

VEM SER

HTML e CSS

Aula 01 - Introdução

As instrutoras



Cristina Jung
Desenvolvedora Web
cristina.jung@dbcccompany.com.br



Mayra Amaral
Desenvolvedora Web
mayra.amaral@dbcccompany.com.br

Conteúdo da aula

Case Styles

Client-side vs Server-side

Front-End vs Back-End

A tríade HTML, CSS e JavaScript

Editores de código e IDEs

Introdução à HTML

Semântica e estrutura

Listas, formulários e tabelas

Inserindo conteúdo multimídia nas páginas

Introdução conceitual

Introdução prática



Case Styles (ou convenções de nomenclatura)

Case Styles são as boas práticas que devemos seguir ao **nomear variáveis** e demais nomes (classes, IDs, componentes, etc) no nosso código. As principais são:

- **camelCase:** nomeDaVariavel -> primeira palavra começa com letra minúscula e demais palavras começam com a primeira maiúscula. Usada principalmente no **JavaScript e Java**;
- **PascalCase:** NomeDoComponente -> primeira letra de cada palavra maiúscula. Usada principalmente para **classes** (POO) e **nomes de componentes no React**;
- **kebab-case:** nome-do-elemento -> espaço entre palavras substituído pelo hífen (-). Usado principalmente no **CSS**, também para **nome de arquivos** e para **links e URLs**;
- **snake_case:** nome_da_variavel -> espaço entre palavras substituído pelo *underline*/sublinhado (_). Usado principalmente no **Python e C**;

Case Styles (ou convenções de nomenclatura)

Algo comum entre praticamente todas as linguagens é nomear constantes absolutas (não mudam nunca!) no padrão **SCREAMING_SNAKE_CASE**:

Deve ser nomeado assim: NOME_DA_CONSTANTE.

Todas as palavras em maiúsculo e com espaço substituído por *underline*/sublinhado (_).

Exemplo:

```
const VALOR_DE_PI = 3.141592;
```

Para saber mais 🔍

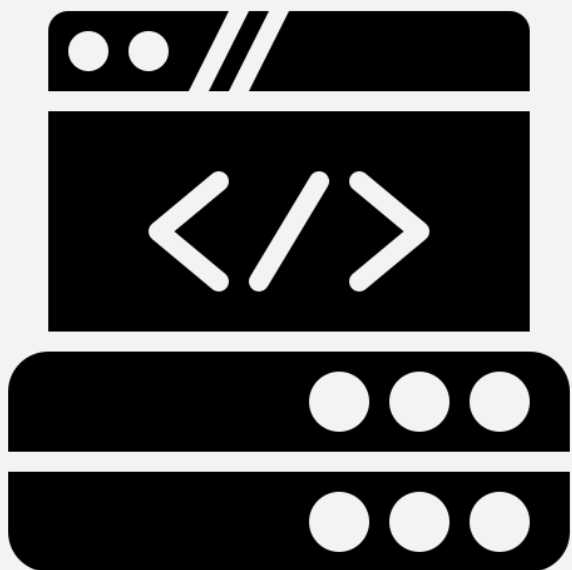
 <https://www.alura.com.br/artigos/convencoes-nomenclatura-camel-pascal-kebab-snake-case>

Percebam que a Alura está usando **kebab-case** na URL





Client-side
vs
Server-side

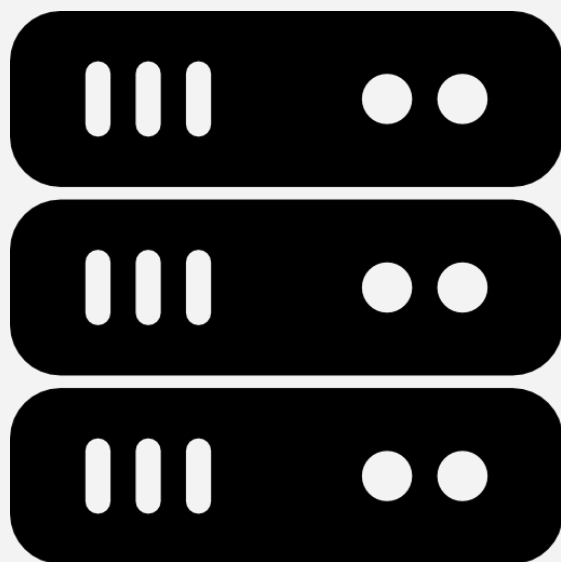


Client-side

Sempre que falamos sobre **client-side**, estamos nos referindo ao que está acessando a nossa aplicação.

Por exemplo, quando acessamos uma página através do Google Chrome, o *client* é o browser.

Em termos técnicos, *client-side* é a parte da nossa aplicação que está sendo executada no computador do usuário.



Server-side

Server-side se refere a parte de processamento realizada em um *web server*.



***Web
server???***

O que é um *web server*?

Um **web server** ou simplesmente servidor, é um **computador dedicado** à execução de aplicações e serviços. Ele é conectado através da internet, por isso recebe o nome de *web server*.



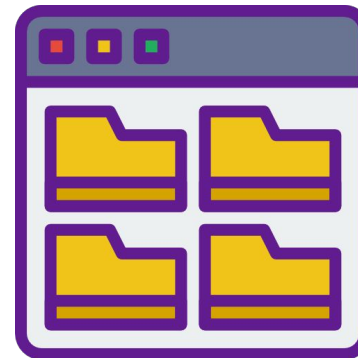
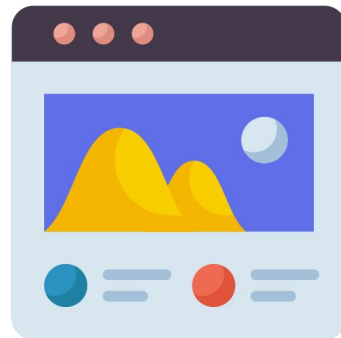
A large, dark blue outline of a speech bubble or document icon, featuring a folded top-right corner and a tail pointing towards the bottom-left.

Webpages vs Websites

Diferença entre webpage e website

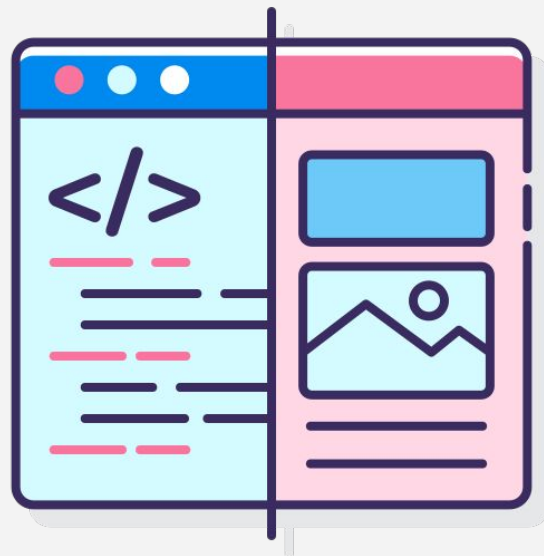
Uma **webpage** é uma página, contém imagens, textos, *links*. Geralmente construída a partir de HTML, CSS e JavaScript.

Um **website** é um **conjunto** de *webpages*.





Front-end
VS
Back-end



Front-end

O desenvolvedor *front-end* é responsável por desenvolver e implementar a interface da aplicação.

A interface **interage diretamente** com o usuário, por esta razão, devemos nos preocupar sempre com a experiência do usuário (UX).



Back-end

O desenvolvedor back-end é responsável por criar, desenvolver e implementar a parte lógica e estrutural da aplicação.

O *front-end* e o *back-end*, atualmente, são “**desacoplados**”. Ou seja, um não depende do outro. A forma como eles se comunicam é através de uma **API** (*Application Programming Interface*).

As linguagens que utilizamos no *front-end*
são: **HTML**, **CSS** e **JavaScript**



HTML *(HyperText Markup Language)*

É uma **linguagem de marcação**, utilizamos para **estruturar** o **conteúdo** da nossa página ou site.

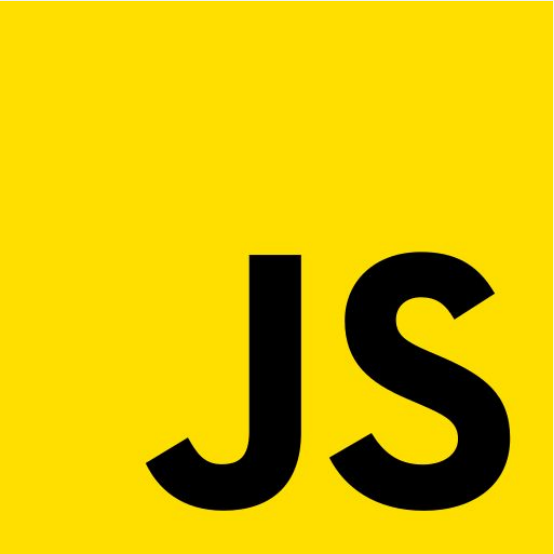
Assim como podemos organizar a estrutura de um documento utilizando a ferramenta Word, podemos fazer isso utilizando o HTML (ou outras linguagens de marcação).

Outros exemplos de linguagens de marcação: Markdown (bastante usada no Github), LaTeX, XML.



CSS *(Cascading Style Sheet)*

É uma **linguagem de estilização**, utilizamos para estilizar o conteúdo que previamente estruturamos utilizando o HTML.



JS

JavaScript

Não confundir com Java. Não tem nada a ver com Java.

É uma **linguagem de programação**, utilizamos para escrever *scripts* (assim são chamados os códigos JS) para as nossas páginas, adicionando lógica a elas e aumentando a **interatividade** que o usuário terá.

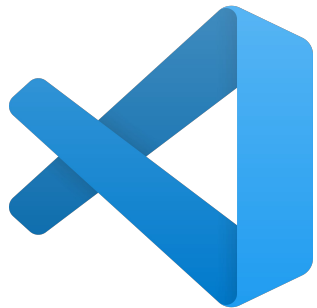
A large, dark blue speech bubble with a white interior, containing the text. The bubble has a thick border and a small tail pointing towards the bottom left.

Editores
de código
e IDEs

Editores de código

Assim como utilizamos o Word ou o Google Docs para documentos, utilizamos editores de código para escrever e editar código. Esses editores têm ferramentas que facilitam e agilizam o processo de desenvolvimento.

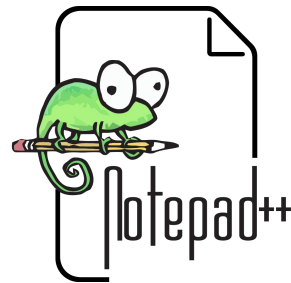
Exemplo de editores:



VS Code



Sublime
Text



IDEs (*Integrated Development Environment*)

Como o próprio nome sugere, são **ambientes** de desenvolvimento. Normalmente uma IDE possui todas as ferramentas de desenvolvimento necessárias para determinado segmento ou linguagem já integradas, por exemplo: compiladores, *debuggers* e um editor de código.

Alguns exemplos de IDE são:



Introdução à HTML

O HTML usa **elementos** para descrever a estrutura das páginas

O que são esses elementos?

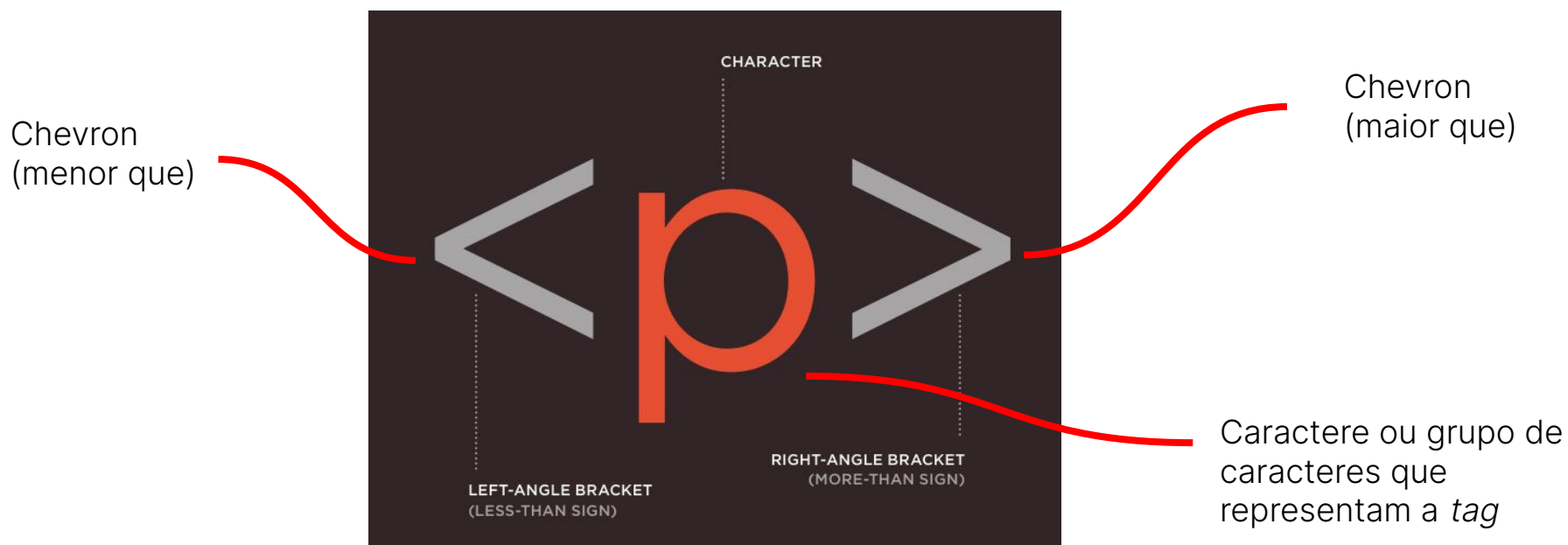
As **tags** são a forma pela qual estruturamos um documento HTML.

Tipos de *tags*:

- *Tags* de abertura;
- *Tags* de fechamento;
- *Tags* de fechamento automático.

Tags de abertura

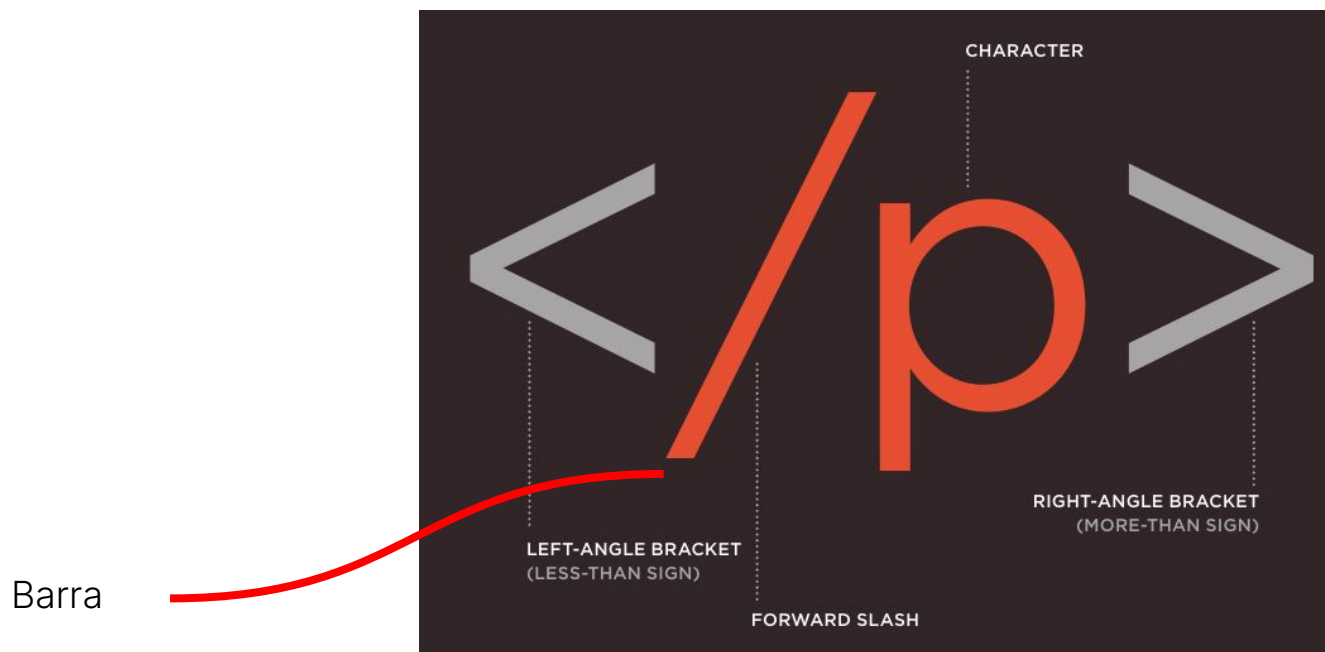
Demarca o início de um determinado conteúdo **que será exposto na página.**



Créditos da imagem: Duckett, Jon. **HTML & CSS: Design and Build Websites.**

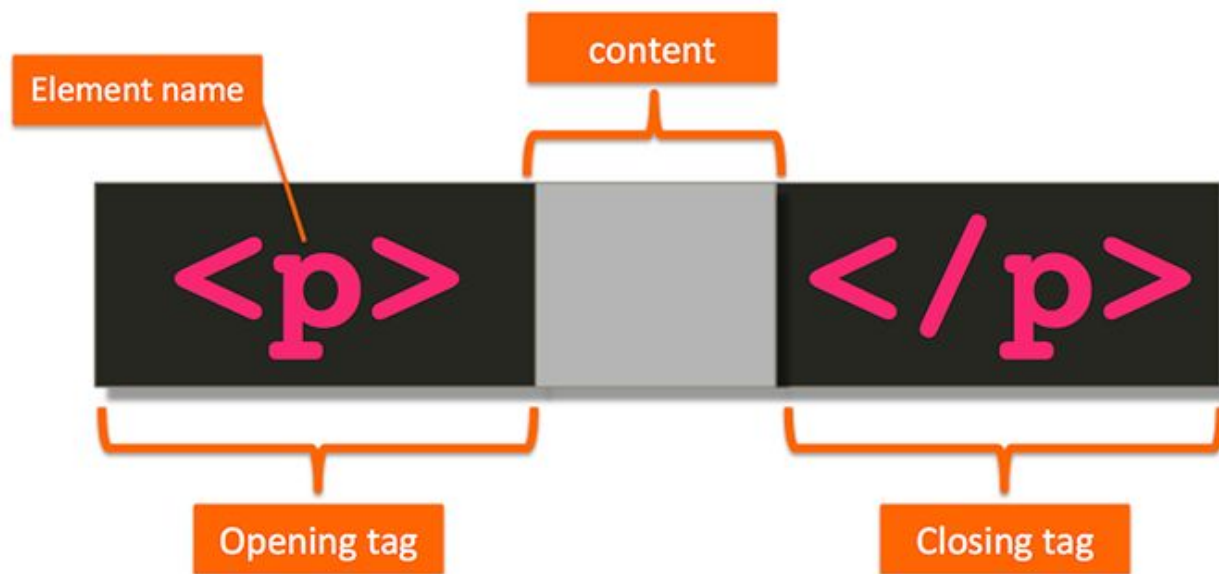
Tags de fechamento

Demarcam o fim de determinado conteúdo. Cada "bloco" de conteúdo é acompanhado por um par de *tag* de abertura e *tag* de fechamento.



Créditos da imagem: Duckett, Jon. **HTML & CSS: Design and Build Websites.**

Par de *tags* de abertura e fechamento



Tag de fechamento automático (self closing tags)

São *tags* que não precisam de fechamento. A partir da versão 5 do HTML, a barra passou a ser opcional, sendo estas duas sintaxes permitidas:

```
<br>
```

```
<br />
```

Atributos

Atributos complementam as *tags*. Fazendo uma analogia simples, são como “adjetivos” das *tags*.

Atributos possuem um par de **nome** e **valor**, conforme mostra o exemplo abaixo:



```
1 <p lang="pt-br">Vem Ser</p>
```

The diagram shows an HTML tag `<p lang="pt-br">Vem Ser</p>` inside a code editor window. A red line points from the word "Nome" to the attribute name `lang`. Another red line points from the word "Valor" to the attribute value `"pt-br"`.

Estrutura de um documento HTML

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```


Tag head



```
1 <head>
2   <meta charset="UTF-8" />
3   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
4   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
5   <title>Document</title>
6 </head>
```

Tag meta

A *tag meta* fornece informações aos mecanismos de busca e aos *browsers* sobre o nosso site.

Por exemplo, no caso anterior, o atributo ***charset*** informa que o conjunto de caracteres que utilizamos no site é o UTF-8, que contém boa parte das acentuações necessárias nas linguagens ocidentais.

Já o atributo ***http-equiv*** e o ***content*** (na linha 3) fornecem uma espécie de compatibilidade mínima com navegadores *Internet Explorer*, para computadores antigos.

Por último, os atributos ***name*** e ***content*** (na linha 4) fornecem informações acerca da largura que o documento HTML deve adotar, configurando a largura do dispositivo que acessar como padrão e configurando a escala de zoom no celular como zero.

*Tag **body***

Essa *tag* contém tudo aquilo que vai ser mostrado no **corpo** da página, como o nome sugere.

Podemos ter parágrafos, links, imagens, seções, citações e vários outros recursos.

Tags de título

São *tags* que devemos utilizar para os títulos das nossas página.

As *tags* de título começam sempre com **h** e vão de **h1** até **h6**.
O *h* vem de *heading*.

The screenshot shows a code editor on the left with a file named 'index.html'. It contains six lines of HTML code, each defining a heading tag followed by the word 'Título':

```

1 <h1>Título</h1>
2 <h2>Título</h2>
3 <h3>Título</h3>
4 <h4>Título</h4>
5 <h5>Título</h5>
6 <h6>Título</h6>

```

On the right, a browser window displays the rendered output. The address bar shows 'https://5iq58s.csb.app/'. The page content shows six instances of the word 'Título', each rendered in a different font size and weight corresponding to its heading level (h1 through h6).

<https://codesandbox.io/s/titulos-5iq58s>

Tags de conteúdo

<code><p></p></code>	→ parágrafo
<code></code>	→ linha única
<code></code>	→ itálico
<code></code>	→ negrito
<code></code>	→ imagem
<code><a></code>	→ link
<code><cite></cite></code>	→ citações pequenas
<code><blockquote></blockquote></code>	→ citações que tomem um parágrafo inteiro
<code><abbr></abbr></code>	→ acrônimos
<code><address></address></code>	→ endereços



Semântica
e estrutura

O que é semântica em uma página web?

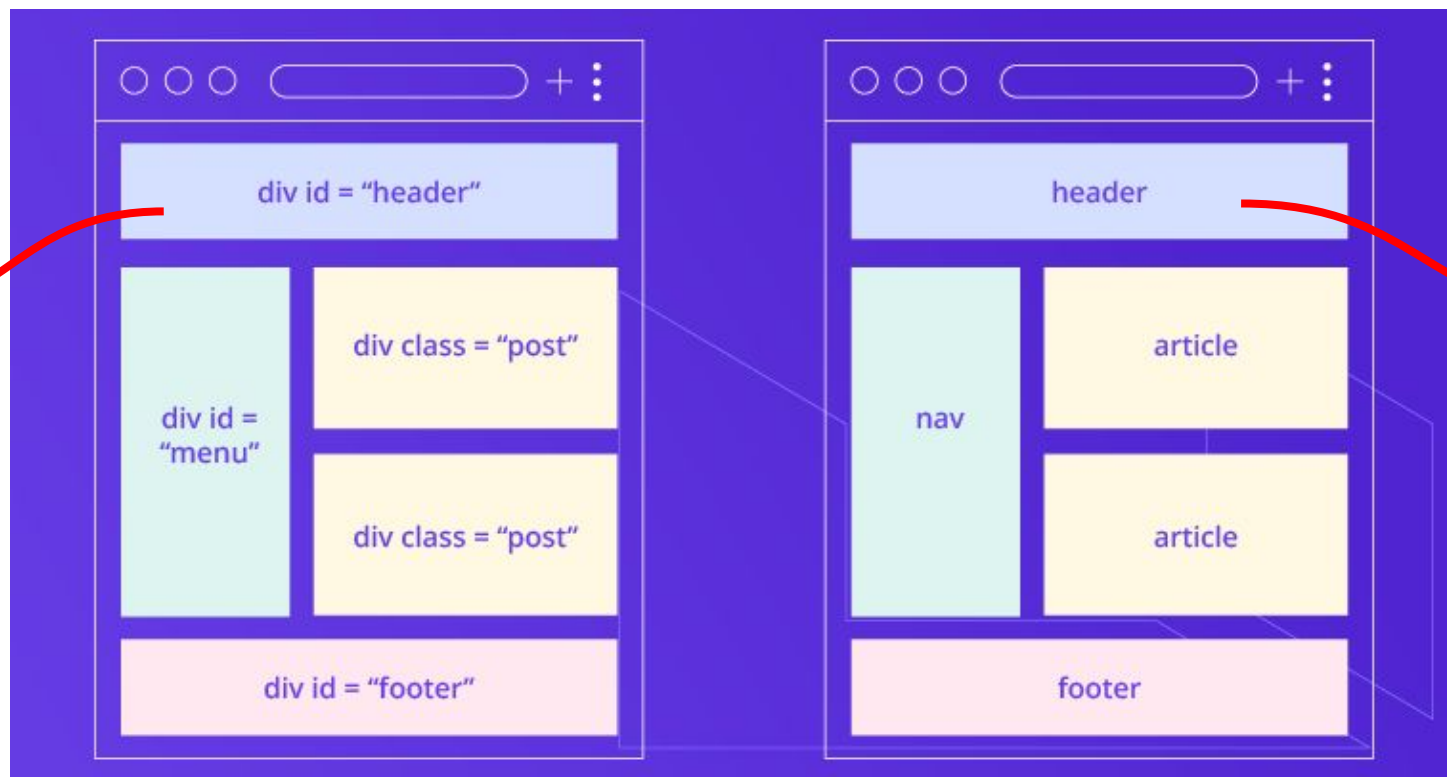
Os **elementos semânticos** são carregados de **sentido**. Ou seja, eles não apenas são responsáveis pela estrutura da página, como também nos “contam uma história”.

Esses elementos descrevem seus significados para o navegador, para os mecanismos de busca e também para os desenvolvedores.

Eles também auxiliam no SEO das páginas (*Search Engine Optimization*), que consiste na otimização do código HTML para ranqueamento nos mecanismos de busca (ex.: Google, Bing).

Estrutura de uma página web

Estrutura
não semântica



Estrutura
semântica

Tags de estrutura

<code><header></header></code>	→ cabeçalho da página
<code><main></main></code>	→ seção principal da página, usar apenas uma vez no site!
<code><section></section></code>	→ seção da página
<code><article></article></code>	→ artigo
<code><aside></aside></code>	→ seção lateral
<code><nav></nav></code>	→ lista de navegação do site
<code><footer></footer></code>	→ rodapé

Documentação do HTML que fala sobre só usar uma única *main* no site:

<https://html.spec.whatwg.org/multipage/grouping-content.html#the-main-element>



Listas

Listas ordenadas

A *tag* que define as listas ordenadas é ``, sendo ol um acrônimo para *ordered list*. Os itens da lista são definidos por ``, sendo li um acrônimo para *list item*.

The screenshot shows a code editor on the left with the following HTML code:

```

1 <h1>Minhas linguagens favoritas</h1>
2 <ol>
3   <li>JavaScript</li>
4   <li>Python</li>
5   <li>C</li>
6 </ol>

```

On the right, a browser window displays the rendered output at the URL `https://5th5kk.csb.app/`. The browser shows the title "Minhas linguagens favoritas" and an ordered list:

1. JavaScript
2. Python
3. C

<https://codesandbox.io/s/listas-5th5kk>

Listas não ordenadas

As listas não ordenadas são definidas pela tag ``, sendo ul um acrônimo para *unordered list*. Os itens também são definidos por ``.

```
<h1>Frameworks JavaScript</h1>
<ul>
  <li>Next</li>
  <li>Angular</li>
  <li>Vue</li>
  <li>Svelte</li>
</ul>
```

Frameworks JavaScript

- Next
- Angular
- Vue
- Svelte

<https://codesandbox.io/s/listas-5th5kk>



Tabelas

Tabelas

Tabelas nos permitem escrever dados de maneira **tabular** e **estruturada**.

Para isso, usamos a tag **<table></table>**.

A tag **<tr></tr>** representa as linhas da tabela (tr: acrônimo de *table row*).

A tag **<td></td>** representa as colunas da tabela (<td: acrônimo de *table data*).

Tabelas

index.html x

```

1 <h1>Instrutoras de Front-End</h1>
2 <table border="1">
3   <tr>
4     <td>Nome</td>
5     <td>Stack</td>
6   </tr>
7   <tr>
8     <td>Mayra</td>
9     <td>React, Node, Python</td>
10  </tr>
11  <tr>
12    <td>Cris</td>
13    <td>React, Next</td>
14  </tr>
15 </table>

```

Browser Tests Terminal

https://fgd2dt.csb.app/

Instrutoras de Front-End

Nome	Stack
Mayra	React, Node, Python
Cris	React, Next

<https://codesandbox.io/s/tabelas-fgd2dt>

Tabelas semânticas

Há uma nova forma de criar tabelas, que é utilizando as *tags* semânticas:

`<thead></thead>` → **cabeçalho da tabela**

`<tbody></tbody>` → **corpo da tabela**

`<tfoot></tfoot>` → **rodapé da tabela**

Reescrevendo a tabela utilizando *tags semânticas*

index.html x

```

1 <h1>Instrutoras de Front-End</h1>
2 <table border="1">
3   <thead>
4     <tr>
5       <th>Nome</th>
6       <th>Stack</th>
7     </tr>
8   </thead>
9   <tbody>
10    <tr>
11      <td>Mayra</td>
12      <td>React, Node, Python</td>
13    </tr>
14    <tr>
15      <td>Cris</td>
16      <td>React, Next</td>
17    </tr>
18  </tbody>
19 </table>

```

Browser
Tests
Terminal

https://d04zhg.csb.app/

Instrutoras de Front-End

Nome	Stack
Mayra	React, Node, Python
Cris	React, Next



Formulários

Formulários

A tag `<form></form>` representa uma seção de controles interativos. Eles nos permitem receber dados do usuário, que são submetidos a algum servidor web e armazenados num banco de dados.

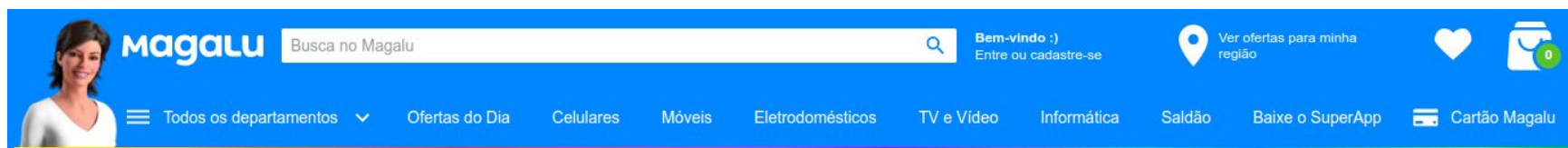
Newsletter

First Name

Last Name

Email Address

Exemplo de formulário



Identificação

Quero criar uma conta

E-mail

Continuar

Dúvidas? [fale conosco](#)

Já sou cliente

E-mail, CPF ou CNPJ

Senha

Continuar

[Esqueci minha senha](#)

Use sua rede social para se conectar*



Seus dados pessoais serão respeitados de acordo com a nossa [política de privacidade](#). *Nada será publicado em sua timeline. Serviço válido somente para pessoas físicas.
Em caso de dúvidas, acesse nossa central de atendimento.

Tags utilizada nos formulários

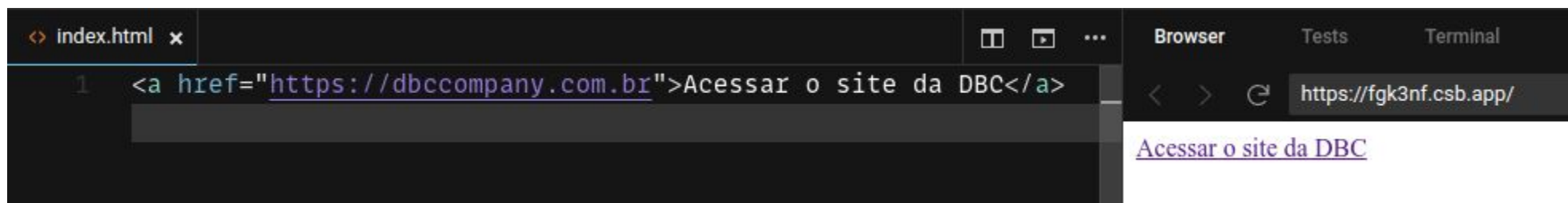
- `<form></form>` → **inicializa o formulário**
- `<input />` → **campo que recebe os dados do usuário**
- `<label></label>` → **título dos campos**
- `<select></select>` → **caixa de seleção**
- `<textarea />` → **campo de muitas linhas**



Conteúdo
multimídia

Links

Para isso usamos a tag `<a>` e o atributo *href* (*hypertext reference*).

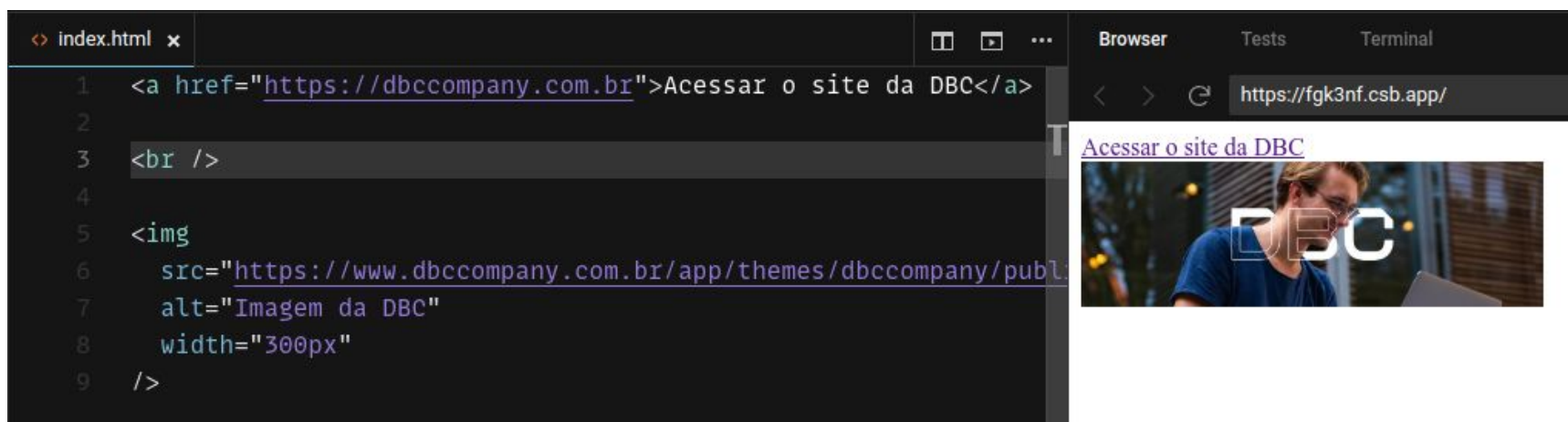


The screenshot shows a code editor on the left and a web browser on the right. In the code editor, the file 'index.html' is open, and the following HTML code is visible on line 1:

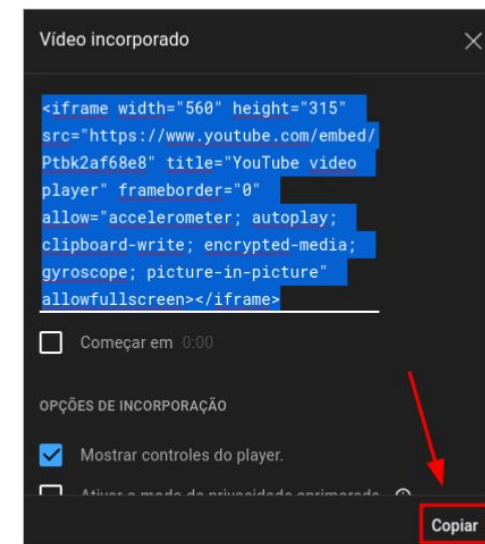
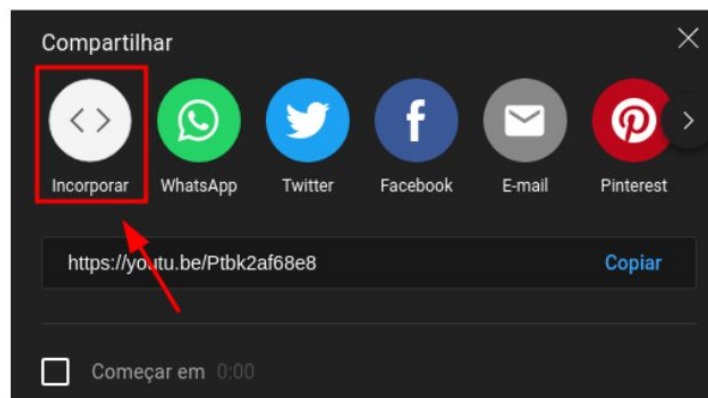
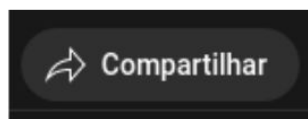
```
<a href="https://dbccompany.com.br">Acessar o site da DBC</a>
```

The browser window on the right shows the URL 'https://fgk3nf.csb.app/' in the address bar. Below the address bar, the text 'Acessar o site da DBC' is displayed as a clickable link.

Imagens



Incorporando vídeos





Referências da aula

Livro: **HTML & CSS: Design and Build Websites** - Jon Duckett

Curso: HTML & CSS in depth - Meta



Let's *Tech Up Together*