



SELETORES

Lembre-se que as declarações são uma parte fundamental do CSS porque aplica um estilo a um elemento selecionado.

Mas como você decide quais elementos terão o estilo? **Com seletor.**

Um seletor é usado para direcionar os elementos HTML específicos a serem estilizados pela declaração.

Um seletor com o qual você já deve estar familiarizado é o seletor de tipo. Assim como o nome sugere, o seletor de tipo corresponde ao tipo do elemento no documento HTML.

Exemplo:

```
p {  
  color: green;  
}
```

Esta é uma instância do uso do seletor de tipo! O tipo de elemento é `p`, que vem da tag `<p>` HTML.

Como os tipos de elemento são frequentemente referidos pelo nome da tag de abertura, o seletor de tipo às vezes é chamado de nome da tag ou seletor de elemento.

Universal

Você aprendeu como o seletor de tipo seleciona todos os elementos de um determinado tipo. Bem, o seletor universal seleciona todos os elementos de qualquer tipo.

O seletor universal usa o `*` caractere no mesmo lugar onde você especificou o seletor de tipo em um conjunto de regras, assim:

```
* {  
  font-family: Verdana;  
}
```

No código acima, cada elemento de texto na página terá sua fonte alterada para Verdana.

Class

O CSS não se limita a selecionar elementos por seu tipo. Como você sabe, os elementos HTML também podem ter atributos. Ao trabalhar com HTML e CSS, um atributo de classe é uma das formas mais comuns de selecionar um elemento.

Por exemplo, considere o seguinte HTML:

```
<p class='descodificadas'>Trilha 1 </p>
```

O elemento de parágrafo no exemplo acima tem um `class` atributo dentro da tag de abertura do `<p>` elemento. O `class` atributo é definido como `'descodificadas'`.

Para selecionar este elemento usando CSS, podemos criar um conjunto de regras com um seletor de classe de *.descodificadas*.

```
.descodificadas {  
  border: 1px black solid;  
}
```

Para selecionar um elemento HTML por sua classe usando CSS, um ponto (.) que deve ser prefixado ao nome da classe. No exemplo acima, a classe é *descodificadas*, então o seletor CSS para ela é *.descodificadas*.

Várias classes

Podemos usar CSS para selecionar o `class` atributo de um elemento HTML pelo nome. E até agora, selecionamos elementos usando apenas um nome de classe por elemento. Se cada elemento HTML tivesse uma única classe, todas as informações de estilo para cada elemento exigiriam uma nova classe.

Felizmente, é possível adicionar mais de um nome de classe ao `class` atributo de um elemento HTML.

Por exemplo, talvez haja um elemento de título que precise ser verde e em negrito.

Você poderia escrever dois conjuntos de regras CSS assim:

```
.green {  
  color: green;  
}
```

```
.bold {
```

```
font-weight: bold;  
}
```

Então, você pode incluir essas duas classes em um elemento HTML como este:

```
<h1 class='green bold'> ... </h1>
```

Podemos adicionar várias classes ao class atributo de um elemento HTML separando-as com um espaço. Isso nos permite misturar e combinar classes CSS para criar muitos estilos exclusivos sem escrever uma classe personalizada para cada combinação de estilo necessária.

ID

Podemos usar CSS para selecionar o id atributo de um elemento HTML pelo nome.

Se cada elemento HTML tivesse uma única classe, todas as informações de estilo para cada elemento exigiriam uma nova classe.

```
<h1 id='large-title'>Descodificadas</h1>
```

Ao contrário de class que aceita vários valores e pode ser usado amplamente em um documento HTML, um elemento id pode ter apenas um único valor e ser usado apenas uma vez por página.

Para selecionar o ID de um elemento com CSS, adicionamos o id nome com um sinal numérico (#).

Por exemplo, se quiséssemos selecionar o elemento HTML no exemplo acima, ficaria assim:

```
#large-title {  
    font-family: Verdana;  
}
```

O id nome é large-title, portanto o seletor CSS para ele é #large-title.

Atributo

Você deve se lembrar de que alguns elementos HTML usam atributos para adicionar detalhes ou funcionalidades extras ao elemento. Alguns atributos familiares podem ser href e src, mas existem muitos mais — incluindo classe id!

O seletor de atributo pode ser usado para direcionar elementos HTML que já contêm atributos. Elementos do mesmo tipo podem ser direcionados de forma diferente por seu atributo ou valor de atributo. Isso alivia a necessidade de adicionar novo código, como os atributos class ou .id.

Os atributos podem ser selecionados de forma semelhante aos tipos, classes e IDs.

```
[href] {  
    color: magenta;  
}
```

A sintaxe mais básica é um atributo entre colchetes. No exemplo acima: [href] segmentaria todos os elementos com um href atributo e definiria color como magenta.

E pode ficar mais específico a partir daí adicionando valores de tipo e/ou atributo.

Uma maneira é usando type[attribute*=value].

Em resumo, este código seleciona um elemento onde o atributo contém qualquer instância do valor especificado.

Vamos dar uma olhada em um exemplo.

```
<img src='/images/seasons/cold/winter.jpg'>  
<img src='/images/seasons/warm/summer.jpg'>
```

O código HTML acima renderiza dois elementos , cada um contendo um src atributo com um valor igual a um link para um arquivo de imagem.

```
img[src*='winter'] {  
  height: 50px;  
}
```

```
img[src*='summer'] {  
  height: 100px;  
}
```

Agora dê uma olhada no código CSS acima. O seletor de atributo é usado para segmentar cada imagem individualmente.

O primeiro conjunto de regras procura um img elemento com um atributo de src que contém a string 'winter' e define height como 50px.

O segundo conjunto de regras procura um img elemento com um atributo de src que contém a string 'summer' e define o height como 100px.

Observe como nenhuma nova marcação HTML (como uma classe ou id) precisou ser adicionada, e ainda conseguimos modificar os estilos de cada imagem de forma independente. Esta é uma vantagem de usar o seletor de atributos!

Pseudo class

Você pode ter observado como a aparência de certos elementos pode mudar, ou estar em um estado diferente, após determinadas interações do usuário.

Por exemplo:

Quando você clica em um elemento `<input>`, uma borda azul é adicionada mostrando que ele está em foco .

Quando você clica em um `<a>` link azul para visitar outra página, mas quando você retorna o texto do link fica roxo.

Quando você está preenchendo um formulário e o botão de envio fica acinzentado e desabilitado. Mas quando todos os campos estiverem preenchidos, o botão fica com a cor indicando que está ativo.

Estes são todos exemplos de seletores de pseudo-classe em ação. Na verdade, `:focus`, `:visited`, `:disabled`, `:hover` e `:active` são todas pseudo classes.

Fatores como interação do usuário, navegação no site e posição na árvore do documento podem dar aos elementos um estado diferente com pseudo classe.

Uma pseudo classe pode ser anexada a qualquer seletor. É sempre escrito como dois pontos `:seguido` de um nome.

Por exemplo `p:hover`.

```
p:hover {  
  background-color: lime;  
}
```

No código acima, sempre que o mouse passar sobre um elemento de parágrafo, este parágrafo terá um plano de fundo cor de limão.

Outros exemplos:

```
input:focus {  
  background-color: yellow;  
}
```

```
a:active {  
  background-color: yellow;  
}
```

```
input[type="text"]:disabled {  
  background: #dddddd;  
}
```

```
a:visited {  
  color: pink;  
}
```

Mais sobre Classes e IDs

O CSS pode selecionar elementos HTML por seu tipo, classe e ID. Classes e IDs têm propósitos diferentes, o que pode afetar qual você usa para estilizar elementos HTML.

As classes CSS devem ser reutilizadas em muitos elementos. Ao escrever classes CSS, você pode estilizar elementos de várias maneiras misturando classes.

Por exemplo, imagine uma página com dois títulos. Um título precisa ser em negrito e azul, e o outro precisa ser em negrito e verde.

Em vez de escrever regras CSS separadas para cada título que repetem o código um do outro, é melhor escrever uma regra CSS `.bold`, outra `.green` e uma `.blue`. Então você pode dar a um título as regras `.bold` e `.green` e a outra as regras `.bold` e `.blue`.

Enquanto as classes podem e devem ser usadas muitas vezes, um ID deve estilizar apenas um elemento.

Os IDs substituem os estilos de tipos e classes. Como os IDs substituem esses estilos, eles devem ser usados com moderação e apenas em elementos que precisam sempre aparecer iguais.

Especificidade

Especificidade é a ordem pela qual o navegador decide quais estilos CSS serão exibidos. Uma prática recomendada em CSS é estilizar elementos usando o menor grau de especificidade para que, se um elemento precisar de um novo estilo, seja fácil de substituir.

ID é o seletor mais específico em CSS, seguido por classes e, por fim, tipo.

Por exemplo, considere o seguinte HTML e CSS:

```
<h1 class='headline'>Decodificadas</h1>
```

```
<h1 >Decodificadas2</h1>
```

```
h1 {  
  color: red;  
}
```

```
.headline {  
  color: firebrick;  
}
```

No código de exemplo acima, a cor do título seria definida como firebrick, pois o seletor de classe é mais específico que o seletor de tipo.

Se um atributo de ID (e seletor) fosse adicionado ao código acima, os estilos no corpo do seletor de ID substituiriam todos os outros estilos do título.

Com o tempo, à medida que os arquivos crescem com o código, muitos elementos podem ter IDs, o que pode dificultar a edição do CSS, pois um novo estilo mais específico deve ser criado para alterar o estilo de um elemento.

Para facilitar a edição de estilos, é melhor estilizar com um seletor de tipo, se possível. Caso contrário, adicione um seletor de classe. Se isso não for específico o suficiente, considere usar um seletor de ID.

Encadeamento

Ao escrever regras CSS, é possível exigir que um elemento HTML tenha dois ou mais seletores CSS ao mesmo tempo.

Isso é feito combinando vários seletores, que chamaremos de encadeamento.

Por exemplo, se houvesse uma classe chamada *special* para o elemento `<h1>`, o CSS ficaria assim:

```
h1.special {  
  
}
```

O código acima selecionaria apenas os elementos `<h1>` com uma classe chamada *special*. Se um elemento `<p>` também tiver uma classe chamada *special*, a regra do exemplo não estilizará o parágrafo.

Combinador Descendente

Além de encadear seletores para selecionar elementos, o CSS também oferece suporte à seleção de elementos aninhados em outros elementos HTML, também conhecidos como descendentes.

Por exemplo, considere o seguinte HTML:

```
<ul class='main-list'>
  <li> ... </li>
  <li> ... </li>
  <li> ... </li>
</ul>
```

Os elementos aninhados `` são descendentes do elemento `` e podem ser selecionados com o combinador descendente da seguinte forma:

```
.main-list li {
}
```

No exemplo acima, `.main-list` seleciona o elemento com a classe `.main-list` (o elemento ``).

Os descendentes `` são selecionados adicionando ao `` o seletor, separados por um espaço. Isso resulta em `.main-list li` como o seletor final.

Selecionar elementos dessa maneira pode tornar nossos seletores ainda mais específicos, garantindo que eles apareçam no contexto que esperamos.

Encadeamento e Especificidade

Adicionar mais de uma tag, classe ou ID a um seletor CSS aumenta a especificidade do seletor CSS.

Por exemplo, considere o seguinte CSS:

```
p {
  color: blue;
```

```
}
```

```
.main p {  
  color: red;  
}
```

Ambas as regras CSS definem como um elemento `<p>` deve parecer. Como `.main p` tem uma classe e um tipo `p` como seletor, apenas os elementos `<p>` dentro do elemento `.main` aparecerão red.

Isso ocorre apesar se houver outra regra mais geral que afirma que os elementos `<p>` devem ser blue.

Vários seletores

Para tornar o CSS mais conciso, é possível adicionar estilos CSS a vários seletores CSS de uma só vez. Isso evita escrever código repetitivo.

Por exemplo, o código a seguir tem atributos de estilo repetitivos:

```
h1 {  
  font-family: Georgia;  
}
```

```
.menu {  
  font-family: Georgia;  
}
```

Em vez de escrever `font-family: Georgia` duas vezes para dois seletores, podemos separar os seletores por uma vírgula para aplicar o mesmo estilo a ambos, assim:

```
h1, .menu {  
    font-family: Georgia;  
}
```

Ao separar os seletores CSS com uma vírgula, tanto os elementos `<h1>` quanto os elementos com a classe `menu` receberão o estilo `font-family: Georgia`.

Em resumo:

- O CSS pode selecionar elementos HTML por tipo, classe, ID e atributo.
- Todos os elementos podem ser selecionados usando o seletor universal.
- Um elemento pode ter diferentes estados usando o seletor de pseudo classe.
- Várias classes CSS podem ser aplicadas a um elemento HTML.
- As classes podem ser reutilizáveis, enquanto os IDs podem ser usados apenas uma vez.
- Os IDs são mais específicos do que as classes e as classes são mais específicas do que o tipo. Isso significa que os IDs substituirão qualquer estilo de uma classe e as classes substituirão qualquer estilo de um seletor de tipo.
- Vários seletores podem ser encadeados para selecionar um elemento. Isso aumenta a especificidade, mas pode ser necessário.
- Elementos aninhados podem ser selecionados separando os seletores com um espaço.
- Vários seletores não relacionados podem receber os mesmos estilos separando os nomes dos seletores com vírgulas.

Div: o elemento de divisão do HTML

A tag **Div** é um dos elementos mais conhecidos do HTML. Até o lançamento do HTML5, era muito utilizado para dividir todas as sessões e todos os blocos do

documento. Porém, com a atualização da versão 5 do HTML, começou a dividir o espaço com as demais tags semânticas.

Ainda assim, é um elemento muito importante e muito utilizado.

Para que serve uma Div no HTML

O elemento **div** define uma divisão ou seção em um documento HTML. O elemento **div** é freqüentemente usado como um contêiner para outros elementos, o que facilita na estilização de blocos. Portanto, ele é definido com um contêiner genérico para conteúdo de fluxo.

Diante disso, a **div** é muito utilizada quando precisamos agrupar elementos sem a necessidade do uso de um elemento semântico do HTML5.

Isso acontece pois o elemento **div** não possui um valor semântico. Portanto, não representa nada para o navegador e para os mecanismos de buscas.

Como é muito utilizado para agrupar elementos, acaba sendo um facilitador para organizar informações dentro de um layout.

Dessa forma, é possível formatar e manipular os elementos, inclusive a própria **div**, através do CSS de uma forma organizada. Geralmente é acompanhada de atributos de **ID** e **Class**, para poder facilitar essa organização e formatação.

A sintaxe do elemento **div** é através da tag de abertura **<div>** e de fechamento **</div>**. Por padrão, a **div** gera uma line-break (quebra de linha) automática ao ser utilizada. Isso ocorre pois possui como display padrão um elemento de bloco, ou seja display **block**. Portanto, para utilizar a **div**, basta colocar os elementos que quiser dentro da tag **<div></div>** como na sintaxe abaixo:

<div>

...

</div>

Dessa forma, vamos utilizar algumas divisões para poder separar os parágrafos.

Exemplo:

```
<div>  
<p>Divisão 01 </p>  
</div>  
  
<div>  
<p>Divisão 02 </p>  
</div>  
  
<div>  
<p>Divisão 03 </p>  
</div>
```

Claro que podemos incluir quaisquer outros elementos HTML, ou seja, outras tags, dentro de uma div.

Atributos da div

O elemento div costuma ter entre seus principais atributos:

- **id** – Define uma ID;
- **class** – Define uma classe;
- **title** – Define um título;
- **height** – Define uma altura;
- **width** – Define uma largura.

Porém, com exceção do ID, da class e do title, todos os demais atributos já são considerados depreciados. Portanto, dessa forma, podemos definir a altura, a largura e a cor de fundo através do próprio estilo CSS, assim como diversas outras características.

Veja o exemplo do uso da div abaixo com id e class:

```
<div id="cabecalho">
```

```
<h2>Meu documento sobre Div HTML</h2>
</div>
<div id="corpo">
  <div class="sessao_conteudo">
    <h3>Meu conteúdo 01</h3>
    <p>Meu paragrafo contendo texto</p>
    <p>Meu paragrafo contendo texto</p>
  </div>
  <div class="sessao_conteudo">
    <h3>Meu conteúdo 02</h3>
    <p>Meu paragrafo contendo texto</p>
    <p>Meu paragrafo contendo texto</p>
  </div>
</div>
<div id="rodape">
  <p>Todos os direitos reservados</p>
</div>
```

Observe que nosso documento HTML foi dividido com três divs pais, cada uma contendo um **id** única (**cabecalho**, **corpo** e **rodape**). Dentro da nossa div com a id **corpo**, recebemos duas divisões para o conteúdo. Observe que essas divisões receberam uma mesma classe: **sessao_conteudo**. Portanto, podemos estilizar as divs através de suas ids, e ainda podemos estilizar todas as divs que contêm a mesma classe de uma só vez. Para IDs, referenciamos no CSS através do “#”, por exemplo: **#cabecalho**. Já para classes, reconhecemos através de um “.”, ou seja, conforme este exemplo: **“.sessao_conteudo”**.

Vejamos então o seguinte exemplo de estilo CSS para o documento apresentado no exemplo anterior:


```
<style>
  #cabecalho {
    background-color: #cccccc;
    width: 600px;
    padding: 40px 2%;
    color: white;
    text-align: center;
  }
  #corpo {
    background-color: red;
    width: 600px;
    padding: 10px 2%;
  }
  #rodape {
    width: 600px;
    padding: 10px 2%;
    background-color: #888888;
    color: white;
    text-align: center;
  }
  .sessao_conteudo {
    text-align: center;
    width: 100%;
    background-color: green;
  }
</style>
```

Saiba mais:

https://www.w3schools.com/cssref/css_selectors.asp

<https://www.maujor.com/tutorial/guia-completo-seletores-css3.php>

https://youtu.be/zXfTjPrMC_0

<https://youtu.be/WPtRX4n0UIs>

Seletores para links

<https://www.maujor.com/tutorial/csslinks.php#l7>