


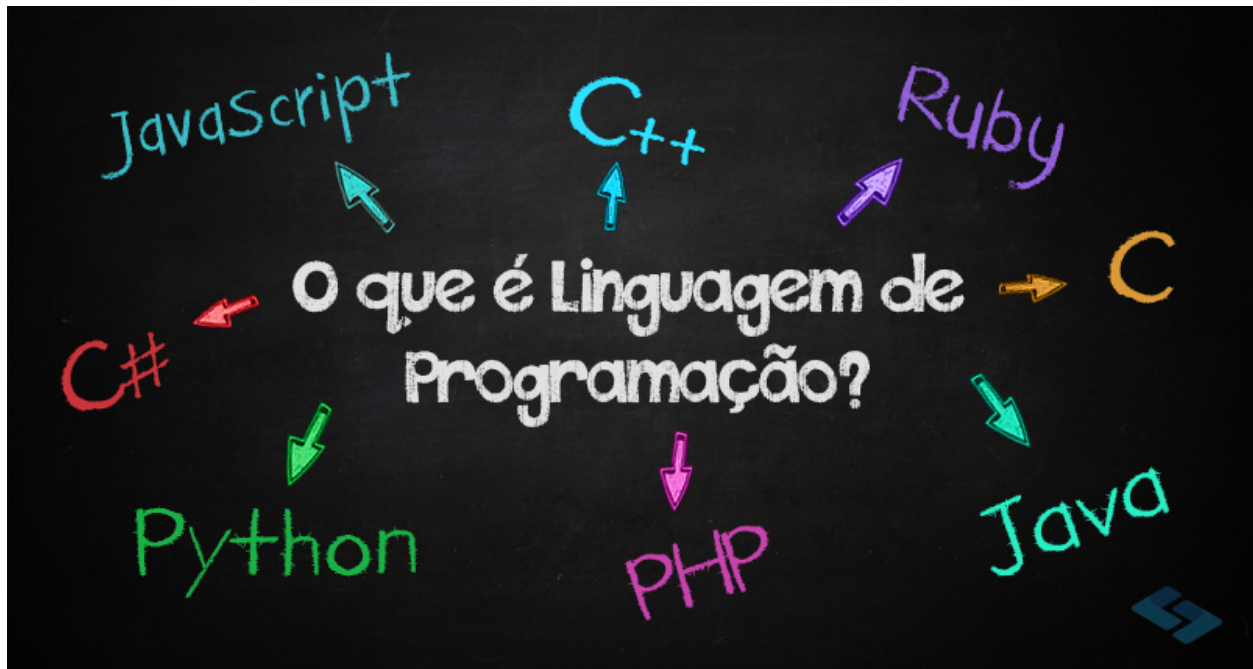


# Como Funciona HTML e CSS

 Nota	Pode parecer estranho ver essas duas tecnologias sendo escritas utilizando artigos femininos, porém, em tradução livre, “ <b>HTML</b> ” e “ <b>CSS</b> ” representam “ <b>Linguagem de Marcação de Hipertexto</b> ” e “ <b>Folha de Estilo em Cascata</b> ”, respectivamente, que são, apesar das controvérsias, termos femininos.
 Created by	 ArthurSobreira

## “Programar em HTML e CSS”

▼ Antes de explicar porque a expressão acima está errada, entenderemos primeiro qual o conceito de **Linguagem de Programação**, e por qual motivo **HTML** e **CSS** não são consideradas parte desse conjunto. Em termos gerais, uma **Linguagem de Programação** é uma **estrutura formal que usa de instruções para permitir ao programador(a) controlar o comportamento físico e lógico de um computador.**



▼ Agora, por que **HTML** e **CSS** não são consideradas, efetivamente, **Linguagens de Programação**? A resposta é simples: **porque nenhuma das duas funciona por meio de instruções**, além de **não possuírem a capacidade de criar funcionalidades dinâmicas**, como todas as linguagens de programação possuem.

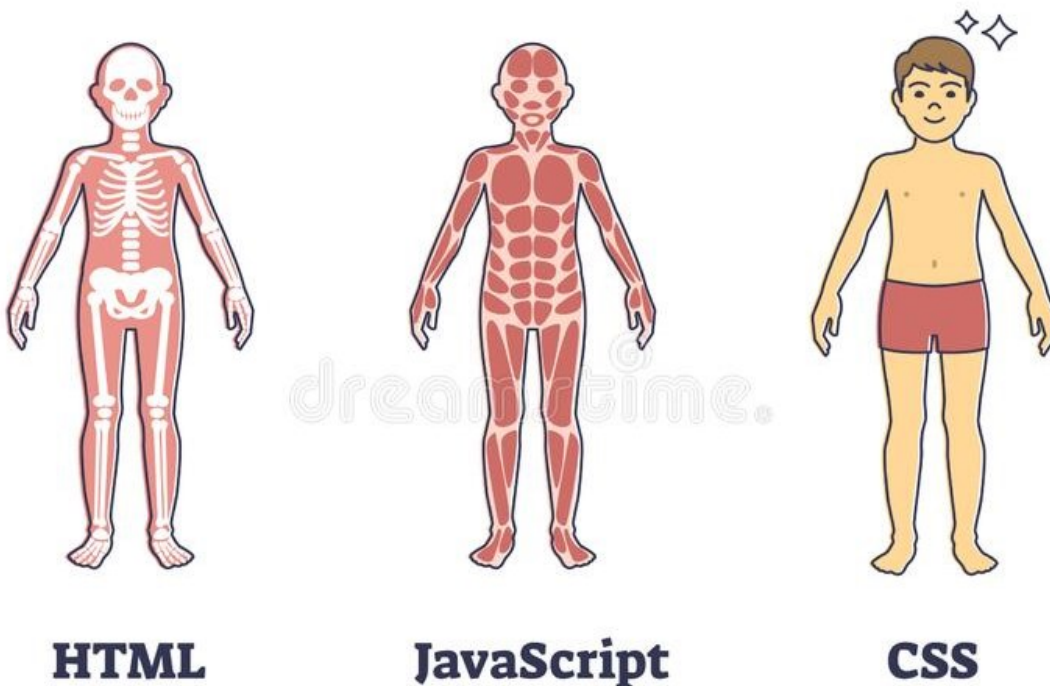
▼ Além disso, o termo “**programação**” envolve **estruturas especializadas** que dependem do uso de **variáveis simples e compostas**, **condições**, **laços**, etc. Uma vez que **HTML** e **CSS** não possuem essas estruturas, as mesmas não são consideradas **Linguagens de Programação**.



Resumindo, **HTML** e **CSS** são consideradas **Linguagens**, mas não **Linguagens de Programação**. Por esse motivo não é correto dizer que você “programa” por meio delas.

## Para que servem HTML, CSS e JS

▼ O **HTML**, **CSS** e **JavaScript** são os **três blocos de construção principais de qualquer página web**. Eles trabalham juntos, mas, como explicaremos a seguir, desempenham **papéis distintos** quando se trata de codificação e **desenvolvimento web**. Para ilustrar a dinâmica entre essas linguagens, utilizaremos um corpo humano como exemplo:



▼ Basicamente, o **HTML** representa o esqueleto do site, sendo utilizado para **estruturar os elementos da página**. Já o **CSS** representa a pele, cabelo, roupas e

tudo relacionado ao **design/estilo** da página. Por último, o **JavaScript** representa a musculatura do site, ou seja, está relacionado ao **movimento/interação** da página. Ainda baseados no exemplo acima, veremos a seguir a funcionalidade de cada uma dessas linguagens, e como a dinâmica entre elas fundamenta o **Desenvolvimento Web**:

## ▼ HTML



- A linguagem **HTML: Hyper Text Markup Language** (Linguagem de Marcação de Hipertexto) compõe a maior parte das páginas web e dos aplicativos online. Essa linguagem define o **significado** e a **estrutura do conteúdo de uma página**. Hipertexto refere-se a **links que conectam páginas da web umas às outras**, seja dentro de um único site ou entre sites.
- A **HTML** funciona baseada em marcações específicas chamadas **tags**. Uma tag é um conjunto de palavras entre sinais de **colchete angular**, conforme representado a seguir:

```
<p>Exemplo de parágrafo</p>
```

- Vale ressaltar que a maioria das tags possuem uma **abertura** e um **fechamento** (pode-se identificar isso pela presença de barra ( / ) no fechamento da tag. Além disso, as tags também podem possuir **atributos** e **valores**, que irão **configurar seu comportamento**:

```
<a href="destino.html">Exemplo de link</a>
```

- Uma mesma tag pode possuir vários parâmetros, cada um com seu respectivo valor. Entretanto, algumas tags não possuem a necessidade de conteúdo interno, e por isso não possuem fechamento, como é o caso das tags `<br>` ou `<img>`.
- Com o surgimento da versão 5 da **HTML**, algumas **tags** simplesmente **deixaram de existir** ou tornaram-se **obsoletas**. Uma tag obsoleta pode até estar funcionando, mas a própria **W3C não recomenda que elas sejam usadas pelos profissionais**.

---

## ▼ CSS



- Como visto anteriormente, as **CSS: Cascading Style Sheets** (**Folhas de Estilo em Cascata**) são usadas para **configurar um resultado visual** dos elementos **HTML**. Basicamente, essa linguagem é responsável por separar a **parte estrutural** da aplicação (que ficará nas mãos do **HTML**) da **parte estética** (que ficará por conta das **CSS**).
- As configurações das **CSS** são realizadas através dos **seletores**, que representam o **elemento que queremos estilizar** (podendo ser uma tag, uma classe, um identificador, etc), a **propriedade será o que iremos alterar** (color, font-size, width, etc) e o **valor de alteração** (red para color, 12pt para font-size, Arial para font-family, etc).

```
h1 {  
    font-family: Arial;  
}
```

- Toda propriedade **deve possuir um valor**, e eles devem ser separados por **dois pontos**, além de que, é obrigatório que se coloque **ponto-e-vírgula** no final de uma declaração para indicar que ela se encerrou.

---

## ▼ JavaScript



- Para completar a grande tríade do desenvolvimento web, temos o **Javascript**. Essa linguagem é usada para **adicionar movimento às páginas web**, além de permitir o **processamento e transformação de dados enviados e recebidos**. Ele permite criar conteúdos que se atualizam de **forma dinâmica e animada**, dando vida às aplicações que antes eram apenas estruturadas com **HTML** de forma estática.
- Sempre que vir uma caixinha de alerta apitando em um site, ou um mecanismo de autocomplete em algum campo, saiba que é o **Javascript** agindo:

```
alert('Hello World');
```

- Vale ressaltar que, assim como no **CSS**, um código **Javascript** pode ser escrito tanto dentro do **HTML** quanto em um **arquivo separado**, sendo importado dentro do atributo desse mesmo elemento quando necessário.

---

## Estrutura Básica de um documento HTML

▼ Ao criar um novo documento **HTML**, devemos sempre escrever a **estrutura básica de um documento desse formato**. Geralmente os editores de código já têm atalhos para iniciar os documentos **HTMLs** com essa estrutura, logo, você não precisa se preocupar em decorá-la.

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
    <title>Documento</title>
  </head>
  <body>
    <h1>Olá Mundo</h1>
  </body>
</html>
```

▼ Veremos a seguir uma simples definição de cada elemento que compõe uma estrutura básica de um documento **HTML**:

▼ **<!DOCTYPE html>** não representa uma tag **HTML**, mas uma instrução para o navegador e outros programas que podem ler seu site, que o código encontrado ali é um código **HTML**.

▼ **<html>** tag que delimita um documento **HTML**, serve como um container que engloba todos os outros elementos do código.

▼ **<head>** contém informações que não são transpostas visivelmente para o usuário/leitor do documento. São dados implícitos, de uso e controle do documento: vinculação com outros arquivos, aplicação de lógica de programação de scripts e metadados.

- **<meta>** define metadados, ou seja, informações sobre dados de um documento **HTML**. São usadas para especificar o conjunto de caracteres, o autor do documento, as configurações da janela de visualização, etc.
- **<title>** tag que define o título do documento. O título deve ser composto apenas por texto e é mostrado na barra de título do navegador ou na aba da página.

▼ **<body>** trata-se do documento em si, ou seja, a informação legível para o usuário do documento. É todo e qualquer texto que se deseja apresentar,



assim como toda e qualquer forma de mídia de saída (imagens, sons, conteúdo multimídia, etc).

- **<h1>** tag utilizada para definir cabeçalhos em documentos **HTML**, deve representar o título/assunto principal de toda a página.



Como dito anteriormente, boa parte dos editores de **código/IDEs** possuem atalhos para criação dessas estruturas base **HTML**, de toda forma, entender a função de cada **tag** é de suma importância a qualquer desenvolvedor web.