

<head>

</head>

<body>

Tópicos Especiais em Desenvolvimento de Software I

Aula 4

*Formulários em HTML e pseudo-classe / pseudo-
elementos*

</body>

<footer>

>

</footer>

<head>

Formulários e HTML

</head>

<body>

- A especificação da linguagem HTML inclui uma série de elementos destinados a construção de formulários para páginas e sistemas web.
- Esses elementos são utilizados, principalmente, para o envio de dados/informações/arquivos/etc por parte dos usuários:
 - Ou seja, para que o usuário possa enviar dados do lado do cliente para o lado do servidor.
- Tornando a utilização de formulários imprescindível em sistemas web.

</body>

<footer>

>

</footer>

<head>

Tag <form>

</head>

<body>

- A tag **<form>** é utilizada para descrever e delimitar a existência de um formulário no documento HTML. Sintaxe básica:

```
<form>
```

```
...
```

```
[elementos de formulário]
```

```
...
```

```
</form>
```

- Importante: permite-se vários formulário por página, porém, não é permitido aninhá-los, geralmente, considera-se apenas o primeiro.

</body>

<footer>

>

</footer>

<head>

Atributos

</head>

<body>

- Principais atributos:
 - **id**: String com identificação única do elemento
 - **action**: URI que define para “onde” será enviada (submetida) as informações do formulário
 - **method**: Define o método de envio dos dados do formulário, como por exemplo:
 - POST dados enviados no corpo da requisição;
 - GET dados enviados pela URL.

Mais informações e atributos sobre formulários:

<https://developer.mozilla.org/docs/Web/HTML/Element/form>

<footer>

>

</footer>

<head>

Componentes de um formulário

</head>

<body>

- Elementos que são tradicionalmente inseridos entre as tags `<form>...</form>` para permitir que o usuário entre com dados, a citar:
- `input [text | password | radio | checkbox | reset | submit | hidden]`
- `textarea`
- `select`
- `option`
- `optgroup`

</body>

<footer>

>

</footer>

<head>

Componentes de um formulário

</head>

<body>

- Continuando:
 - button
 - label
 - fieldset
 - legend

Importante: Assim como nos demais elementos HTML, também é possível aplicar aos elementos de formulário uma grande gama de estilos utilizando CSS

<footer>

>

</footer>

<head>

Exemplo

</head>

<body>

```
<form id="frm-add-aluno" action="dados.html" method="get">
  <label for="txt-rga">RGA:</label>
  <input type="text" id="txt-rga" name="txt-rga" />
  <br />
  <label for="txt-nome-completo">Nome Completo:</label>
  <input type="text" id="txt-nome-completo" name="txt-nome-completo" />
  <br />
  <label for="psswd-senha-aluno">Senha</label>
  <input type="password" id="psswd-senha-aluno" name="psswd-senha-aluno" />
  <br />
  <span>Curso:</span>
  <br />
  <input type="radio" id="rd-curso-si" name="rd-curso" />&nbsp;  
  <label for="rd-curso-si">Sistemas de Informação</label>
  <br />
  <input type="radio" id="rd-curso-co" name="rd-curso" />&nbsp;  
  <label for="rd-curso-co">Ciência da Computação</label>
  <br />
  <button type="submit">Cadastrar</button>
</form>
```

</body>

<footer>

>

</footer>

<head>

Exemplo

</head>

<body>

- Resultado:

RGA:

Nome Completo:

Senha

Curso:

- ☒ Sistemas de Informação
- ☐ Ciência da Computação

Importante: assim como os demais elementos HTML, os elementos de formulário também podem ser estilizados utilizando com CSS da forma que quisermos (fonte, borda, display, margem, etc)

<footer>

>

</footer>

<head>

HTML5

</head>

<body>

- Elementos e atributos relacionados a formulários introduzidos no HTML5:
 - *Input type*:
 - *color, date, datetime, number, range, etc.*
 - Atributos:
 - *autocomplete, autofocus, formaction, list, multiple, placeholder, required, contenteditable, min, max, step, etc.*
 - Elementos:
 - *datalist, keygen, output*

</body>

<footer>

>

</footer>

<head>

Pseudo-classes

</head>

<body>

- Pseudo-classe é uma palavra chave adicionada aos seletores CSS com o objetivo de permitir a definição de estilos para condições (casos) específicos, como por exemplo, quando um elemento estiver ativo, quando o cursor do *mouse* estiver sobre um elemento, entre outros.

Sintaxe:

```
seletor:pseudo-classe {  
    propriedade: valor;  
}
```

</body>

<footer>

>

</footer>

<head>

Pseudo-classes

</head>

<body>

- Exemplo:

```
div:hover {  
    background-color: #008800;  
}
```

<https://developer.mozilla.org/pt-BR/docs/Web/CSS/Pseudo-classes>

https://www.w3schools.com/css/css_pseudo_classes.asp

</body>

<footer>

>

</footer>

<head>

Pseudo-classes

</head>

<body>

- Algumas pseudo-classes interessantes:

:hover

Instantâneo em que o cursor do mouse é passado sobre um elemento

:link

Pseudo classe referente aos links “dentro” de um elemento

:visited

Links já visitados

:active

Links ativos e o instante após clicar e antes de “soltar” o cursor do mouse sobre um link / botão

Importante: para um funcionamento adequado (onde as propriedades não se sobrescrevam de maneira indesejada) a ordem adequada é: LVHA-order: *:link* — *:visited* — *:hover* — *:active*
<https://developer.mozilla.org/pt-BR/docs/Web/CSS/:active>

<footer>

>

</footer>

<head>

Exemplo

</head>

<body>

```
body { color: black }
a:link { color: CornflowerBlue }
a:visited { color: DarkSlateBlue }
a:hover { color: DarkOrchid }
a:active { color: GreenYellow }
```

<body>

<p>

Exemplo do comportamento do link alterado
utilizando pseudo-classes

Pseudo Classes.

</p>

</body>

</body>

<footer>

>

</footer>

<head>

Pseudo-classes

</head>

<body>

- ...mais pseudo-classes interessantes:

:first-child()

Faz referência somente aos elementos que sejam o **primeiro filho** do seu (elemento) pai

:last-of-type()

Faz referência somente aos elementos que sejam o **último filho** do seu (elemento) pai

:nth-child(*n*)

Faz referência somente aos elementos que sejam ***n-ésimo* filho** do seu (elemento) pai

:not(*seletor*)

Faz referência a “todos” os elementos com **exceção** dos elementos definidos pelo ***seletor***

</body>

<footer>

>

</footer>

<head>

exemplo

</head>

<body>

```
li:not(.do-contras) {  
  color: Gainsboro;  
}  
  
li:first-child {  
  color: green;  
}  
  
li:last-child {  
  color: DarkRed;  
}  
  
li:nth-child(3) {  
  color: DarkViolet;  
}
```

```
<ul>  
  <li>Lorem Ipsum</li>  
  <li>Lorem Ipsum</li>  
  <li>Lorem Ipsum</li>  
  <li>Lorem Ipsum</li>  
  <li class="do-contras">Lorem Ipsum</li>  
  <li>Lorem Ipsum</li>  
</ul>
```

</body>

<footer>

>

</footer>

<head>

pseudo-elementos

</head>

<body>

- Uma vez que pseudo-classes nos permite adicionar estilos à condições (casos específicos), os **pseudo-elementos**, por sua vez, nos permite especificar o estilo “partes específicas” de um elemento.
- Por exemplo o seguinte parágrafo:

```
<p> Lorem ipsum dolor sit, amet consectetur adipisicing elit.  
Nostrum cum non rem, est nobis voluptate sint ab atque  
perferendis ipsum nisi, porro exercitationem ratione accusantium,  
vero velit eaque harum quidem.  
Eveniet vero exercitationem beatae minima nostrum dolore deserunt  
nemo praesentium quaerat voluptatum. Non saepe natus eum molestias  
repudiandae libero eaque animi suscipit modi, accusantium  
consequatur excepturi, perferendis obcaecati animi libero?</p>
```

Não há nenhum elemento referente à primeira linha ou primeira letra do texto do parágrafo, porém é possível nos referirmos a elas utilizando os pseudo-elementos **::first-line** e **::first-letter**

<https://developer.mozilla.org/pt-BR/docs/Web/CSS/Pseudo-elements>

</body>

<footer>

>

</footer>

<head>

Exercício

</head>

<body>

- Criar um formulário com pelo menos 10 (grupos de) elementos de formulários:
 - Elementos obrigatórios:
 - 2 inputs do tipo texto
 - 1 área de texto
 - 1 select com pelo menos 3 opções
 - checkbox e radiobutton
 - Deve haver pelo menos um botão (que não entra na conta dos 10 elementos)
 - Formatar / estilizar o formulário com pelo menos:
 - 2 seletores aninhados
 - 2 seletores com pseudo-classes
 - 2 seletores com pseudo-elementos

</body>

<footer>

>

</footer>