

INSTITUTO FEDERAL

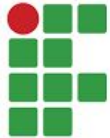
Catarinense

Campus Camboriú

Programação Web

Professor: Euclides Paim

euclides.paim@ifc.edu.br



INSTITUTO FEDERAL

Catarinense

Campus Camboriú

Fundamentos de HTML

Professor: Euclides Paim

euclides.paim@ifc.edu.br

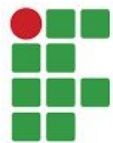
Desenvolvimento Web I

Sumário

- **Fundamentos de HTML**
 - Atributos globais;
 - Classes;
 - Id;
 - Iframes;
 - Head;
 - Elementos semânticos.



<html>

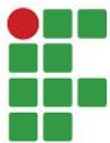


INSTITUTO FEDERAL
Catarinense
Campus Camboriú

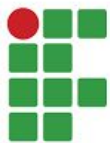
Desenvolvimento Web I



- **Atributos globais.**
 - **Atributos globais** são aqueles comuns a todos os elementos da HTML, ou seja, podem ser definidos para qualquer elemento da linguagem. Vejamos quais são os atributos globais mais comuns:
 - **id:** define o ID de um elemento. O ID é um identificador único de um elemento. É como se fosse um RG.
 - **class:** define as classes de um elemento. Uma classe não é única como o ID, e um elemento pode ter múltiplas classes, separadas por espaços.
 - **lang:** define o idioma do conteúdo do elemento e de seus atributos textuais.
 - **style:** define estilos para o elemento. Não é recomendado usar essa tag, visto que o HTML deve ser usado para definir a estrutura do documento, e não a sua apresentação.
 - **title:** define um texto auxiliar do elemento, que geralmente fornece informações adicionais sobre ele, e é apresentado como uma tooltip, que é revelada ao passar o mouse sobre o elemento.
 - **contenteditable:** define o conteúdo do elemento como editável.
 - **draggable:** permite que um elemento seja arrastado.



- O atributo global **class**
 - O atributo global **class** é uma lista das classes que um elemento possui, separadas por espaços.
 - Vários elementos HTML podem compartilhar a **mesma** classe.
 - Um mesmo elemento HTML pode pertencer a **mais de uma** classe.
 - O atributo **class** é frequentemente usado para **apontar** para um nome de classe em uma **folha de estilo**.
 - Ele também pode ser usado pelo **JavaScript** para acessar e manipular elementos com o nome de classe específico.



- **Classes HTML**

- Classes permitem que elementos HTML possam ser identificados e acessados posteriormente por seletores de classe CSS e funções ou métodos DOM do *JavaScript*.
- Apesar da especificação não impor requisitos sobre os nomes de classes, é considerada **boa prática** usar nomes que descrevam o **propósito** semântico do elemento, em vez de sua representação. Exemplo: *atributo* para descrever um atributo em vez de *itálico*, mesmo que um elemento desta classe possa ser representado por *itálico*.
- Nomes baseados em semântica permanecem coerentes mesmo que a representação da página mude.

- **Exemplo 1:**

- No exemplo a seguir, temos três elementos **<div>** com um atributo de classe com o valor "**city**". Todos os três elementos **<div>** serão estilizados igualmente de acordo com a definição de estilo **.city** na seção head:

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  border: 2px solid black;
  margin: 20px;
  padding: 20px;
}
</style>
</head>
<body>

<div class="city">
  <h2>London</h2>
  <p>London is the capital of England</p>
</div>

<div class="city">
  <h2>Paris</h2>
  <p>Paris is the capital of France</p>
</div>

<div class="city">
  <h2>Tokyo</h2>
  <p>Tokyo is the capital of Japan</p>
</div>

</body>
</html>
```


- **Exemplo 2:**

- No exemplo a seguir, temos dois elementos **** com um atributo class com o valor de "note". Ambos os elementos **** terão o mesmo estilo de acordo com a definição de estilo **.note** na seção **head**:

```
<!DOCTYPE html>
<html>
<head>
<style>
.note {
  font-size: 120%;
  color: red;
}
</style>
</head>
<body>

<h1>My <span class="note">Import
ant</span> Heading</h1>
<p>This is
some <span class="note">importan
t</span> text.</p>

</body>
</html>
```

- **Dica:** O atributo **class** pode ser usado em qualquer elemento HTML.
- **Obs.:** o nome da classe diferencia maiúsculas de minúsculas!

- **Sintaxe para o CSS**

- Para declarar uma classe no CSS escrevemos um ponto (.), seguido por um nome de classe. Em seguida, definimos as propriedades CSS entre chaves {}:

```
<style>
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>
```

- **Múltiplas classes**
 - Elementos HTML podem pertencer a mais de uma classe.
 - Para definir várias classes, separe os nomes das classes com um espaço, por exemplo:
`<div class = "city main">`
 - O elemento será estilizado de acordo com todas as classes especificadas.
 - No exemplo a seguir, o primeiro elemento **`<h2>`** pertence à classe **city** e também à classe **main**, e obterá os estilos CSS de ambas as classes:

```
<h2 class="city main">London</h2>
```

```
<h2 class="city">Paris</h2>
```

```
<h2 class="city">Tokyo</h2>
```



- Elementos **diferentes** podem compartilhar a **mesma** classe
 - Elementos HTML diferentes podem apontar para o mesmo nome de classe.
 - No exemplo a seguir, `<h2>` e `<p>` apontam para a classe "**city**" e compartilharão o **mesmo** estilo:

```
<h2 class="city">Paris</h2>  
<p class="city">Paris is the capital of France</p>
```

- **Uso do atributo de classe em JavaScript**
 - O nome da classe também pode ser usado pelo *JavaScript* para realizar certas tarefas para elementos específicos.
 - O JavaScript pode acessar elementos com um nome de classe específico com o método:
getElementsByClassName()

```
<script>
function myFunction() {
  var x = document.getElementsByClassName("city");
  for (var i = 0; i < x.length; i++) {
    x[i].style.display = "none";
  }
}
</script>
```

- **Atributo HTML id**

- O atributo HTML **id** é usado para especificar um **id** único para um elemento HTML.
- Você não pode ter mais de um elemento com o mesmo **id** em um documento HTML.

- **Usando o atributo id**

- O atributo **id** é usado para apontar para uma declaração de estilo específica em uma folha de estilo.
- Ele também é usado pelo JavaScript para acessar e manipular o elemento com o **id** específico.
- A sintaxe para **id** é: escreva um caractere hash (#), seguido por um nome de id. Em seguida, defina as propriedades CSS entre chaves {}.
- No exemplo a seguir, temos um elemento **<h1>** que aponta para o nome de id "**myHeader**". Este elemento **<h1>** será estilizado de acordo com a definição de estilo **#myHeader** na seção head.

Nota: O nome do **id** diferencia maiúsculas de minúsculas, deve conter pelo menos um caractere e **não** deve conter espaços em branco (espaços, tabulações, etc.).

- **Exemplo:**

```
<!DOCTYPE html>
<html>
<head>
<style>
#myHeader {
    background-color: lightblue;
    color: black;
    padding: 40px;
    text-align: center;
}
</style>
</head>
<body>

<h1 id="myHeader">My Header</h1>

</body>
</html>
```

- **Diferença entre classe e ID**

- Um nome de **classe** pode ser usado por **vários** elementos HTML, enquanto um nome de **id** deve ser usado apenas por **um** elemento HTML na página.

- **Marcadores HTML com ID e *links***

- Os **marcadores** HTML são usados para permitir que os leitores acessem partes específicas de uma página da web.
- Os marcadores podem ser úteis se sua página for muito longa.
- Para usar um marcador, você deve primeiro criá-lo e, em seguida, adicionar um *link* para ele.
- Dessa forma, quando o *link* for clicado, a página rolará para o local com o **marcador**.

- **Exemplos:**

- Primeiro, criamos um marcador com o atributo **id**..

```
<h2 id="C4">Chapter 4</h2>
```

- Em seguida, adicionamos um *link* para o marcador ("*Pule para o Capítulo 4*"), dentro da mesma página:

```
<a href="#C4">Jump to Chapter 4</a>
```

- Ou adicione um link para o favorito ("*Pule para o Capítulo 4*"), de outra página:

```
<a href="html_demo.html#C4">Jump to Chapter 4</a>
```

- **Usando o atributo id em JavaScript**

- O atributo **id** também pode ser usado pelo JavaScript para realizar algumas tarefas para aquele elemento específico.
- O JavaScript pode acessar um elemento com um **id** específico com o método **getElementById()**:

```
<script>
function displayResult() {
    document.getElementById("myHeader").innerHTML = "Have a nice day!";
}
</script>
```

- **Iframes HTML**

- Um **iframe HTML** é usado para exibir uma página da web dentro de uma página da web.
- A tag HTML **<iframe>** especifica um quadro embutido.
- Um quadro embutido é usado para incorporar outro documento ao documento HTML atual.

- **Sintaxe:**

```
<iframe src="url" title="description">
```

Dica: É uma boa prática sempre incluir um atributo ***title*** para o `<iframe>`. Isso é usado por leitores de tela para ler qual é o conteúdo do iframe.



- **Iframes** - Remover a borda
 - Por padrão, um *iframe* tem uma **borda** ao seu redor. Para remover a borda, adicione o atributo *style* e use a propriedade **CSS border**.
 - Com CSS, você também pode alterar o tamanho, estilo e cor da borda do iframe.
 - Um iframe pode ser usado como o quadro de destino de um link.

```
<iframe  
  src="demo_iframe.htm"  
  style="border:none;"  
  title="Iframe Example">  
</iframe>
```



- **O elemento HTML Head**

- O elemento HTML **<head>** providencia informações gerais (**metadados**) sobre o documento, incluindo seu **título** e links para **scripts** e **folhas de estilos**.
- O elemento HTML `<head>` é um contêiner para os seguintes elementos: `<title>`, `<style>`, `<meta>`, `<link>`, `<script>` e `<base>`.
- Os metadados HTML são dados sobre o documento HTML. Os metadados **não são exibidos**.
- Os metadados normalmente definem o título do documento, conjunto de caracteres, estilos, scripts e outras metainformações.

- O elemento HTML `<link>`

- O elemento `<link>` define o relacionamento entre o documento atual e um recurso externo. A tag `<link>` é mais frequentemente usada para vincular a folhas de estilo externas:

```
<link rel="stylesheet" href="mystyle.css">
```

- O elemento HTML `<meta>`

- O elemento `<meta>` é normalmente usado para especificar o **conjunto de caracteres**, a descrição da página, as **palavras-chave**, o autor do documento e as configurações da janela de visualização (**viewport**).
- Os metadados não serão exibidos na página, mas são usados por navegadores (como exibir conteúdo ou recarregar a página), por mecanismos de pesquisa (palavras-chave) e outros serviços da web.

- **Exemplos**

- Define o conjunto de caracteres usado:

```
<meta charset="UTF-8">
```

- Define um conjunto de palavras-chave para mecanismos de busca:

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

- Define uma descrição para a página web:

```
<meta name="description" content="Free Web tutorials">
```

- Recarrega o documento a cada 30 segundos:

```
<meta http-equiv="refresh" content="30">
```

- Ajusta a janela de visualização para escalar conforme o tamanho do dispositivo:

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

- **Configurando a janela de visualização**

- A janela de visualização (**viewport**) é a área visível do usuário em uma página da web. Isso varia de acordo com o dispositivo - será menor em um telefone celular do que na tela de um computador.
- Devemos incluir o seguinte elemento **<meta>** em todas as suas páginas da web:

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

- Isso fornece ao navegador instruções sobre como controlar as dimensões e a escala da página.
- A parte **width=device-width** define a largura da página para seguir a largura da tela do dispositivo (que irá variar dependendo do dispositivo).
- A parte **initial-scale=1.0** define o nível de zoom inicial quando a página é carregada pela primeira vez pelo navegador.

- O elemento HTML **<base>**

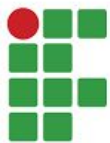
- O elemento **<base>** especifica o URL base para todos os URLs relativos em uma página.
- A tag **<base>** deve ter um **href** ou um atributo **target** presente, ou ambos.
- Só pode haver um único elemento **<base>** em um documento!
- Exemplo:

```
<head>
  <base href="https://www.w3schools.com/" target="_blank">
</head>

<body>
  
  <a href="tags/tag_base.asp">HTML base Tag</a>
</body>
```

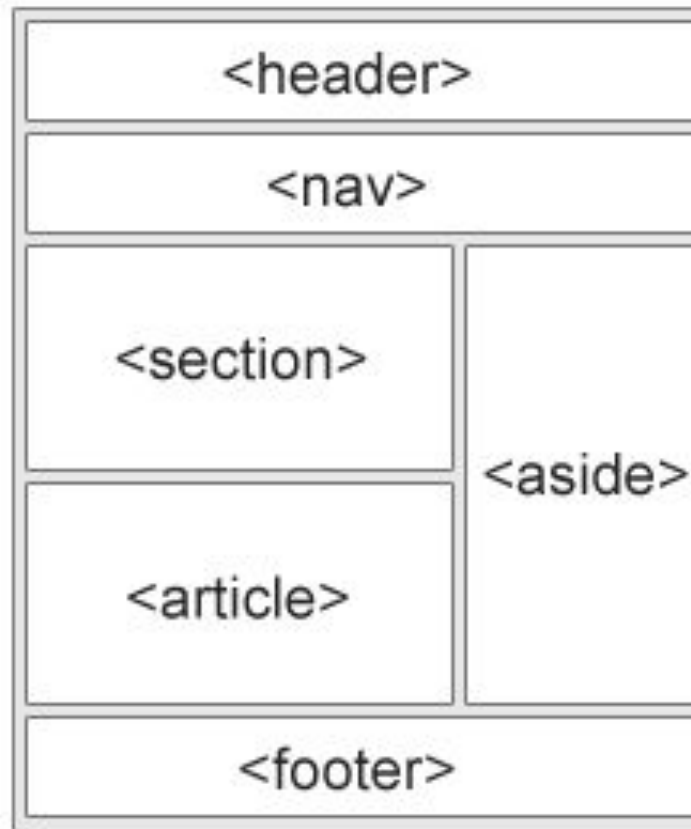
- **Elemento semânticos HTML**

- O que são elementos semânticos?
 - Um elemento semântico descreve claramente seu significado para o navegador e o desenvolvedor.
 - Exemplos de elementos **não** semânticos: <div> e - Não dizem nada sobre seu conteúdo. Exemplos de elementos **semânticos**: <form>, <table> e <article> - definem claramente seu conteúdo.
- Muitos sites contêm código HTML como:
 - <div id = "nav">
 - <div class = "header">
 - <div id = "footer">
 - Que indicam navegação, cabeçalho e rodapé. Em HTML, existem alguns elementos semânticos que podem ser usados para definir diferentes partes de uma página da web.



- **Elemento semânticos HTML**

- `<article>`
- `<aside>`
- `<details>`
- `<figcaption>`
- `<figure>`
- `<footer>`
- `<header>`
- `<main>`
- `<mark>`
- `<nav>`
- `<section>`
- `<summary>`
- `<time>`



- **Elemento semânticos HTML**

- **<article>** - O elemento <article> especifica conteúdo independente e autocontido. Um artigo deve fazer sentido por si só e deve ser possível distribuí-lo independentemente do resto do site.
- Exemplos de onde um elemento <article> pode ser usado:
 - Postagem de fórum;
 - Postagem de blog;
 - Artigo de jornal;
- **<header>** - O elemento <header> representa um contêiner para conteúdo introdutório ou um conjunto de *links* de navegação.
- Um elemento <header> normalmente contém:
 - um ou mais elementos de título (<h1> - <h6>)
 - informações de autoria
 - logotipo ou ícone
- **Nota:** Podemos ter vários elementos <header> em um documento HTML. No entanto, <header> **não** pode ser colocado **dentro** de um <footer>, <address> ou outro elemento <header>.

- **Elemento semânticos HTML**

- **<footer>** - O elemento <footer> define um rodapé para um documento ou seção. Um elemento <footer> normalmente contém:
 - informações de autoria
 - informações de copyright
 - informações de contato
 - mapa do site
 - voltar ao início
 - *links* de documentos relacionados
- Podemos ter vários elementos <footer> em um documento. Exemplo:

```
<footer>  
  <p>Author: Hege Refsnes</p>  
  <p><a href="mailto:hege@example.com">hege@example.com</a></p>  
</footer>
```

- **Elemento semânticos HTML**

- **<figure>** - A tag <figure> especifica conteúdo independente, como ilustrações, diagramas, fotos, listagens de código, etc.
- A tag **<figcaption>** define uma legenda para um elemento <figure>. O elemento <figcaption> pode ser colocado como o primeiro ou o último filho de um elemento <figure>.
- O elemento define a imagem ou ilustração em si.

```
<figure>  
    
  <figcaption>Fig1. - Trulli, Puglia, Italy.</figcaption>  
</figure>
```

- **Por que elementos semânticos?**

- De acordo com o W3C: "Uma Web semântica permite que os dados sejam compartilhados e reutilizados em aplicativos, empresas e comunidades." Abaixo está uma lista de alguns dos elementos semânticos mais comuns em HTML:
 - **<article>** - Define o conteúdo independente e autocontido
 - **<aside>** - Define o conteúdo além do conteúdo da página
 - **<details>** - Define detalhes adicionais que o usuário pode visualizar ou ocultar
 - **<figcaption>** - Define uma legenda para um elemento <figure>
 - **<figure>** - Especifica o conteúdo autocontido, como ilustrações, diagramas, fotos, listagens de código, etc.
 - **<footer>** - Define um rodapé para um documento ou seção
 - **<header>** - Especifica um cabeçalho para um documento ou seção
 - **<main>** - Especifica o conteúdo principal de um documento
 - **<mark>** - Define texto destacado
 - **<nav>** - Define links de navegação
 - **<section>** - Define uma seção em um documento
 - **<summary>** - Define um título visível para um elemento <details>
 - **<time>** - Define uma data/hora

Desenvolvimento Web I

Resumo

- **Fundamentos de HTML**
 - Atributos globais;
 - Classes;
 - Id;
 - Iframes;
 - Head;
 - Elementos semânticos.





Referências Básicas

SILVA, Maurício Samy. CSS3: desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3. São Paulo: Novatec, 2012.

SILVA, Maurício Samy. HTML 5: a linguagem de marcação que revolucionou a web. São Paulo: Novatec, 2011.

NIEDERAUER, Juliano. Desenvolvendo websites com PHP: aprenda a criar websites dinâmicos e interativos com PHP e bancos de dados. 2. ed. rev. e atual. São Paulo: Novatec, 2011.

Referências Complementares

FLANAGAN, David. **o guia definitivo**. . O Really. 2012

SILVA, Maurício Samy. **Criando sites com HTML: sites de alta qualidade com HTML e CSS**. . Novatec. 2010

SOARES, Wallace. **PHP 5: conceitos, programação e integração com banco de dados**. . Érica. 2010

DALL'OGGIO, Pablo. **PHP: programando com orientação a objetos**. . Novatec. 2009

DEITEL, Paul J. Ajax,. **Rich Internet applications e desenvolvimento Web para programadores**. . Pearson Prentice Hall. 2009

IEPSEN, Edécio Fernandes. **Lógica de Programação e Algoritmos com JavaScript**. Novatec. 2018.

Referências na Internet

<https://www.w3schools.com>

<https://developer.mozilla.org/pt-BR/docs/Web>

<https://illustrated.dev/advancedjs>