



# Apostila de HTML e CSS para Ciência da Computação

## Parte 1: HTML

### Mecânica de um Servidor Web

Um servidor web é um sistema que hospeda sites e disponibiliza-os para serem visualizados na internet. Quando um usuário solicita um site, o servidor web processa essa solicitação e envia os arquivos necessários para o navegador do usuário.

Para entender melhor o que um servidor web faz, vamos considerar um exemplo simplificado:

- Imagine que você quer acessar o site `www.exemplo.com` no seu navegador. Você digita o endereço no campo de URL e pressiona Enter.
- O navegador envia uma solicitação ao servidor web que hospeda o site `www.exemplo.com`, usando o protocolo HTTP (Hypertext Transfer Protocol).
- O servidor web recebe a solicitação e verifica se o site existe e está disponível. Se sim, ele procura pelo arquivo HTML principal do site, que normalmente é chamado de `index.html`.
- O servidor web envia o arquivo `index.html` para o navegador, junto com outros arquivos associados, como imagens, folhas de estilo ou scripts.
- O navegador recebe os arquivos e começa a renderizar a página, seguindo as instruções contidas no código HTML, CSS e JavaScript.
- O resultado é a página web que você vê na sua tela, que pode conter texto, imagens, links, botões, vídeos, formulários ou qualquer outro elemento que o desenvolvedor do site tenha incluído.

Este é um processo básico que ocorre toda vez que você visita um site. Existem muitos detalhes técnicos que não foram abordados aqui, mas o objetivo é dar uma visão geral de como um servidor web funciona.



## Como o Navegador Recebe e Interpreta a Página

O navegador recebe os arquivos do site (geralmente HTML, CSS, JavaScript e arquivos de mídia) do servidor web. O navegador interpreta o código HTML para estruturar a página, o CSS para estilizá-la e o JavaScript para adicionar funcionalidades interativas.

O navegador recebe os arquivos do site (geralmente HTML, CSS, JavaScript e arquivos de mídia) do servidor web. O navegador interpreta o código HTML para estruturar a página, o CSS para estilizá-la e o JavaScript para adicionar funcionalidades interativas.

O código HTML consiste em elementos denominados tags, que indicam o tipo e o conteúdo de cada parte da página. Algumas tags são estruturais, ou seja, definem a organização básica da página, como:

Claro! As tags estruturais em HTML são usadas para indicar o tipo de conteúdo que está sendo inserido na página e sua relação com o resto do conteúdo. Aqui estão alguns exemplos:

1. `<html>`: Esta é a tag raiz de um documento HTML. Todo o código HTML deve estar dentro desta tag.
2. `<head>`: Esta tag contém metadados sobre o documento HTML, como o título da página e links para scripts e folhas de estilo CSS.
3. `<body>`: Esta tag contém o conteúdo principal que é visível para os usuários quando eles visitam a página.
4. `<header>`, `<main>`, `<footer>`: Essas tags são usadas para definir a estrutura básica de uma página web.
5. `<article>`, `<section>`, `<aside>`, `<nav>`: Essas tags são usadas para definir diferentes partes de uma página web.

Essas são apenas algumas das muitas tags estruturais disponíveis em HTML. Elas ajudam a tornar o código mais legível e também são importantes para a acessibilidade e SEO.



As tags semânticas em HTML são usadas para descrever o tipo de conteúdo que está sendo inserido na página, proporcionando um significado claro tanto para os navegadores quanto para os desenvolvedores. Aqui estão alguns exemplos:

1. `<header>`: Define o cabeçalho de um documento ou seção.
2. `<nav>`: Define um conjunto de links de navegação.
3. `<main>`: Especifica o conteúdo principal de um documento.
4. `<article>`: Define um conteúdo independente e autossuficiente, como um post de blog.
5. `<section>`: Define uma seção em um documento.
6. `<aside>`: Define conteúdo à parte do conteúdo ao qual está próximo.
7. `<footer>`: Define o rodapé de um documento ou seção.
8. `<figure>` e `<figcaption>`: Usados juntos para marcar conteúdo relacionado a uma imagem ou ilustração e sua legenda.

Essas tags ajudam a tornar o código HTML mais legível e acessível, além de serem úteis para SEO (Search Engine Optimization).

As tags estruturais e semânticas ajudam o navegador a apresentar a página de forma adequada e também melhoram a acessibilidade e a otimização para motores de busca.

### Resumo de Tags Estruturais Importantes:

- `<!DOCTYPE html>`: Define o documento como HTML5.
- `<html>`: Elemento raiz de uma página web.
- `<head>`: Contém metadados e links para folhas de estilo e scripts.
- `<body>`: Contém o conteúdo visível da página web.
- `<div>`: Define uma divisão ou seção.
- `<p>`: Define um parágrafo.



## Tutorial: Criando uma Página Vazia

1. Abra o VS Code e crie um novo arquivo chamado `index.html`.
2. Comece com a estrutura básica de uma página HTML:

```
<!DOCTYPE html>
<html>
<head>
  <title>Minha Primeira Página</title>
</head>
<body>

</body>
</html>
```

3. Salve o arquivo e abra-o em um navegador. Você verá uma página em branco com o título “Minha Primeira Página”.



## Parte 2: CSS

### O que é CSS

CSS (Cascading Style Sheets) é uma linguagem usada para descrever a aparência de um documento escrito em HTML. Ele permite que você controle coisas como cores, fontes, layout e muito mais.

### Principais Componentes e Classes

- **Seletores:** Permitem que você selecione elementos específicos para estilizar.
- **Propriedades:** São as características que você deseja alterar, como cor, tamanho da fonte, etc.
- **Valores:** São os valores que você deseja aplicar às propriedades.
- **Classes:** Permitem que você defina estilos que podem ser reutilizados em vários elementos.

### CSS Inline ou em Arquivo Separado

CSS pode ser aplicado diretamente nos elementos HTML (inline) ou colocado em um arquivo separado. Geralmente, é melhor prática colocar o CSS em um arquivo separado, pois torna o código mais organizado e fácil de manter.

### Tutorial: Estilizando a Página

1. Crie um novo arquivo chamado `styles.css`.
2. Adicione algum CSS para estilizar o corpo da página:

```
body {  
  background-color: lightblue;  
  font-family: Arial, sans-serif;  
}
```

3. No arquivo `index.html`, adicione um link para a folha de estilo no elemento `<head>`:

```
<head>  
  <title>Minha Primeira Página</title>  
  <link rel="stylesheet" href="styles.css">  
</head>
```



4. Salve ambos os arquivos e recarregue a página no navegador. Você verá que a página agora tem um fundo azul claro e a fonte foi alterada para Arial.

## Expandindo o código

### Agora tente o seguinte:

1. **Adicione mais conteúdo à página:** Adicione mais elementos à sua página, como cabeçalhos (<h1>, <h2>, etc.), links (<a>), imagens (<img>), listas (<ul>, <ol>, <li>) e tabelas (<table>, <tr>, <td>). Veja como cada um deles afeta a estrutura da página.
2. **Estilize os novos elementos:** Use CSS para alterar a aparência dos novos elementos que você adicionou. Experimente diferentes propriedades, como `color`, `font-size`, `background-color`, `border`, `margin`, `padding`, etc.
3. **Crie classes CSS:** Crie algumas classes CSS e aplique-as a diferentes elementos na página. Veja como você pode reutilizar estilos e manter seu CSS DRY (Don't Repeat Yourself).
4. **Experimente Layouts CSS:** Experimente diferentes técnicas de layout CSS, como Flexbox ou Grid. Tente criar um layout de várias colunas para a sua página.
5. **Adicione interatividade:** Adicione alguma interatividade à sua página usando pseudo-classes CSS como `:hover`, `:active`, `:focus`.
6. **CSS Responsivo:** Use media queries para fazer sua página se adaptar a diferentes tamanhos de tela. Verifique como a página se comporta em um dispositivo móvel versus um desktop.
7. **Refatoração:** Revise seu código HTML e CSS. Existe algo que você poderia melhorar? Talvez você possa simplificar seu CSS ou tornar seu HTML mais semântico.

### Aprofundamento:

Agora, aproveite e comece a fazer o Tutorial em: [HTML Tutorial \(w3schools.com\)](https://www.w3schools.com/html/html_tutorial.asp)

Lembre-se, a melhor maneira de aprender é fazendo. Então, mãos à obra e divirta-se codificando!