

CSS – Seletores por Seleção

Um seletor representa uma estrutura. Essa estrutura pode ser usada como uma condição (por exemplo: em uma regra CSS) que determina quais são os elementos casados na árvore do documento pelo seletor ou uma descrição do fragmento HTML ou XML que corresponda àquela estrutura.

Seletores podem ser desde um simples nome de elemento até uma rica descrição contextual.



CSS – Seletores por Seleção - Usabilidade

O CSS é uma linguagem de criação de estilos e a utilizamos para moldar a aparência dos elementos em uma página web através de regras. Essas regras, por sua vez, são compostas por pares de propriedades e valores que quando aplicadas aos elementos HTML alteram suas características visuais originais.

Ex: `p {color: blue;}` – Onde o “p” é o seletor.

As CSS usam seletores para atrelar propriedades de estilização aos elementos de um documento.



CSS – :root

:root se equipara à raiz de uma árvore, que por sua vez representa o documento. Aplicado ao HTML, :root representa o elemento <html> e é idêntico ao seletor html, exceto que sua especificidade é mais alta.

```
:root {  
  --main-bg-color: brown;  
}
```

Observe o CSS o exemplo acima. A cor de fundo é definida brown em vários lugares. Ao declarar uma propriedade personalizada na :root será usada em todo o documento.



CSS – :nth-child()

:nth-child() seleciona elementos com base em suas posições em um grupo de elementos irmãos.

```
/* Seleciona um a cada quatro elementos  
de qualquer grupo de elementos irmãos,  
começando do quarto elemento (4, 8 12, etc.). */  
:nth-child(4n) {  
  color: lime;  
}
```



CSS – :nth-last-child()

:nth-last-child() seleciona um ou mais elementos com base em sua posição entre um grupo de elementos-irmãos, contando a partir do último.

```
/* Seleciona todo quarto elemento  
em qualquer grupo de irmãos,  
contando do último para o primeiro */  
:nth-last-child(4n) {  
    color: lime;  
}
```

:nth-last-child() é essencialmente o mesmo que usar :nth-child, porém conta os itens de trás para a frente, e não do início para o final.



CSS – :nth-of-type()

:nth-of-type() corresponde a um ou mais elementos de um dado tipo, baseado em sua posição entre um grupo de irmãos.

```
/* Selecciona cada quarto elemento <p>  
   entre qualquer grupo de irmãos */  
p:nth-of-type(4n) {  
    color: lime;  
}
```

:nth-of-type é especificada com um único argumento, o qual representa o padrão para a correspondência dos elementos.



CSS – :nth-last-of-type()

:nth-last-of-type() combina elementos de um determinado tipo, com base em sua posição entre um grupo de irmãos, contando a partir do final.

```
/*Seleciona cada quarto elemento <p>  
entre qualquer grupo de irmãos,  
contando regressivamente a partir do último*/  
p:nth-last-of-type(4n) {  
  color: lime;  
}
```

A nth-last-of-type é especificada com um único argumento, que representa o padrão para combinar elementos, contando a partir do final.



CSS – :last-child

:last-child representa qualquer elemento que é o último filho de seu elemento pai.

```
li:last-child {  
  background-color: lime;  
}
```

- Esse item não é afetado pelo estilo.
- Esse também não.
- **Esse item é! :)**



CSS – :first-of-type

:first-of-type representa o primeiro elemento de seu tipo entre os filhos de seu elemento pai.

```
/* Selecciona qualquer <p> que seja o primeiro elemento  
   desse tipo entre seus irmãos */
```

```
p:first-of-type {  
  color: red;  
}
```

Originalmente definido, o elemento selecionado tinha que ter um pai. Desde o Seletores Nível 4, isso não é mais necessário.



CSS – :last-of-type

:last-of-type representa o último irmão com o nome de elemento dado na lista de filhos de seu elemento pai.

```
p em:last-of-type {  
  color: lime;  
}
```



CSS – :only-child

:only-child representa um elemento sem nenhum elemento-irmão.

```
/* Selecciona cada <p>, mas apenas se for o */  
/* único filho de seu elemento-pai */  
p:only-child {  
    background-color: lime;  
}
```

Originalmente, foi definido que o elemento selecionado deveria ter um elemento-pai.



CSS – Seletores por Seleção

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="seletores.css"/>
    <style>
      div{border:solid 1px; height:100px;}
    </style>
  </head>
  <body>
    <div id="barra_1"></div>
    <div id="barra_2">
      <p>Paragrafo 1</p>
      <p>Paragrafo 2</p>
    </div>
    <div id="barra_3">
      <ul>
        <li>item 1</li>
        <li>item 1</li>
        <li>item 1</li>
      </ul>
    </div>
    <div id="barra_4">
      <p>Paragrafo 3</p>
    </div>
  </body>
</html>
```



CSS – Seletores por Seleção

```
div:nth-child(1){  
  background: red;  
}  
  
div:nth-last-child(2){  
  background: #AB82FF;  
}  
  
li:nth-of-type(1){  
  color: green;  
}  
  
li:nth-last-of-type(1){  
  color: yellow;  
}  
  
div:last-child{  
  background: orange;  
}  
  
div:first-of-type{  
  border: solid 5px green;  
}  
  
div:last-of-type{  
  border: solid 5px brown;  
}  
  
p:only-child{  
  font-size: 30px;  
}  
  
ul:only-of-type{  
  text-decoration: underline;  
}
```



CSS – Seletores por Estado

Estes, por sua vez, funcionam com base no estado atual do controle (vazio, habilitado, desabilitado etc.). A seguir, cada um é explicado, facilitando o entendimento.



CSS – Seletores por Estado - Usabilidade

Com a chegada das CSS3, aplicar estilos a elementos individualmente se tornou muito mais fácil. Os novos seletores permitem identificar elementos com base em seus atributos, posição em que aparecem no “elemento pai”, estado (habilitado/desabilitado), entre outros.



CSS – Seletores por Estado

- E~F: seleciona todo elemento F que seja precedido por um elemento E.

CSS – :empty

:empty representa qualquer elemento que não tenha filhos. Aqui, filhos podem ser outros elementos ou blocos de texto (incluindo espaços em branco). Comentários e instruções de processamento não são considerados ao se determinar se um elemento é vazio.

```
/* Selecciona qualquer <div> que não tenha conteúdo */  
div:empty {  
    background: lime;  
}
```



CSS – :target

:target representa um único elemento (o elemento alvo) com uma id correspondente ao fragmento da URL.

/ Selecciona um elemento com a ID correspondente ao fragmento da URL */*

```
:target {  
  border: 2px solid black;  
}
```

CSS – :enabled

:enabled representa qualquer elemento ativado. Um elemento é ativado se puder ser ativado (selecionado, clicado, digitado etc.) ou aceitar o foco. O elemento também possui um estado desativado, no qual não pode ser ativado nem aceitar foco.

```
/* Selecciona qualquer <input> ativado */  
input:enabled {  
    color: blue;  
}
```

Um exemplo pode ser a troca da cor do texto e do botão <input> verde quando ativada e cinza quando desativada.



CSS – :disabled

:disabled representa qualquer elemento desativado. Um elemento é desativado se não puder ser ativado (selecionado, clicado, digitado etc.) ou aceitar o foco. O elemento também possui um estado habilitado, no qual ele pode ser ativado ou aceitar o foco.

```
/* Selecciona apenas <input> desabilitados */  
input:disabled {  
    background: #ccc;  
}
```

Um exemplo poderá ser o uso de um formulário básico de envio. :disabled poderá mostrar a cor de campos ativos e desativados nos campos de faturamento.



CSS – :checked

:checked de um seletor CSS representa um elemento radio (<input type="radio">), checkbox (<input type="checkbox">) ou option (<option> em um <select>) que está marcada ou alternado para um estado ligado. O usuário pode mudar seu estado clicando no elemento, ou selecionando um valor diferente, nesse caso a pseudo-classe :checked não se aplica a esse elemento, mas vai para um elemento relevante.

```
input:checked {  
  margin-left: 25px;  
  border: 1px solid blue;  
}
```

Exemplo de uso pode ser os todos os botões do tipo radio que estão ativos.



CSS – ::selection

::selection aplica estilos para partes do documento que foram selecionadas pelo usuário (tal como clicar e arrastar o ponteiro do mouse através de um texto).

```
::selection {  
  color: gold;  
  background-color: red;  
}
```

Exemplo de uso poderá de texto selecionado dourado com o fundo vermelho.



CSS – :not(X)

:not(X), é uma notação funcional que recebe um seletor simples X como argumento. Ela seleciona um elemento que não é representado por seu argumento. X não pode conter outro seletor de negação.

```
p:not(.classico) { color: red; }  
body *:not(p) { color: green; }
```

Resultado de uso:

Um pouco de texto.

Um pouco mais de texto.

Mais um texto.



CSS – Seletores por Estado

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="seletores.css"/>
    <style>
      div{border:solid 1px; height:100px;}
    </style>
  </head>

  <body>
    <h1>Selecione esse texto</h1>
    <p><a href="#Texto1">Link para o Texto 1</a></p>
    <p><a href="#Texto2">Link para o Texto 2</a></p>
    <div>
      <p id="Texto1">Esse e o TEXTO 1</p>
      <p id="Texto2">Esse e o TEXTO 2</p>
    </div>
    <div id="divVazia"></div>
    <div>
      <span>Span antes do input</span><br/>
      <input type="checkbox" checked="checked" value="Opcao 1">Opcao 1
      <input type="checkbox" value="Opcao 2">Opcao 2
      <input type="text" disabled="disabled" value="desabilitado"/>
      <input type="text" value="habilitado"/><br/>
      <span>Span apos o input</span>
    </div>
  </body>
</html>
```



CSS – Seletores por Estado

```
:target{
  color: green;
  font-weight: bold;
}

::selection{
  color: red;
}

.empty{
  border: solid 1px;
  height: 50px;
  background: gray;
}

input:checked{
  text-decoration: underline;
}

h1{
  font-family: "Courier New";
}

:not(h1){
  font-family: "Arial";
}

input-span{
  color: orange;
}

input:disabled{
  border: solid 1px red;
  background-color: silver;
  height: 25px;
}

input:enabled{
  border: solid 1px blue;
  background-color: #fff;
  height: 25px;
}
```

