**O atributo pattern**

Vimos que o HTML5 trouxe validação para formulários sem a necessidade de JavaScript, com novos tipos e o atributo required. Além disso, temos também o atributo pattern.

No atributo pattern, podemos especificar qual padrão deve seguir um determinado campo, para que o usuário preencha-o corretamente. Especificamos esse padrão usando uma linguagem chamada *expressão regular*. Para garantir que um campo só tenha letras e números, por exemplo, podemos fazer:

<input pattern="[A-Za-z0-9]\*">

Repare que colocamos os conjuntos de caracteres permitidos entre colchetes (letras maiúsculas, minúsculas e dígitos, nessa ordem) e um asterisco depois. Esse asterisco significa que podemos repetir quantas vezes quisermos um caractere desse conjunto. Se quisermos garantir uma quantidade mínima de letras e números, podemos fazer

<input pattern="[A-Za-z0-9]{8,}">

Com esse padrão, agora nosso usuário precisa digitar pelo menos 8 letras ou números para a entrada ser válida.

Com base nisso, altere o campo "Nome" para que o usuário só possa entrar um nome contendo letras (maiúsculas e minúsculas), espaços e apóstrofe, com no mínimo 4 letras. Como resposta, forneça o código do <input> alterado.

Como queremos um valor com letras maiúsculas e minúsculas, além de espaços e apóstrofo, precisamos colocar esses caracteres entre colchetes no atributo pattern do campo "Nome":

<input id="nome" type="text" name="nome" required autofocus pattern="[A-Za-z ']">

Porém, como queremos que ele possa digitar qualquer nome com 4 caracteres ou mais, precisamos colocar as chaves após os colchetes indicando essa quantidade:

<input id="nome" type="text" name="nome" required autofocus pattern="[A-Za-z ']{4,}">

Vale observar que um nome pode ter ainda outros caracteres, incluindo letras acentuadas, portanto essa é uma simplificação do que faríamos num site real.

**Descrevendo o pattern**

Fizemos um padrão personalizado para o campo "Nome", garantindo dados mais confiáveis dos nossos visitantes. Porém, quando um dado inválido é inserido nesse campo, a mensagem de erro é muito genérica! Vamos melhorar isso.

Num campo como o "Nome", com o atributo pattern, podemos usar o atributo title para descrever esse padrão. Por exemplo:

<input pattern="[0-9]\*" title="Preencha somente com números">

Com base nisso, altere o código do campo "Nome" colocando uma mensagem descritiva do formato.

O código final do campo "Nome" deve ficar assim:

<input id="nome" type="text" name="nome"

required autofocus pattern="[A-Za-z ']{4,}"

title="O nome precisa ter pelo menos 4 caracteres">

E a mensagem de erro deve ser algo como:

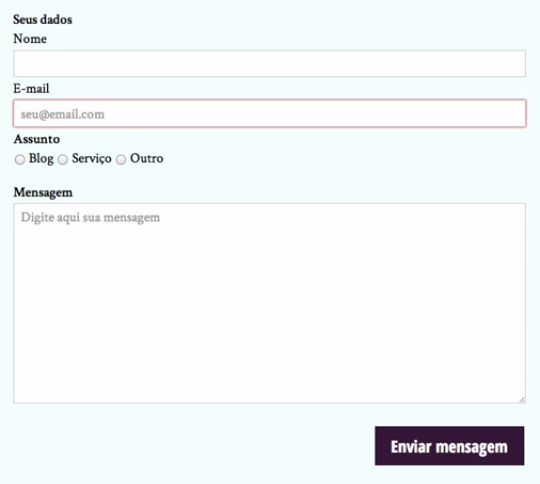
Preencha o campo com o formato requisitado: O nome precisa ter pelo menos 4 caracteres

**Estilizando formulários**

Vamos deixar nosso formulário de contato mais bonito para nossos visitantes. Queremos mudar os seguintes aspectos:

* As legendas dos grupos de campo e do campo "Mensagem" devem estar em negrito e espaçadas do grupo de campos anterior.
* Deixar os rótulos dos campos sempre em cima deles, *exceto* para os assuntos.
* Os campos de texto devem ocupar a largura inteira da tela (*exceto* o campo para o assunto "Outro"), ter uma borda cinza clara, espaçamento interno e usar a mesma fonte do resto da página.
* O campo de texto que tiver o foco do teclado deve ficar com o fundo levemente amarelo.
* Os campos que estiverem inválidos devem ter uma sombra vermelha (*dica*: use a pseudoclasse :invalid).
* O botão para enviar o formulário deve ficar na parte inferior direita da página, após o formulário, e ter o mesmo esquema de cores da barra lateral com a fonte usada nos títulos e levemente maior do que o texto normal. Além disso, o botão não deve ter borda.
* Quando o botão for selecionado pelo teclado ou quando o mouse estiver sobre ele, o fundo deve mudar para a cor #8C1D3D.

O resultado final deve ser parecido com o da imagem abaixo:



Seguindo passo a passo os requisitos do enunciado:

* As legendas dos grupos de campo e do campo "Mensagem" devem estar em negrito e espaçadas do grupo de campos anterior.

/\* Podemos usar um seletor de atributo para selecionar

apenas a legenda desse campo e não dos outros \*/

legend,

label[for="mensagem"] {

font-weight: bold;

margin-top: 1em;

}

* Deixar os rótulos dos campos sempre em cima deles, *exceto* para os assuntos.

/\* As tags <label> que têm o atributo "for" são

os rótulos dos campos sem ser os assuntos \*/

label[for] {

display: block;

}

* Os campos de texto devem ocupar a largura inteira da tela (*exceto* o campo para o assunto "Outro"), ter uma borda cinza clara, espaçamento interno e usar a mesma fonte do resto da página.

/\* Selecionamos apenas os campos de texto \*/

input[type="text"],

input[type="email"],

textarea {

/\* Força o navegador a usar a mesma fonte da página \*/

font-family: inherit;

/\* Inclusive o tamanho dela \*/

font-size: 100%;

padding: .25em .5em;

width: 100%;

box-sizing: border-box;

border: 1px solid #ccc;

}

/\* alguns ajustes para o campo "Outro": deixar ele na mesma linha dos outros assuntos \*/

fieldset > fieldset {

display: inline;

}

/\* deixar o rótulo dele inline de novo \*/

label[for="outro"] {

display: inline;

}

/\* deixar só esse campo sem ocupar a tela toda \*/

input[name="outro-assunto"] {

width: auto;

}

* O campo de texto que tiver o foco do teclado deve ficar com o fundo levemente amarelo.

input:focus,

textarea:focus {

background-color: #FFD;

}

* Os campos que estiverem inválidos devem ter uma sombra vermelha.

input:invalid {

box-shadow: 0 0 3px red;

}

* O botão para enviar o formulário deve ficar na parte inferior direita da página, após o formulário, e ter o mesmo esquema de cores da barra lateral com a fonte usada nos títulos e levemente maior do que o texto normal. Além disso, o botão não deve ter borda.

button {

/\* Precisamos tirar a borda que o navegador coloca por padrão \*/

border: 0;

padding: .5em 1em;

font-family: "Open Sans Condensed", sans-serif;

background-color: #3C1D3D;

color: white;

font-size: 1.2em;

/\* Faz com que o botão apareça na direita \*/

margin-left: auto;

margin-top: 1em;

}

* Quando o botão for selecionado pelo teclado ou quando o mouse estiver sobre ele, o fundo deve mudar para a cor #8C1D3D.

button:hover,

button:focus {

background-color: #8C1D3D;

}

Adicionamos um campo de texto ao lado da opção "Outros" para que o usuário especifique o assunto quando selecionar esta opção. Sendo assim, o campo não é útil antes que esta opção seja selecionada.

Tendo isso em mente, podemos esconder o campo por padrão, mostrando-o somente quando a opção "Outros" for selecionada, e podemos fazer isso com o que vimos até agora e a pseudoclasse :checked, que o navegador adiciona a elementos de formulário selecionados.

O *radio* usado para o assunto "Outro" tem o atributo value com esse texto. Então, usando os seletores de atributo e de irmão, conseguimos selecionar o campo e escondê-lo com a propriedade display:

input[value="Outro"] ~ input {

display: none;

}

Agora, quando esse *radio* for selecionado, a pseudoclasse será adicionada a ele. Podemos usar essa pseudoclasse para selecionar novamente o campo de texto ao lado e mudar o valor da propriedade display:

input[value="Outro"]:checked ~ input {

display: inline;

}

Dos tipos com casos de uso mais interessantes, temos:

* **date**, usado para que o usuário possa fornecer uma data como seu aniversário ou data de nascimento.
* **datetime**, para que o usuário possa fornecer, além da data, um horário, útil para que ele forneça um momento para ser lembrado de algo.
* **datetime-local**, para quando a data e o horário que o usuário fornecer sejam independentes de local ou quando o sistema é feito para trabalhar sempre no mesmo fuso horário, como por exemplo quando o usuário precisa fornecer a data e o horário de nascimento dele no horário de Brasília.
* **hidden**, usado para enviar informações extra para o servidor além daquelas fornecidas pelo usuário, como um identificador do usuário gerado pelo sistema.
* **image**, usado para enviar os dados do formulário com um botão gráfico, pode ser usado para não apenas ter um botão mais bonito, como também para fazer uma interação mais dinâmica com o usuário, em que o resultado da ação depende do local onde ele clica na imagem, uma vez que esse tipo de botão envia as coordenadas do clique para o servidor.
* **month**, usado para que o usuário forneça um mês e um ano, útil para que ele forneça a validade de um cartão de crédito, por exemplo.
* **number**, para que o usuário forneça um número como sua idade ou a quantidade de um produto num carrinho de compras.
* **range**, usado para que se possa escolher um valor numérico numa faixa de valores possíveis, normalmente sem muita preocupação com a precisão, como por exemplo quando se deseja que o usuário dê uma nota para um serviço.