

Iniciante em Programação T5 - ONE

HTML5 e CSS3 parte 3: trabalhando com formulários e tabelas

Aprendizado – Começando uma nova página

Campo onde fazemos a busca, nada mais é do que um formulário.

Formulário: Responsável por pegar um conteúdo, enviar para uma outra página ou para algum lugar com o conteúdo que foi passado.

Sempre que estamos enviando informações que o usuário digitou de uma página para outra, nós estamos usando o formulário.

Importante: A informação digitada em uma página, vai para uma nova página a partir do momento do preenchimento do formulário e da busca com o enter.

O que aprendemos?

Nesta aula, vimos:

- ☐ Uma revisão do conteúdo aprendido no treinamento anterior
- ☐ Uma introdução ao projeto do treinamento
- ☐ A criação da página de contato
- ☐ Um pouco sobre os formulários

Aprendizado – Começando um formulário

Campos de dados para capturar informações do usuário.

Tag do formulário é a `<form>`

Dentro do formulário, nós temos as entradas de dados do usuário, e quando o usuário está entrando algum dado, ele está fazendo um `<input>` de informação.

Toda tag de `input` tem que ter um tipo (`type`), e a primeira informação que nós queremos absorver do usuário é o nome dele. Então o input tem sempre um tipo, e nesse caso o tipo é o `texto`, o usuário vai digitar texto em um campo.

```
<form>  
  <input type="text">  
</form>
```

Outra informação muito importante dentro de um formulário é o nome daquele campo. Se queremos que o usuário insira o nome dele, o nome completo dele, o nome e sobrenome, precisamos dizer isso aqui, "Nome e sobrenome".

Dentro do formulário, nós sempre usamos uma etiqueta para o `input`. É um `<label>`Nome e sobrenome`</label>` para um `input`. Esse par sempre anda junto. Eu tenho um `label` e tenho um `input`. Eu tenho uma informação sobre para que serve aquela entrada de dados e efetivamente a entrada de dados.

```
<form>  
  <label>Nome e sobrenome</label>  
  <input type="text">  
</form>
```

Aprendizado – Começando um formulário

Mas eu estou falando que isso é um par de informações, como eles então se conectam?

Toda vez que eu tenho um `input`, eu boto um identificador nele `<input type="text" id="nomesobrenome">`.

Um `input` geralmente é único, eu vou ter uma entrada do nome e sobrenome do usuário.

Label é uma etiqueta para um `input`, então `<label for="nomesobrenome">` que eu tenho aqui no meu `input`.

`for` e `id` -> Esse par anda junto.

`<form>`

`<label for="nomesobrenome">Nome e sobrenome</label>`

`<input type="text" id="nomesobrenome" >`

`</form>`

Aprendizado – Começando um formulário

Como enviar esses dados?

Todo formulário sempre vai ter um outro input, esse não precisa ter **label**, que vai ser do tipo `<input type="submit">`.

Ou seja, de envio, onde eu mando aqueles dados para um servidor, e-mail, ou para uma próxima página. Ao salvar recarregar, teremos um botão com o texto **"Enviar"**.

Como o meu navegador está em português, o sistema operacional está em português, ele traduz a palavra **"submit"**.

Como garantir que essa palavra seja sempre **"Enviar"**, independente da língua que meu usuário estiver usando?

Dizer que esse campo tem um valor, o **value** dele vai ser agora **"value="Enviar formulário">**

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

Essa é a estrutura básica que vamos ter sempre criando um formulário: a tag **"form"**, um **input**, que pega qualquer tipo de informação, e o **input** para enviar aqueles dados.

O **label** serve para criarmos uma estrutura melhor no nosso formulário, botando uma etiqueta para cada um dos campos onde temos entrada de dados do usuário. Ou seja, a tag `<label>` é uma etiqueta para a entrada de dados, para o `<input>`.

Aprendizado – Começando um formulário

Vimos que todo elemento tem um tipo de display. Ou ele é um display do tipo **inline** ou é um **display do tipo block**. Por padrão, os elementos como o parágrafo são display do tipo block, ou seja, eles **ocupam 100% da largura da página**. O elemento do tipo inline ocupa só o tamanho do seu conteúdo.

Todos os campos de um formulário são do tipo inline, eles só ocupam o tamanho do seu conteúdo. Após o **label**, imediatamente vem o **input**, e após o **input** imediatamente vem outro **label**. Para quebrar a estrutura todos os campos, que são os **labels** do formulário, devem ser block “display: **block**”.

```
form label {  
    display: block;  
}
```

Precisamos também que todos os inputs do meu formulário sejam block.

```
form input {  
    display: block;  
}
```

O que aprendemos?

Nesta aula, aprendemos:

- ☐ A criar um formulário HTML
 - A tag que o representa é a `<form>`
- ☐ A tag `<input>`, para a entrada de dados do usuário
- ☐ A criar uma etiqueta para o input, com a tag `<label>`
- ☐ A conectar um input com o seu label
 - Colocamos um id para o input e associamos esse id ao atributo for do label
- ☐ Alguns tipos de input, como text e submit
- ☐ Que label possui o display inline e o input possui display inline-block
- ☐ A estilizar o nosso formulário

Aprendizado – Tipos de campos diferentes

Entrada de texto -> campo de mensagem: campo de mensagem não é uma entrada simples de texto, não usamos a tag `input` nesse campo. Nós usamos a tag `<textarea>` para o usuário ter a possibilidade de colocar mais de uma linha de texto, um conteúdo complexo. O `textarea`, diferente da tag de `input`, é uma tag de conteúdo. Então ela abre e fecha `<textarea></textarea>`.

Configurar no HTML quantas colunas e quantas linhas de texto essa caixa vai ter.

`"cols="30"` -> O número de `colunas (cols)` no campo é de 30, vai ter uma `largura` grande para inserir o conteúdo.

`"rows="10"` -> O número de `linhas (rows)` é de 10, para ele ter uma `altura` boa.

```
<textarea cols="70" rows="10" ></textarea>
```

Se quisermos que ela tenha uma largura maior que seja proporcional ao `input`, aumentar a quantidade de `colunas (cols)`

Para completar esse campo, tá faltando o `label`, a etiqueta dizendo qual é o conteúdo daquela `textarea`. E para isso nós vamos usar a mesma tag que usamos no `input`, que é a tag do `label` `<label for="mensagem">Mensagem</label>`. Já vimos que para o `for` do `label` funcionar, eu preciso ter um `id` no meu elemento. Então vou ter aqui um `id="mensagem"`.

```
<label for="mensagem">Mensagem</label>
```

```
<textarea cols="70" rows="10" id="mensagem" ></textarea>
```

Campo do "Selecionar" -> Três opções de seleção, é chamado de Radio Box. Esse `input` tem a configuração de serem três opções referentes a um único campo.

Aprendizado – Tipos de campos diferentes

E como isso se dá no HTML?

Criar um input que vai ter o tipo “radio”, e assim como no input do submit, ele vai ter um valor, e o primeiro vai ser “email”, <input type=“radio” value=“email”>.

Para que o texto apareça, colocar um label. Antes do input, adicionar um label chamado “Email”. Esse label vai ter um for=“radio-email”. Colocar esse mesmo conteúdo em uma id chamado radio-email, <input type=“radio” value=“email” id=“radio-email”>.

```
<label for="radio-email">Email</label>  
<input type="radio" value="email" id="radio-email">
```

Quando criamos um input do tipo radio ele deixar marcar várias opções, mas ele precisa ter uma única identificação, um único termo para que eles todos permaneçam sempre um marcado. Eles precisam todos fazer parte de um grupo, para que a configuração de quando um clica o outro desclica funcione.

Para isso, nos três itens do input criar uma nova configuração, que é uma nova propriedade, o name=“contato”.

Essa configuração faz com que todos eles passem a ser do mesmo grupo, e, ao clicar em um, ele desmarca o outro.

```
<label for="radio-email">Email</label>  
<input type="radio" name="contato" value="email" id="radio-email" >
```

Checkbox: Quando temos um label e um input eles formam um par, mas existe uma estratégia que pode ser feita para poupar esse trabalho, que é adicionar o input dentro do label.

Criar um input do tipo checkbox dentro do label.

```
<label for=""><input type="checkbox"> Gostaria de receber nossas novidades por email?</label>
```

Aprendizado – Tipos de campos diferentes

No CSS, quando criamos uma configuração específica para um elemento e queremos replicar a mesma configuração para outro elemento, usamos a vírgula.

form input,

form textarea {

display: **block**;

margin: **0 0 20px**;

padding: **10px 25px**;

width: **50%**;

}

O **textarea** e o **input** agora têm as mesmas configurações de espaçamento interno, espaçamento externo, de largura e de visualização do tipo de display.

Colocando o **input** dentro do **label**:

```
<label for="radio-email"><input type="radio" name="contato" value="email" id="radio-email">Email</label>
```

Aprendizado – Tipos de campos diferentes

Quando temos duas tags, nós temos o dobro da força do seletor. Todo seletor é configurado a partir de uma força.

<i>inline</i>	<i>id</i>	<i>class</i>	<i>tag</i>
1000	100	10	1

p 1
form p 1+1 =2
.teste 10

Imagine que nessas três caixas nós temos a primeira sendo o identificador, a segunda sendo a classe, e por último a tag.

Tag -> força 1

Classe -> força 10

Identificador -> força 100.

Toda vez que temos um seletor daquele tipo, "**p**", isso quer dizer que a força desse seletor é 1.

Quando temos o "**form p**" a força disso é 1+1, a força disso é 2. Então 2, como é mais forte que o 1, o estilo aplicado vai ser o segundo, do "**form p**".

.teste é uma classe aplicada no **p** -> A classe tem uma força 10, ela é superior a esses dois marcadores que colocamos aqui. Então "**.teste** 10".

Marcador com configuração -> **p.teste**

Só os parágrafos que têm aquela classe vão ter essa cor.

Somamos, a classe tem a força 10, e a tag tem a força 1, então com 11 ele vai ser mais forte que o teste especificamente.

O mais forte deles é o **identificador**, tem a **força 100**. Então sempre que estamos criando CSS, precisamos pensar em o quão específico é o nosso marcador e o quão forte ele vai ser para que não seja sobrescrito por qualquer outro, e que não cometamos nenhum erro no nosso código.

Aprendizado – Tipos de campos diferentes

A única forma de alterarmos isso e alguma coisa mais forte que o identificador é quando temos o **estilo inline**.

O estilo inline é no HTML. Então se adicionarmos uma propriedade "style", e colocarmos o color igual a roxo, no nosso navegador, ao recarregar, a cor vai ser roxo.

Nada substitui o inline, ele é muito específico, ele está no elemento, e por isso ele é o mais forte. Ele teria o que seria equivalente a uma força 1000, e nada substitui isso.

Aprendizado – Tipos de campos diferentes

O seletor não é um campo do tipo input, ele é um campo do tipo `select`, e um campo do tipo `select` tem dentro dele várias opções.

As opções são campos de conteúdo, e aqui eu vou colocar os valores:

`<option>`Manhã`</option>`, uma outra opção que é a `<option>`Tarde`</option>` e uma outra opção que é a `<option>`Noite`</option>`.

A estrutura da tag `<select>` é composta de um ou mais `<option>`.

Nesta aula, vimos:

- ☐ O textarea, para entradas de texto de mais de uma linha
- ☐ O input do tipo radio
- ☐ Como agrupar vários input do tipo radio, impedindo que mais de um input seja selecionado
- ☐ O input do tipo checkbox
- ☐ Que podemos criar um input dentro de um label, assim associando-os
- ☐ Mais estilizações para a nossa página
- ☐ Como funciona a hierarquia no CSS
- ☐ O select, que é seletor, um campo de seleção de um item, e o option, que representa cada opção do seletor

Aprendizado – Melhorando a semântica

INPUTS PARA CELULARES

Mobile Input Types: Mostra visualmente o resultado esperado quando estamos usando um tipo de HTML5 de input em um celular.

SITE: <http://mobileinputtypes.com/>

Quando temos um input do tipo texto, o teclado permanece o mesmo, o teclado padrão do celular.

Aprendizado – Melhorando a semântica

INPUTS PARA CELULARES

input o tipo "email" -> Ao invés de uma barra de espaço muito grande, a barra de espaço é dividida e terá um arroba, um ponto. Em alguns teclados e sistemas operacionais, terá um ".com". Isso facilita a vida do usuário na hora de cadastrar o seu e-mail.

```
<input type="email">
```

input tipo "tel" -> Troca o teclado para teclado numérico

```
<input type="tel">
```

input tipo "number" -> Muda para o teclado de número e o usuário consegue entrar os dados facilmente. Exemplo: Inserção de CEP, CPF em algum formulário de compra de um site. O ideal é que se input seja do tipo number para entrar com os dados rapidamente.

```
<input type="number">
```

input tipo "password" -> Transforma o conteúdo digitado em um asterisco ou em uma bolinha para que não seja possível visualizar diretamente nossa senha.

```
<input type="password">
```

input tipo "date" -> Abre o seletor de data do telefone, sendo possível rolar para achar o dia, mês e ano.

```
<input type="date">
```

input tipo "datetime" -> Além da data, abre o teclado, onde é possível escolher o dia, as horas e segundos.

```
<input type="datetime">
```

input tipo "month" -> É possível selecionar o mês e o ano.

```
<input type="month">
```

input tipo "busca" -> Irá uma caixa de busca. É uma mudança bem sutil é que ao invés de ser "Go" ou "Done", o campo de envio vira "Pesquisar", "Buscar".

```
<input type="search">
```

Aprendizado – Melhorando a semântica

DADOS IMPORTANTES NOS INPUTS

Regras para deixar campos obrigatórios

No HTML5, sempre que temos um `input` e queremos que ele seja preenchido, nós podemos colocar na tag do `input` uma palavra reservada chamada "**required**", que quer dizer "obrigatório".

Ao tentar enviar o formulário, ele volta para aquele campo e me diz que aquele campo precisa ser preenchido, ou seja, antes de enviar o formulário, aquele campo precisa ter qualquer conteúdo. Tornamos o **preenchimento daquele campo obrigatório**.

placeholder="" -> Sugestão de preenchimento

checked -> Sugestão de marcação

É útil fazer quando temos um checkbox e queremos que ele seja marcado, também usamos a palavra reservada "**checked**" no `input` do checkbox.

Aprendizado – Melhorando a semântica

MELHORANDO A SEMÂNTICA DO FORMULÁRIO

Dentro do formulário, quando queremos agrupar os campos e quando queremos ter um título para isso, temos tags específicas, que deixam nosso formulário melhor escrito. A primeira é a divisão, quando temos um campo e um texto, ou vários campos referentes a alguma coisa, não usamos a tag `div`, usamos a tag `fieldset`. Ela é referente à configuração de um ou mais campos referentes a um assunto específico.

Dentro de um `fieldset` não temos parágrafos, nós temos o título, e o título de um `fieldset` é chamado de `legend`.

`alt=""` nas imagens -> Campo importante para acessibilidade.

`Alt` é uma alternativa. Descreve a imagem.

Nesta aula, aprendemos:

- ☐ Alguns tipos de inputs para celular: email, tel, number, password, date, datetime, month e Search
- ☐ Como não permitir que um campo não seja preenchido, através do atributo `required`
- ☐ Como exibir uma sugestão de preenchimento para os campos, através do atributo `placeholder`
- ☐ Como deixar uma opção marcada por padrão nos nossos input radio e checkbox, através do atributo `checked`
- ☐ Como estruturar melhor o nosso código com `fieldset` e `legend`
- ☐ Como adicionar uma alternativa à imagem, descrevendo-a, com o atributo `alt`

Aprendizado – CSS AVANÇADO

O QUE SÃO TRANSIÇÕES?

Propriedade transition: Podemos dizer o tempo de transição de uma cor para outra.

Propriedade cursor: **pointer** -> mostra ao usuário uma mãozinha e ele consegue saber que é clicável.

ENTENDENDO TRANSFORMAÇÕES

Propriedade transform: Passa o mouse por cima e aumenta o elemento

Propriedade transform: **scale()** aumenta um elemento proporcionalmente.

Propriedade transform: **rotate()** "gira" um elemento em seu eixo.

Nesta aula, aprendemos:

- ❑ Como estilizar o botão de envio de formulário
- ❑ A realizar transições nos nossos elementos, com a propriedade CSS transition
- ❑ A modificar o estilo do ponteiro do mouse, quando passar por cima de determinado elemento, através da propriedade CSS cursor
- ❑ A realizar transformações nos nossos elementos, como aumentar proporcionalmente a escala de determinado elemento ou rotacioná-lo, através da propriedade CSS transform

Aprendizado – Estrutura de Tabelas

TABELA BÁSICA

<table>

tag para tabela: Uma tabela é uma sequência de linhas e colunas

<tr>

É uma linha

<td>

Células

.

TAGS SEMÂNTICAS PARA TABELAS

Primeira linha -> É a mais importante, é a linha que dita o conteúdo daquela coluna, que mostra o que está sendo apresentado naquela tabela. É a linha mais importante, onde temos o cabeçalho das informações da tabela. E para isso podemos marcar com uma tag mais específica, uma tag mais semântica, que nos diz que aquilo é o cabeçalho.

Dentro de uma tabela, podemos separá-la em três: o cabeçalho, o conteúdo e o rodapé.

Cabeçalho **<thead>**

Dentro do cabeçalho, para dizer que aquela célula é uma célula do cabeçalho, ao invés de usarmos a tag **<td>**, usamos ainda a tag **<th>**, que é uma célula do **head** da tabela. Com isso, nós conseguimos ter um cabeçalho bem mais semântico.

Corpo da tabela **<tbody>**

As tags **<thead>**, **<tbody>** e **<tfoot>** ajudam a deixar o conteúdo da tabela mais bem dividido e mais semântico.

Aprendizado – Estrutura de Tabelas

PARA SABER MAIS: JUNTANDO CÉLULAS DA TABELA

As tabelas também nos oferecem a possibilidade de juntar células e montar um visual diferente. Por exemplo, quando uma linha, que deveria ter 5 células, passa a mostrar só "uma célula".

Esse efeito é conseguido através da propriedade `colspan=X`, onde X é o número de células que você quer agrupar.

Portanto, em uma tabela de 5 colunas, para ter uma célula única na linha, usamos um código assim:

```
<tr>  
  <td colspan="5">Rio de Janeiro</td>  
</tr>
```

Nesta aula, aprendemos:

- ❑ A criar uma tabela HTML
 - A tag `table`, que representa a tabela
 - A tag `tr`, que representa a linha da tabela
 - A tag `td`, que representa a célula da tabela
 - A tag `thead`, que representa o cabeçalho da tabela
 - A tag `tbody`, que representa o corpo da tabela
 - A tag `th`, que representa a célula do cabeçalho da tabela
 - A tag `tfoot`, que representa o rodapé da tabela
- ❑ A estilizar a tabela

Aprendizado – Estrutura de Tabelas

CONCLUSÃO

- ❑ Acessibilidade da imagem com alternativa.
- ❑ Criação de formulário com a tag do form.
- ❑ Tag do label
- ❑ Tag do input
- ❑ Tipos de input
- ❑ Relação entre uma label e um input.
- ❑ Classes novamente
- ❑ Input obrigatório
- ❑ Texto de exemplo no input
- ❑ Criação de caixa para o usuário escrever um texto
- ❑ Criação de boa semântica dentro do formulário com o fieldset e com o legend
- ❑ Criação de um tipo, uma seleção de várias opções dentro do formulário
- ❑ Uso checkbox, os input do tipo radio
- ❑ Criação de um botão e estilização.
- ❑ Tabela semântica
- ❑ Criação de duplicação do mesmo estilo para dois elementos
- ❑ Uso do display block
- ❑ Uso das fontes e dos espaçamentos
- ❑ Criação de transições e transformações nos elementos.