



Tópicos especiais em CSS

Prof. Guilherme Alves

CSS

O **CSS** é a sigla para o termo em inglês (*Cascading Style Sheets*) que, traduzido para o português, significa folha de estilo em cascatas. Estes códigos podem ser reaproveitados na criação de páginas web sendo utilizados para estilizar elementos.





Specificity

Especificidade é o meio pelo qual os navegadores decidem quais valores de propriedade **CSS** são os mais relevantes para um elemento. Quando várias declarações têm a mesma especificidade, a última declaração encontrada é aplicada ao elemento. Quando o `!important` é utilizado em uma declaração de estilo, essa declaração substitui quaisquer outras declarações. Este por sua vez deve ser utilizado somente em casos muito específicos e deve ser consideradas outras alterações antes desta decisão.

Seletores



```
1  * {  
2      /* Todos os elementos do documento */  
3  }  
4  
5  div {  
6      /* Elemento HTML */  
7  }  
8  
9  #meu-id {  
10     /* ID do elemento HTML */  
11 }  
12  
13 .minha-classe {  
14     /* Seleção de classe */  
15 }
```

Outros seletores



```
1  div > p {} /* seleciona somente os filhos diretos do elemento */  
2  div + p {} /* seleciona somente o elemento imediatamente após o elemento */  
3  div ~ p {} /* seleciona qualquer elemento vindo após o elemento */
```

<https://developer.mozilla.org/pt-BR/docs/Web/CSS/Pseudo-elements>
<https://developer.mozilla.org/pt-BR/docs/Web/CSS/Pseudo-classes>



Shortcuts



```
1  margin:
2    [margin-top/margin-bottom]
3    [margin-left/margin-right];
4
5  margin:
6    [margin-top]
7    [margin-left]
8    [margin-bottom]
9    [margin-right];
```



Shortcuts



```
1 padding:
2   [padding-top/padding-bottom]
3   [padding-left/padding-right];
4
5 padding:
6   [padding-top]
7   [padding-left]
8   [padding-bottom]
9   [padding-right];
```



Shortcuts



```
1 background:
2   [background-color]
3   [background-image]
4   [background-repeat]
5   [background-attachment]
6   [background-position-x]
7   [background-position-y];
```




Shortcuts



```
1 font:
2   [font-style]
3   [font-weight]
4   [font-size]
5   [font-family];
```

Variáveis



```
1  :root {  
2      --color: black;  
3  }  
4  
5  selector {  
6      --color: white; /* override */  
7      background: var(--color, black); /* fallback */  
8  }
```

Position

```
1  /*
2  valor padrão de todos os elementos
3  */
4  position: static;
5
6  /*
7  se comporta igualmente ao static, a não ser que
8  adicione as propriedades (top, right, bottom e left)
9  */
10 position: relative;
11
12 /*
13 posicionado relativamente ao viewport, isso significa
14 que ele sempre ficará no mesmo lugar mesmo que haja
15 rolagem na página. Assim como o relative, as propriedades
16 (top, right, bottom e left) também são utilizadas
17 */
18 position: fixed;
19
20 /*
21 este valor se comporta como o fixed, porém tendo como
22 referência a posição do elemento relativo mais próximo
23 de onde está contido, ao invés do viewport. Se um elemento
24 absolute não possuir elementos ancestrais posicionados
25 relativamente, ele utilizará o body como referência
26 */
27 position: absolute;
```

Display

```
1  /*
2  um elemento de linha pode preencher algum texto dentro
3  de um parágrafo sem quebrar o fluxo daquele parágrafo
4  */
5  display: none;
6
7  /*
8  o elemento de bloco sempre começa em uma nova linha e se
9  expande pra esquerda e direita o tanto quanto for possível
10 */
11 display: block;
12
13 /*
14 normalmente utilizado para esconder e exibir elementos
15 sem realmente os remover nem recriar
16 */
17 display: inline;
18
19 /*
20 você pode criar um grid de caixas que preenchem a largura do
21 navegador e quebram suavemente. Pode ser utilizada a propriedade
22 vertical-align para setar o alinhamento dos elementos
23 */
24 display: inline-block;
```

Display: flex

```
1 display: flex;
2
3 flex: 1; /* define a proporção de expansão flexível nos filhos */
4 flex-wrap: nowrap; /* informa se quebra linha */
5 flex-direction: [row/column]; /* direção da disposição */
6
7 /* nos permite alinhar o container */
8 /* - start */
9 /* - end */
10 /* - center */
11 /* - stretch */
12 /* - space-between */
13 /* - space-around */
14 /* - space-evenly */
15 justify-content: center; /* alinha no mesmo eixo do flex-direction */
16 align-content: center; /* alinha no eixo contrário do setado no flex-direction */
17
18 /* nos permite alinhar os itens do container no espaço disponível a célula */
19 /* - start */
20 /* - end */
21 /* - center */
22 /* - stretch */
23 justify-items: center; /* alinha no mesmo eixo do flex-direction */
24 align-items: center; /* alinha no eixo contrário do setado no flex-direction */
25
26 /*
27 aplicado diretamente no item e não ao container,
28 nos permitindo subscrever apenas um item da grid
29 */
30 justify-self: center;
31 align-self: center;
```

Display: grid

```
1  display: grid;
2
3  gap: 10px; /* espaçamento entre linhas e colunas */
4  row-gap: 10px; /* espaçamento entre linhas */
5  column-gap: 10px; /* espaçamento entre as colunas */
6  grid-gap: 10px; /* espaçamento entre linhas e colunas */
7  grid-row-gap: 10px; /* espaçamento entre linhas */
8  grid-column-gap: 10px; /* espaçamento entre colunas */
9
10 grid-template-rows: 350px 1fr; /* proporção de ocupação das linhas */
11 grid-template-columns: 2fr 1fr 2fr; /* proporção de ocupação das colunas */
12 grid-template-columns: repeat(4, 1fr); /* 1fr 1fr 1fr 1fr */
```