

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.

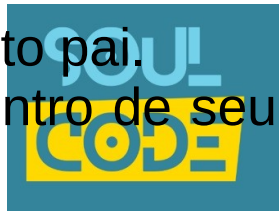
Seletores podem ser desde um simples nome de elemento até uma rica descrição contextual.

A tabela a seguir apresenta um sumário da sintaxe dos seletores:



CSS – Seletores por Seleção

- `:root`: seleciona o elemento raiz do documento, no caso, é a tag HTML.
- `E:nth-child(N)`: seleciona o elemento do tipo E que seja o N-ésimo filho do seu elemento pai.
- `E:nth-last-child(N)`: semelhante ao anterior, mas começando a contagem de baixo para cima.
- `E:nth-of-type(N)`: seleciona o elemento que seja o N-ésimo do seu tipo.
- `E:nth-last-of-type(N)`: semelhante ao anterior, mas contando de baixo para cima.
- `E:last-child`: seleciona o elemento E que seja o último filho do elemento pai.
- `E:first-of-type`: seleciona o elemento E que seja o primeiro de seu tipo dentro do elemento pai.
- `E:last-of-type`: semelhante ao anterior, mas com o último elemento.
- `E:only-child`: seleciona o elemento E que seja o único filho do seu elemento pai.
- `E:only-of-type`: seleciona o elemento que seja o único filho deste tipo dentro de seu 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

- E:empty: seleciona o elemento E que esteja vazio, sem nenhum elemento filho.
- :target: seleciona o objeto alvo de um link interno (aqueles que são precedidos por # e que direcionam o usuário para um elemento na página).
- E:enabled: seleciona o elemento E que esteja habilitado para o usuário
- E:disabled: semelhante ao anterior, mas para elementos desabilitados.
- E:checked: seleciona elementos que estejam marcados (radios e checkboxes).
- ::selection: formata o conteúdo selecionado pelo usuário.
- :not(P): seleciona os elementos que não atendem ao seletor simples P.
- E~F: seleciona todo elemento F que seja precedido por um elemento E.



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>Selezione 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;
}
```



CSS – Font-Size

Píxeis

Definir o tamanho da fonte em valores de pixel (px) é uma boa escolha quando você precisa de precisão de pixel. Um valor de px é estático. Esta é uma maneira independente do sistema operacional e entre navegadores de literalmente dizer aos navegadores para renderizar as letras exatamente no número de pixels de altura que você especificou. Os resultados podem variar ligeiramente entre os navegadores, pois eles podem usar algoritmos diferentes para obter um efeito semelhante.

As configurações de tamanho de fonte também podem ser usadas em combinação. Por exemplo, se um elemento pai for definido como 16px e seu elemento filho for definido como larger, o elemento filho será maior do que o elemento pai na página.



CSS – Font-Size

Ems

Outra forma de definir o tamanho da fonte é com em valores. O tamanho de um em valor é dinâmico. Ao definir a font-size propriedade, um em é igual ao tamanho da fonte que se aplica ao pai do elemento em questão. Se você não definiu o tamanho da fonte em nenhum lugar da página, então é o padrão do navegador, que geralmente é 16px. Portanto, por padrão, 1em = 16px e 2em = 32px. Se você definir font-size 20px no elemento do corpo, 1em = 20px e 2em = 40px. Observe que o valor 2 é essencialmente um multiplicador do tamanho atual em.



CSS – Font-Size

Ems

Para calcular o equivalente em para qualquer valor de pixel necessário, você pode usar esta fórmula:

$$\text{em} = \text{desired element pixel value} / \text{parent element font-size in pixels}$$



CSS – Font-Size

Ems

Por exemplo, suponha que o tamanho da fonte do corpo da página esteja definido como 16 px. Se o tamanho da fonte desejado for 12px, você deve especificar 0,75em (porque $12/16 = 0,75$). Da mesma forma, se você deseja um tamanho de fonte de 10px, especifique 0,625em ($10/16 = 0,625$); para 22px, especifique 1,375em ($22/16$).

O em é uma unidade muito útil em CSS, pois adapta automaticamente seu comprimento em relação à fonte que o leitor escolher usar.



CSS – Font-Size

Ems

Um fato importante a ter em mente: os valores em são compostos. Pegue o seguinte HTML e CSS:

```
html {  
  font-size: 62.5%; /* font-size 1em = 10px on default browser */  
}  
span {  
  font-size: 1.6em;  
}
```

```
<div>  
<span>Outer <span>inner</span> outer</span>  
</div>
```



CSS – Font-Size

Ems

Supondo que o padrão do navegador font-size seja 16px, as palavras “externo” seriam renderizadas em 16px, mas a palavra “interno” seria renderizada em 25,6px. Isso ocorre porque o interior `` é font-size é 1.6em que é em relação ao do seu pai font-size, que por sua vez é relativo a seu pai de font-size. Isso geralmente é chamado de composição .



CSS – Font-Size

Rems

Esses tipos de valores foram inventados a fim de contornar o problema de composição. Rems são relativos ao html elemento raiz, não ao elemento pai. Em outras palavras, ele permite que você especifique um tamanho de fonte de maneira relativa, sem ser afetado pelo tamanho do pai, eliminando assim a composição.



CSS – Font-Size

Rems

O CSS abaixo é quase idêntico ao exemplo anterior. A única exceção é que a unidade foi alterada para rem.

```
html {  
  font-size: 62.5%; /* font-size 1em = 10px on default browser settings */  
}  
span {  
  font-size: 1.6rem;  
}
```



CSS – Font-Size

Rems

Em seguida, aplicamos esse CSS ao mesmo HTML, que se parece com isto:

```
<span>Outer <span>inner</span> outer</span>
```



CSS – Font-Size

Rems

Neste exemplo, as palavras "outer inner outer" são todas exibidas em 16 px (assumindo que o navegador font-size foi deixado com o valor padrão de 16 px).



CSS – Font-Variant

Defina um parágrafo para uma fonte em caixa alta:

```
p.small {  
  font-variant: small-caps;  
}
```



CSS - Font-Variant

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {
  font-variant: normal;
}

p.small {
  font-variant: small-caps;
}
</style>
</head>
<body>
<h1>The font-variant Property</h1>

<p class="normal">My name is Hege Refsnes.</p>
<p class="small">My name is Hege Refsnes.</p>

</body>
</html>
```



CSS - Font-Weight

Defina diferentes espessuras de fonte para três parágrafos:

```
p.normal {  
  font-weight: normal;  
}
```

```
p.thick {  
  font-weight: bold;  
}
```

```
p.thicker {  
  font-weight: 900;  
}
```



CSS - Font-Weight

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {
  font-weight: normal;
}

p.light {
  font-weight: lighter;
}

p.thick {
  font-weight: bold;
}

p.thicker {
  font-weight: 900;
}
</style>
</head>
<body>

<h1>The font-weight Property</h1>

<p class="normal">This is a paragraph.</p>
<p class="light">This is a paragraph.</p>
<p class="thick">This is a paragraph.</p>
<p class="thicker">This is a paragraph.</p>

</body>
</html>
```



CSS – Espaçamento entre Linhas

Defina a altura da linha para diferentes elementos <div>:

```
div.a {  
  line-height: normal;  
}
```

```
div.b {  
  line-height: 1.6;  
}
```

```
div.c {  
  line-height: 80%;  
}
```

```
div.d {  
  line-height: 200%;  
}
```



CSS – Espaçamento entre Linhas

```
<!DOCTYPE html>
<html>
<head>
<style>
div.a {
  line-height: normal;
}

div.b {
  line-height: 1.6;
}

div.c {
  line-height: 80%;
}

div.d {
  line-height: 200%;
}
</style>
</head>
<body>
```

<h1>The line-height Property</h1>

<h2>line-height: normal (default):</h2>

<div class="a">This is a paragraph with a standard line-height.

The standard line height in most browsers is about 110% to 120%.</div>

<h2>line-height: 1.6 (recommended):</h2>

<div class="b">This is a paragraph with the recommended line-height.

The line height is here set to 1.6. This is a unitless value;

meaning that the line height will be relative to the font size.</div>

<h2>line-height: 80%:</h2>

<div class="c">This is a paragraph with a smaller line-height.

The line height is here set to 80%.</div>

<h2>line-height: 200%:</h2>

<div class="d">This is a paragraph with a bigger line-height.

The line height is here set to 200%.</div>

</body>
</html>



CSS – Espaçamento entre Palavras

Ao criar o seu código HTML, você poderá inserir espaços entre os caracteres, palavras, números ou outros elementos. Isso ajudará a organizar melhor o seu documento. No entanto, o navegador só reconhecerá o primeiro espaço e os demais serão ignorados. Se precisar mostrar mais de um espaço, você deverá adicionar um código ou usar uma tag.



CSS – Letter-spacing

Define o espaçamento entre letras para os elementos <h1>, <h2> e <h3>:

```
h1 {  
  letter-spacing: 3px;  
}
```

```
h2 {  
  letter-spacing: 2px;  
}
```

```
h3 {  
  letter-spacing: -1px;  
}
```



CSS – Letter-spacing

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  letter-spacing: 3px;
}

h2 {
  letter-spacing: 2px;
}

h3 {
  letter-spacing: -1px;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>

</body>
</html>
```



CSS – Word-wrap

Especifica que o espaço entre as palavras nos elementos `<p>` deve ser de 30 pixels:

```
p {  
  word-spacing: 30px;  
}
```



CSS – Word-wrap

```
<!DOCTYPE html>
<html>
<head>
<style>
p.a {
  word-spacing: normal;
}

p.b {
  word-spacing: 30px;
}

p.c {
  word-spacing: 1cm;
}
</style>
</head>
<body>

<h1>The word-spacing Property</h1>

<h2>word-spacing: normal:</h2>
<p class="a">This is some text. This is some text.</p>

<h2>word-spacing: 30px:</h2>
<p class="b">This is some text. This is some text.</p>

<h2>word-spacing: 1cm:</h2>
<p class="c">This is some text. This is some text.</p>

</body>
</html>
```



CSS – White-space

Demonstra diferentes valores da propriedade de espaço em branco:

```
p.a {  
  white-space: nowrap;  
}
```

```
p.b {  
  white-space: normal;  
}
```

```
p.c {  
  white-space: pre;  
}
```



CSS – White-space

```
<!DOCTYPE html>
<html>
<head>
<style>
p.a {
  white-space: nowrap;
}
```

```
p.b {
  white-space: normal;
}
```

```
p.c {
  white-space: pre;
}
</style>
</head>
<body>
```

```
<h1>The white-space Property</h1>
```

```
<h2>white-space: nowrap:</h2>
<p class="a">
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
</p>
```

```
<h2>white-space: normal:</h2>
<p class="b">
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
</p>
```

```
<h2>white-space: pre:</h2>
<p class="c">
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
</p>
```

```
</body>
</html>
```

