



Introdução à linguagem CSS



**Certified
Developer**
The Ultimate Tech Degree

DigitalHouse >
Coding School



Temas

1

**Definição e
sintaxe**

2

CSS e HTML

3

Seletores

4

**Usando o Inspetor
de propriedades**

5

**Propriedades
básicas**

1 | Definição e sintaxe

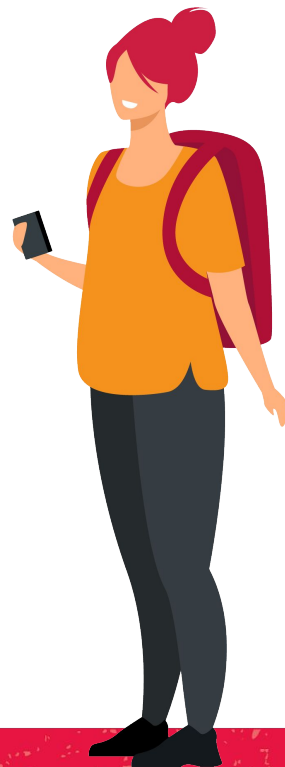


O que significa CSS?

CSS

Cascading
Style
Sheets

Folhas
de estilo
em cascata





Para que servem as folhas de estilo?



As folhas de estilo servem para **estilizar** o nosso conteúdo HTML. Com o CSS, podemos modificar as cores, o background, as tipografias, larguras, alturas... Também podemos gerar animações e transições.



2

CSS e HTML



Como adicionar estilo à página HTML?

HTML



CSS



Contamos com **3 métodos** para vincular nossos arquivos CSS com o documento HTML.



Vinculação externa

Essa é a mais utilizada, já que escrevemos todos os nossos estilos em um arquivo CSS. Feito isso, vinculamos esse arquivo usando a tag `<link>` dentro do `<head>` do nosso documento.



Atributo + valor

Indica a relação entre os documentos a serem ligados.
O valor é sempre o mesmo.

html

```
<link href="css/estilos.css" rel="stylesheet">
```

Atributo + valor

Rota de localização da minha folha de estilo.



Vinculação interna

É feita através da tag `<style>` dentro do `<head>`.

NÃO RECOMENDADA 😞👎



html

```
<head>
  <meta charset="UTF-8">
  <style>
    body {background: blue}
  </style>
</head>
```



Vinculação em linha

Usando o atributo `style` em cada elemento do nosso HTML.

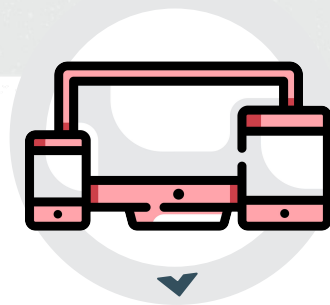
NÃO RECOMENDADA 😞👎



```
html
```

```
<p style="color: red">...</p>
```

3 | Seletores



O que são seletores no CSS?

Trabalhando com vinculação externa, precisamos selecionar o elemento sobre o qual desejamos aplicar uma propriedade de estilo.

Seletor

Indica sobre qual ou quais elementos aplicaremos a propriedade.

```
css  body { font-family: sans-serif }
```

Propriedade

Define qual propriedade do(s) elemento(s) vamos modificar.

Valor

Especificará o comportamento que a propriedade vai ter.

Seletor Universal

O seletor universal irá selecionar todos os elementos HTML.

Para chamá-lo a partir do CSS, usamos o `*`.

css

```
* { color: blue }
```

Seletores de elemento

O seletor de elemento vai chamar o elemento HTML com o mesmo nome da **tag** que chamamos do nosso CSS.

html

```
<p>Primeiro parágrafo</p>
```

Para chamá-lo a partir do CSS, usamos o **nome da tag**.

css

```
p { color: gray }
```

Seletores de classe

O seletor de classe irá chamar o elemento HTML que é associado a um atributo **class**.

Podemos atribuir tantas classes quanto quisermos a um mesmo elemento HTML. Para fazer isso, só precisamos separá-los com um espaço.

```
html <h3 class="titulo noticias">Notícias</h3>
```

Para selecionar uma classe a partir do CSS, usamos o `.` seguido do **nome da classe**.

```
css .titulo { font-size: 22px }
```

Seletores de ID

Este seletor irá selecionar o elemento HTML que é associado ao atributo **id**.
Recomenda-se a utilização de nomes únicos para cada id, sem repetí-los ao longo do documento.

```
html    <h3 id="saudacao">Olá!</h3>
```

Para chamá-lo a partir do CSS, usamos o # seguido do **nome do ID**.

```
css     #saudacao { color: blue }
```


Seletores de grupos

É possível aplicar uma mesma regra a vários elementos, classes e ids de uma só vez.

html

```
<h2>Subtítulo em amarelo</h2>  
<p>Parágrafo em amarelo também</p>
```

Para chamá-los a partir do CSS, você só precisa adicionar **os seletores separados por vírgula**, cada um com a sintaxe correspondente (id, classe ou elemento).

css

```
h2, p { color: yellow }
```

Seletores descendentes

Podemos selecionar certos elementos que sejam descendentes de algum outro elemento especificado anteriormente.

Observe os diferentes elementos `a` a seguir:

html

```
<nav>
  <strong><a href="www.link1.com">Primeiro link</a></strong>
  <a href="www.link2.com">Segundo link</a>
</nav>

<p>Parágrafo com um <a href="www.link3.com">terceiro link</a> </p>
```

Seletores de todos os descendentes

Para seleccionar qualquer descendente a partir do CSS, escrevemos o **seletor ancestral seguido do seletor descendente, separados por um espaço**.

A seguir, estamos seleccionando apenas os elementos `a` que sejam descendentes do elemento `nav` - inclusive os indiretos, se houver!

```
css    nav a { color: blue }
```

Aplicando a propriedade acima ao código HTML anterior, apenas o primeiro e o segundo link serão estilizados com a cor azul.



Seletores de descendentes diretos

Para seleccionar apenas descendentes diretos, escrevemos o **seletor ancestral seguido do seletor descendente, separados por um `>`**.

A seguir, estamos seleccionando apenas os elementos `a` que sejam descendentes diretos do elemento `nav`.

```
css    nav > a { color: blue }
```

Aplicando a propriedade acima ao código HTML anterior, apenas o segundo link será estilizado com a cor azul.



Seletores adjacentes

Podemos ainda selecionar um elemento que seja adjacente de algum outro elemento especificado anteriormente. Observe os diferentes elementos `p` a seguir:

html

```
<div>
  <p>Primeiro parágrafo</p>
</div>

<p>Segundo parágrafo</p>

<p>Terceiro parágrafo</p>
```

Seletores de todos os adjacentes

Para seleccionar todos os elementos adjacentes, escrevemos o **seletor anterior seguido do seletor adjacente, separados por um ~**.

A seguir, estamos seleccionando todos os elementos `p` que sejam adjacentes a um elemento `div`:

```
css  div ~ p { color: blue }
```

Aplicando a propriedade acima ao código HTML anterior, ambos o segundo parágrafo e o terceiro parágrafos serão estilizados com a cor azul.



Seletor de adjacente direto

Para selecionar apenas o primeiro elemento adjacente, escrevemos o **seletor anterior seguido do seletor adjacente, separados por um +**.

A seguir, estamos selecionando apenas o elemento `p` que seja imediatamente adjacente ao elemento `div`:

```
css    div + p { color: blue }
```

Aplicando a propriedade acima ao código HTML anterior, apenas o segundo parágrafo será estilizado com a cor azul.



Seletores combinados

Estes seletores afetam o(s) elemento(s) que cumpre(m) todas as condições estabelecidas. No exemplo a seguir, vamos chamar o elemento **h2** designado para a classe `subtitulo`.

html

```
<h2>Subtítulo sem estilo</h2>  
<h2 class="subtitulo">Subtítulo em vermelho</h2>
```

Para chamá-los a partir do CSS, você só precisa adicionar **um seletor ao lado do outro**, sem separadores, cada um com a sintaxe correspondente (id, classe ou tag).

css

```
h2.subtitulo { color: red }
```




O CSS **sempre vai priorizar** os seletores mais específicos para aplicar os estilos.



4

**Usando o Inspetor de
propriedades**



O que é inspeção de código?

HTML



CSS

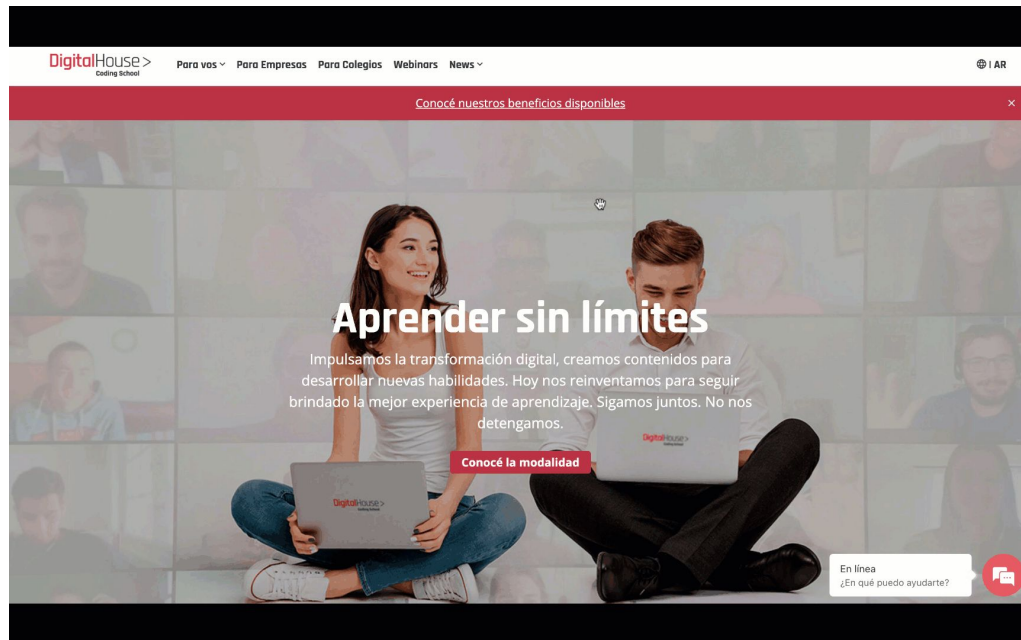


Ao acessar ou visualizar uma página web, podemos habilitar o inspetor do navegador para examinar e modificar o código HTML e CSS.



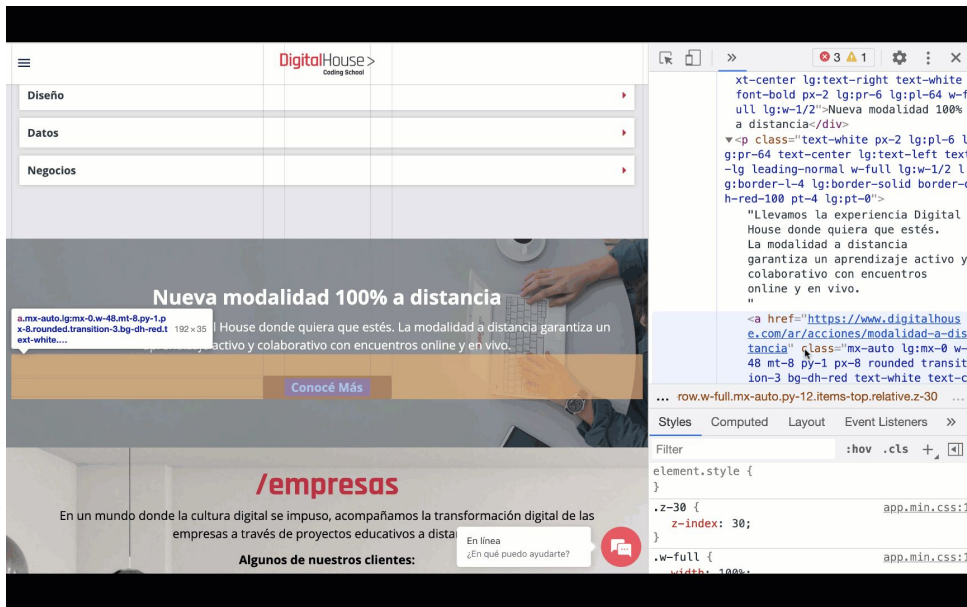


Basta acionar a tecla F12 ou clicar com o botão direito do mouse em qualquer área da página e selecionar a opção “Inspecionar”.





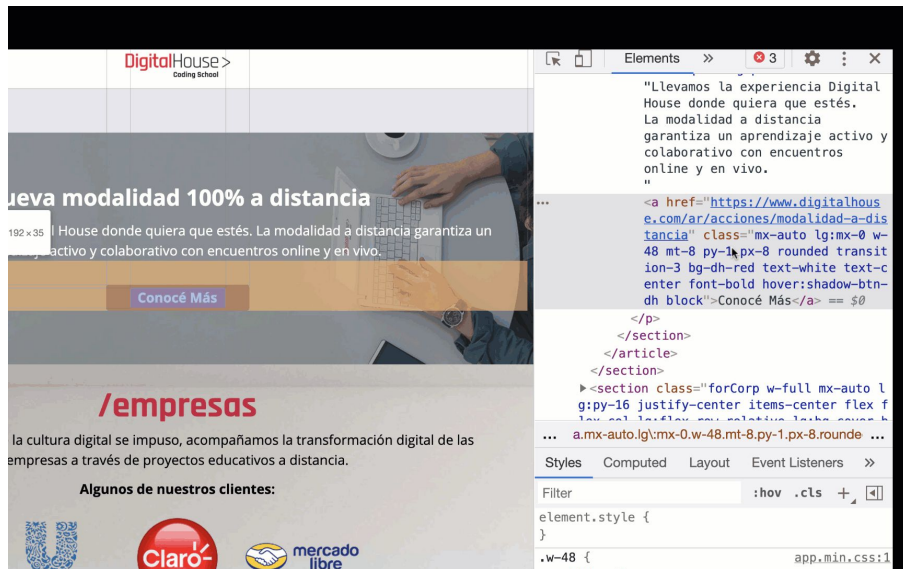
Ao posicionar o cursor no código HTML, o elemento correspondente é destacado.





Podemos editar o código em
inspeção e visualizar as
alterações imediatamente.

**Essas edições encontram-se
apenas em nosso navegador
e são perdidas ao recarregar
a página.**



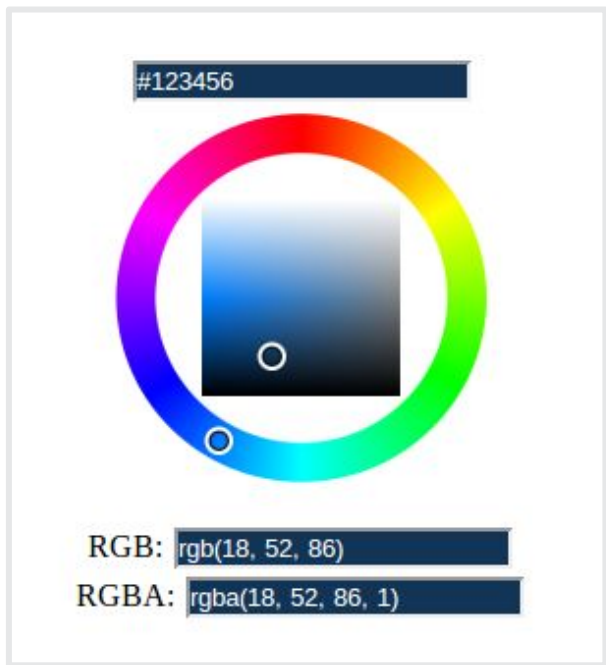


Podemos ver e modificar também todos os estilos criados, inclusive os que não estão sendo aplicados ou sejam inválidos. Isso pode nos ajudar a entender porque certos estilos não estão se aplicando.

Styles	Computed	Layout	Event Listeners	DOM Breakpoints	>>
Filter				:hov .cls	+
element.style {					
}					
.w-48 {					app.min.css:1
width: 12rem;					
}					
.text-white {					app.min.css:1
color: #fff;					
}					
.text-center {					app.min.css:1
text-align: center;					
}					
.px-8 {					app.min.css:1
padding-left: 2rem;					
padding-right: 2rem;					
}					
.py-1 {					app.min.css:1
padding-top: .25rem;					
padding-bottom: .25rem;					
}					

5 | Propiedades básicas

Formatos de cor



Os seguintes formatos podem ser aplicados a qualquer propriedade CSS que receba cor:

Nome (podemos identificar as cores pelo seu nome em inglês, veja todos [aqui](#)).

Ex: purple

RGB

rgb(255, 100, 50)

RGBA

rgba(122, 50, 125, **0.5**).

O último *número* representa a *opacidade* que o elemento terá e varia de 0 a 1. Quanto menor o número, maior a transparência.

Hexadecimal

#f05331

Cor do texto

O atributo `color` nos permite atribuir uma **cor ao texto de um elemento**.
Recebe como valor qualquer dos formatos de cor permitidos.

```
css    h2 { color: purple }
```

```
css    h2 { color: #3459ff }
```

```
css    p { color: rgb(12, 34, 32) }
```

Cor de fundo

O atributo `background-color` nos permite atribuir uma cor para o fundo de um elemento.

Ele recebe como valor qualquer um dos formatos de cor permitidos (como vimos anteriormente).

```
css  p { background-color: tomato }
```

```
css  p { background-color: #3459ff }
```

```
css  p { background-color: rgb(12, 34, 32) }
```

DigitalHouse>
Coding School