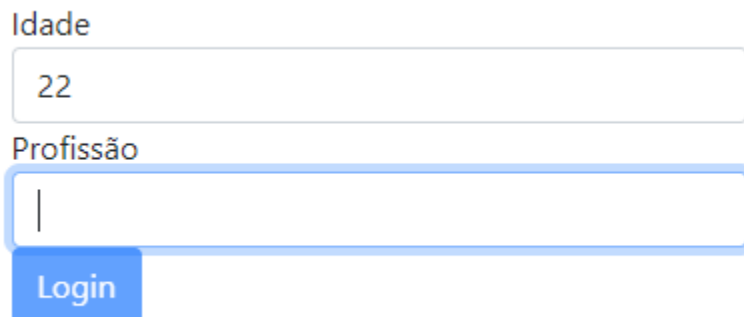
Idade

Profissão

Login

Navegador

Como não há dados, o botão está desabilitado.



Idade

Profissão

Login

Navegador

O primeiro campo foi preenchido, porém o segundo ainda não; logo, o botão permanece desabilitado.

Codificação Front-End

Idade

Profissão

Navegador

Após o preenchimento dos dois *inputs*, o botão foi habilitado. Agora, sigamos para um exemplo utilizando o operador OR. O código no HTML será o mesmo, o que iremos alterar será a lógica.

```
$('#formExample, idade, #profissao').change(function(){  
  var idade = $('#idade').val();  
  var profissao = $('#profissao').val();  
  if(idade > 18 || profissao != ""){  
    $('.btn').removeAttr('disabled');  
  }else{  
    $('.btn').attr('disabled');  
  }  
});
```

Jquery

Note que basta apenas uma condição ser verdadeira.

Idade

Profissão

Navegador

Exemplo de negação:

```
$('#formExample, idade, #profissao').change(function(){  
    var idade = $('#idade').val();  
    var profissao = $('#profissao').val();  
    if(!(idade > 18) && (profissao != "")){  
        $('.btn').removeAttr('disabled');  
    }else{  
        $('.btn').attr('disabled');  
    }  
});
```

Jquery

No navegador:

Idade

Profissão

Login

Navegador

Note que 19 é maior que 18, porém mesmo assim o botão continua desabilitado. Como a condição de idade foi negada, o valor lógico é alterado; logo, valores menores que 18 liberam o acesso. Observe:

Idade

Profissão

Login

Navegador