

# Conceitos da WEB: Fundamentos e Tecnologias Essenciais



# Tabelas de Caracteres

ASCII & UTF-8



# Introdução

- Os computadores não entendem letras, números ou símbolos diretamente – eles trabalham apenas com números.
- Para que o computador entenda o que escrevemos são necessários padrões de codificação de caracteres.
- Duas das codificações mais importantes são:
  - ASCII
  - UTF-8



# O que é uma Tabela de Caracteres?

- Uma tabela de caracteres ou charset é um conjunto padronizado de regras que mapeia cada caractere a um número.
- Esses números são armazenados como bytes.



# ASCII (American Standard Code for Information Interchange)

Criada em meados de 1960.

Usa 7 bits para representar os caracteres.

Permitindo 128 caracteres diferentes (0 a 127).

[ASCII table - Table of ASCII codes, characters and symbols](#)

Code Point	Caractere
65	A
66	B
97	a
98	b
48	0
49	1
32	(espaço)
10	\n (nova linha)



# UTF-8 (Unicode Transformation Format – 8bit)

- Pode representar todos os caracteres do mundo.
- Inclui acentos, emojis, caracteres chineses, árabes, dentre outros.
- Usa de 1 a 4 bytes por caractere.

Caractere	UTF-8
A	41
ã	C3 A3
€	E2 82 AC
😊	F0 9F 98 8A

[SYMBL \(🌐\\_🌐\) Symbols, Emojis, Characters, Scripts, Alphabets, Hieroglyphs and the entire Unicode](#)



# Windows-1252

Extensão do ASCII

Usa 8 bits.

Suporta caracteres latinos com acentos.

Usado principalmente em sistemas ocidentais.

Charset	Nome	Utilização
CP1252	Windows Latin 1	Ocidental/Europeu
CP1250	Windows Latin 2	Europa Central
CP1251	Windows Cyrillic	Russo, Sérvio, Búlgaro
CP1253	Windows Greek	Grego
CP1254	Windows Turkish	Turco
CP1255	Windows Hebrew	Hebraico
CP1256	Windows Arabic	Árabe
CP1257	Windows Baltic	Báltico
CP1258	Windows Vietnamese	Vietnamita



# HTML

## HyperText Markup Language





# HTML



## O que é?

### Linguagem de Marcação

HTML é uma linguagem de marcação essencial para criar e estruturar o conteúdo da web, permitindo que os desenvolvedores organizem informações.

### Elementos e Tags

Usando elementos e tags, o HTML possibilita a definição de títulos, parágrafos, links e imagens, formando a base de uma página da web.



# Tags HTML

- Tags HTML são comandos ou instruções que dizem ao navegador como exibir o conteúdo.

```
<nome-da-tag> conteúdo </nome-da-tag>
```

```
<!-- Tags de conteúdo -->
```

```
<p>Este é um parágrafo</p>
```

```
<h1>Título principal</h1>
```

```
<strong>Texto em negrito</strong>
```

```
<!-- Tags vazias (Self-closing) -->
```

```
<br>
```



# Exemplo de estrutura

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <h1>Estrutura básica de um documento HTML</h1>
    <p>
      Um documento HTML é composto por diversas tags que definem sua estrutura e
      conteúdo. A seguir, um exemplo básico:
    </p>
  </body>
</html>
```



# CSS

## Cascading Style Sheets



# CSS



## O que é?

### Definição de CSS

CSS é uma linguagem utilizada para a estilização de um documento HTML.

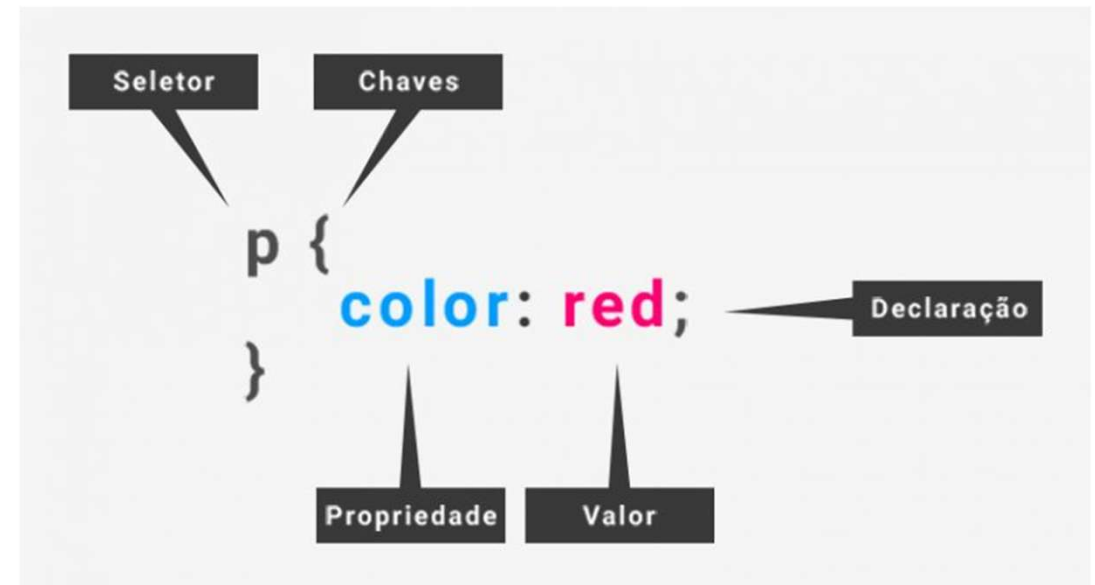
### Exibição em Dispositivos

CSS define como os elementos HTML devem ser apresentados, como cores, fontes, layouts e outros aspectos visuais.



# Como é composto?

- Um arquivo CSS típico contém uma combinação de seletores e declarações organizadas para estilizar uma página da web.



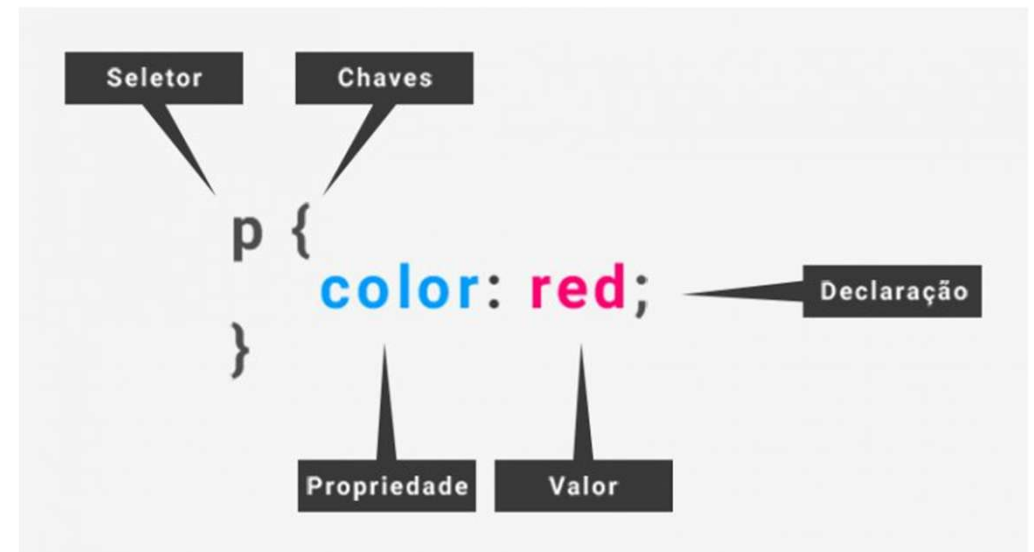
# Como é composto?

## Seletores CSS

É o que define quais elementos do HTML serão estilizados. Os seletores são usados para apontar elementos HTML específicos que você deseja estilizar em seu documento.

## Declarações CSS

As declarações definem as propriedades, como cor, fonte e espaçamento, que serão aplicadas aos elementos selecionados.



# JavaScript





# JavaScript

**Executado no Navegador:** JavaScript é uma linguagem de script que pode ser executada diretamente no navegador, sem a necessidade de compilação.

**Interatividade do Usuário:** A linguagem possibilita a reação de páginas web a ações do usuário, como cliques e movimentos do mouse, tornando a navegação mais dinâmica.



# JavaScript

- **Orientada a Objetos:** Suporta programação orientada a objetos, embora de forma prototípica em vez de baseada em classes.
- **Interoperabilidade:** Pode ser integrado com HTML e CSS para criar páginas web dinâmicas e interativas.



# JavaScript

- **Dinamicamente Tipada:** Não é necessário declarar explicitamente o tipo das variáveis; o tipo é determinado em tempo de execução.
- **Assíncrona e Event-Driven:** Utiliza eventos e callbacks para operações assíncronas, como requisições AJAX. Promises e async/await foram introduzidos para lidar com operações assíncronas de maneira mais eficiente.



# Uso do JavaScript

**JavaScript no Frontend:** No frontend, JavaScript é essencial para criar páginas web interativas, melhorando a experiência do usuário através de elementos dinâmicos.

**JavaScript no Backend:** No backend, JavaScript, especialmente com Node.js, permite a criação eficiente de servidores e APIs para aplicações web.



# Conceito de Backend & Frontend



# Frontend & Backend

- O desenvolvimento do software para a web é geralmente dividido em duas áreas principais: Backend e Frontend
- Ambas desempenham papéis vitais na criação de aplicações web.



# Backend

É a parte da aplicação que lida com a lógica de negócio e o armazenamento de dados.



- **Linguagens de Programação:**

- **JavaScript** (Node.js): Ambiente de execução JavaScript do lado do servidor.
- **Python** (Django, Flask): Linguagem versátil e popular para desenvolvimento web.
- **Ruby** (Ruby on Rails): Framework completo para desenvolvimento de aplicações web.





- **Frameworks é Bibliotecas:**

- Express (Node.js)
- Django
- Flask

ex

 Flask

django



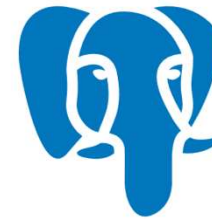
- **Banco de Dados:**

- **Relacionais (SQL):**

- MySQL, PostgreSQL, SQLite

- **Não Relacionais (Documentos):**

- MongoDB, Redis, Cassandra

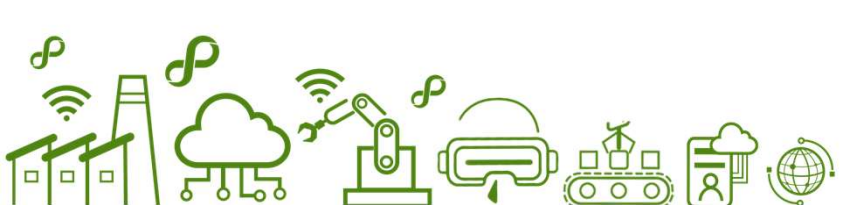


<https://icons8.com/icons>



# Responsabilidade de um Desenvolvedor Backend

- Desenvolver e manter a lógica de negócios da aplicação.
- Gerenciar e interagir com o banco de dados para armazenamento e recuperação de dados.
- Implementar APIs (Application Programing Interfaces) para comunicação entre frontend e backend.



# Responsabilidade de um Desenvolvedor Backend

- Garantir a segurança da aplicação, incluindo autenticação e autorização.
- Monitorar e otimizar a performance do servidor.
- Garantir a escalabilidade de manutenção do servidor.



# Frontend

Frontend é tudo que o usuário vê, ou seja, quando você acessa um site e vê botões, menus, animações, isso é o frontend.



# Tecnologias e Ferramentas Comuns no Frontend

- **Tecnologias Principais:** JavaScript, HTML e CSS são as principais tecnologias usadas para desenvolver o frontend e criar interfaces dinâmicas.

- **HTML (HyperText Markup Language):**

- Estrutura e organiza o conteúdo na web



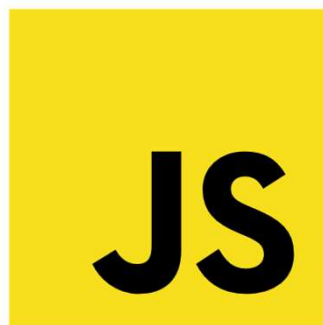
- **CSS (Cascading Style Sheets):**

- Estiliza e organiza a apresentação do conteúdo HTML.



- **JavaScript:**

- Adiciona interatividade e dinamismo às páginas web.
- Permite a manipulação do DOM (Document Object Model)



- **Frameworks e Bibliotecas JavaScript:**

- **React:** Biblioteca para construção de interfaces.
- **Angular:** Framework completo para desenvolvimento de aplicações web.
- **Vue.js:** Framework progressivo para construção de interfaces de usuário.

