



# Fundamentos em Desenvolvimento Front End

## Capítulo 1. Ecossistema

Prof. Raphael Gomide

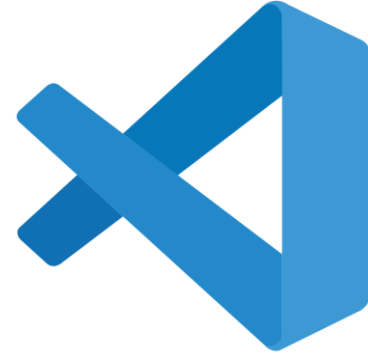


## **Aula 1.1. Introdução ao Visual Studio Code**

## ☐ Visual Studio Code:

- Instalação.
- Extensões.
- Principais configurações.
- Dicas de utilização.

- <https://code.visualstudio.com/>
- Principais características:
  - Principal editor de código utilizado nos dias atuais.
  - Excelente suporte ao git.
  - Terminal de comandos integrado.
- Acompanhe o professor:
  - Instalação da ferramenta.
  - Instalação de extensões.
  - Configuração essencial.
  - Dicas de utilização.



## ☑ Visual Studio Code:

- Mais conhecido como VSCode.
- Principal editor de código JavaScript nos dias atuais.
- Multiplataforma.
- Excelente suporte ao git.
- Excelente conjunto de ferramentas e extensões.
- Suporte ao Emmet.
- Terminal integrado.

# Próxima aula

- ☐ Introdução ao Node.js.



## **Aula 1.2. Introdução ao Node.js**

## ☐ Node.js.

- Introdução.
- Instalação.
- Testes de execução.



- <https://nodejs.org/>
- Ecosystema de desenvolvimento JavaScript.
- Permite, por exemplo:
  - Criação de scripts (CLI – Command Line Interface).
  - Manipulação de arquivos e pastas.
  - Criação de servidores web e API's (Back End).
  - Interação com Bancos de Dados.
  - Instalação de pacotes e bibliotecas através do NPM (Node Package Manager).
- Acompanhe o professor:
  - Instalação da ferramenta.
  - Testes de execução.



## ☑ Node.js:

- Ecossistema para desenvolvimento e execução de código JavaScript fora da web (computadores pessoais, servidores, etc.).
- Utilizado como CLI de diversas ferramentas (Angular, React, Vue, etc.).
- Utilizado para a criação de servidores web.

- ❑ Introdução ao NPM e à biblioteca live-server.



## **Aula 1.3. Introdução ao NPM e à biblioteca live-server**

## ❑ NPM:

- Introdução.
- Utilização básica.

## ❑ Biblioteca live-server:

- Introdução.
- Instalação.
- Testes.



- Node Package Manager.
- Gerenciador de pacotes do Node.js.
- Possui também um site que é, na verdade, um repositório de pacotes.
- <https://www.npmjs.com/>
- Ferramenta de linha de comando.
- Pré-requisito: Node.js instalado e configurado.
- Instalação **padrão** de pacotes:
  - `npm install nome_do_pacote`
- Instalação **global** de pacotes:
  - `npm install -g nome_do_pacote`

## Live-server

A simple development http server with live reload capability.

- <http://tapiov.net/live-server/>
- Biblioteca para o Node.js, instalável via NPM.
  - Comando: `npm install -g live-server`
- Cria um servidor web simples rapidamente.
- Muito útil para desenvolvimento.
- Auto refresh.
- Acompanhe o professor:
  - Instalação da ferramenta.
  - Testes.

## ☒ NPM:

- Grande repositório de bibliotecas construídas com Node.js.
- Comando para manipulação de pacotes.

## ☒ Live-server:

- Pacote do Node.js para a criação de servidor web para desenvolvimento.



# Próxima aula

- ☐ Noções de HTML.



## **Aula 1.4. Noções de HTML**

## □ HTML:

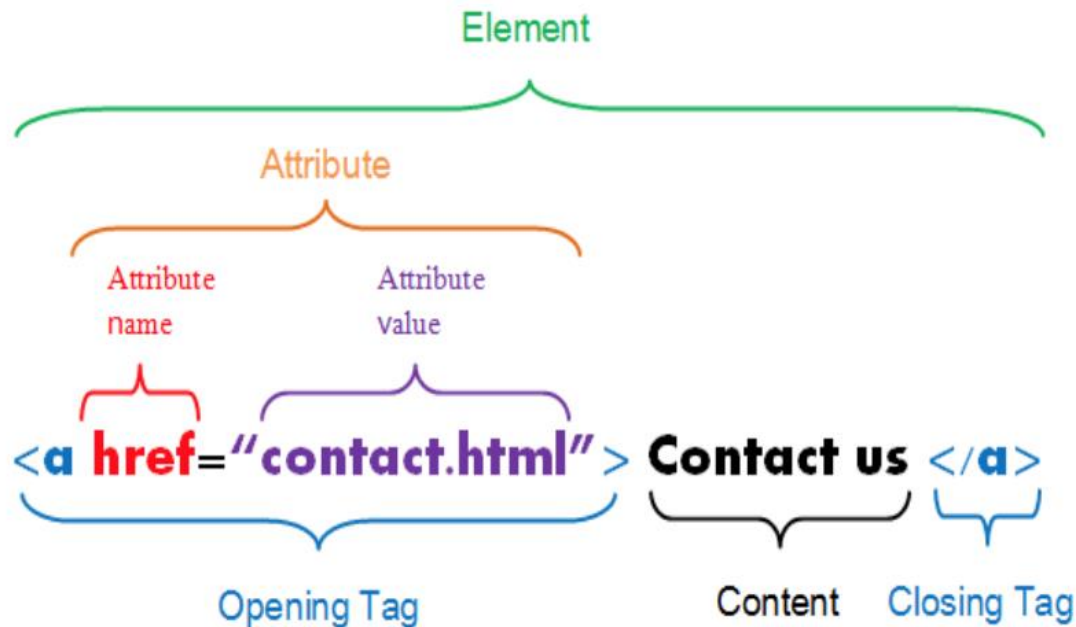
- Introdução.
- Elementos x tags x atributos x valores x conteúdo.
- Principais tags.
- Caminho absoluto x caminho relativo.
- Demonstração de criação de conteúdo HTML.

HTML



- Hyper Text Markup Language.
- **Não** pode ser considerada uma **linguagem de programação**.
- É, na verdade, uma linguagem de **marcação**.
- Utilizada de forma **declarativa** para a estruturação de conteúdo na web.
- Define **elementos**, que são delimitados por **tags**.
- **Tags** podem possuir **atributos** e **conteúdo**.
- **Tags** com conteúdo devem ser encerradas.
- **Atributos** possuem **valores**.
- **Muita** importância **semântica**.
- **Pouca** importância **visual**.
- Excelente fonte de documentação - <https://developer.mozilla.org/pt-BR/docs/Web/HTML>

# Estrutura de um elemento HTML



Estrutura de um elemento HTML – [Fonte](#)

# Principais tags HTML para marcação

- `<p>` → parágrafos.
- `<h1>` a `<h6>` → títulos.
- `<span>` → trechos a serem destacados.
- `<div>` → divisões da página.
- `<img>` → definição de imagens.
- `<table>` `<tr>` `<td>` → definição de tabelas, linhas e colunas.
- `<ul>` `<li>` → listas e itens de lista.
- `<strong>` e `<em>` → ênfase no texto.
- `<a>` → âncoras (links).

- Muito utilizados em imagens e links.
- Caminho absoluto:
  - Local **absoluto** do recurso (arquivo, imagem, etc.) **em disco**.
  - Em geral, só funciona no ambiente original do site/página HTML.
  - Portanto, **evite**.
- Caminho relativo:
  - Local **relativo** do recurso (arquivo, imagem, etc.) **em relação aos demais arquivos do projeto**.
  - Pasta local → `'./'`
  - Pasta pai/mãe → `'../'`
  - Utilize sempre caminhos relativos.
  - Assim, as referências ao seu projeto funcionarão **em qualquer ambiente**.

- Acompanhe o professor.
- Serão demonstrados os seguintes conteúdos:
  - Estruturação de uma página HTML.
  - Utilização das principais tags.
  - Diferenciação entre caminho absoluto e caminho relativo.



- ✓ 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

- ☐ Noções de CSS.



## **Aula 1.5. Noções de CSS**

## □ CSS:

- Introdução.
- Sintaxe.
- Elementos x classes x id's.
- CSS Reset.
- Exemplos com:
  - Estilização de texto.
  - Estilização de cores.



- Cascading Style Sheets.
- Permite a estilização do conteúdo HTML.
- Foco no conteúdo visual.
- Utilizada de forma declarativa.
- Permite a alteração de cores, estilos de texto e posicionamento de elementos.
- Pode ser definido no HTML de três formas:
  - Atributo **style**.
  - Tag **style**.
  - Arquivo externo.

```
2  h1 selector
3  { declaration
4      padding: 10px;
5      margin: 0 auto;
6      color: blue;
7  } property value
8
```

Sintaxe do CSS – [Fonte](#)


- **Elemento:**

- Estiliza todos os elementos conforme identificador.
- Representado pelo nome da tag.

```
p {  
  color: blue;  
}
```


- **Classe:**

- Estiliza todos os elementos que possuam a classe (atributo **class**).
- Representado no CSS por um ponto (.).

```
.destaque {  
  color: green;  
}
```

- **Id:**

- Estiliza o elemento que possui o id (atributo **id**).
- Geralmente deve ser feito para um único elemento.
- Representado no CSS por uma hashtag (#).

```
#menu {  
  color: yellow;  
}
```

- Técnica para remover o CSS padrão dos navegadores.
- Auxilia na padronização visual do site.
- Deve ser feita antes de qualquer outra declaração de CSS.
- Há diversas maneiras para se fazer o CSS Reset.
- Uma das técnicas mais utilizadas é a inserção do arquivo reset.css, criado por Eric Meyer, que pode ser obtido [aqui](#).



- Acompanhe o professor.
- Serão demonstrados os seguintes conteúdos:
  - Integração de HTML e CSS.
  - CSS Reset.
  - Estilização de texto.
  - Estilização de cores.

- ☑ CSS é uma linguagem de marcação.
- ☑ Utilizada para a estilização de conteúdo HTML na web.
- ☑ Foco em estilos, cores e posicionamento.
- ☑ É possível aplicar estilização em elementos, classes e id's.
- ☑ É recomendada a utilização de CSS Reset antes de qualquer outra estilização.

- ❑ Capítulo 2 – Introdução ao JavaScript.