INSTITUTO FEDERAL
Sul-rio-grandense | Câmpus Pelotas

LPW - Linguagem de Programação para WEB

CSS

Aula 8

Professor Sérgio Rodrigues

tsi tecnologia em sistemas para internet

CSS

Formulários

FORMULÁRIOS

Revisando todo formulário começa com uma tag <form>.

```
<form action="#" method="post"></form>
```

Para agrupar todos os campos do formulário, usa-se a tag <fieldset> (grupo de campos).

```
<form action="#" method="post">
    <fieldset>
        <!-- campos -->
    </fieldset>
</form>
```

FORMULÁRIOS

Para mais versatilidade e facilidade na criação das CSS, é recomendável englobar o rótulo do campo (label) e o campo com um outro elemento.

```
<div class="campo">
    <label for="nome">Nome</label>
    <input type="text" name="nome" id="nome">
</div>
```

A tag label é a tag mais apropriada para justamente "rotular" os campos de um formulário.

O atributo **for** serve para associar o rótulo ao campo desejado.

A div com a classe "campo" será para estilização CSS.

FORMULÁRIOS

A tag fieldset que vimos antes também será utilizada para agrupar campos lado a lado, através da classe "grupo".

```
<fieldset class="grupo">
    <div class="campo">
        <label for="nome">Nome</label>
        <input type="text" name="nome" id="nome">
    </div>
    <div class="campo">
        <label for="snome">Sobrenome</label>
        <input type="text" name="snome" id="snome">
    </div>
</fieldset>
```

FORMULÁRIOS

A tag input é extremamente versátil. Com o atributo type, conseguimos configurá-la de várias formas diferentes.

```
<input type="text" name="nome" size="30" maxlength="30">
```

Outros atributos:

minlength: número mínimo de caracteres a ser preenchido;
required: indica campo de preenchimento obrigatório;
value: indica o que ficará preenchido no campo por padrão;
disabled: desabilita o campo, evitando o seu preenchimento.

FORMULÁRIOS

- Caso queiramos um campo de envio de arquivos, escrevemos:

```
<input type="file" name="arquivo" size="30">
```

Para campos de escolha de alternativas, temos `radio` e `checkbox`:

```
<input type="radio" name="radio"> Escolha 1
<input type="radio" name="radio" checked> Escolha 2
<input type="radio" name="radio"> Escolha 3
<br>
<input type="checkbox" name="checkbox"> Escolha 1
<input type="checkbox" name="checkbox" checked> Escolha 2
<input type="checkbox" name="checkbox"> Escolha 3
```

FORMULÁRIOS

Para botões tipo "radio" e "checkbox", uso um label com a classe "checkbox". Assim, todo o campo, incluindo o rótulo e o botão, fica automaticamente clicável, melhorando a sua usabilidade (neste caso não precisamos do atributo for):

```
<label class="checkbox">
    <input type="radio" name="sexo" value="masculino"> Masculino
</label>
<label class="checkbox">
    <input type="radio" name="sexo" value="feminino"> Feminino
</label>
```

FORMULÁRIOS

A tag <button>, essa tag é mais fácil de trabalhar e de aplicar estilos que, por exemplo, uma tag input tipo "submit". Suas vantagens incluem uma aparência mais consistente entre navegadores e a possibilidade de inserir conteúdo variado entre as tags de abertura e fechamento (ex: texto e imagens).

```
<button type="submit" class="botao submit">Enviar</button>
```

Veja o exemplo completo:

exe8_4

FORMULÁRIOS

Aplicar CSS a formulários é simples.

```css
* {
    margin: 0;
    padding: 0;
}

fieldset {
    border: 0;
}

body, input, select, textarea, button {
    font-family: sans-serif;
    font-size: 1em;
}
```

Estilo reset (*) zerar as margens, além de retirar a borda padrão dos fieldsets. Padronizar a fonte e o tamanho de fonte nos campos.

exe8_5

FORMULÁRIOS

Melhorando o formulário.

```
.campo {
    margin-bottom: 1em;
}

.campo label {
    margin-bottom: 0.2em;
    color: #666;
    display: block;
}                   /*Como todos os campos estão agrupados dentro destes element
                        estilos como margens de forma uniforme */

fieldset.grupo .campo {
    float:  left;
    margin-right: 1em;
}                           /* Flutuamos os campos lado a lado Cidade e Estado */
```

Listas

- CSS proporciona propriedades especiais que são projetados para listas. Geralmente é mais conveniente usar estas propriedades sempre que puder.
- Para especificar o estilo para uma lista, use o list-style propriedade para especificar o tipo de marcador.
- O seletor na sua regra de CSS pode selecionar os elementos de item de lista (por exemplo, <li>), ou pode selecionar o elemento primário da lista (por exemplo, <ul>) de modo a que os elementos da lista herdam o modelo.

Listas não Ordenadas

Em uma lista não ordenada, cada item da lista é marcado da mesma forma.

Exemplos de marcadores, e aqui está como seu navegador exibe-os:

- disc
- circle
- square

Listas Não Ordenadas

```
    <style>
        li.open {
            list-style: circle;
        }
        li.closed {
            list-style: disc;
        }
    </style>
</head>
<body>


<ul>
  <li class="open"> São Paulo </ li>
  <li class="closed"> Rio de Janeiro </ li>
  <li class="closed"> Curitiba </ li>
  <li class="open"> Florianópolis </ li>
  <li class="closed"> Porto Alegre </ li>
</ ul>
```

exe8_6

Listas Ordenadas

Em uma lista ordenada , cada item da lista é marcado diferentemente para mostrar a sua posição na sequência.

Use a propriedade list-style para especificar o tipo de marcador:

- decimal
- lower-roman
- upper-roman
- lower-latin
- upper-latin

Listas Ordenadas

```
  <title>Exemplo - Lista  Ordenada</t
  <style>
    ol.info {
            list-style: decimal;
        }
  </style>
</head>
<body>


<ol class="info">
  <li>Flamengo</li>
  <li>Palmeiras</li>
  <li>Santos</li>
  <li>Grêmio</li>
  <li>São Paulo</li>
</ol>
```

exe8_7

List-style - Shorthand

```
<style>
  ul {
  list-style:  inside url("./imagens/brasil.svg");
  }
</style>
```

inside

e

outside

exe8_7 B

Contadores

Você pode usar contadores para numerar quaisquer elementos, não somente itens da lista. Por exemplo, em alguns documentos você pode querer numerar cabeçalhos, parágrafos ou sub-titulos.

Nota: Alguns navegadores não suportam contadores.

Contadores

Para especificar a numeração, você precisa de um *contador* com um nome que você especificar. Exemplo: num

Em alguns elementos antes da contagem é começar, reinicie o contador com a propriedade counter-reset e o nome do seu contador.

Em cada elemento que o contador é incrementado, use a propriedade counter-increment e o nome do seu contador.

Contadores

Para mostrar seu contador, adicione o pseudo-elemento :: before ou :: after para o seletor e usar o conteúdo da propriedade.

No valor do conteúdo de propriedade, especifique counter () com o nome do contador.

Contadores

```
<style>
    body {
        counter-reset:num;
    }
    p.numbered::before{
     content: counter(num) ": ";
     counter-increment: num;
     font-weight: bold;
    }
</style>
```

exe8_8

Menu

CSS possibilita definir uma variedade infinita de layouts e efeitos para um menu de navegação.

A tag <nav> define um conjunto de links de navegação.

Primeiramente faça um elemento NAV (antigamente usaríamos a tag DIV) e atribua um ID. Neste exemplo nossa NAV se chamará "menu".

Dentro desta NAV, faça uma lista com as opções do menu.

Menu

```
<nav id="menu">
    <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">Produtos</a></li>
        <li><a href="#">Missão</a></li>
        <li><a href="#">Links</a></li>
        <li><a href="#">Contato</a></li>
    </ul>
</nav>
```

Menu

- Agora que já fizemos a estrutura do menu, vamos formatá-lo com CSS.

```
<style>
    #menu ul {
                padding:0px;
                margin:0px;
                background-color:#EDEDED;
              list-style:none;
            }
</style>
```

exe8_9

Menu

- Para o menu ficar horizontal, temos que fazer as suas opções ficarem uma ao lado da outra, para isso, basta atribuir um display:inline; para a tag LI isso fará todas as opções ficarem em uma linha horizontal:

```
#menu ul li { display: inline; }
```

Menu

Menu vertical definindo um novo visual.

```
<style>
.vertical-menu {
  width: 200px;
}
.vertical-menu a {
  background-color: #eee;
  color: black;
  display: block;
  padding: 12px;
  text-decoration: none;
}
.vertical-menu a:hover {
  background-color: #ccc;
}
```

exe8_10

Cabeçalho

O Cabeçalho podemos tratar como um Box Model, assim como, o rodapé. Sendo assim é possível realizar vários efeitos com as propriedades border, margin, padding, entre outras.

Veja o exemplo a seguir:

Cabeçalho

Cabeçalho número 1

Cabeçalho número 2

Cabeçalho número 3

Cabeçalho número 4

Cabeçalho

```css
/* Header/Logo Title */
.header {
  padding: 60px;
  text-align: center;
  background: #1abc9c;
  color: white;
  font-size: 30px;
}
```

exe8_12

Box-sizing

A propriedade box-sizing do CSS nos permite incluir o preenchimento e a borda na largura e altura total de um elemento.

Por padrão, a largura e a altura de um elemento são calculadas assim:

width + padding + border = Largura do elemento
height + padding + border = Altura do elemento

Isso significa: quando você define a largura/altura de um elemento, o elemento geralmente parece maior do que você definiu (porque a borda e o preenchimento do elemento são adicionados à largura/altura especificada do elemento).

exe8_13