Autofocus: fonte : http://diveintohtml5.com.br/detect.html

AUTO FOCO DE FORMULÁRIO

Os web sites podem usar JavaScript para dar foco ao primeiro campo de um formulário automaticamente. Por exemplo, a página principal do Google.com dá auto foco no campo de busca para você poder iniciar sua busca sem ter que posicionar o cursor do mouse no campo de busca. Enquanto isso pode ser conveniente para a maioria das pessoas, pode vir a ser incômodo para usuários experientes ou pessoas com necessidades especiais. Se você tentar pressionar a barra de espaço esperando que a página seja rolada para baixo, não ocorrerá conforme o esperado, pois o foco já está em um campo. (Acaba digitando um espaço no campo ao invés de rolar a página.) Se você der foco a um campo diferente antes de a página terminar de carregar, o script de auto foco pode “gentilmente” mover o foco de volta ao campo original ao fim do carregamento da página, atrapalhando seu fluxo de trabalho fazendo com que você digite no lugar errado.

Como o auto foco é feito através do JavaScript, pode ser trabalhoso ter que lidar com todos esses delicados casos, e não há muito o que fazer para evitar a página web de “roubar” o foco.

Para resoler esse problema, a HTML5 apresenta um atributo de autofocus em todos os controles de formulário . O atributo autofocus faz exatamente o que o seu nome diz: move o foco para um campo específico. Mas por ser apenas uma marcação ao invés de script, o comportamente será consistente entre todos os sites. Além disso, os desenvolvedores de navegadores podem oferecer formas de desabilitar o comportamente de auto foco.

A verificação de suporte ao auto foco utiliza a técnica de detecção #2. Se o seu navegador possui suporte ao auto foco, o objeto DOM criado para representar um elemento <input> terá uma propriedade autofocus (mesmo que você não inclua o atributo autofocus em seu HTML). Se o seu navegador não possui suporte ao auto foco, o objeto DOM criado para um elemento <input> não terá uma propriedade autofocus. Você pode detectar suporte ao auto foco usando essa função:

function supports\_input\_autofocus() {

var i = document.createElement('input');

return 'autofocus' in i;

}

Ao invés de escrever essa função, você pode usar o Modernizr (1.1 ou superior) para detectar o suporte a campos com auto foco.

↶ verificando suporte ao auto foco

if (Modernizr.input.autofocus) {

// autofocus funciona!

} else {

// sem suporte a autofocus :(

// contornar com uma solução com script

}

Fonte: http://www.kadunew.com/blog/html/html5-atributo-autofocus

O atributo autofocus é um atributo booleano, ou seja, pode ou não ser definido em um elemento. É usado em campos de formulários. **Especifica que um elemento deve receber automaticamente o foco quando a página é carrega**. Permitindo que o usuário comece digitar sem ter que manualmente dar o foco (clicar) no elemento.  
  
Os elementos que podem receber esse atributo são: button, input, keygen, select e textarea. Não deve haver mais de um elemento no documento com o atributo autofocus especificado.

Se você ainda não entende o que é isso, por favor, vá até o [Google](http://www.google.com.br/) e perceba que ao entrar na página o campo de busca já recebe o foco. Você digita o termo da busca de pesquisa sem primeiro clicar no campo input. Definir o foco para caixa de texto automaticamente é sem dúvida um acréscimo a[usabilidade do site](http://www.kadunew.com/blog/web-design/6-dicas-para-melhorar-a-legibilidade-e-usabilidade-do-seu-site).

Há outras maneira de aplicar o foco automático em um campo de formulário, com JavaScript, [jQuery](http://www.kadunew.com/blog/category/jquery), por exemplo.

O atributo autofocus é suportado em todos os principais navegadores, com exceção do Internet Explorer.

No exemplo a seguir, o elemento input será focado assim que a página seja carregada.

[view plaincopy to clipboardprint?](http://www.kadunew.com/blog/html/html5-atributo-autofocus)

1. **<input** maxlength="255" name="busca" value="" autofocus**>**
2. **<input** type="submit" value="Pesquisar"**>**

Autocomplete fonte: http://www.kadunew.com/blog/html/html5-atributo-autofocus

O atributo autocomplete é novo na HTML5 . A função do atributo é habilitar ou desabilitar o autopreenchimento dos elementos form e input.

Alguns navegadores oferecem a funcionalidade de preencher ou sugerir opções de preenchimento de um campo de formulário. Essa sugestão é feita com base no que foi digitado anteriormente pelo usuário.

Definir se um campo de formulário deve ou não ter seu autopreenchomento ativo é extremamente importante, uma vez que nunca sabemos onde nossos usuários estarão acessando esse formulário. Eles podem preencher seus dados a partir de um computador seguro ou um computador público. Independentemente do nível de segurança, alguns valores de campo nunca devem ser lembrado, por exemplo, número de cartão de crédito, CPF, etc.

Valores do atributo

on: valor Default/Padrão.O navegador irá preencher automaticamente os valores com base em valores digitado anteriormente pelos usuários.

off: o navegador não completa automaticamente entradas nos campos do formulário.

É possível ter autocomplete igual a on para o formulário, e off para campos de entrada específicos, ou vice-versa.

O exemplo abaixo desabilita o autopreenchimento de todo formulário, já que o atributo está sendo aplicado no elemento de formulário.

**<form** method="post" autocomplete="off"**>**

Esse outro exemplo mostra como você pode isolar um único campo para prevenir o autocomplete:

**<input** type="text" name="cartao\_c" autocomplete="off" **/>**

**Placeholder fonte: http://diveintohtml5.com.br/detect.html**

**Além dos novos tipos de entrada, a HTML5 inclui vários pequenos ajustes aos formulários já existentes. Uma melhoria é colocar um placeholder em um campo de entrada . O placeholder é exibido dentro de um campo de entrada enquanto o campo estiver vazio e sem foco. Assim que você clica no campo (ou navega com o tab até ele), o placeholder some. O capítulo sobre formulários web tem screenshots se você está tendo dificuldades em visualizá-lo.**

**A verificação de suporte ao placeholder utiliza a técnica de detecção #2. Se o seu navegador possui suporte à colocar placeholders em campos de entrada, o objeto DOM criado para representar o elemento <input> terá uma propriedade placeholder (mesmo que você não inclua um atributo placeholder em seu HTML). Se o seu navegador não possui suporte ao placeholder, o objeto DOM criado para um elemento <input> não terá uma propriedade placeholder.**

**function supports\_input\_placeholder() {**

**var i = document.createElement('input');**

**return 'placeholder' in i;**

**}**

**Ao invés de escrever essa função, você pode utilizar o Modernizr (1.1 ou superior) para detectar suporte ao placeholder.**

**↶ checa pelo texto de placeholder**

**if (Modernizr.input.placeholder) {**

**// seu placeholder já deve estar visível!**

**} else {**

**// sem suporte ao placeholder :(**

**// implemente uma solução com script**

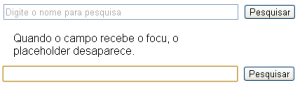
**}**

**Fonte: http://html5brazil.wordpress.com/tag/placeholder/**

1. Placeholder Text

O placeholder é um texto que aparece dentro do campo de texto e assim que o campo recebe o focu ou deixa de ser vazio ele some.

Exemplo:

[](http://html5brazil.files.wordpress.com/2010/12/placeholder.png)

Para fazer o placeholder este é o código:

<input type=”text” placeholder=”Digite o nome para pesquisa…”>

Caso o navegador não suporte o atributo placeholder ele será ignorado.

Maxlength fonte: <http://www.linhadecodigo.com.br/artigo/3443/introducao-ao-html-elementos-de-formularios-html.aspx>

**Listagem 2:** Exemplo de text

|  |  |
| --- | --- |
| 1 | <input type="text" value="Seu nome: " maxlength="15" size=”50” accesskey=”a” /> |
| Atributo dentro do text: | Descrição: |
| maxlength | Serve para definir o maior número de caracteres que podem ser digitados; |
| value | É usado para definir a mensagem que aparecerá no campo textual logo após a inicialização da página; |
| size | É usado para definir o tamanho do campo de texto; |
| accesskey | Serve para definir um atalho para o campo de texto. |

O atributo accesskey serve para criar um atalho através da combinação “alt+...”; no exemplo que dei, o atalho é alt+a. É importante ressaltar que os visitantes não têm como saber desse atalho se não for explicitamente informado.

**Listagem 3:** Uso de password

|  |  |
| --- | --- |
| 1 | <input type="password"  maxlength="15" /> |

**reset**

O próximo atributo é o reset, ele só funciona quando conectado ao precisa resetar. Então vamos pô-lo dentro da tag form, assim, conectando-o ao seu objetivo de ação, como se segue no exemplo abaixo:

**Listagem 4:** Exemplo do uso de reset

|  |  |
| --- | --- |
| 1  2  3  4 | <form>  <input type="text"  maxlength="15" size="50" accesskey="a" />  <input type="reset" value= "Formatar texto" />  </form> |

Pattern fonte: <http://blog.popupdesign.com.br/validando-formularios-like-a-boss-com-html5/>

## Pattern

No atributo Pattern nós usamos uma **Expressão Regular** para determinar qual padrão de preenchimento o formulário deve seguir.

Por exemplo: se quero um formulário em que o usuário escreve uma sequência de 3 números e 3 letras em Uppercase, uso o seguinte pattern:

|  |  |
| --- | --- |
| 1  2  3 | ...  <input pattern="[0-9]{3}[A-Z]{3}" />  ... |

Neste outro exemplo, o usuário deverá colocar um número decimal, onde ele usa a vírgula como separador e duas casas decimais:

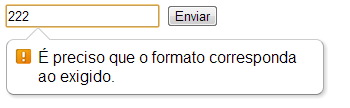
|  |  |
| --- | --- |
| 1  2  3 | ...  <input pattern="[0-9],[0-9][0-9]" />  ... |

Fonte: <http://www.debugado.com.br/validando-formularios-com-html5/>

## Validando formulário HTML5 com a função pattern.

A função **pattern**, nos permite determinado um formado desejado a ser digitado pelo usuário. Supondo que temos um campo deCPF, e queremos que o usuário preencha o campo seguindo a formatação de um CPF. Exemplo 111.111.111-11. Segue exemplo abaixo:

|  |  |
| --- | --- |
| 1 | <**input** type="text" pattern="[0-9]{3}.[0-9]{3}.[0-9]{3}-[0-9]{2}" required="" /> |

[](http://www.debugado.com.br/wp-content/uploads/2013/07/debugado-validacoes-021.png)

Perceba que se você não seguir o exemplo determinado, uma mensagem de aviso aparece informando que o campo não foi  
preenchido corretamente.

Fonte: <http://www.kadunew.com/blog/html/validacao-de-formulario-com-html5>

## pattern = “Expressão”

Com uso do atributo pattern você pode exigir que o usuário digite valores em um padrão feito com[expressão regular](http://www.phpit.com.br/artigos/entendendo-as-expressoes-regulares.phpit), ou seja, o que for digitado no compo input deverá satisfazer a expressão regular criada pelo desenvolvedor.

Explicarei o atribudo criando uma expressão para entrada de um número de telefone. O valor inserido no campo de input deverá iniciar com (xx) logo terá um espaço em branco com uma sequência de quatro dígitos, um traço, e para terminar mais uma sequência de quatro dígitos, ficando assim: (xx) xxxx-xxxx.

Abaixo o código para criar a condição descrita acima.

[view plaincopy to clipboardprint?](http://www.kadunew.com/blog/html/validacao-de-formulario-com-html5)

1. **<form>**
2. **<label>**Telefone: **<input** required name="tel" pattern="\([0-9]{2}\)[\s][0-9]{4}-[0-9]{4}"**></label>** (xx) xxxx-xxxx
3. **<input** type=submit value=Enviar**>**
4. **</form>**

**\([0-9]{2}\)** usuário deve digitar dois números de 0 até 9 entre parênteses.

**[\s][0-9]{4}-[0-9]{4}** um espaço e duas sequência de números divididas por um traço.

Em [HTML5](http://www.kadunew.com/blog/category/html5) há um valor para o atributo type que é o tel. Ao utilizar esse valor não há máscara de formatação ou validação, já que não há um padrão para números de telefones. Ele serve apenas para informar que o valor do campo é um número de telefone.

Abaixo outro exemplo utilizando o atributo pattern, agora validando CEP

[view plaincopy to clipboardprint?](http://www.kadunew.com/blog/html/validacao-de-formulario-com-html5)

1. **<form>**
2. **<label>**CEP: **<input** required name="cep" pattern="[0-9]{5}-[0-9]{3}"**></label>** (xx) xxxx-xxxx
3. **<input** type=submit value=Enviar**>**
4. **</form>**

Required : fonte: <http://www.kadunew.com/blog/html/validacao-de-formulario-com-html5>

## Atributo required

Utilizamos esse atributo para informar ao navegador que um elemento do formulário é de preenchimento obrigatório, ou seja, o usuário não consegue enviar o formulário se não preencher o campo com o **atributo required**. Seu uso é exclusivo com os elementos input, select e textarea.

Uma mensagem é informada ao usuário em uma **tentativa de envio de formulário sem o preenchimento de um campo** com required. Esse atributo apenas verifica se o campo foi preenchido ou não. Quero dizer que, qualquer caractere será aceito.

[view plaincopy to clipboardprint?](http://www.kadunew.com/blog/html/validacao-de-formulario-com-html5)

1. **<form** method="post" action=""**>**
2. **<label** for="nome"**>**Nome: **</label><input** id="nome" type=text required name=nome**>**
3. **<input** type=submit value="OK"**>**
4. **</form>**

Fonte: <http://www.debugado.com.br/validando-formularios-com-html5/>

## Validando de formulário HTML5 com ****required****

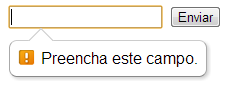
**Observação:** Vale a pena lembrar que para fazer testes de **validação** de **formulário**, os seus campos precisam estar inseridos dentro de um“form” com um botão do tipo “submit”. Segue exemplo abaixo:

|  |  |
| --- | --- |
| 4  5  6  7 | <**form** action="valida.php" method="post">  <**input** type="text" />  <**input** type="submit" value="Enviar" />  </**form**> |

O required é uma funcionabilidade que transforma um determinado campo de formulário obrigatório. Sendo assim, obriga o usuário a preenche-lo por padrão. Digite o exemplo abaixo e tente enviar o formulário sem preencher o campo.

|  |  |
| --- | --- |
| 1 | <**input** type="text" required="" /> |

Note que surgiu uma mensagem, informando que o campo precisa ser preenchido.

[](http://www.debugado.com.br/wp-content/uploads/2013/07/debugado-validacoes-01.png)

Novalidate fonte: <http://pt.slideshare.net/fagnerlima91/html5-04-novos-elementos-e-atributos>

1. Atributos Alguns elementos ganharam novos atributos: • O atributo autofocus pode ser especificado nos elementos input (exceto quando há atri-buto hidden atribuído), textarea, select e button. • A tag a passa a suportar o atributo media como a tag link. • A tag form ganha um atributo chamado novalidate. Quando aplicado o formulário pode ser enviado sem validação de dados. • O elemento ol ganhou um atributo chamado reversed. Quando ele é aplicado os indica-dores da lista são colocados na ordem inversa, isto é, da forma descendente. • O elemento fieldset agora permite o atributo disabled. Quando aplicado, todos os filhos de fieldset são desativados. 11/8/2013 Fagner S. de Lima - HTML 5 19

Fonte: <http://www.wufoo.com/html5/attributes/11-novalidate.html?num=4>

<form **novalidate**>

<input type=number name=num step=5>

</form>

Fonte: <http://html5brazil.wordpress.com/tag/html5/>

 existe a necessidade de ativar a validação de formulário no HTML5, ela está ativada por padrão.

Para desativar a validação deve-se especificar o atributo novalidate.

Exemplo:

<form novalidate>

<input type="email" id="enderecoEmail">

<input type="submit" value="Enviar">

</form>

Os navegadores estão implementado aos poucos o suporte a validação de formulários do HTML5. O Firefox 4 terá uma implementação completa desta funcionalidade, mas infelizmente o Chrome e Safari possuem uma implementação incompleta. Eles realizão a validação dos campos do formulário, caso exista algum erro não se permite enviar o formulário mas também não avisam o porque. Eles irão jogar o focu para o campo com erro e não exibirão a mensagem de erro como o firefox4.

Fonte: <http://www.w3schools.com/tags/att_form_novalidate.asp>

# HTML <form> novalidate Attribute

[[HTML form Tag Reference](http://www.w3schools.com/tags/tag_form.asp) HTML <form> tag](http://www.w3schools.com/tags/tag_form.asp)

## Example

Indicate that the form is not to be validated on submit:

<form action="demo\_form.asp" novalidate>  
E-mail: <input type="email" name="user\_email">  
<input type="submit">  
</form>

[**Try it yourself »**](http://www.w3schools.com/tags/tryit.asp?filename=tryhtml5_form_novalidate)

## Browser Support

Internet Explorer Firefox Opera Google Chrome Safari

The novalidate attribute is supported in Internet Explorer 10, Firefox, Opera, and Chrome.

**Note:** The novalidate attribute is not supported in Internet Explorer 9 and earlier version, or in Safari.

## Definition and Usage

The novalidate attribute is a boolean attribute.

When present, it specifies that the form-data (input) should not be validated when submitted.

## Differences Between HTML 4.01 and HTML5

The novalidate attribute is new in HTML5.

## Differences Between HTML and XHTML

In XHTML, attribute minimization is forbidden, and the novalidate attribute must be defined as <form novalidate="novalidate">.

## Syntax

<form novalidate>

[[HTML form Tag Reference](http://www.w3schools.com/tags/tag_form.asp) HTML <form> tag](http://www.w3schools.com/tags/tag_form.asp)

Expressões regulares: fonte: <http://wbruno.com.br/html/validando-formularios-apenas-com-html5/>

Validar formulários é chato, tedioso e trabalhoso. Felizmente alguém olhou isso e resolveu incluir dentro da especificação do html, alguns atributos e valores novos muito interessantes.

Se usarmos corretamente, e estudarmos Expressões Regulares, é possível fazer uma validação simples sem escrever nenhuma linha de javascript. Vou deixar abaixo alguns snippets da tag input, utilizando o type correto (veja aqui todos os [types novos da html5](http://html5doctor.com/html5-forms-input-types/)) e um uso do atributo **pattern** para os qual eu escrevi algunms ERs.  
  
**Apenas letras**

1. <input type="text" required="required" name="text" pattern="[a-z\s]+$" />

**Apenas números**

1. <input type="text" required="required" name="numbers" pattern="[0-9]+$" />

**Data**

1. <input type="date" required="required" maxlength="10" name="date" pattern="[0-9]{2}\/[0-9]{2}\/[0-9]{4}$" min="2012-01-01" max="2014-02-18" />

**Hora**

1. <input type="time" required="required" maxlength="8" name="hour" pattern="[0-9]{2}:[0-9]{2} [0-9]{2}$" />

**Campos genéricos de texto**

1. <input type="text" required="required" name="name" />

**Telefone**

1. <input type="tel" required="required" maxlength="15" name="phone" pattern="\([0-9]{2}\) [0-9]{4,6}-[0-9]{3,4}$" />

**Email**

1. <input type="email" required="required" class="input-text" name="email" pattern="[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$" />

## placeholder

Lembre-se de usar o placeholder nos seus campos em que você precise “dar alguma dica” para o usuário de como ele deve preenchê-lo

## Estilizar os inputs

Faça testes utilizando as pseudo classes

1. input:required:invalid {}
2. input:required:valid {}

## Personalizar as mensagens de erro

Encontrei este artigo bem completo e interessante: [Validando formulários like a boss com HTML5](http://blog.popupdesign.com.br/validando-formularios-like-a-boss-com-html5/). Onde é mostrado o atributo:required x-moz-errormessage=”Ops. Não esqueça de preencher este campo.”, logicamente exclusivo do Firefox.

E para webkit, o css:

1. ::-webkit-validation-bubble {/\*Insira aqui seu CSS.\*/}
2. ::-webkit-validation-bubble-message {}
3. ::-webkit-validation-bubble-arrow {}
4. ::-webkit-validation-bubble-arrow-clipper {}

## That’s all folks!

Em alguns inputs, mesmo com o type definido, eu forcei a validação no pattern, pois um pode ser implementado sem o outro pelos browsers, e existe um “bug” no type email, em que no Chrome ele aceita um email do tipo: **email@a**, sem obrigar que o cliente informe o domínio do site. (.com, .net…)