

## **Cliente X Server**

**Clientes** são as máquinas usadas para acesso aos sites (computadores, celulares, tablets).

**Server** (servidores) são as fontes de onde vem os arquivos e informações.

Os sites e aplicativos são formados por três etapas

- 1: Imagem e organização que é todo o conteúdo que irá ter no site HTML.
- 2: Desing, organização das imagens, aparência do site e sua formatação visual CSS.
- 3: Programação, garantir o funcionamento correto do site e as interações necessárias JavaScript ou alguma outra forma de linguagem.

**HTML** (Hyper Text Markup Language): Linguagem de Marcação de Hipertexto.

Linguagem de marcação que define a estrutura e o conteúdo das páginas da web.

São os conteúdos que vai no site, os textos, imagens, tabelas e todas as informações no geral.

**CSS**: São as cores, sombras, tamanhos, posicionamento, paleta de cores, fontes e tudo que envolva o visual do site ou app.

**JavaScript** (ou outras linguagens) Menus, interação, validações e tudo o que funciona no site.

Durante a criação dos arquivos (pastas, imagens, vídeos, nomes) tudo o que for ser salvo no arquivo, usar apenas letras minúsculas.

**Front-End**: Dev responsável pela parte que vai ter contato com o cliente, as informações, visual/formatação e funcionalidade do site/app. No caso toda informação que é fornecida ao cliente.

**Back-End**: Dev responsável pela comunicação entre as funções do front junto ao servidor do qual o site/app é ligado.

## Linguagem: VS Code HTML

Base de código: Usar tecla “!” para mostrar base.

<!DOCTYPE html> Indica o formato

<html lang="pt-br"> Idioma

<head> Cabeça do site

<meta charset="UTF-8"> Caracteres ocidentais

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title> Título do documento

</head> Cabeça fechamento

<body> Corpo abertura

</body> Corpo fechamento

</html> Fim do site

```
<h1>Testando carga e imagem</h1>

  <p>Podemos carregar também, imagens que
  estão em sub-pastas</p>

<p>Podemos carregar imagens externas</p>

```

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
  <link rel="shortcut icon"
href="ico-monkey.ico" type="image/x-
icon">
  <title>Test Favicon</title>
</head>
```

```
<h1>Entendendo Portugal</h1>
<h2>Introdução</h2>
<h3>Justificação da tematica</h3>
```

## Entendendo Portugal

### Introdução

### Justificação da tematica

### Formas de add imagem

Usar= `img` gera: `<img src="" alt="">`

Após o “src” é colocado a imagem com base no nome dado na pasta do arquivo.

O “alt” é a alternativa caso não seja possível abrir a imagem, podendo ser usado para leitores de tela e acessibilidade.

Pode ser copiado também a URL da imagem desejada.

### Add imagem ao link da aba do site

Colocar comando no “<head>”

Usar: `link:favicon`, gera: `link`

`rel="shortcut icon"`

`href="favicon.ico" type="image/x-  
icon">`

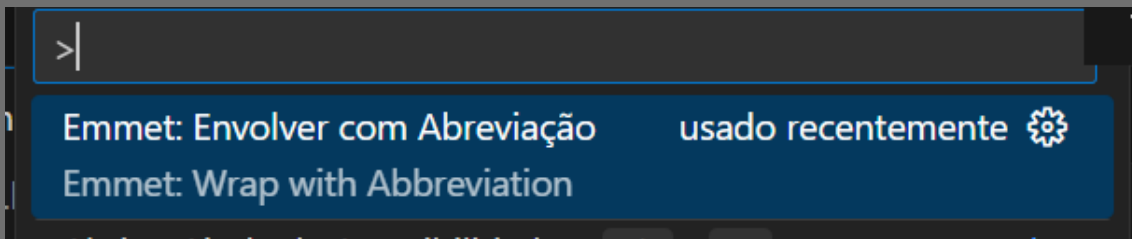
Apagar o “favicon.ico” e colocar o nome do arquivo desejado.

### Hierarquia de títulos

O <h1> é o assunto principal, enquanto o <h2> é o seu conteúdo, como tema daquele único parágrafo e o <h3> um sub-assunto dentro da temática de <h1> e <h2>.

Usada para dividir textos em ordem de conteúdos e dentro de suas respectivas classes.

\*Dica: Ir apenas até o <h6> para não deixar muito longo e confuso.



\*Dica: Selecionar uma sentença e apertar CTRL+SHIFT+P abre uma caixa com a opção Emmet: Wrap with Abbreviation (Envolver com abreviação) onde você pode escolher qual tag usar para aquele ponto, evitando criar uma nova tag no início e fim.

CTRL+SHIFT+P+Enter+(tag desejada)+Enter

```
<p>Nessa frase temos um <b>temos em  
negrito usando a tag B (não semântica)</  
b></p>  
<p>Nesta frase tenho um <strong> termo em  
destaque</strong> usando a tag STRONG  
(semântica).</p>  
<h2>Itálico / Ênfase</h2>  
<p>Nesta frase, temos um <i>termo em  
itálico</i> usando a tag I (não semântica)  
</p>  
<p>Nesta frase, <em>termo um termo</em>  
em ênfase usando a tag EM </p>
```

### Negrito e Itálico

Negrito: Usam-se as Tags `<b>` `<b>` ou `<strong>` `<strong>`.

Recomendável: `<strong>` `<strong>`.

Itálico: Usam-se as Tags `<i>` `<i>` ou `<em>` `<em>`.

Recomendável: `<em>` `<em>`.

\*Dica: Se for forma: Usar sem semântica, se for sentido, usar com semântica.

```
<h1>Titulo</h1>  
<h2>Subtitulo</h2>  
<h3>Dentro do subtitulo</h3>  
<P>Conteúdo do paragrafo paragrafo</P>  
<strong>Negrito com sentido</strong>  
<b>Negrito com forma</b>  
<em>Itálico com sentido</em>  
<i>Itálico com forma</i>  
  
CTRL+SHIFT+P+Enter+(tag desejada)+Enter (Atalho)  
<link rel="shortcut icon" href="" type="image/x-icon">  
Tag acima, add simbolo de link
```

(Tags aprendidas até o momento – 04/11/24)

```
<style>  
  mark {background-color: coral;}  
</style>
```

A tag `<style>` serve para formatar todas as tags que virão a seguir, nesse caso, a tag `<mark>` que é de marcação, todas as futuras tags de marcação virão com a formatação desejada.

`<p>Podemos criar tambem <mark>um texto marcado</mark> usando a tag MARK</p>`  
`<p>E no outro parágrafo temos <mark>texto marcado</mark> no final</p>`

Podemos criar tambem **um texto marcado** usando a tag MARK

E no outro parágrafo temos **texto marcado** no final

Tag “<mark>” usada para fazer marcação nas palavras do texto. Lembrando que por ser em HTML, a formatação não é semântica, é apenas a forma.

## Texto deletado

podemos marcar um ~~texto como excluido~~ para indicar que ele deve ser lido mas não considerado

## texto inserido

podemos considerar um texto como inserido para dar uma ênfase e indicar que ele foi adicionado depois

O sublinhado tambem pode ser criado pela tag U (não semantica)

## Texto sobrescrito

para inserir coisas do tipo  $x^{20}+3$

## Texto subscripto

inserir coisas do tipo H<sub>2</sub>O

## Texto deletado (riscado)

Tag: `<del> </del>`

## Texto inserido (sublinhado)

Tag: `<ins> </ins>`

## Texto sublinhado S/semântica

Tag: `<u> </u>`

## Texto sobrescrito (cima)

Tag: `<sup> </sup>`

## Texto subscripto (baixo)

Tag: `<sub> </sub>`

```
<pre>
  <code>
    num = int(input('Digite um número'))
    if num % 2 == 0:
      print(f'0 número {num} é PAR)
    else:
      print(f'0 número {num} é ÍMPAR)
      print('Fim do programa')
  </code>
</pre>
```

O comando `document.getElementById( 'test')`

```
num = int(input('Digite um número'))
if num % 2 == 0:
  print(f'0 número {num} é PAR)
else:
  print(f'0 número {num} é ÍMPAR)
  print('Fim do programa')
```

A tag `<pre> </pre>` vai servir para manter a quebra de linha no texto envelopado.

A tag `<code> </code>` muda a formação da letra, boa para ser usada para descrever códigos.

Tag `<pre> </pre>` aplicada na quebra de linhas.

Tag `<code> </code>` aplicada na formatação de texto: `document.getElementById`

Tag: `<q> </q>` é utilizada para dar aspas ao texto.

```
<p>Estou estudando <abbr title="HyperText Markup Language">HTML</abbr> e <abbr title="Cascading Style Sheet">CSS</abbr></p>
```

Tag: <abbr> </abbr> serve para mostrar o significado das siglas usadas nos textos.

Envelopa a sigla que deseja que tenha o significado e coloca a Tag, ao ser inserida, aparece a informação quando passa o cursor em cima. Após digitada, aparece da seguinte forma:

<abbr title="">. Entre as aspas coloca o significado. (exemplo abaixo)

Estou estudando HTML e CSS

HyperText Markup Language

## Texto invertido

Tag: <abbr title="HyperText Markup Language">.

```
<h1>Trabalhando com listas</h1>
<h2>Listas ordenadas</h2>
  <ol type="1" start="4">
    <li>Acordar</li>
    <li>Tomar café</li>
  </ol>
  <ol type="I"> <!--1 A a I i-->
    <li>Tomar café</li>
    <li>Escovar os dentes</li>
  </ol>
<h2>Listas não ordenadas</h2>
  <ul type="square"> <!--disc circle square-->
    <li>Tomar café</li>
    <li>Escovar os dentes</li>
  </ul>
```

### Listas

Tag: <ol> </ol> define uma lista seguindo ordem, numérica ou alfabética.

Tag: <li> </li> define os itens da lista e é colocado antes de cada item, não existe a necessidade do fechamento da tag. Existe a opção numérica, alfabética e com algarismo romano.

Tag: <ul> </ul> é usado para listas sem ordenação, usando símbolos no início dos nomes. Opção de círculo, bola e quadrado.

### Listas ordenadas

4. Acordar
5. Tomar café
6. Escovar os dentes

- I. Tomar café
- II. Escovar os dentes
- III. ir para faculdade

### Listas não ordenadas

- Tomar café
- Escovar os dentes
- ir para faculdade

```

<h2>Meus jogos favoritos</h2>
<ol>
  <li>Nitendo</li>
  <ul type="square">
    <li>Mario Bros</li>
    <li>dokey</li>
  </ul type="square">
  <li>Nes</li>
  <ul type="square">
    <li>Shadow</li>
    <ul type="circle">
      <li>Gaius</li>
      <li>Valus</li>
    </ul>
  </ul>
  <li>Playstaition</li>
  <ul type="square">
    <li>Atari</li>
  </ul>
</ol>

```

## Listas 2

Podem ser criados sub-listas dentro das listas, para citar itens, como no exemplo uma lista de videogames e dentro do modelo o nome dos jogos.

Para isso usam-se as Tags informadas acima.

`<ol> </ol>` para informar a organização de uma lista seguindo uma ordem numérica ou alfabética.

`<li> </li>` para os itens da lista no geral e `<ul> </ul>` quando quiser fazer os objetos secundários apenas com marcações e não uma ordem. Segue exemplo abaixo da formatação aplicada.

1. Nitendo
  - Mario Bros
  - dokey
2. Nes
  - Shadow
    - Gaius
    - Malus
    - Valus
3. Playstaition
  - Atari

```

<h2>Lista de definições</h2>
<dt>HTML</dt>
<dd>Linguagem de marcação para criação de conteúdo de um site</dd>
<dt>CSS</dt>
<dd>Linguagem de marcação para criação do desing do site</dd>
<dt>JavaScript</dt>
<dd>Linguagem de programação para as funcionalidades do site</dd>

```

## Lista de definições

Tag: `<dt> </dt>` serve para a palavra a ser definida.

Tag: `<dd> </dd>` para dar a definição.

Podendo ser usada para dicionários ou exemplificação.

Ex abaixo.

## HTML

Linguagem de marcação para criação de conteúdo de um site

## CSS

Linguagem de marcação para criação do desing do site

## JavaScript

Linguagem de programação para as funcionalidades do site

## Links

Organização de código é importante para que o site seja achado no momento da busca.

```
<h1>Usando Links externos</h1>
<p>Você pode acessar <a href="https://gustavoguanabara.github.io">meu
repositorio publico no GitHub</a></p>
<p>Você também pode acessar o <a
href="https://www.instagram.com/miguelfelipes98/" target="_blank"
rel="external">Meu instagram</a></p>
```

Links externos, onde ocorre o direcionamento para outro site.

Tag usada: `<a>` `</a>`, o “A” representa “âncora”.

Após usado, abre a Tag

`<a href="" target="" rel="">`

Em: href: Você add o link

Em: target: `_blank`

Em: rel: `external`

A tag sendo feito apenas com o `<a>` `</a>` substitui a pagina que você está acessando, enquanto a tag completa: `<a href="" target="" rel="">` abre uma nova guia.

Em: target: `_blank` é o que define a abertura em outra guia

Ex: `<a href="https://www.instagram.com/miguelfelipes98/" target="_blank" rel="external">`

## Link interno

Telca “Ctrl” + espaço, quando estiver dentro do “href” já mostra as paginas internas do arquivo.

```
<h2>Usando links internos</h2>
<p>Está é a primeira pagina do site, se
quiser pode acessar minha <a href="pag002.
html" rel="next" target="_self"> segunda
pagina</a></p>
<p>Se joga na nossa <a href="noticias/
pag003.html" rel="next">pagina de
noticias</a></p>
```

Segue o mesmo caso da tag de

cima `<a href="pag002.html"`

A tag poderia ser fechada aí.

O nome `pag002.html` eu que dei a nova pasta de arquivo.

As tags: `rel="next"`

`target="_self">` são para

semântica. O “Next” mostra que irá para uma pagina seguinte e o `_self` mostra que é uma página do próprio site.

```
<p>Volta para <a href=" ../index.
html">primeira pagina
</a></p>
<p>Volte para <a href=" ../pag002.
html">segunda</a></p>
```

É usado: `../` para mostrar que o arquivo a ser selecionado para o link está dentro de uma pasta, como que fosse para voltar na pasta do VS Code.

## Links para Download

```
<h1>Links para dowload</h1>
<ul>
  <li><a href="Livro/HTML - Material de
apoio.pdf" type="application/
pdf">Livro HTML</a></li>
  <li><a href="Curso.rar"
download="meulivro.zip"
type="application/zip">Livro VScod</
a></li>
</ul>
```

Para baixar arquivos, primeiro colocar os arquivos dentro da pasta onde o site está sendo criado.

Seguir com a tag `<a href=` e colocar o nome do arquivo.

A tag poderia ser apenas:

```
<li><a href="Curso.rar">
```

Incluimos as tags:

```
download="meulivro.zip"
```

```
type="application/zip" por
```

semântica (ajudar em busca)

O arquivo em PDF é lido no próprio navegador enquanto o arquivo ZIP é baixado.