**PARA COMEÇAR**

Este material aborda conceitos relacionados à implementação de interface gráfica (Front-End) de sistemas online (sites).

No decorrer de seus estudos, esperamos que você desenvolva as seguintes capacidades:

* **Utilizar técnicas de interação e codificação, considerando particularidades e funcionalidades da linguagem**.
* **Utilizar linguagem para manipulação e validação de dados na interface**.
* **Aplicar linguagens de programação no desenvolvimento de interface, seguindo os requisitos do projeto**.

Para desenvolver tais capacidades, você deverá estudar os seguintes temas:

* Linguagem de marcação HTML.
* Linguagem de estilos CSS.
* Linguagem de programação JavaScript

Esse estudo será necessário para que você resolva a situação-problema a seguir. Então, avance para conhecê-la.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

**SITUAÇÃO-PROBLEMA**

Você assumiu uma demanda como programador para o desenvolvimento do site de uma loja de eletrônicos que irá vender jogos e já realizou as seguintes ações:

* levantou os requisitos do projeto, analisando as necessidades do cliente a partir de um questionário específico;
* relacionou esses requisitos com os princípios de User Experience (UX) e User Interface (UI);
* implementou e apresentou ao cliente o protótipo da interface gráfica.

Agora, está na hora de partir para a parte prática e iniciar o desenvolvimento do projeto. Para atender as expectativas do cliente, você deverá publicar os conteúdos no site (texto, imagens, vídeos etc), definir a aparência das páginas e criar aplicações e funcionalidades, para deixar o site mais dinâmico, a fim de mostrar mensagens e informações que permitam uma melhor interação com os usuários.

*Clique nos botões a seguir para conhecer os desafios dessa situação-problema:*

Desafio 1

Desenho de um cachorro

Descrição gerada automaticamente com confiança média

Desenvolver o Front-End do site utilizando a linguagem de marcação HTML5.

Desafio 2

Desenho de um cachorro

Descrição gerada automaticamente com confiança média

Desenvolver o Front-End responsivo do site utilizando o framework do CSS3 - Bootstrap;

Desafio 3

Desenho de um cachorro

Descrição gerada automaticamente com confiança média

Aplicar a linguagem de programação JavaScript para implementar eventos no código em HTML que constituirão o site.

Tela de computador com letras brancas

Descrição gerada automaticamente com confiança média

**DESAFIO 1**

Nesta etapa, você deverá resolver o Desafio 1:

* **Desenvolver o Front-End do site utilizando a linguagem de marcação HTML5**.

Para isso, você estudará os seguintes conteúdos:

* **HTML5** – estrutura de documentos HTML, formatação, etiquetas para links, listas numeradas e não numeradas, tabelas, formulários, imagens, áudio e vídeo.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**DESENVOLVIMENTO PRÁTICO**

Após especificar os requisitos a serem atendidos no projeto de um site e construir o protótipo (rafe) de sua interface, está na hora de iniciar o desenvolvimento prático de seu Front-End.

*Clique no player e assista ao vídeo sobre o primeiro passo do Front-End.*

Interface gráfica do usuário, Site

Descrição gerada automaticamente

**SINTAXE DA TAG**

Uma tag é formada por um conjunto de caracteres que representa um elemento. Ela contém instruções que serão lidas e interpretadas pelo navegador ou browser, que irá [renderizá-las](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/aula-1.html) e apresentar a estrutura visual ao usuário.

As tags são escritas sempre dentro dos sinais de Chevron (sinal de "maior que" e "menor que"), ou seja: < >. Ex.: <body>.

**IMPORTANTE**

Algumas tags não necessitam de fechamento, como **<br>**, **<img>** ou **<meta>**, entre outras. Você pode usar essas tags abertas ou fechadas (como no exemplo a seguir): seu código irá funcionar corretamente.

**<br>** **<meta charset="UTF-8"/>** **<img src="" />**

**ESTRUTURA BÁSICA DE DOCUMENTOS HTML**

O HTML possui uma estrutura padrão para que o navegador possa ler as tags e renderizar o código, ou seja, transformar o HTML em uma página da web com um visual mais amigável e agradável para o usuário.

A estrutura de uma linguagem de marcação HTML é realizada por meio do processo de aninhamento. Aninhar significa colocar uma tag dentro de outra.

O HTML5 acrescenta novas tags como section, video, header, footer, main etc. Sua identificação é a tag .

**PENSE NISSO...**

Complicou? Nem tanto...

Pense em um processo de organização utilizando caixas, por exemplo. Podemos colocar uma ou mais caixas dentro de outra, certo? O processo de aninhamento segue esse mesmo princípio. As tags são caixas que armazenam algum conteúdo, que pode incluir imagens, texto, vídeo etc. Às vezes, você precisará colocar caixas dentro de outras, ou seja, você terá uma ou mais tags aninhadas dentro de outra tag.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Nesse exemplo, temos duas tags: **<body>** e **<p>**.  
*Clique nos destaques para saber mais sobre cada uma delas*.

Nesse caso, o elemento é o parágrafo, delimitado pelas tags **<p>** e **</p>**, que está aninhado dentro do corpo do HTML, delimitado pelas tags **<body>** e **</body>**.

Interface gráfica do usuário, Site

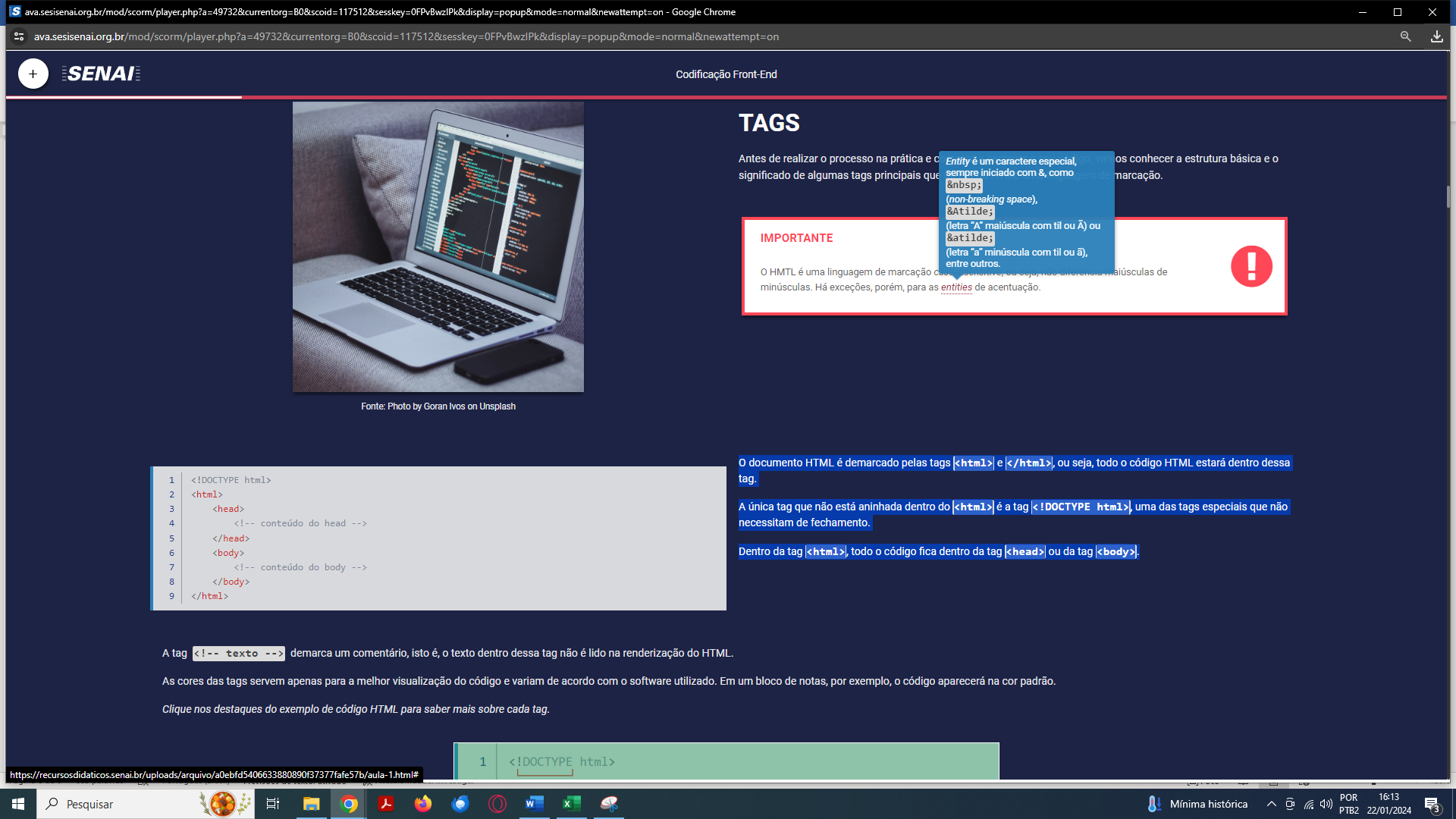
Descrição gerada automaticamente

**TAGS**

Antes de realizar o processo na prática e começar a escrever o seu código, vamos conhecer a estrutura básica e o significado de algumas tags principais que fazem do HTML uma linguagem de marcação.

**IMPORTANTE**

O HMTL é uma linguagem de marcação *case-insensitive*, ou seja, não diferencia maiúsculas de minúsculas. Há exceções, porém, para as [*entities*](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/aula-1.html) de acentuação.



O documento HTML é demarcado pelas tags **<html>** e **</html>**, ou seja, todo o código HTML estará dentro dessa tag.

A única tag que não está aninhada dentro do **<html>** é a tag **<!DOCTYPE html>**, uma das tags especiais que não necessitam de fechamento.

Dentro da tag **<html>**, todo o código fica dentro da tag **<head>** ou da tag **<body>**.

A tag  **<!-- texto -->**  demarca um comentário, isto é, o texto dentro dessa tag não é lido na renderização do HTML.

As cores das tags servem apenas para a melhor visualização do código e variam de acordo com o software utilizado. Em um bloco de notas, por exemplo, o código aparecerá na cor padrão.

*Clique nos destaques do exemplo de código HTML para saber mais sobre cada tag.*

**<!DOCTYPE html>**

A tag **<!DOCTYPE>** declara qual é o tipo de documento para que o navegador possa renderizá-lo corretamente. É uma tag obrigatória e vem sempre na primeira linha do código.

No exemplo, a tag **<!DOCTYPE html>** declara que o documento é do tipo HTML.

**<html>**

Delimita um documento HTML, **abrindo com** **<html>** **na segunda linha e fechando com** **</html>** **na última linha**. Neste exemplo, na tag de abertura, há o [atributo](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/aula-1.html) **lang="pt-br"**, indicando que a linguagem principal do documento é português.

**<head>**

*Head* é o cabeçalho do documento HTML. **Esse cabeçalho não é exibido para o usuário**, ou seja, tudo que estiver aninhado dentro da tag **<head>** é lido pelo navegador, mas não é renderizado visualmente.

Dentro da tag **<head>** estão as tags que fornecem informações, como título da página, fontes, favicons (ícones) e [metadados](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/aula-1.html), como autor do código, propriedades da página, codificação de caracteres e palavras-chaves. Além disso, na tag **<head>** também estão as tags de links para arquivos e instruções para programas externos, como bibliotecas, CSS e informações para o mecanismo de busca do Google.

**<meta>**

**Tag de metadados, aninhada dentro da tag <head>**. No exemplo, a informação **charset="utf-8"** garante a compatibilidade do código com os caracteres de padrão latino-americano. Assim, o navegador vai conseguir ler corretamente caracteres quem contêm cedilha e acentuações.

**<title>**

Define o título mostrado na barra de título do navegador ou na aba da página.

No exemplo, a informação **Meu primeiro site** será mostrada no navegador como título da página.

**<body>**

**Body** é o corpo do documento HTML visível para o usuário. Dentro dessa tag deverá constar todo o conteúdo que será exibido no navegador, como títulos, textos, imagens e links. Os elementos da tag **<body>** serão estilizados por meio da linguagem CSS.

**SAIBA MAIS...**

Há uma quantidade considerável de tags e abordaremos apenas algumas principais. Pesquise outras tags e mantenha-se atualizado, pois a área de tecnologia sempre apresenta novidades.

Uma sugestão de pesquisa é o site W3Schools, que traz tutoriais, referências e exemplos de HTML, CSS, JavaScript entre outras linguagens.

[Clique aqui](javascript:popup('https://www.w3schools.com/python/default.asp',%7bw:625,h:460%7d)) para conhecê-lo (site em inglês).

**ATRIBUTOS DAS TAGS**

Há alguns comportamentos que são padrão para algumas tags. Por exemplo, a fonte, por padrão, é gerada na cor preta, e o parágrafo, sempre alinhado à esquerda.

Os atributos servem para mudar o padrão da tag, personalizando-a, permitindo que ela tenha determinado comportamento ou característica.

**Há atributos que são específicos de uma tag e há muitos atributos que são comuns a várias tags**. Por exemplo, o atributo **charset**, que indica o tipo de caracteres do documento, é específico da tag **meta**. Já o atributo **src**, que indica a origem do elemento, pode ser atribuído a diversas tags.

**A maioria das tags comporta um ou mais atributos, mas há tags que não comportam nenhum**. A tag de imagem pode ter atributos de altura, largura e origem da imagem, entre outros. A tag **<head>** não aceita atributos.

**Por fim, a maioria das tags tem atributos próprios**. A tag de parágrafo, por exemplo, não aceita o atributo **src** de origem, próprio de tags de imagem, vídeo ou áudio.

Confira na tabela a seguir alguns exemplos de atributos.

| **Atributo** | **Característica** | **Exemplo** |
| --- | --- | --- |
| class | Aplicada para formatar vários elementos de uma vez. | **<h1 class="titulo\_post">** |
| id | É aplicada para formatar um único elemento. Não pode começar com número ou hífen, por exemplo: id=“123” ou id=“-987”, nem com símbolos, como, id=“@123” ou id=“&A12b”. | **<h2 id="destaque\_header">** |
| style | Permite incluir estilos de CSS dentro da tag. | **<table style="border: 1px solid black">** |
| lang | Define o idioma principal. | **<html lang="pt-br">** |
| title | Define o título do elemento. | **<img src="img/bolo1.jpg" title="Bolo caseiro de chocolate">** |
| alt | Define um texto alternativo e, por isso, é muito utilizado para descrição de imagens. | **<img src="img/bolo1.jpg" alt="A imagem mostra um bolo de chocolate sobre uma mesa de madeira.">** |
| hidden | Oculta o elemento. | **<p hidden>Trecho de texto em aprovação</p>** |
| align | Define o padrão de alinhamento, como: *right* (direita), *center* (centralizado), *left* (esquerda) e *justify* (justificado). | **<h2 align="center">Título centralizado</h2>** |
| width | Determina a largura do elemento. | **<img src="img/bolo1.jpg" width="100%">** |
| height | Define uma altura para o elemento. | **<img src="img/bolo1.jpg" height="300px">** |

Os atributos **hidden**, **align**, **width** e **height** estão caindo em desuso, devido à manipulação deles diretamente no CSS.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**DICA**

Os atributos são constituídos de um nome e um valor e são sempre alocados na tag de abertura, após a identificação do elemento da tag. O valor do atributo está entre o par de aspas duplas. Fique atento às aspas se usar um editor de texto genérico, pois elas podem ser diferentes dependendo do programa e podem inviabilizar o código.

<a href="https://www.w3schools.com/" target="\_blank">https://www.w3schools.com/</a>

<img src="logo.png" alt="HTML 5 Logo" width="100%" height="100"/>

<p id="especial">Lorem ipsum</p>

<!-- a, img e p são tags href, target, src, alt, width, height e id são atributos

os valores estão entre aspas duplas -->

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**CRIANDO SEU PRIMEIRO ARQUIVO HTML**

Neste momento, vamos criar juntos o primeiro arquivo HTML. Há várias maneiras de criar um documento HTML, pois podemos criá-lo em qualquer programa de edição de texto.

Neste tópico, vamos abordar duas maneiras para fazer o mesmo exercício:

**Primeira maneira**

* Assistir ao vídeo com o tutorial.
* Realizar o processo em seu computador.

**Segunda maneira**

* Acompanhar o primeiro passo a passo.
* Realizar o exercício utilizando a ferramenta disponibilizada.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**METADADOS DA HEAD**

Como já vimos, a **tag** **<head>** é o cabeçalho e contém informações que são acessadas pelo navegador, mas não são visíveis para o usuário.

No exemplo a seguir, temos dentro da tag **<head>**:

* a tag **<meta>** (de metadados), com o atributo charset, cujo valor é "**<UTF-8>**";
* a tag **<title>**, sem atributos, cujo conteúdo é "Meu primeiro site";
* mais 4 tags de metadados, com atributos iguais ([*name*](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/aula-1.html) e [*content*](https://recursosdidaticos.senai.br/uploads/arquivo/a0ebfd5406633880890f37377fafe57b/aula-1.html)) e valores diferentes.

**NAME** – É O ATRIBUTO QUE ESPECIFICA O NOME DO METADADO E O TIPO DE INFORMAÇÃO QUE ELE CONTÉM

**CONTENT** – ESPECIFICA O CONTEÚDO DO METADADO

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**"description"**

Esse metadado fornece aos mecanismos de busca uma breve descrição do conteúdo do site. No exemplo, essa breve descrição está no atributo **content** ("Codificação Front-End, SENAI").

Essa comunicação entre o site e os mecanismos de busca é chamada de *Search Engine Optimization* ou SEO. Uma meta tag de **description** bem feita é uma – mas não a única – ferramenta importante para um SEO eficiente.

**"keywords"**

Significa "palavras-chave", em português. Os navegadores leem essas palavras e exibem o site como sugestão em resultados de pesquisa. Não existe um limite de palavras-chave, porém acrescente palavras que estejam relacionadas à proposta do projeto.

**"author"**

Nome do(s) programador(es) do documento HTML.

**"viewport"**

Esse metadado indica para o navegador a largura para renderização, que deve se adaptar de acordo com o dispositivo de acesso (responsividade).

É imprescindível determinar que o **viewport** mostre o conteúdo adaptado à largura de exibição da tela e não à resolução, pois devemos considerar a rotação de tela.

Nesse exemplo, determinamos os valores da content como:

* **width=device-width**: determina que o layout se adapte automaticamente à largura de tela do dispositivo; e
* **initial-scale=1.0**: define a escala inicial do **viewport**.

Interface gráfica do usuário, Texto, Aplicativo, Email

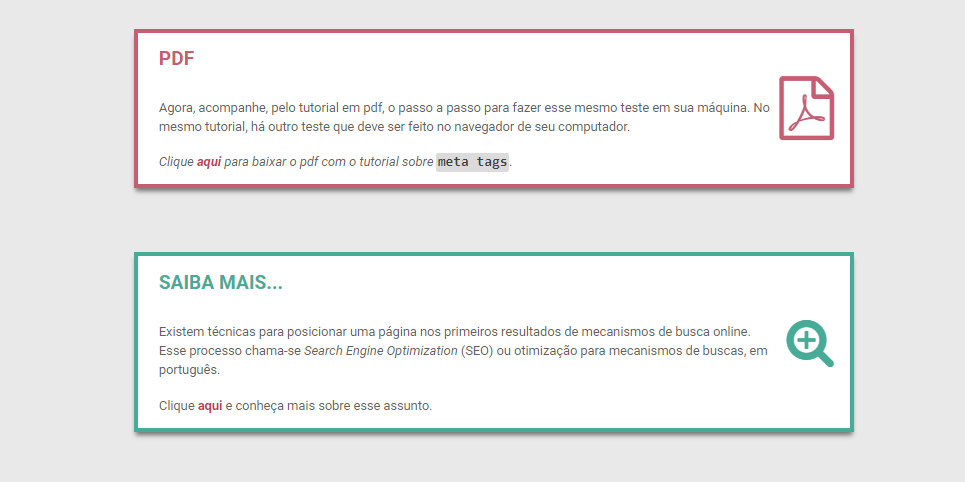
Descrição gerada automaticamente

**UM OLHAR PRÁTICO**

Vamos comprovar na prática que as **meta tags** da **head** não são exibidas para o usuário.

Para isso, use a ferramenta online de renderização. Substitua, na linha 8, o trecho **seu nome** pelo seu nome completo, por exemplo. É importante manter todas as aspas duplas presentes nas tags.

Note que não há nenhuma mudança visível na segunda janela, que mostra a renderização do código.



Interface gráfica do usuário, Texto

Descrição gerada automaticamente com confiança média

**ELEMENTOS DO BODY**

Agora que você conheceu melhor algumas tags do **head**, aprenderá o significado e o uso de alguns elementos e **atributos que fazem parte do body**.

Vamos começar com as seguintes tags:

* **header**
* **nav**
* **div**
* **section**
* **article**

Diagrama

Descrição gerada automaticamente

**Navegação <nav>**

Reúne links para a navegação (menu principal); normalmente está incluída dentro do header.

No exemplo, a tag **nav** está assim:

<nav id="menu">

<a href="index.html">home</a>

<a href="interna01.html">interna01</a>

<a href="interna02.html">interna02</a>

</nav>

**Cabeçalho <header>**

É o cabeçalho do documento; geralmente agrupa um conjunto de links de navegação (menu) e os títulos.

No exemplo, a tag **header** está assim:

<header>

<!-- tag nav aqui -->

<h1>Bem-vindo à página de exemplo</h1>

<h2>Complemente seus estudos</h2>

</header>

**Divisão <div>**

É utilizada para fins de estilo; pode englobar outras tags em seu conteúdo, como textos, imagens ou títulos.

No exemplo, temos mais de uma **div**; as quais contêm uma imagem e um atributo: **style="height:100px; overflow:hidden"**.

<div style="height:100px; overflow:hidden">

<img src="https://pixabay.com/pt/photos/c%C3%B3digo-html-digital-codifica%C3%A7%C3%A3o-1076536/"

alt="código HTML" width="100%">

</div>

**Seção <section>**

Uma seção de um documento, geralmente, contém título; pode ser utilizada para descrever seções/tópicos de um documento.

Há várias maneiras de usar a tag **section**. No exemplo, ela está assim:

<section>

<h3>Resumo</h3>

<p>Aqui, veremos algumas tags importantes do body, como: header, nav, section, article e footer,

além de algumas dicas.</p>

<!-- tag article aninhada aqui -->

</section>

**Artigo <article>**

É utilizada para um conteúdo independente, por exemplo, uma postagem ou artigo. É recomendado identificar utilizando um título.

No exemplo, a tag **article** está assim.

<article>

<h4>Dicas</h4>

<p>As tags abordadas aqui são importantes, mas não são as únicas. Pesquise outras tags e seus

atributos e mantenha-se sempre informado.</p>

</article>

**Rodapé <footer>**

Define o rodapé e geralmente contém informações de autoria (nome e contato do autor) e data de criação do conteúdo.

Observe que, no exemplo da tag **footer**, há uma **div** com o atributo **style="float:left"** e outra com **float:right**. Pesquise essa estilização.

<footer>

<div style="float:left">

<p>autor: SENAI-SP</p>

</div>

<div style="float:right">

<p>março de 2021</p>

</div>

</footer>

Interface gráfica do usuário

Descrição gerada automaticamente

<!doctype html>

<html>

<head>

<meta charset=utf-8>

<title>Meu primeiro site</title>

</head>

<body>

<header>

<nav id="menu">

<a href="index.html">home</a>

<a href="interna01.html">interna01</a>

<a href="interna02.html">interna02</a>

</nav>

<h1>Bem-vindo à página de exemplo</h1>

<h2>Complemente seus estudos</h2>

</header>

<div style="height:100px; overflow:hidden">

<img src="https://cdn.pixabay.com/photo/2015/12/04/14/05/code-1076536\_960\_720.jpg" alt="código HTML" width="100%" />

</div>

<section>

<h3>Resumo</h3>

<p>Aqui, veremos algumas tags importantes do body, como: header, nav, section, article e footer, além de algumas dicas.</p>

<article>

<h4>Dicas</h4>

<p>As tags abordadas aqui são importantes, mas não são as únicas. Pesquise outras tags e seus atributos e mantenha-se sempre informado.</p>

</article>

</section>

<footer>

<div style="float:left">

<p>autor: SENAI-SP</p>

</div>

<div style="float:right">

<p>março de 2021</p>

</div>

</footer>

</body>

</html>

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**FORMATAÇÃO DE TEXTO E TÍTULOS**

Além do uso da tag **<p>** para criação de textos, conheça algumas tags para criação de títulos. Atente-se que existe uma hierarquia, sendo o **<H1>** o mais importante e o **<H6>** o menos importante.

Há também algumas tags utilizadas para criar efeitos na formatação do texto.



**SAIBA MAIS...**

**Tag ênfase x Tag itálico**

A renderização é a mesma, mas o significado semântico é diferente. **<em>** representa ênfase, enquanto **<i>** representa uma mudança no texto, sem ênfase. Por exemplo:

**<em>**Saia agora!**</em>**  
Limpe sua **<em>**própria**</em>** sujeira.  
Neste caso, um software leria o texto com ênfase.

A **<i>**Rainha Mary**</i>** velejou ontem.  
O itálico não indica ênfase, apenas indica que Rainha Mary não é uma pessoa, e sim o nome de um navio.

Texto

Descrição gerada automaticamente com confiança média

**LISTAS**

Lista é uma forma de organizar um conjunto de itens, sejam textos, hiperlinks, ícones, etc. Esse recurso é muito usado para construir menus de navegação, blocos de textos que sejam tópicos ou quaisquer outras informações agrupadas.

Há três tipos de listas: **desordenadas**, **ordenadas** e de **definição**.

As **listas desordenadas** são iniciadas com a tag **<ul>** e cada item dessa lista é marcado com um *bullet* **<li>**.

As **listas ordenadas** são iniciadas com a tag **<ol>** e cada item é marcado com números **<li>**, por padrão. Para marcar os itens com letras, utilize a tag **<ol type="a">** e para marcar com números romanos, **<ol type="I">**.

As **listas de definição** são iniciadas com a tag **<dl>**, seguida de **<dt>** e **<dd>**.

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

<title>Meu primeiro site</title>

</head>

<body>

<h1>Listas</h1>

<ul>Lista desordenada

<li>item 1</li>

<li>item 2</li>

<li>item 3</li>

</ul>

<ol>Lista ordenada (padrão)

<li>item 1</li>

<li>item 2</li>

<li>item 3</li>

</ol>

<ol type="a">Lista ordenada com letra

<li>item 1</li>

<li>item 2</li>

<li>item 3</li>

</ol>

<ol type="I">Lista ordenada com número romano

<li>item 1</li>

<li>item 2</li>

<li>item 3</li>

</ol>

<dl>Lista de definição

<dt>tag head</dt>

<dd>tag title</dd>

<dd>tags meta</dd>

<dt>tag body</dt>

<dd>tag header</dd>

<dd>conteúdo site</dd>

<dd>tag footer</dd>

</dl>

</body>

</html>



**ETIQUETAS PARA LINKS**

Etiquetas para links, também chamadas de âncoras, são criadas para que o usuário, ao realizar uma ação, seja direcionado para uma página interna ou externa específica. Usar **#** entre aspas duplas no href (**href="#"**)  significa que não irá direcionar para nenhum local.

É possível inserir etiquetas em qualquer elemento da página, como textos, títulos, imagens e botões. Para isso, basta utilizar a tag **<a>**, que possui um atributo de **href** no qual será inserido o valor referente ao endereço para onde o usuário será direcionado ao clicar.

Veja um exemplo de código:

<nav id="menu">

<a href="index.html">home</a>

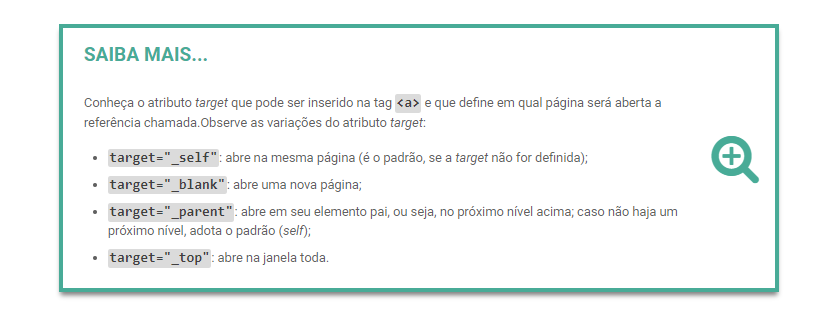
</nav>

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**IMPORTANTE**

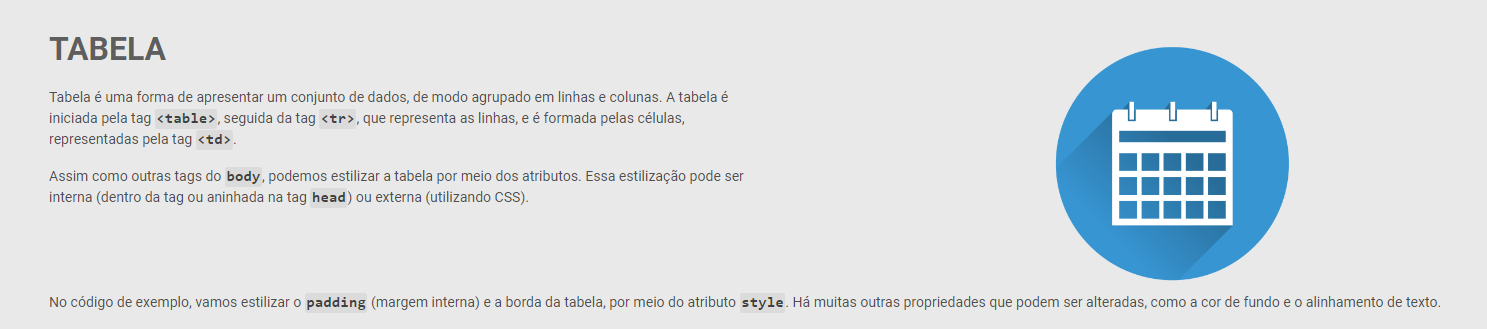
Outra etiqueta de link é a tag **<link>**. Diferente da tag **<a>**, que irá na tag **<body>**, a tag **<link>** irá no cabeçalho da estrutura do código. Nela, é possível adicionar um recurso interno ou externo, como chamar um arquivo CSS. Essa tag é mais utilizada para vincular folhas de estilo.



**SAIBA MAIS...**

Conheça o atributo *target* que pode ser inserido na tag **<a>** e que define em qual página será aberta a referência chamada.Observe as variações do atributo *target*:

* **target="\_self"**: abre na mesma página (é o padrão, se a *target* não for definida);
* **target="\_blank"**: abre uma nova página;
* **target="\_parent"**: abre em seu elemento pai, ou seja, no próximo nível acima; caso não haja um próximo nível, adota o padrão (*self*);
* **target="\_top"**: abre na janela toda.



**TABELA**

Tabela é uma forma de apresentar um conjunto de dados, de modo agrupado em linhas e colunas. A tabela é iniciada pela tag **<table>**, seguida da tag **<tr>**, que representa as linhas, e é formada pelas células, representadas pela tag **<td>**.

Assim como outras tags do **body**, podemos estilizar a tabela por meio dos atributos. Essa estilização pode ser interna (dentro da tag ou aninhada na tag **head**) ou externa (utilizando CSS).

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**FORMULÁRIOS**

Um formulário permite a interação entre o usuário e o sistema; dessa forma, é possível fazer a coleta de dados do público e enviá-los a um servidor para serem processados e salvos em um banco de dados, por exemplo. Um formulário deve conter um **campo de texto (input)**, **checkbox (caixa de seleção)**, **RadioButton** ou **botão**.

Para isso, no layout, é necessário conter os campos com as informações que o desenvolvedor deseja obter, atentando-se para deixar o formulário objetivo, para que o usuário não gaste muito tempo e/ou desista do preenchimento.

Na imagem, temos um exemplo de um formulário para realizar um cadastro, em que se solicita o preenchimento dos campos de **nome completo, e-mail, empresa, telefone, UF, cidade, assunto, categoria de usuário** e **mensagem**. Esses dados são salvos em um banco de dados e, por meio da lógica de uma linguagem de programação, são resgatados e confirmados se existem, para liberar o acesso.

O formulário pode ser utilizado para que o usuário entre em contato com a empresa, em uma enquete, em um cadastro de login, na recuperação de uma senha, na alteração de dados, entre outros.

Você pode iniciar com uma estrutura mais simplificada, apenas abrindo a tag **<form>**. A tag **<label>** é utilizada para representar uma legenda. O **for** é um atributo da tag cujo objetivo é permitir que, quando o usuário clique na palavra nome, selecione o **input** (caixa de texto), pois o valor do **id** é igual ao valor do **for**. Caso o **id** seja diferente do valor do **for** ou vice-versa, quando o usuário clicar no nome, não selecionará o **input**. O **input** é o campo de entrada do usuário.

*Verifique o código na primeira janela e observe sua respectiva renderização*. Observe que algumas das interações necessitam de recursos adicionais para funcionarem corretamente.

<body>

<form>

<label for="name">Nome</label>

<input type="text" id="name">

<p>O atributo type="text" está especificando o tipo de valor que aquele input receberá.</p>

<hr><br>

<input type="button" value="Clique aqui">

<p>Type="button" cria um botão simples e que é atribuído para um evento, no qual o usuário pode clicar. Não é necessário criar esse input dentro de um form. </p>

<hr><br>

<label for="checkbox-example">Exemplo type checkbox</label>

<input type="checkbox" id="checkbox-example">

<p>Type="checkbox" é utilizado para que o usuário escolha mais de uma opção.</p>

<hr><br>

<label for="date-example">Exemplo type date</label>

<input type="date" id="date-example">

<p>Type="date" é utilizado para que o usuário possa escolher uma data.</p>

<hr><br>

<label for="email-example">Exemplo type e-mail</label>

<input type="email" id="email-example">

<p>Type="email" é utilizado para que o usuário possa passar uma entrada cujo valor seja um e-mail. Ao usar o atributo required (estudaremos posteriormente sua funcionalidade), o texto que não tenha @ não é reconhecido como type="email".</p>

<hr><br>

<label for="file-example">Exemplo type file</label>

<input type="file" id="file-example">

<p>Type="file" é utilizado para que o usuário possa fazer upload de um arquivo. </p>

<hr><br>

<label for="number-example">Exemplo type number</label>

<input type="number" id="number-example">

<p>Type="number" é utilizado para que o usuário possa passar uma entrada cujo valor seja um número. </p>

<hr><br>

<label for="password-example">Exemplo type password</label>

<input type="password" id="password-example">

<p>Type="password" é utilizado para que o usuário insira uma entrada cujo valor seja o de uma senha. Os caracteres inseridos aparecem como ocultos. </p>

<hr><br>

<label for="radio-example">Exemplo type radio</label>

<input type="radio" id="radio-example">

<p>Type="radio" são bolinhas em que o usuário irá fazer uma seleção. </p>

<hr><br>

<label for="range-example">Exemplo type range</label>

<input type="range" id="range-example">

<p>Type="range" é uma opção dinâmica, na qual o usuário pode especificar um valor numérico que não deve ser inferior a um determinado valor.</p>

<hr><br>

<label for="search-example">Exemplo type search</label>

<input type="search" id="search-example">

<p>Type="search" é utilizado para que o usuário possa passar uma entrada que permitirá realizar uma pesquisa dentro do site. Para realizar essa pesquisa, será feito um filtro, no qual será criada uma lógica no Back-End e integrada ao Front-End.</p>

<hr><br>

<input type="submit" id="btn-enviar" name="btn-enviar" required>

<p>Type="submit" é utilizado para que o usuário possa enviar uma entrada.</p>

<hr><br>

<label for="tel-example">Exemplo type tel</label>

<input type="tel" id="tel-example">

<p>Type="tel" é utilizado para que o usuário possa passar uma entrada cujo valor seja um número.</p>

</form>

</body>

Texto

Descrição gerada automaticamente

A diferença entre atributos **types** **checkbox** e **radio** é o fato de **radio** apresentar bolinhas para que o usuário selecione apenas uma opção, enquanto o **checkbox** apresenta quadrados para que ele selecione mais de uma opção.

Exemplo:  
Caso tenhamos a seguinte pergunta: "Qual o seu gênero?", e que ela apresente como respostas possíveis: feminino, masculino ou outro, o usuário, que tem apenas um gênero, deverá escolher apenas uma opção, portanto, o **type** escolhido para esse **input** será **radio**.

Caso tenhamos uma pergunta mais aberta, por exemplo: "Quais suas principais habilidades?", cujas respostas sejam: HTML5, CSS3, Wordpress, PHP, JavaScript, Jquery, Java, C#, Photoshop, Illustrator e InDesign; o **type** será **checkbox**, uma vez que o usuário pode ter conhecimento sobre mais de uma opção.

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

**IMAGEM**

A maneira mais comum de inserir imagens é utilizando as tags **figure** e **img**.

A tag **figure**, além de agrupar imagens, pode adicionar ilustrações, diagramas e listagens. A tag **figcaption** define uma legenda para a imagem.

A tag **img** possui o atributo **src** (source), que é a fonte ou o caminho – local ou online – da imagem. O atributo **alt** é uma breve descrição da imagem não mostrada no navegador, mas interpretada pelo browser.

Texto

Descrição gerada automaticamente

**IMPORTANTE**

O tamanho da imagem pode ser definido pelos atributos **width** (largura) e **height** (altura), usando medidas fixas em pixels ou em porcentagem. Então, se o tamanho da imagem for definido em px (pixels), o tamanho da imagem ficará fixo e a imagem terá barras de rolagem, caso seja acessada em dispositivos com telas menores. Usando a porcentagem, o tamanho da imagem se adapta ao tamanho da tela do dispositivo, mantendo a proporção. Por ser mais responsiva, a porcentagem é a medida mais utilizada.

<body>

<figure class="banner-home">

<!-- aqui a tag figure pode agrupar outras imagens nessa class, mas nesse exemplo só há uma imagem -->

<img src="https://cdn.pixabay.com/photo/2017/06/29/10/28/games-2453777\_960\_720.jpg"

alt="banner da home" width="50%" >

<!-- o src é uma imagem online, mas poderia ser um caminho local. Experimente salvar esse código em um arquivo HTML local e chamar uma imagem em seu computador -->

<!-- o atributo alt é utilizado em ferramentas de acessibilidade -->

<!-- a largura está definida em porcentagem.

 Experimente trocar por uma medida em pixels -->

<figcaption> Banner da home</figcaption>

</figure>

</body>

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**Dica**

Você pode também criar um arquivo HTML e verificar a renderização no navegador.

Confira outras dicas:

1. Não utilize imagens encontradas em mecanismos de busca devido aos direitos autorais.
2. Há diversos sites de imagens grátis, como o Pixabay e o FreePik.
3. Salve suas imagens em uma pasta e renomeie-as com nomes curtos.
4. Se a extensão do seu arquivo não estiver aparecendo, clique no arquivo com o botão direito e clique em "Propriedades".
5. Lembre-se de otimizar suas imagens, convertendo a resolução para 72 dpi, para que o site fique mais leve. Isso pode ser realizado com o tinypng.com.

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

**ÁUDIO**

As tags de **<áudio>** e **<vídeo>** do HTML5 foram criadas para acabar com a dependência de plugins de reprodução. Com elas, podemos inserir e configurar atributos para essas tags do mesmo modo que fazemos com a tag **img**. Analise a seguir um exemplo de código.

<audio>

<source src="audio.mp3" type="audio/mpeg">

</audio>

Texto

Descrição gerada automaticamente

**VÍDEO**

Para adicionar um arquivo de vídeo em uma página HTML, utilize a tag **<video>** para incorporar o código.

Além da tag **<video>**, é necessário adicionar o atributo **width** (largura), **height** (altura) e **controls**. Assim como a tag **<audio>**, o atributo **controls** permite ao usuário controlar a reprodução do vídeo.

Dentro da tag **<video>** está aninhada a tag **source**, que determina a fonte do vídeo, que pode ser um caminho local ou uma URL. Veja um exemplo a seguir.

<video width="320" height="240" controls>

<source src="video.mp4">

</video>

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

**FERRAMENTA PARA IMPLEMENTAÇÃO DE CÓDIGO**

Como você percebeu ao usar o bloco de notas, não são necessárias ferramentas sofisticadas para iniciar o desenvolvimento de um código. Para esse curso, porém, vamos utilizar o programa Visual Studio Code, que pode otimizar seu trabalho.

O UI Design aborda a maneira com a qual o usuário irá interagir ou controlar o produto, serviço ou sistema, aplicando um design voltado mais à forma visual.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente