



Fundamentos de Desenvolvimento de Sistemas

Capítulo 1 - HTML

Dev & Data

Prof. Raphael Gomide

Aula 1.1. HTML

HTML

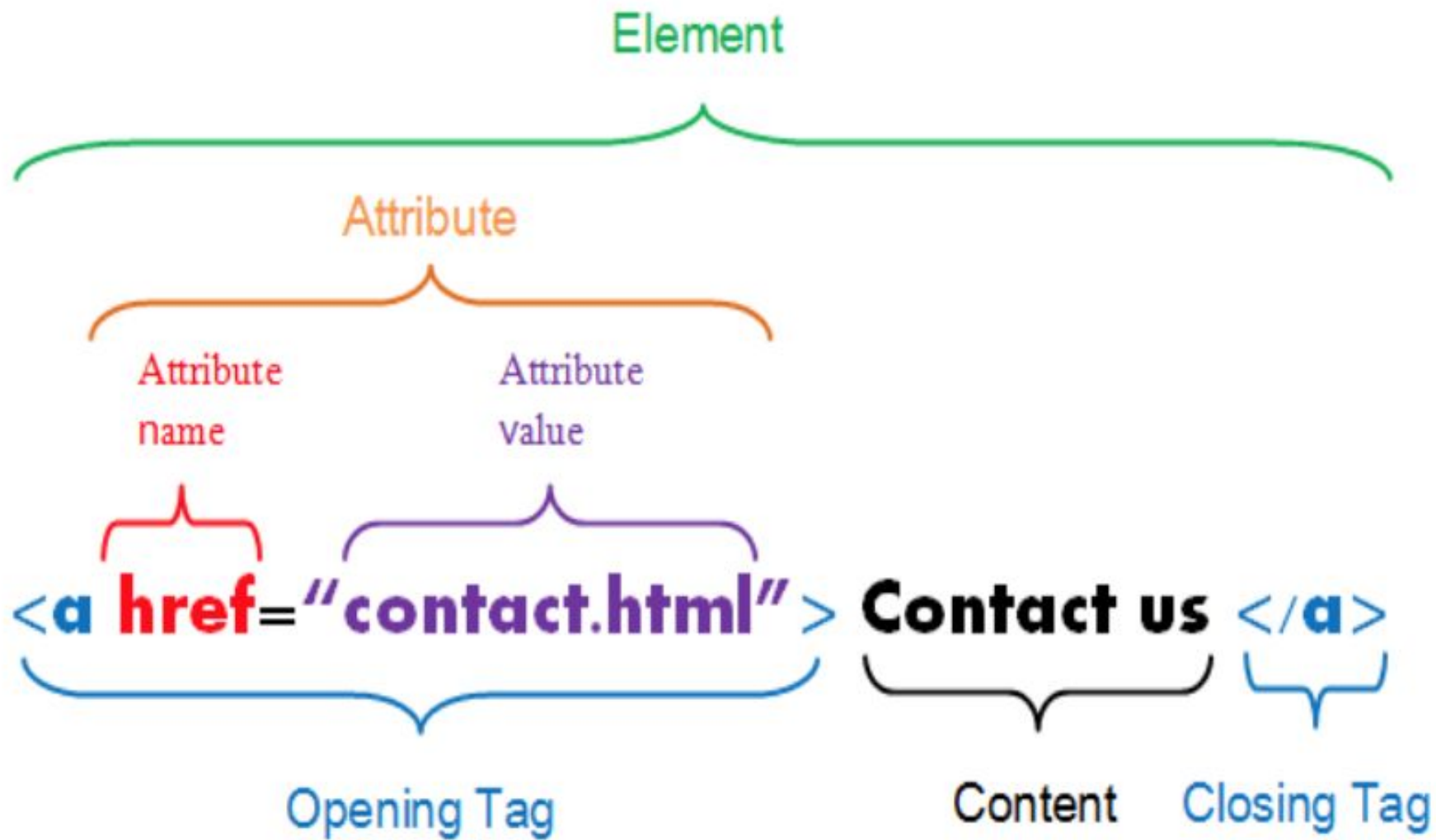
- *Hyper Text Markup Language*
- Linguagem de **marcação** para hipertexto.
- Hipertexto: conteúdo dinâmico (links).
- Encontra-se atualmente na versão 5.
- HTML **não** pode ser considerada uma linguagem de programação!
- Excelente fonte de documentação:
<https://developer.mozilla.org/pt-BR/docs/Web/HTML>



HTML

- Permite a definição **declarativa** de elementos.
- É utilizada para compor **semanticamente** uma página Web.
- É constituída por **elementos**.
- Elementos são delimitados por **tags**.
- A grande maioria das *tags* possuem **conteúdo**.
- Além disso, as *tags* podem possuir **atributos**.
- Os atributos possuem **valores**.





HTML – principais tags

- `<h1>` a `<h6>` □ títulos.
- `<p>` □ parágrafos.
- `<div>` □ divisões blocantes (quebra de linha).
- `` □ divisões não blocantes.
- `` □ definição de imagens.
- `<table>` `<thead>` `<tbody>` `<tr>` `<td>` □ definição de tabelas, linhas e colunas.
- `` `` e `` □ listas e itens de lista.
- `` e `` □ ênfase no texto.
- `<a>` □ âncoras (links).
- `<input>` □ entrada de texto.
- `<button>` □ botão.
- `` □ imagens.

Conclusão

- HTML é uma linguagem de marcação.
- Utilizada para a definição de conteúdo na web.
- Foco em semântica.
- Composta de elementos.
- Elementos são delimitados por *tags*.
- Elementos com conteúdo precisam de *tag* de fechamento.
- *Tags* podem possuir atributos.
- Atributos podem possuir valores.

Próxima aula

Práticas com HTML.



Fundamentos de Desenvolvimento de Sistemas

Capítulo 2 - CSS

Dev & Data

Prof. Raphael Gomide

Aula 2.1. CSS

CSS

- *Cascading Style Sheets*.
- Folhas de estilo em cascata.
- Tecnologia para o *layout* das aplicações web.
- Foco no conteúdo **visual**.
- Utilizam os **atributos class** e **id**.
- Excelente fonte de documentação:
<https://developer.mozilla.org/pt-BR/docs/Web/CSS>



CSS

- Seletores de classe são precedidos pelo símbolo de pontuação "."
- Seletores de id são precedidos pelo símbolo de *hashtag* "#"



```
h1 selector  
{  
    padding: 10px; declaration  
    margin: 0 auto;  
    color: blue;  
}  
property value
```

CSS – principais propriedades

- margin ☐ margem externa.
- padding ☐ margem interna.
- border ☐ borda.
- display ☐ posicionamento de elementos.
- font-size ☐ tamanho de fonte.
- font-family ☐ tipo de fonte.
- background-color ☐ cor de fundo.
- color ☐ cor do texto.
- cursor ☐ tipo de cursor do mouse.
- list-style-type ☐ tipo de estilo de listas .
- width ☐ largura.
- height ☐ altura.

Conclusão

- CSS é uma linguagem de estilização.
- Utilizada para a definição de estilos e layout na web.
- Foco em aspecto visual.
- Composta de seletores, propriedades e valores.

Próxima aula

Práticas com CSS.



Fundamentos de Desenvolvimento de Sistemas

Capítulo 3 - DOM

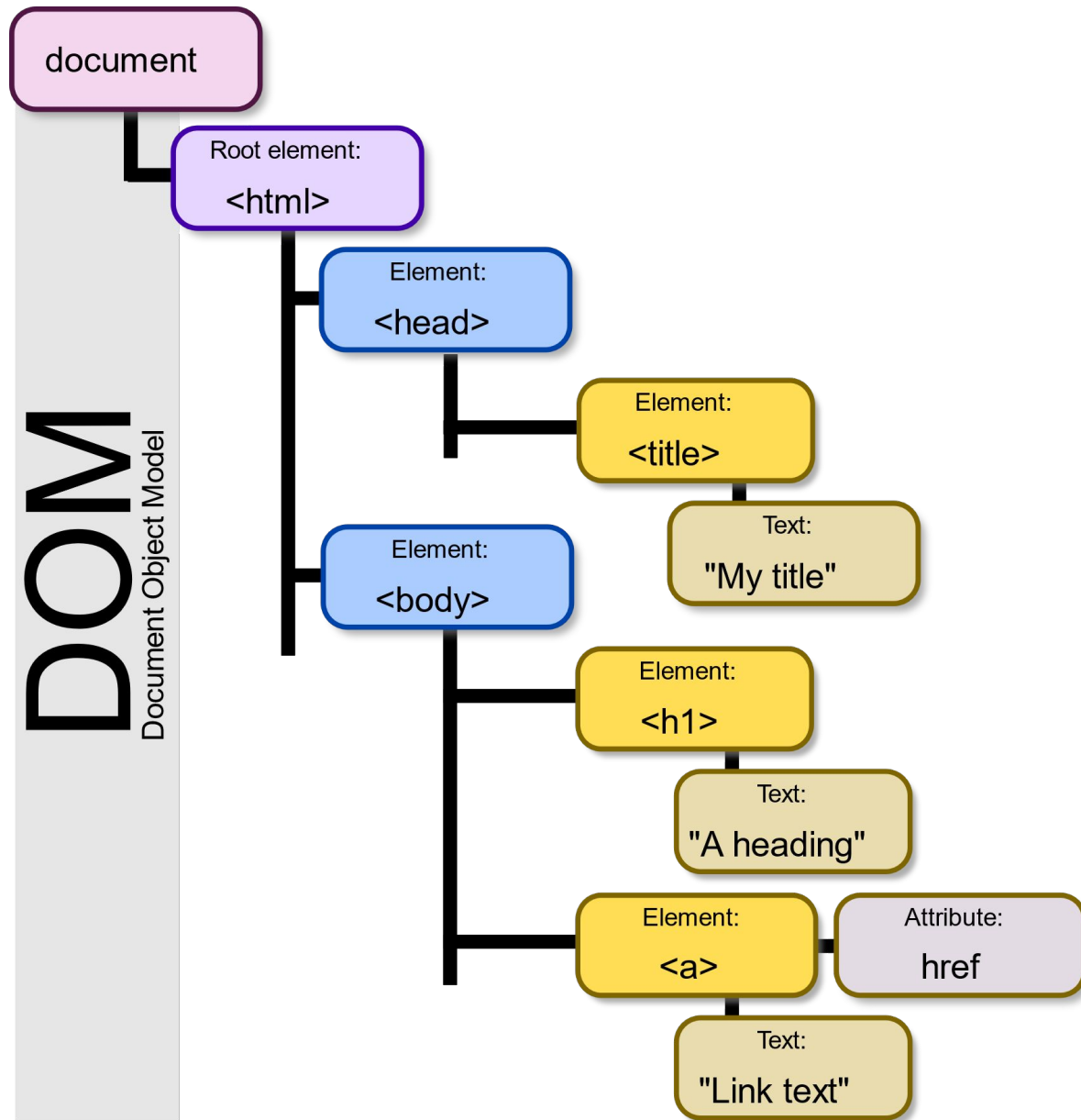
Dev & Data

Prof. Raphael Gomide

Aula 3.1. DOM

DOM

- *Document Object Model.*
- Modelo de Objeto de Documento.
- Estrutura hierárquica de conteúdo HTML.
- É possível manipular o DOM e CSS com JavaScript.
- Excelente fonte de documentação:
https://developer.mozilla.org/pt-BR/docs/Web/API/Document_Object_Model



JavaScript – principais comandos para manipular o DOM

- `document.querySelector` ☐ seletor de um único elemento (tag, class, id).
- `document.qurerySelectorAll` ☐ seletor de *n* elementos.
- `document.createElement` ☐ criação de elementos.
- `appendChild` ☐ inserção de filhos.
- `addEventListener` ☐ inclusão de um “escutador” de evento (**quando**).
- `classList.add` ☐ inclusão de classe CSS.
- `classList.remove` ☐ exclusão de classe CSS.

Conclusão

- É possível manipular o DOM e CSS com JavaScript.
- Isso permite que os apps web sejam mais dinâmicos e interativos.

Próxima aula

Práticas com manipulação do DOM utilizando HTML, CSS e JavaScript.



Fundamentos de Desenvolvimento de Sistemas

Capítulo 4 - Introdução ao LocalStorage

Dev & Data

Prof. Raphael Gomide

Aula 4.1. Introdução ao LocalStorage

LocalStorage

- Funcionalidade para armazenamento de dados nos navegadores.
- Simples e, portanto, limitada.
- Armazena conteúdo no formato de *string*.
- Os dados podem ser excluídos pelo navegador ou usuário.
- Excelente fonte de documentação:
<https://developer.mozilla.org/pt-BR/docs/Web/API/Window/localStorage>

Principais comandos relacionados ao LocalStorage

- `window.localStorage.getItem(key)` □ recupera dados relacionados à *key*
- `window.localStorage.setItem(key)` □ define dados relacionados à *key*.
- `JSON.parse(string)` □ Transforma *string* em JSON. Muito usado com **`getItem`**.
- `JSON.stringify(objeto)` □ Transforma JSON em *string*. Muito usado com **`setItem`**.

Conclusão

- **LocalStorage** é uma das possibilidades de armazenamento de dados em aplicações web.
- É simples e também limitado.
- Recomendado para dados **não** sensíveis e também cuja perda não implica em problemas com a aplicação.

Próxima aula

Prática utilizando LocalStorage.



Fundamentos de Desenvolvimento de Sistemas

Capítulo 5 - Introdução ao Netlify Drop

Dev & Data

Prof. Raphael Gomide

Aula 5.1. Introdução ao Netlify Drop

Netlify Drop netlify

- Uma solução simples para deploy (implantação) de aplicações web.
- Acesso: <https://app.netlify.com/drop>
- Basta arrastar a pasta com os arquivos HTML, CSS e JavaScript.
- Será criada uma URL aleatória de acesso à aplicação.
- Com cadastro é possível modificar a URL.

Conclusão

- O **Netlify Drop** é uma das soluções simples e sem custo para implantação (deploy) de aplicações Web.

Próxima aula

Prática utilizando o Netlify Drop.