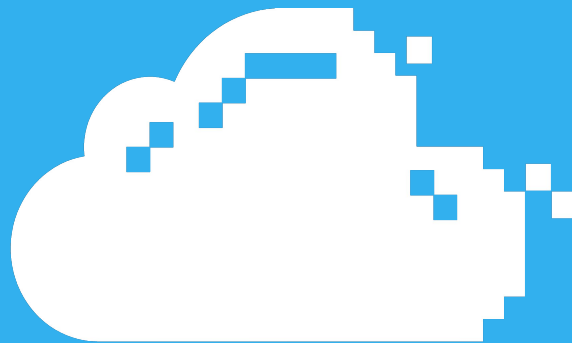


PROCESSO SELETIVO

AULA 6

Unidades de CSS
Formulários



BitWeb_{jr}

Unidades de CSS



Unidades Absolutas

Unidades absolutas possuem um valor fixo e sempre serão exibidas exatamente no tamanho indicado.

Não é recomendado usar unidades absolutas para exibição em telas por causa da grande variedade de tamanhos de tela.

Unidades absolutas são úteis quando o tamanho do meio é previamente conhecido como um layout de impressão por exemplo.

Tabela de Unidades Absolutas

cm	Centímetros
mm	Milímetros
in	Polegadas(1in = 96px = 2.54cm)
px*	Pixels(1px = 1/96th de 1in)
pt	Pontos(1pt = 1/72 de 1in)
pc	Picas(1pc = 12pt)

* Pixels são relativos a tela do aparelho

Unidades Relativas

O tamanho de uma unidade relativa é relativa a outra propriedade.

Unidades relativas escalam melhor entre meios diferentes.

Apesar da utilidade dessas unidades, elas são mais difíceis de usar e quando usadas incorretamente podem ter consequências não previstas.

Tabela de Unidades Relativas

em	Relativo ao tamanho de font do elemento
ch	Relativo a largura do caracter “0”
rem	Relativo ao tamanho de font do elemento raiz
vw	Relativo a 1% da largura da viewport
vh	Relativo a 1% da altura da viewport
vmin	Relativo a 1% da menor dimensão da viewport
vmax	Relativo a 1% da maior dimensão da viewport
%	Relativo ao elemento pai

Formulários



Formulários

Formulários permitem que os **usuários insiram dados** na página e retornar esses dados para o web site.

Eles são criados usando o elemento **<form>**

```
<form action="/meuHandler" method="">
```

```
</form>
```


Elemento <form>

Este elemento **define formalmente o formulário**. Os **atributos são opcionais** mas é boa prática implementar o **atributo action** e o **atributo method**.

```
<form action="/meuHandler" method="get">
```

```
</form>
```

Atributo action

O **atributo action** define o endereço(URL) para enviar os dados do formulário.

```
<form action="/meuHandler" method="get">
```

```
</form>
```

Atributo method

O **atributo method** define o **método HTTP** usado para mandar os dados. O atributo pode ter o valor **get**, que passa os dados pela URL, ou **post**, que passa os dados pelo corpo do pedido.

```
<form action="/meuHandler" method="get">
```

```
</form>
```

Label e Input

Um **campo** de um formulário é formado por um **par de elementos** `<label>` e `<input>`. A ligação entre eles é feita pelo **atributo** `for` do `<label>` e o **atributo** `id` de `<input>`.

```
<form action="/meuHandler" method="get">  
  <label for="nome">Nome:</label>  
  <input type="text" id="nome" name="user_name">  
</form>
```

Label

O elemento `<label>` representa uma legenda de um campo do formulário. **Associar o `<label>` usando o atributo `for` provê vantagens como ajudar leitores de tela e associar o evento de click do campo com o label.**

```
<form action="/meuHandler" method="get">  
  <input type="checkbox" id="terms" name="terms_of_use">  
  <label for="terms">Não li mas aceito os termos.</label>  
</form>
```

Input

O elemento `<input>` é usado para criar uma **interface para o usuário inserir dados**.
A **interface e seu comportamento é definido pelos seus atributos**, em especial o **atributo type**.

```
<form action="/meuHandler" method="get">  
  <label for="email">Email:</label>  
  <input type="email" id="email" name="user_email">  
</form>
```

Input type="text"

O valor padrão do **atributo type** é **text**, que exibe uma **caixa de texto que aceita apenas uma linha**, removendo quebra de linhas.

```
<form action="/meuHandler" method="get">  
  <label for="nome">Nome:</label>  
  <input type="text" id="nome" name="user_name">  
</form>
```

Input type="number"

O valor **number** exibe uma **caixa que aceita números**.

```
<form action="/meuHandler" method="get">  
  <label for="life">Significado da vida:</label>  
  <input type="number" id="life" name="truth">  
</form>
```


Input type="password"

O valor **password** exibe uma **caixa de texto que esconde os caracteres digitados**.

Usado para ocultar a senha na interface.

O exemplo abaixo tem um falha de segurança.

```
<form action="/meuHandler" method="get">  
  <label for="secret">Seu segredo mais profundo:</label>  
  <input type="password" id="secret" name="õ_ô">  
</form>
```

Input type="checkbox"

O valor **checkbox** exibe uma **caixa de seleção** que pode estar **checada** ou **não**.

```
<form action="/meuHandler" method="get">  
  <input type="checkbox" id="human" name="human">  
  <label for="human">Marque se você é humano.</label>  
</form>
```

Input type="radio"

O **valor radio** é similar ao **valor checkbox**, mas deve **ser agrupado** e **apenas um input do grupo pode estar selecionado**. Um grupo de radio é definido pelo **atributo name**.

```
<p>Qual o melhor pokemon?</p>
<input type="radio" id="bat" value="41" name="bestmon">
<label for="bat">Zubat</label>
<input type="radio" id="cool" value="71" name="bestmon">
<label for="cool">Tentacool</label>
```

<fieldset> e <legend>

Podemos **agrupar campos de formularios em um <fieldset>** e adicionar legendas a esses grupos usando o **elemento <legend>**. Esses grupos possuem uso **semântico** e para **estilizar** o grupo e são usados por **tecnologias de assistência**.

```
<fieldset>
  <legend>Qual o melhor pokemon?</legend>
  <input type="radio" id="bat" value="41" name="bestmon">
  <label for="bat">Zubat</label>
  <input type="radio" id="cool" value="71" name="bestmon">
  <label for="cool">Tentacool</label>
</fieldset>
```

Botão de submissão

Para criar um botão para **enviar os dados de um formulário** basta criar um elemento `<button>` com **atributo type** com **valor submit**.

```
<form action="/meuHandler" method="get">  
  <button type="submit">Send your message</button>  
</form>
```