



Fundamentos em Desenvolvimento Front End

Capítulo 3. DOM, Formulários e Eventos

Prof. Raphael Gomide



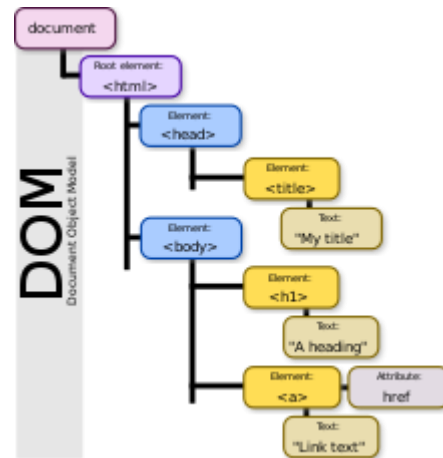
Aula 3.1. Manipulando o DOM com JavaScript

□ DOM:

- Introdução.
- O comando `querySelector`.
- O comando `querySelectorAll`.

DOM – Introdução

- Document Object Model.
- Representa uma árvore de objetos em memória.
- No caso da web, mapeia documentos HTML em objetos.
- Com isso, pode-se facilmente recuperar e modificar dados dos documentos.
- Para mais informações sobre o DOM, acesse [este link](#).



O comando *querySelector()*

- Principal comando para acessar elementos do DOM um-a-um.
- Opções de parâmetros:
 - Elemento / Classes / Id's.
- Acompanhe o professor.

```
> var firstParagraph = document.querySelector('p');  
↵ undefined  
  
> var mediawiki = document.querySelector('.mediawiki');  
↵ undefined  
  
> var content = document.querySelector('#content');  
↵ undefined  
  
> firstParagraph  
↵ ▶ <p>...</p>  
  
> mediawiki  
↵ ▶ {redefineFallbacksForTest: undefined, trackQueue: Array(0), now: f, track: f, trackError: f, ...}  
  
> content  
↵ ▶ <div id="content" class="mw-body" role="main">...</div>
```

O comando *querySelectorAll()*

- Principal comando para acessar elementos agrupados do DOM.
- O retorno é do tipo NodeList. Para converter para array (mais comum), use Array.from
- Opções de parâmetros:
 - Elemento / Classes / Id's.
- Acompanhe o professor.

```
> var allParagraphs = document.querySelectorAll('p');  
⏏ undefined  
  
> allParagraphs  
⏏ ▶ NodeList(16) [p, p, p, p, p, p, p, p, p, p, p, p, p, p, p, p]  
  
> var arrayParagraphs = Array.from(allParagraphs);  
⏏ undefined  
  
> arrayParagraphs  
⏏ ▶ (16) [p, p, p, p, p, p, p, p, p, p, p, p, p, p, p, p]
```

A propriedade *textContent*

- Refere-se ao conteúdo textual de alguns elementos.
- Acompanhe o professor.

Parágrafo 1

```
> var element = |
```

- ☑ Na web, é possível manipular o DOM com JavaScript.
- ☑ **querySelector** é utilizado para se obter um único elemento.
- ☑ **querySelectorAll** é utilizado para se obter vários elementos semelhantes.
- ☑ **textContent** refere-se ao conteúdo textual de diversos elementos.

- ❑ Manipulando o CSS com JavaScript.



Aula 3.2. Manipulando o CSS com JavaScript

❑ CSS com JavaScript:

- A propriedade `style`.
- O comando `classList.add`.
- O comando `classList.remove`.

A propriedade *style*

- É possível modificar o CSS diretamente através da propriedade *style*.
- Entretanto, esta **não** é uma boa prática.
- Acompanhe o professor.
- Não é interessante injetarmos código CSS em nosso JavaScript.
- Uma boa prática seria a adição e remoção de classes.

Parágrafo 1

```
> var p |
```

O comando *classList.add()* e *classList.remove()*

- Com estes comandos, podemos adicionar e remover classes de elementos.
- Acompanhe o professor.

```
<style>
  .good {
    color: blue;
  }
  .great {
    color: green;
  }
  .awesome {
    color: purple;
  }
</style>
</head>
<body>
  <p id="p1">Parágrafo 1</p>
  <p id="p2">Parágrafo 2</p>
  <p id="p3">Parágrafo 3</p>
</body>
```

Parágrafo 1

Parágrafo 2

Parágrafo 3

```
> var p1 = document.getElementById('p1');
p1.classList.add('good');
p1.classList.add('great');
p1.classList.add('awesome');
```

- ☑ Na web, é possível manipular o CSS com JavaScript.
- ☑ Não é uma boa prática manipular o CSS diretamente.
- ☑ É melhor definir classes de CSS em arquivos isolados e invocar **classList.add** e/ou **classList.remove**.

Próxima aula

☐ Formulários HTML.



Aula 3.3. Formulários HTML

☐ Formulários HTML:

- Introdução.
- Principais elementos.
- Manipulação com JavaScript.

- Principal interação com os usuários em sistemas web.
- Tag **<form>**.
- Funcionamento padrão:
 - Botão submit – envia os dados ao servidor;
 - Botão reset – limpa os dados do formulário.
- Funcionamento atual:
 - JavaScript intercepta os dados.
 - Envio de dados ao servidor feito de forma assíncrona.
 - Tag **<button>** também é muito utilizada.

Formulários – Campos de input

- **<textarea>** - usado para grandes quantidades de texto.
- **<input>** - elemento mais utilizado, contendo:
 - Caixas de texto simples.
 - Data/hora.
 - Checkboxes.
 - Cores.
 - Números.
 - E-mail.
 - Botões submit e reset.
 - Outros.
- Renderização varia conforme o navegador.
- Acompanhe o professor.

The screenshot displays a web form with several input fields and validation messages. At the top, there is a text input field containing 'gmailcom' with a validation message 'Insira um URL.' (Insert a URL). Below it, a text input field contains 'teste' and a button labeled 'Enviar'. Another text input field contains 'raphaelyahoo.com' with a validation message 'Inclua um "@" no endereço de e-mail. "raphaelyahoo.com" está com um "@" faltando.' (Include an '@' in the email address. 'raphaelyahoo.com' is missing an '@'). To the right of this field is a dropdown menu showing the number '3'. Below these, there is another 'Enviar' button and a slider control. Further down, there is a date input field showing 'agosto de 2016' and a time input field showing '03:01'. At the bottom, there is a date and time input field showing '06/08/2015 02:03'. To the right of these fields is a calendar widget for August 2015, showing a grid of days with the 25th highlighted.

dom	seg	ter	qua	qui	sex	sáb
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

☑ Formulários:

- Principal forma de interação com o usuário.
- Vários elementos disponíveis.
- Diversos tipos de inputs e botões.
- Principais atributos utilizados pelo JavaScript: class/id e value.

- ❑ Manipulação de eventos.



Aula 3.4. JavaScript – Manipulação de eventos

☐ JavaScript – Manipulação de eventos:

- Introdução.
- Eventos em botões.
- Eventos em inputs.
- Manipulação com JavaScript.
- Evitando envio de dados ao servidor com JavaScript.

- Representam o “quando” das aplicações.
- A implementação é feita através de “escutadores” (*listeners*), com a função **addEventListener**.
- Acompanhe o professor.

Evento	Ativação	Categoria
onload	Após o carregamento	document / window / body
onchange	Conteúdo do elemento alterado	form
onfocus	Elemento recebe foco	form
onblur	Elemento perde o foco	form
onselect	Elemento é selecionado	form
onsubmit	Dados do formulário são enviados ao servidor	form
onkeydown	Tecla pressionada	teclado
onkeypress	Tecla pressionada e solta	teclado
onkeyup	Tecla solta	teclado
onclick	Clique do mouse no elemento	mouse
ondblclick	Clique duplo do mouse no elemento	mouse
onmousemove	Mouse se moveu sobre o elemento	mouse
onmouseout	Mouse saiu do elemento	mouse
onmouseover	Mouse passou sobre o elemento	mouse
onmouseup	Botão do mouse solto sobre o elemento	mouse

JavaScript – Evitando envio de dados ao servidor



- Por padrão, ao clicar em um botão do tipo submit de um formulário HTML:
 - Os dados são enviados ao servidor.
 - A página é recarregada.
- Atualmente, grande parte dos sistemas web funciona como SPA (Single Page Application).
- Assim, deve-se evitar o **refresh** com **evento.preventDefault**.
- Dados **reagem** às interações do usuário instantaneamente.
- Acompanhe o professor.

☒ Eventos:

- Implementam o quando.
- Melhoram a experiência do usuário em sistemas web.

☒ Sistemas web geralmente devem se comportar como SPA's (Single Page Applications).

- ❑ CRUD com HTML/CSS/JavaScript.



Aula 3.5. JavaScript – CRUD com HTML/CSS/JavaScript

☐ Sistema passo-a-passo:

- Implementação de CRUD (Create, Retrieve, Update, Delete).

- Acompanhe o professor.

- ☑ Já possível construir o front end de diversos sistemas com a informação vista até aqui.
- ☑ Entretanto, o JavaScript tem mais a oferecer.
- ☑ O JavaScript moderno melhora a experiência do desenvolvedor.

- ❑ Capítulo 4 – JavaScript Moderno.