

# Cristian Oliveira

## Front End Developer - Scores and Levels

[cristianpoliveira@gmail.com](mailto:cristianpoliveira@gmail.com)

# HTML

O que é HTML

O que é HTML - Os blocos invisíveis

Elementos

Sintaxe

Tags

HTML Estrutura

HTML Atributos

Nesting

Hierarquia

Elementos Block e Inline

Código Limpo e Sustentável

Semântica

# Requisitos

Edit de Texto / IDE (VS Code, Sublime Text, Atom)

Google Chrome

# O que é HTML ?

HTML: hypertext markup language

Especificação

Linguagem de marcação

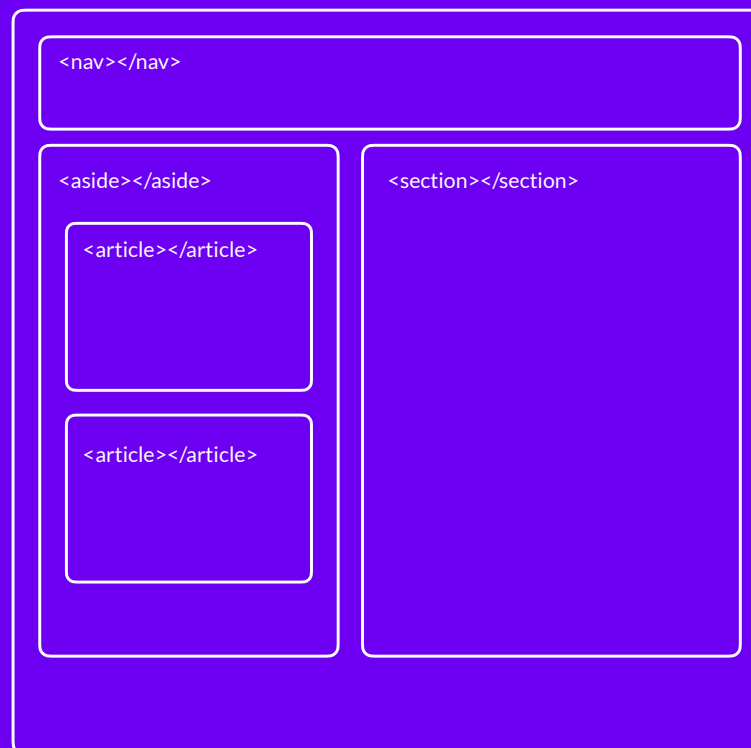
# O que é HTML ?

HTML: hypertext markup language

Especificação

Linguagem de Marcação

Consiste em, diferentes elementos que utilizamos na construção de uma página.



Portanto, “quem” define os standards da web?

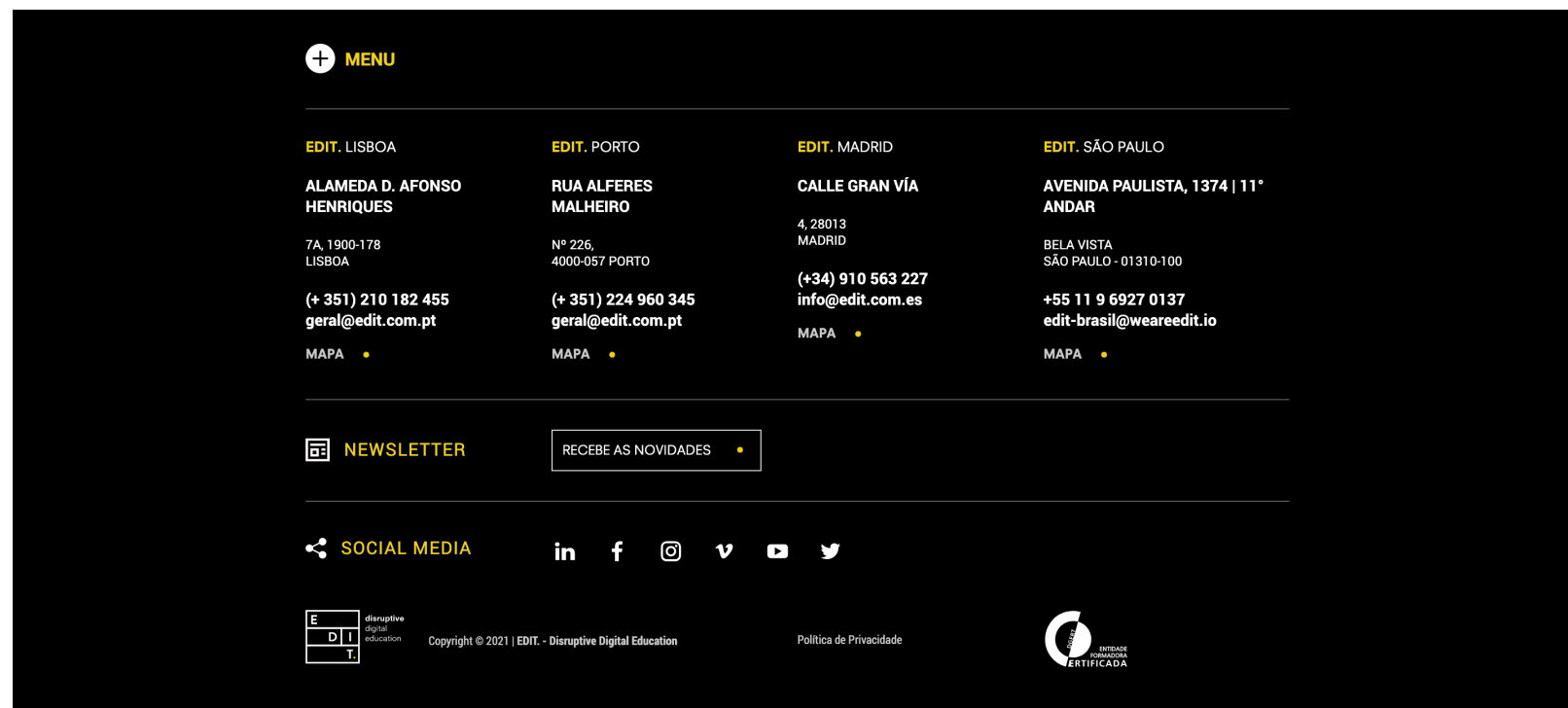
## W3C

O World Wide Web Consortium (W3C) é a principal organização internacional responsável pela World Wide Web (abreviado WWW ou W3).

Ora,  
A parte divertida.

# O que é realmente HTML ?

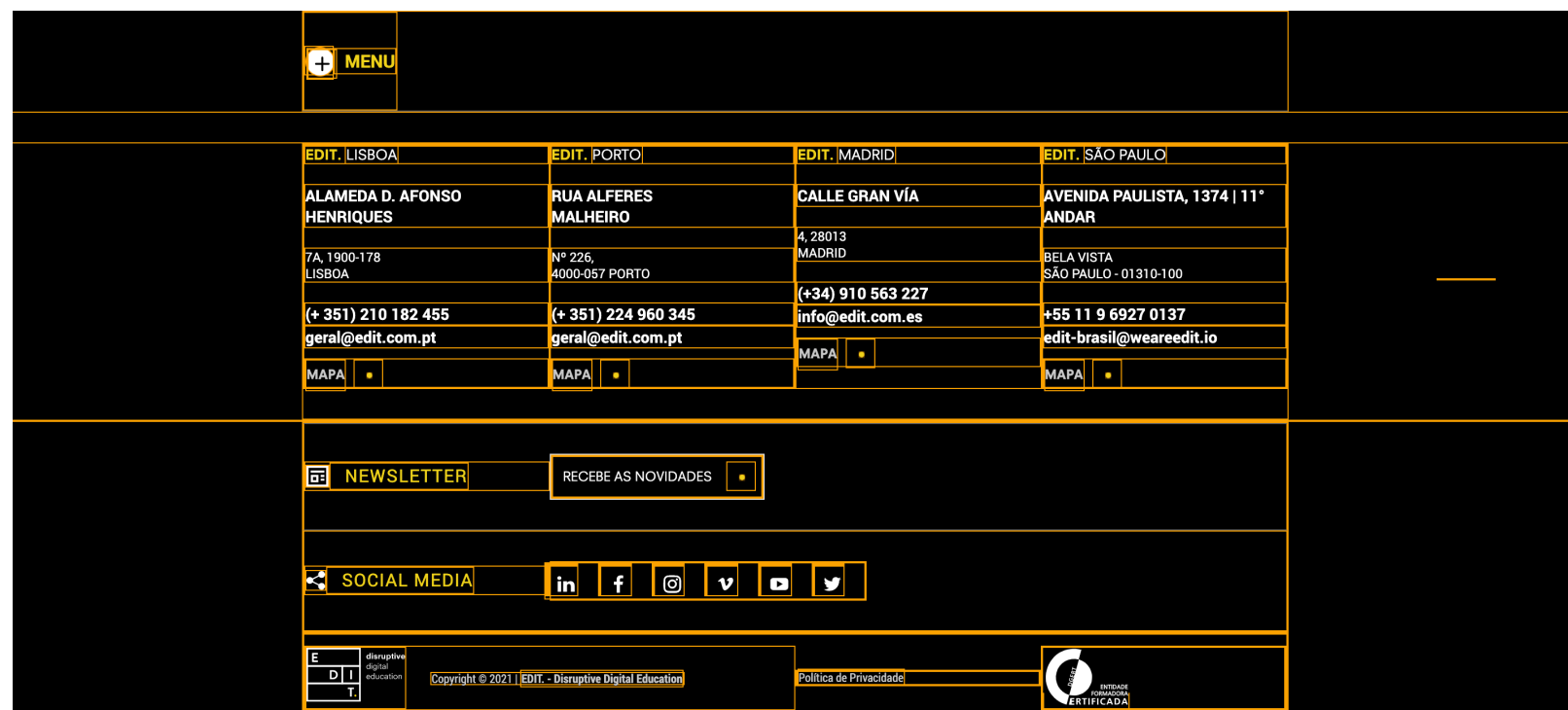
É a ferramenta que usamos para construir estruturas que representam uma página.





# Os “Blocos Invisíveis”

Estruturamos a página com base em elementos, ou grupos de elementos.



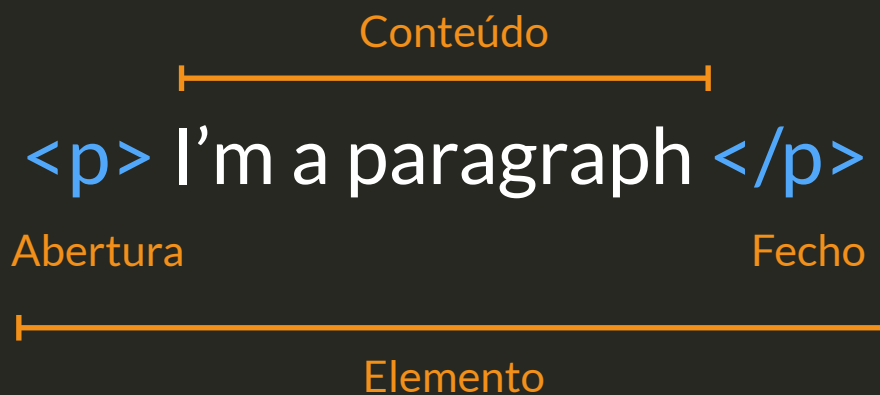
# Os Elementos

Os Elementos são na realidade entidades. São entidades na nossa página.  
Escritas de uma maneira especial.

`<tagname>` conteúdo aqui... `</tagname>`

# Os Elementos

Os Elementos são na realidade entidades. São entidades na nossa página. Escritas de uma maneira especial.



Notem que, tudo o que tem um início tem sempre um fim.

# Os elementos - Tags

Tags são palavras específicas, que representam a entidade que queremos construir.

Algumas são:

<h1>      <h2>      <h3>      <h4>      <h5>      <h6>      <p>      <span>      <a>  
<address>   <article>   <aside>   <blockquote>   <canvas>   <dd>   <div>  
         <dl>   <dt>   <fieldset>   <figcaption>   <figure>   <footer>  
<form>   <h1>–<h6>   <header>   <hr>   <li>   <main>   <nav>   <noscript>  
         <ol>   <p>   <pre>   <section>   <table>   <tfoot>   <video>

E etc.

# Estrutura de um ficheiro HTML

Há alguns requisitos quando criamos uma página.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
</body>
</html>
```

# Estrutura de um ficheiro HTML

Cada tag tem o seu lugar e serve o seu propósito.

```
<!-- A declaração <!DOCTYPE html> define que este documento é HTML5 -->
<!DOCTYPE html>
<!-- a tag <html> é o elemento raiz da página HTML -->
<html lang="en">
  <!-- a tag <head> contém metadata (informação) sobre a página.
  Não será visível na página renderizada.
  -->
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- o <title> especifica o título do documento -->
    <title>Document</title>
  </head>
  <!-- o <body> contém a parte visível da página -->
  <body>
  </body>
</html>
```

# Estrutura de um ficheiro HTML

Cada tag tem o seu lugar e serve o seu propósito.

```
<!-- A declaração <!DOCTYPE html> define que este documento é HTML5 -->
<!DOCTYPE html>
<!-- a tag <html> é o elemento raiz da página HTML -->
<html lang="en">
<!-- a tag <head> contém metadata (informação) sobre a página.
Não será visível na página renderizada.
-->
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
<!-- o <title> especifica o título do documento -->
  <title>Document</title>
</head>
<!-- o <body> contém a parte visível da página -->
<body>
</body>
</html>
```

By the way, eu sou um comentário  
Esta é a minha sintaxe:

**<!-- comment -->**

Não serei visível no documento  
renderizado.

# Hands on deck!

Vamos escrever o nosso primeiro parágrafo.

Hands on deck

**Exercício**



# Atributos HTML

“Semanticamente”, em casos básicos, podemos escrever html com tags simples.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Example</title>
</head>
<body>
  <header></header>
  <nav></nav>
  <aside>
    <article></article>
  </aside>
  <section>
    <article></article>
  </section>
  <footer></footer>
</body>
</html>
```

Side Note

## Semântica

Lembrem-se desta definição.  
Falaremos mais à frente.

# Atributos HTML

**style** é usado para definir estilos como a cor, fonte, tamanho e outros.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Example</title>
</head>
<body>
  <header></header>
  <nav style="background-color: red;"></nav>
  <aside>
    <article>
      <img src="">
    </article>
  </aside>
  <section>
    <article></article>
  </section>
  <footer></footer>
</body>
</html>
```

```
<nav style="background-color: red;"></nav>
```

# Atributos HTML

**src** define a origem da imagem. **width** refere a largura da imagem.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Example</title>
</head>
<body>
  <header></header>
  <nav style="background-color: red;"></nav>
  <aside>
    <article>
      <img src="">
    </article>
  </aside>
  <section>
    <article></article>
  </section>
  <footer></footer>
</body>
</html>
```

```

```

# Atributos HTML

Todos os elementos de HTML podem ter **atributos**

Atributos anexam **informação adicional** sobre os elementos

Atributos são sempre adicionados **na tag de abertura**.

Atributos normalmente, alias, devem, ser adicionados em name/value pairs

ex: **name="value"**

```
<tag attribute="value">
```

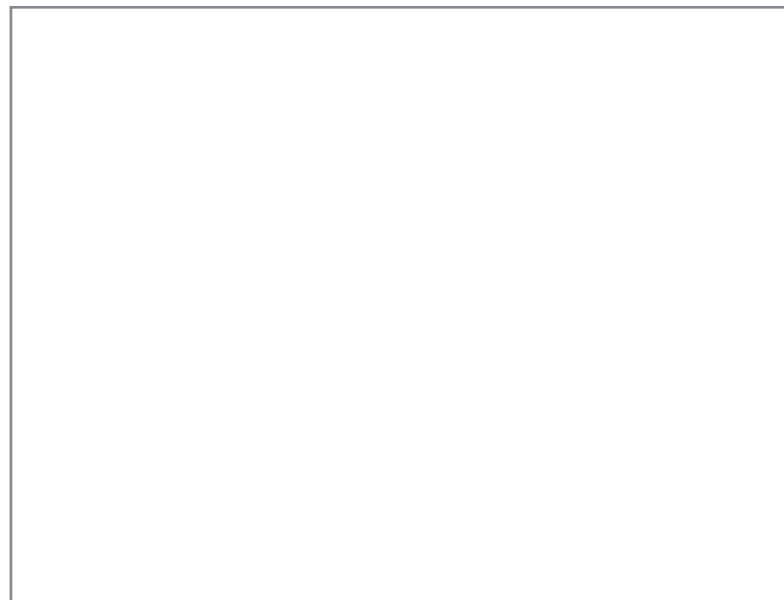
```
<p class="my-class" id="my-id" style="color: red">
```

# Nesting

Considera que criamos a seguinte secção de um blog.

```
<section class="articles_section">  
</section>
```

// articles section



# Nesting

Nesting é quando adicionamos **filhos** ao nosso elemento.

```
<section class="articles_section">  
  <article class="article"></article>  
  <article class="article"></article>  
  [...]  
</section>
```

// articles section

Article

Article

# Hierarquia

Alguns elementos tem a sua própria hierarquia. Seguem “a sua própria linha” de como são representados.

```
<h1>Heading 1</h1>  
<h2>Heading 2</h2>  
<h3>Heading 3</h3>  
<h4>Heading 4</h4>  
<h5>Heading 5</h5>  
<h6>Heading 6</h6>  
<p>Paragraph</p>
```

# Heading 1

## Heading 2

### Heading 3

#### Heading 4

##### Heading 5

###### Heading 6

Paragraph

# Hierarquia

Alguns elementos tem a sua própria hierarquia. Seguem “a sua própria linha” de como são representados.

```
<ul>
  <li>List Item 1</li>
  <li>List Item 2</li>
  <li>List Item 3</li>
</ul>
```

- List Item 1
- List Item 2
- List Item 3



# Hierarquia

Alguns elementos tem a sua própria hierarquia. Seguem “a sua própria linha” de como são representados.

```
<ol>
  <li>List Item 1</li>
  <li>List Item 2</li>
  <li>List Item 3</li>
</ol>
```

1. List Item 1
2. List Item 2
3. List Item 3

# Elementos - Block and Inline

**Block Elements** - Utilizam o espaço disponível no viewport. Começam sempre numa nova linha.

```
<div>  
  Content 1  
</div>  
<div>  
  Content 2  
</div>  
<div>  
  Content 3  
</div>
```

Content 1

Content 2

Content 3

# Elementos - Block and Inline

**Inline Elements** - Utilizam só, o espaço necessário ao seu conteúdo.

```
<span>  
  Just Span  
</span>  
<span>  
  I'm a bigger span  
</span>
```

Just Span

I'm a bigger span

# Código Limpo e Sustentável

Não esmagues. Usa o espaço. Respira.

```
<div><ul><li>List Item 1</li><li>List Item 2</li><li>List Item 3</li></ul></div>
```

```
<div>  
  <ul>  
    <li>List Item 1</li>  
    <li>List Item 2</li>  
    <li>List Item 3</li>  
  </ul>  
</div>
```

# Código Limpo e Sustentável

Indentar o código.

```
<div>
<ul>
<li>List Item 1</li>
<li>List Item 2</li>
<li>List Item 3</li>
</ul>
</div>
```

```
<div>
  <ul>
    <li>List Item 1</li>
    <li>List Item 2</li>
    <li>List Item 3</li>
  </ul>
</div>
```

# Código Limpo e Sustentável

Sê Consistente.

```
<div class="my_class">  
<div>  
<div class="my-class">  
<div>
```

```
<div class="my_class">  
<div>  
<div class="my_class">  
<div>
```

# Código Limpo e Sustentável

Dá significado ao nome que dás aos teus elementos. Pensa neles com amor e carinho.

```
<div class="articles_section">  
  <article class="small_article">  
  </article>  
</div>
```

```
<div class="ar">  
  <article class="s_art">  
  </article>  
</div>
```

# Código Limpo e Sustentável

Escreve código estruturado! Tens todas as ferramentas que precisas.

```
<div class="my_class">  
  <div class="heading_1">  
    BIG TITLE  
  </div>  
</div>
```

```
<div class="my_class">  
  <h1>BIG TITLE</h1>  
</div>
```



# Código Limpo e Sustentável

Semântica?

Nota

**Semântica**

Cada elemento tem o seu significado.

Uma nav é uma nav. Define o propósito de navegação.

Uma div não tem significado semântico, mas estruturalmente é uma divisão.

Uma section é uma secção.

O h1 está relacionado com hierarquia tipográfica. Deve ser usado num contexto tipográfico.

Um p é um parágrafo, um bloco de texto.

**Semântica** introduz significado ao código que escreves.

Pensa antes de escrever.

Irás poupar-te muito tempo e terás menos dores de cabeça.

# Hands on deck!

Vamos estruturar a página de um jornal.

Hands on deck

**Exercício**