Formulários

Há várias maneiras de se criar um formulário na web, porém, é preciso considerar vários fatores para que este formulário não seja apenas uma página com um amontoado de campos, mas sim a porta de entrada para o usuário criar uma conta no seu site, entrar em contato solicitando um serviço ou apenas realizar uma pesquisa.

Desenvolver um formulário amigável ao usuário é o ponto de partida para que seu formulário seja o mais objetivo e transparente possível, providenciando ao usuário o resultado esperado. Formulário amigável é aquele desenvolvido visando facilitar a tarefa de um usuário, e não complicá-la, pois cada formulário nada mais é do que uma área destinada à realização de tarefas por parte do usuário.

Quando o caminho for longo

Cada formulário possui por fim um único objetivo. Há formulários de cadastro, de contato, de login e até mesmo um simples campo de busca na verdade trata-se de um formulário de pesquisa. Com um objetivo definido, é hora de se analisar a estrutura que o formulário apresentará, baseado nesse objetivo.

Formulários de login e busca são mais simples de serem estruturados devidamente pela quantidade menor de campos que eles geralmente apresentam. Formulários de contato também acabam sempre tendo a mesma estrutura, que raramente é complicada, mas apresentam mais campos que os formulários de login e busca. O formulário mais complicado e cansativo de se preencher, entretanto é o de cadastro, pois sempre solicita muitas informações de uma única vez e acaba se tornando uma tarefa entediante ao usuário.

A melhor forma de se estruturar um formulário desse tipo é separando-o por seções e facilitando seu preenchimento. Cada seção corresponde a um conjunto de campos relacionados, formando um conjunto de campos para dados pessoais, dados cadastrais (login e senha), etc. Uma seção pode estar separada visualmente de outra seção em uma página ou possuir uma página própria, formando uma etapa para o processo de cadastro.

Uma forma de se agrupar campos relacionados é através da tag *fieldset*, a qual gera uma borda em volta dos campos correspondentes àquela seção e pode ser estilizada de qualquer maneira. A tag *legend* é utilizada em conjunto com a tag *fielset* e apresenta um título para a seção.

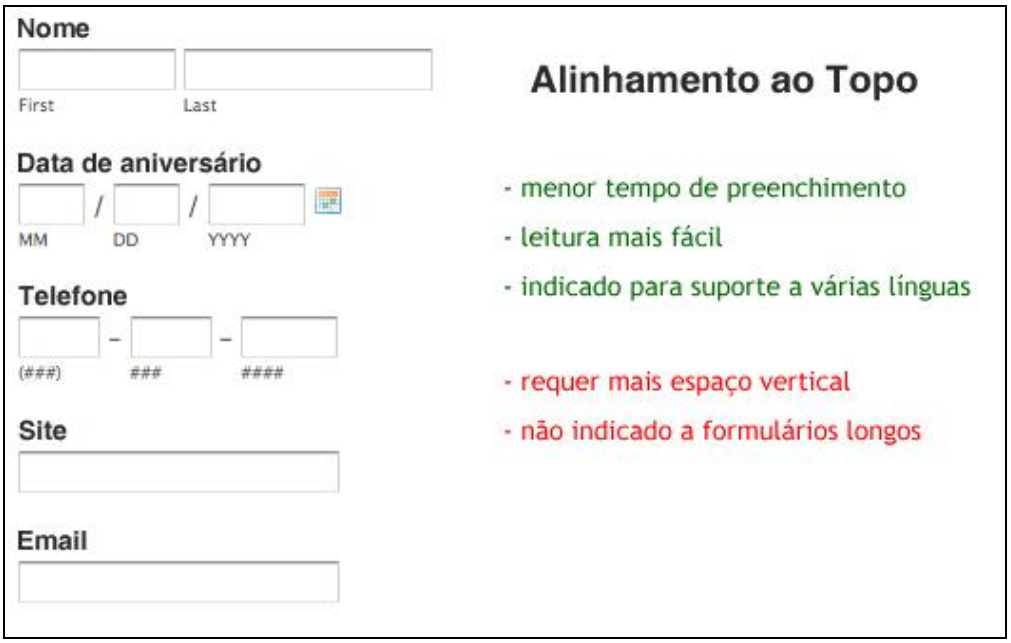
Localização de rótulos

Em um formulário, rótulos são os textos que geralmente acompanham cada campo e ao se desenvolver um formulário, um ponto importante que deve ser considerado, é a localização desses rótulos em relação aos seus campos. A posição desses elementos afeta consideravelmente a legibilidade e usabilidade do formulário, auxiliando na velocidade de conclusão do seu preenchimento, o que interfere diretamente no nível de satisfação do usuário. Não há uma posição certa para o posicionamento desses elementos, tudo depende das necessidades especificas do formulário, assim como do seu tamanho e do design padrão de uma página.

Alinhado ao topo

Como regra geral, alinhar rótulos ao topo de um campo diminui o tempo de preenchimento de um formulário, além de aumentar a legibilidade entre rótulos e campos, pois diminui a quantidade de movimentos realizados pelos olhos. Outra vantagem deste tipo de alinhamento é que ele diminui o espaço horizontal utilizado por cada rótulo e campo. Caso seu formulário ofereça suporte a outras línguas, tal alinhamento evitará quebras horizontais.

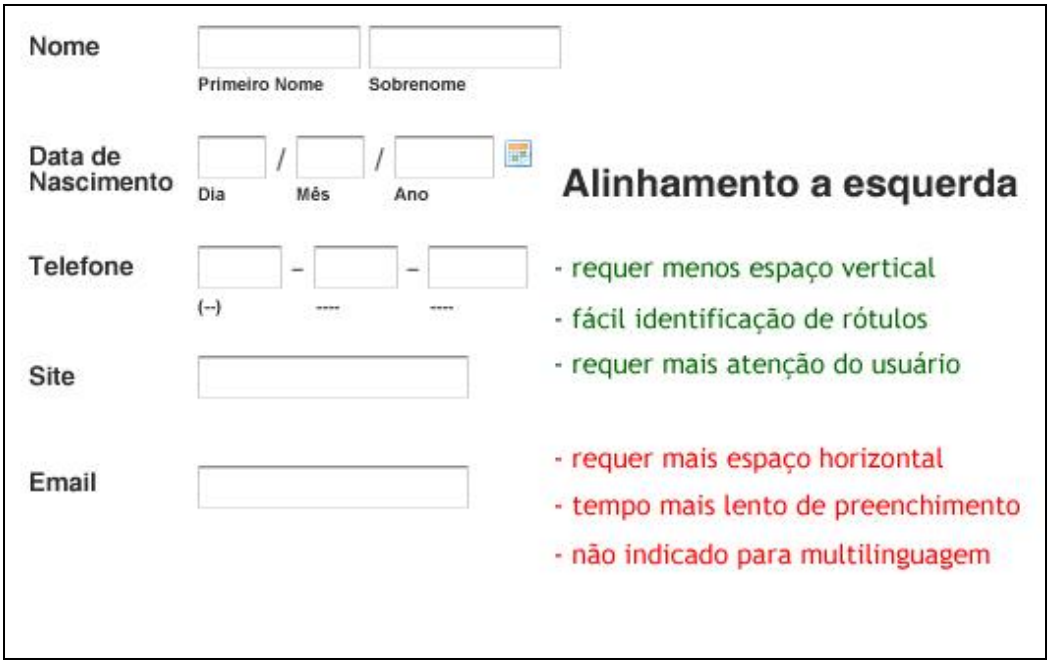
Este alinhamento parece ser uma boa alternativa, mas não é indicado a formulários longos, pois aumentará consideravelmente a altura do formulário, gerando uma rolagem extra, a qual pode ser desnecessária. O tempo de preenchimento é prejudicado quando o formulário apresenta rolagem.



Alinhado a esquerda

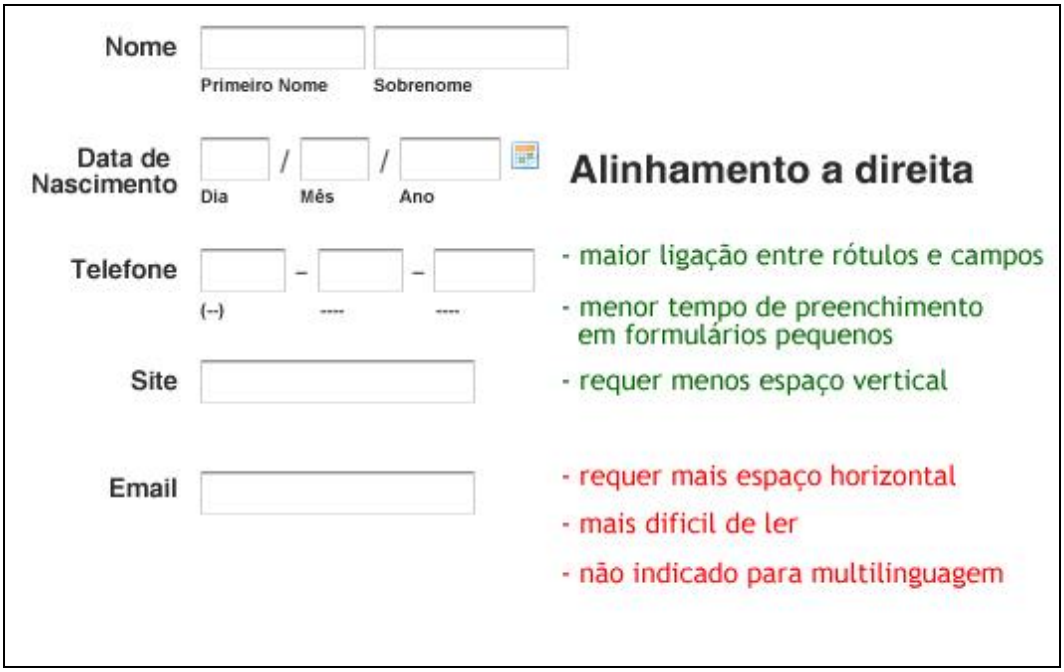
Uma das vantagens de se alinhar rótulos a esquerda é a facilidade gerada na identificação desses rótulos. Uma vez que rótulos e campos se encontram na mesma linha horizontal, os usuários podem rapidamente verificar cada rótulo e iniciar o preenchimento do campo. Esse aspecto é ideal quando o formulário solicita algum dado importante e incomum ao usuário e o rótulo para cada campo precisa ser compreendido claramente. Alinhamento a esquerda também necessita menos espaço vertical, o que implica em menor altura para a página. Entretanto, menos espaço vertical significa mais espaço horizontal, assim, o design do seu site poderá dar a palavra final.

Um dos aspectos negativos desse tipo de alinhamento é que ele gera um tempo mais lento de preenchimento. Isso ocorre devido à distância gerada entre e rótulo e o seu campo. Quanto menor o rótulo, maior será essa distância.



Alinhado a direita

Rótulos com alinhamento à direita possuem as mesmas vantagens e desvantagens que rótulos alinhados a esquerda: possuem mais espaço horizontal e menos espaço vertical. Além disso, o problema com suporte a outras línguas aparece novamente. A grande vantagem com este alinhamento é que ele aumenta a ligação entre o rótulo e seu campo, o que falta ao alinhamento à esquerda. A grande desvantagem, porém, é o desconforto visual causado por este alinhamento, afetando sua legibilidade. Sem uma base visual a sua esquerda, os rótulos ficam difíceis de ler. Em formulários pequenos, este problema não é muito visível.



Utilize *labels*

A não utilização de *labels* no seu formulário pode ser considerado um crime de acessibilidade. *Labels* são tags reponsáveis por definir um rótulo para um campo e não possuem nenhum efeito visual É possível associar um *label* a um campo através do atributo *for*, como no exemplo abaixo:

```
<label for="nome">Nome:</label>
<input type="text" id="nome" />
```

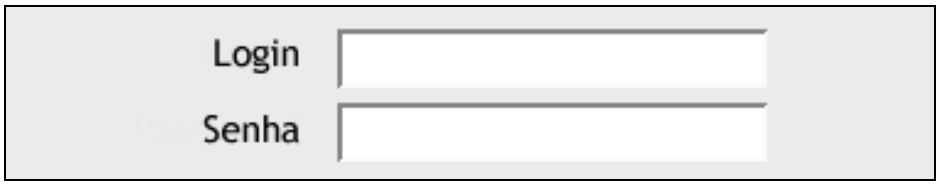
Dessa forma, caso o usuário clique no rótulo o campo a que ele esta associado recebe o foco para preenchimento. Para que isto aconteça, é necessário que o atributo *for* da tag *label* possua o mesmo valor que o atributo *id* da tag *input*. O uso da tag *label* não é apenas semanticamente correto como oferece a oportunidade de estilização dos rótulos via CSS.

***Labels* flutuantes**

Adicionar *float* aos rótulos de um formulário proporciona um controle melhor ao seu posicionamento, oferecendo uma estrutura visual do tipo tabela. Um exemplo simples é flutuar o rótulo à esquerda, alinhar o texto à direita e adicionar um pouco de margem à direita.

```
label{
      float: left;
```

```
        text-align: right;
        margin-right: 15px;
}
```



Atenção a estilos padrões

Vários navegadores possuem um estilo padrão para *inputs* do tipo *button*. Isso oferece uma experiência consistente ao usuário daquele navegador. Mas na maioria das vezes gostaríamos de interferir nesse estilo, para oferecer ao usuário uma experiência mais pessoal em nosso site, independente do navegador que ele utiliza. Uma maneira simples e comum de interferir nesse estilo padrão é através de técnicas do tipo *CSS Reset*, o que incluiria declarações da seguinte forma:

```
* {
        border: none;
}
```

Isso eliminaria o estilo padrão de botão dos navegadores. Ao fazer isso, tenha certeza de providenciar um estilo próprio a estes elementos que transmita ao usuário a sensação de botão naquele elemento.

Redimensionamento de textarea

Navegadores recentes como Chrome, Safari 5.05 e Firefox 4 implementaram a possibilidade do usuário redimensionar o tamanho de um campo do tipo *textarea*. Essa possibilidade pode gerar quebras no layout da página, dependendo do tamanho redimensionado pelo usuário. Para evitar que isso ocorra, é possível declarar um tamanho limite para a largura deste campo através da propriedade *max-width*, da seguinte maneira:

```
textarea{
        width: 200px;
        max-width: 400px;
}
```

Declarada dessa forma, a largura da *textarea* do nosso exemplo terá 200 *pixels* e poderá ser redimensionada até 400 *pixels*, evitando quebras no layout da página.

Sombreamento em campos

Um artifício utilizado por padrão em alguns navegadores como Chrome e Safari, é adicionar uma espécie de sombreamento no campo que está focado no formulário. Esse sombreamento (laranja no Chrome e azul no Safari) pode interferir no visual de um formulário, seja pelas cores utilizadas no layout ou pela temática presente nos campos. Para remover este estilo padrão utilizamos a propriedade *outline*, responsável por gerar uma linha em volta de um elemento (fora das bordas), e declaramos a ele o valor *none*, da seguinte maneira:

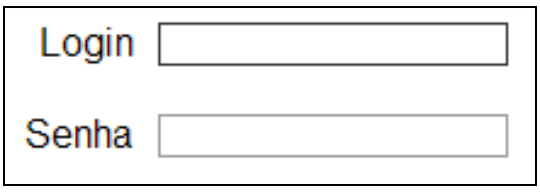
```
input, textarea{
      outline:none;
}
```

Com apenas isso removemos o estilo padrão de um navegador especifico e oferecemos uma experiência mais consistente no uso do formulário, independentemente do navegador utilizado pelo usuário.

O pseudo-seletor :focus

É possível adicionar um estilo a elementos de formulário que sejam exibidos apenas quando o usuário clicar num desses elementos através do pseudo-seletor *:focus*. Este seletor permite aplicar um estilo ao campo que possui o foco de entrada. Como exemplo, podemos alterar a cor da borda do campo ao qual o usuário esta prestes a preencher:

```
textarea, input {
        border: 1px solid #999;
}
textarea:focus, input:focus {
        border: 1px solid #333;
}
```



O pseudo-seletor *:focus* possui suporte na maioria dos navegadores, com restrição apenas para versões de IE abaixo da 8.

Foco no primeiro campo

Assim que a página que possui seu formulário for carregada, dependendo do caso, é muito útil ao usuário que o foco apareça já no primeiro campo. O usuário pode manter suas mãos no teclado para começar a digitar seus dados desse modo, o que lhe poupa tempo e esforço de direcionar o mouse e clicar no campo. A técnica de destacar o campo que possui o foco no formulário contribui para que o usuário saiba claramente que pode começar a digitar.

Os dois trechos de código JavaScript a seguir são responsáveis por colocar o foco em um campo assim que o formulário é carregado. O primeiro exemplo possui a seguinte estrutura:

```
$(document).ready(function(){
      document.formulario.campo.focus();
});
```

Onde *formulario* é o valor do atributo *name* da tag *form* e *campo* é o valor do atributo *id* do *input* que desejamos aplicar o foco. O segundo exemplo utiliza a biblioteca *jQuery* e é mais simples:

```
$(document).ready(function(){
      $('#campo).focus();
});
```

Aqui é apenas declarado que o *input* com *id* igual a campo receberá o foco ao carregar a página.

Adicione dicas

Pode ser muito útil preencher seus campos com alguma dica ou informação. Por exemplo, em um formulário de contato comum você pode ter um campo do tipo *textarea* com o rótulo “Mensagem”. Você pode preencher este campo com uma dica do tipo “Adicione sua mensagem aqui”. Este texto pode ser adicionado dentro da tag *textarea* ou no atributo *value* para um *input* comum:

```
<textarea name="mensagem">Adicione sua mensagem aqui</textarea>
<input type="text" id="email" value="exemplo@email.com" />
```

Do modo descrito acima, sua dica será exibida dentro do campo mas, dessa maneira o usuário terá que remover este texto manualmente, o que não é bom para a usabilidade do formulário. Com um pouco de programação é possível limpar estes campos quando o usuário clicar para preenchê-los. O pequeno código JavaScript abaixo permite limpar os campo assim que o usuário colocá-lo em foco para preenchê-lo.

```
<textarea onfocus="this.value=''; this.onfocus=null;"
name="mensagem">Adicione sua mensagem aqui</textarea>

<input type="text" id="email" value="example@email.com"
onfocus="this.value=''; this.onfocus=null;" />
```

O código acima permite limpar os campos uma única vez, não exibindo mais as dicas quando o usuário tirar o foco do campo.

Exiba a senha

A maioria dos formulários de cadastro solicita o preenchimento de dois campos para a senha. Devido à forma como campos do tipo *password* são exibidos (ocultando os caracteres digitados), é compreensível o preenchimento de dois campos para que o usuário possa ter certeza de que digitou sua senha corretamente da primeira vez.

Na verdade, essa prática comum fere a usabilidade de duas formas:

- Exige que o usuário digite mais e a mesma informação;
- Não prove resposta ao que o usuário digitou;

A exibição de marcadores ao invés dos caracteres digitados pelo usuário num campo de senha já pode ser considerado um grande problema por si mesmo, pois o usuário não consegue visualizar se esta digitando corretamente sua senha. Além disso, solicitar que ele repita este processo mais uma vez implica apenas em fazê-lo digitar mais e desnecessariamente, pois ele pode errar a senha nos dois campos.

Um conceito diferente e eficiente é solicitar o preenchimento do campo senha apenas uma vez e incluir um *checkbox* que exiba a senha caso seja clicado. Dessa forma, o usuário pode verificar se sua senha foi digitada corretamente e então prosseguir com seu cadastro. Essa técnica também é eficiente em formulários de *login*, pois permite ao usuário que acabou de errar sua senha, exibi-la e então conferir qual caractere foi digitado incorretamente e assim corrigi-lo.

Exibir a senha digitada em um campo do tipo *password* é possível através do *plugin showPassword* da biblioteca *jQuery*. Tanto o *script* da biblioteca quanto o *script* do *plugin* devem ser carregados dentro da tag *head* no documento HTML da seguinte maneira:

```
<script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/1.6.2/jquery.min.js"></scr
ipt>
<script type="text/javascript"
src="jquery.showpassword.min.js"></script>
```

Após isso, é preciso adicionar ao atributo *id* da tag *input* referente ao campo senha, o seguinte valor: showpassword. E adicionar o seguinte trecho de código a tag *head*:

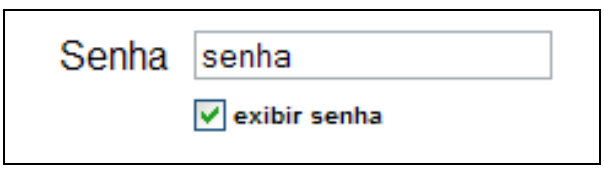
```
$(document).ready(function(){
      $('#showpassword').showPassword();
});
```

Esse código resultará na opção mais básica do plugin, gerando o *checkbox* logo após o campo senha, o que pode ser customizado com duas opções em sua estrutura:

```
$('#showpassword').showPassword(elemento, opções);
```

Em elemento, podemos definir o elemento em que desejamos inserir nosso checkbox através da classe ou id desse elemento. Em opções podemos adicionar três variáveis: *name* e *className* para o *checkbox* e o texto que será exibido ao seu lado. Desse modo, nosso exemplo teria o seguinte aspecto:

```
$(document).ready(function(){
      $('#showpassword').showPassword('.checker', {
            text: ' exibir senha',
            classname: 'myclass',
            name: 'showmypass'
            }
        );
});
```



Botão enviar ou cadastrar

Faça o botão enviar tão grande quanto os campos de texto. O botão em um formulário, independentemente se seu texto é “Enviar” ou “Cadastrar”, não serve apenas para submeter os dados preenchidos ao formulário, mas também para dizer ao usuário qual ação ele está prestes a realizar. Um botão pequeno possui um poder de persuasão menor diante do usuário, tornando-o inseguro sobre a ação que ele pretende tomar. Um botão grande da mais confiança ao usuário e é mais fácil de ser visualizado, assim como seu texto, tornando mais clara a ação que o usuário irá submeter.

Newsletter

A maioria dos sites que oferece assinatura de newsletter aos seus usuários apresenta na página de contato um campo do tipo *checkbox* para isso, geralmente selecionado. Com esse campo selecionado por padrão, tais sites esperam conquistar mais assinantes. Essa assinatura na verdade não tem sentido se não for feita conscientemente pelo usuário, pois caso ele receba uma newsletter sem ter solicitado, provavelmente cancelará seu recebimento ou no pior dos casos, a marcará como spam.

Forçar o usuário a receber newsletters que ele não solicitou por vontade própria não é a melhor forma de estabelecer uma comunicação com ele e ainda pode gerar uma má reputação ao seu site. A melhor forma de oferecer esse serviço é apresentar o campo *checkbox* desmarcado e próximo a ele um link para uma prévia da newsletter. Dessa forma, antes de assinar a newsletter, o usuário poderá ter noção do que está perdendo se não assinar seu conteúdo e será mais fácil analisar o número de usuários que solicitaram tal assinatura porque estavam realmente interessados.

Não use Captchas

Apenas pelo fato de seu site possuir um formulário, ele pode ser vítima de programas maliciosos de spam, adicionar um campo do tipo captcha pode então ser necessário. O que não é necessário é tornar isso um obstáculo ao usuário, repelindo-o de preencher seu formulário. Captchas tradicionais geralmente solicitam ao usuário que decifrem as letras distorcidas em uma imagem e as digite em um campo de texto. Contudo, essas letras não são fáceis de serem compreendidas, tornando-se uma barreira para a usabilidade. Captcha é um método comprovadamente eficiente para evitar spam, mas também é um método irritante ao usuário, pois é comum que estes errem as letras solicitadas no campo e tenham que recarregar a página para atualizar a imagem do captcha.

Uma abordagem diferente é o uso de uma técnica chamada *honeypot technique* (pote de mel) ou captcha invisível. Essa técnica compreende em adicionar um campo ao formulário e remover sua visibilidade através de CSS. Programas maliciosos que costumam enviar spam através de formulários são geralmente programados para preencher a maior quantidade possível de campos dos mesmos. Desse modo, é altamente provável que nosso campo invisível seja preenchido por um desses programas que geralmente ignoram códigos CSS (principalmente em arquivos externos). Um usuário comum deverá deixar este campo vazio, pois não poderá visualizá-lo.

```
<p class="hybot">
      <label for="confirmation">
            Caso esteja visualizando este campo, favor deixá-lo em branco e
recarregue sua página.
      </label>
      <input type="text" name="confirmation" value="" />
</p>

.hybot{
      display:none;
}
```

Quando o formulário for enviado, você pode validar este campo, verificando se ele foi preenchido. Caso tenha sido você pode cancelar o envio do formulário, pois certamente ele foi preenchido por um programa malicioso, caso ele esteja vazio, você pode enviar o formulário normalmente. A validação desse campo deve ser feita tanto no lado cliente (com *JavaScript*) quanto do lado servidor (com PHP ou outra linguagem).

Esse método é menos intrusivo que um captcha comum, pois não é visualizado pelo usuário. Entretanto, em alguns casos como navegadores móveis e leitores de tela, o CSS pode não ser carregado. Nesse caso, exibimos uma mensagem informando ao usuário que não preencha este campo. Essa técnica, porém não é recomendada a sites com grande volume de usuários e que, portanto, são alvos de muitos programas maliciosos que podem contorná-la, alem de que ela não funcionaria em um ataque direcionado ao seu formulário, casos em que captchas tradicionais seriam absolutamente necessários.