

jQuery Validate – Validando Formulários (2019)

O que é o jQuery Validate

Olá! O [jQuery Validate](#) é um plugin para jQuery que facilita a validação de formulários html com javascript.

Criado pelo pessoal do [bassistance.de](#) e previamente chamado de **jQuery Validation** esse plugin foi disponibilizado de forma [open source](#) está em constante atualização. A versão mais atual do [jQuery Validate](#) pode ser encontrada para download em sua [página de releases](#) ou se preferir ela está disponível também no [CDN.js](#)

Configurando o jQuery Validate em sua página

Para utilizar o [jQuery Validate](#), a primeira coisa que você precisa fazer é adicioná-lo a sua página html, porém por se tratar de um *plugin* do [jQuery](#), você precisa também importar o [jQuery](#). Você consegue fazer isso da seguinte forma:

```
<!DOCTYPE html>
<html>
<head>
  <title>Validando Formulários com jQuery Validate</title>
</head>
<body>

  <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.js">
</script>
  <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery-
validate/1.19.0/jquery.validate.js"></script>
</body>
</html>
```

Configurando o jQuery Validate em sua página

Como vocês podem perceber a importação **jQuery** e do **jQuery Validate** foram feitas antes do fechamento da tag `<body>`.

O Formulário

Agora que já temos as dependências importadas em nossa página, vamos criar um simples formulário html com os campos mais comuns de um formulário de cadastro, para que possamos fazer a validação na mesma página, sem recarregar.

Nesse formulário teremos **nome**, **email**, **senha**, **confirmação de senha** e por fim o **aceite de termos e condições de uso**. Com esses 5 campos, vamos conseguir aplicar algumas regras de validação, como **campos obrigatórios**, **campo com email válido**, **campo com confirmação** e por aí vai. Então vamos lá, faça o formulário da assim:

```
<form id="sign_up_form" method="post" action="" >
  <div class="form-group">
    <label for="name">Nome</label>
    <input type="text" class="form-control" id="name" name="name" placeholder="Nome">
  </div>
  <div class="form-group">
    <label for="email">Email</label>
    <input type="text" class="form-control" id="email" name="email" placeholder="Email">
  </div>
  <div class="form-group">
    <label for="password">Senha</label>
    <input type="password" class="form-control" id="password" name="password" placeholder="Senha">
  </div>
  <div class="form-group">
    <label for="password_confirmation">Confirmação de senha</label>
    <input type="password" class="form-control" id="password_confirmation" name="password_confirmation"
placeholder="Confirmação de senha">
  </div>
  <div class="form-check">
    <label class="form-check-label">
      <input type="checkbox" class="form-check-input" name="terms_of_use" id="terms_of_use">
      Termos de uso
    </label>
  </div>
  <button type="submit" class="btn btn-primary">Cadastrar</button>
</form>
```

O Formulário

Validando o Formulário

Com tudo devidamente adicionado ao html, chegou a hora de validar o formulário na mesma página usando javascript, com a ajuda do [jQuery](#) e do [jQuery Validate](#).

Para isso vamos abrir uma tag `<script type="text/javascript"></script>` e adicionar as nossas regras:

```
<script type="text/javascript">
$(document).ready( function() {
  $('#sign_up_form').validate({
    rules:{
      name:{
        required: true,
        minlength: 3
      },
      email: {
        required: true,
        email: true
      },
      password: {
        required: true
      },
      password_confirmation:{
        required: true,
        equalTo: "#password"
      },
      terms_of_use: "required"
    },
    messages:{
      name:{
        required: "O campo nome é obrigatorio.",
        minlength: "O campo nome deve conter no mínimo 3 caracteres."
      },
      email: {
        required: "O campo email é obrigatorio.",
        email: "O campo email deve conter um email válido."
      },
      password: {
        required: "O campo senha é obrigatorio."
      },
      password_confirmation:{
        required: "O campo confirmação de senha é obrigatorio.",
        equalTo: "O campo confirmação de senha deve ser identico ao campo senha."
      },
      terms_of_use: "Para se cadastrar você deve aceitar os termos de uso."
    }
  });
});
</script>
```

Validando o Formulário

Pronto! Agora é só rodar o código no browser e ver tudo funcionando. Com isso a gente conseguiu validar um formulário html, que pode ser feito em PHP, NodeJs ou qualquer outra linguagem de backend. Como tudo acontece no front-end a experiência do usuário fica muito mais agradável.





O código completo apenas é exibido no desktop, pois a leitura fica mais fácil, mas você pode acessar também pelo link direto do [Gist](#).

No código completo, eu adicionei também alguns estilos css, apenas para que a pagina fique mais apresentável.

Para saber mais...

Agora que você já sabe como utilizar o jquery validate, você pode recorrer a documentação para verificar os [diversos métodos de validação](#) que o *plugin* tem por padrão.

Se você gostou desse post, compartilhe com seus amigos, ajude a espalhar conhecimento! Não se esqueça de nos seguir nas redes sociais, por que você não pode ficar de fora!

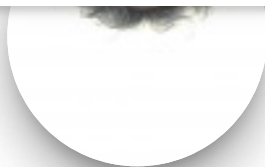
1. [Curta nossa página no facebook](#)
2. [Não perca as atualizações no twitter](#)
3. [Veja as dicas no instagram](#)
4. [Siga nossos repositórios no github](#)
5. [Faça parte da nossa lista de desenvolvedores](#)

Esse post foi publicado em 22 de maio de 2010, e teve uma atualização recente. Este é um post de cunho didático para quem está começando com o uso do jquery e o jquery validate, o código pode sim ser refatorado para melhor leitura, assim como também podemos usar as tags html5 para verificação de campos obrigatórios, email, ect. Enfim, tudo pode ser melhorado, mas devemos ir passo a passo.

Categorias: [JAVASCRIPT](#)

Tags: [broga](#) [javascript](#) [jquery valitation](#) [validação de formulario](#) [validar formulário](#) [validar javascript](#) [validate](#)





@emersonbroga

Formado em Desenvolvimento de Sistemas pela Faculdade Pitágoras e pós-graduado em Gestão Estratégica de Marketing pela PUC Minas trabalhei em diversas agências e atualmente trabalho em projetos internacionais como FOX.com, FXNetworks.com, NatGeo.com entre outros.

0 Comentários

Emerson Carvalho

1 Entrar ▾

♥ Recomendar

🐦 Tweet

f Compartilhar

Ordenar por Mais votados ▾



Iniciar a discussão...

FAZER LOGIN COM

OU REGISTRE-SE NO DISQUS (?)

Nome

Seja o primeiro a comentar.

✉ Inscreva-se D Adicione o Disqus no seu site Adicionar Disqus Adicionar

Pesquisar ...



Participe!

Faça parte da nossa lista de desenvolvedores! Junte-se à nossa lista de desenvolvedores e fique por dentro das últimas novidades! ⤴



Email:

SUBSCRIBE

Redes sociais

Facebook

Twitter

Instagram

YouTube

GitHub

Participe!

Tags

bitbucket **broga** coordenadas developer emersonbroga emersonbrogadev es6
git github githubcampusexperts githubdesign githubeducation githubpages
githubuniverse google iOS Javascript **javascript** javascript30 javascriptdeveloper
javascriptengineer javascriptlearning javascriptlover javascripts javascripttutorial **jquery** node
nodejs Objective-C **PHP** programacao programador programar prototype
react servidor **soudev** svn tools versionamento **Web** webdev Windows
Wordpress Zend Framework





 [Hospedagem BR](#)

 [Hospedagem Gringa](#)

 [Enviar Dinheiro](#)

 [Enviar Dinheiro para amigos](#)

 [Hospede-se](#)

 [Privacidade Online](#)

Posts relacionados



**JAVASCRIPT****[Vídeo] Como usar Promises no Javascript**

Melhore o seu javascript aprendendo a promises. Veja como criar uma promise e como utilizar uma ou várias promises. Entenda o resolve/reject e o then/catch.



JAVASCRIPT

Resumo da Semana

Olá, pessoal! Nesse post de hoje, quero apenas deixar registrado o que foi compartilhado essa semana. for Entenda como funciona o loop "for" no javascript! while Entenda como funciona o loop "while" no javascript! do...while [Leia mais...](#)



JAVASCRIPT

Manipulando DOM com `textContent`, `innerText` e `innerHTML`

Aprenda a manipular o DOM e saiba quando usar `innerHTML`, `innerText` e `textContent`. Veja a diferença e entenda como essas três propriedades funcionam.

FAÇA PARTE!

Hestia | Desenvolvido por Themelsle