

CSS

Pseudo-elementos

Pseudo-elementos

Assim como pseudo-classes, pseudo-elementos são adicionados aos seletores mas em vez de descrever um estado especial, eles permitem que você aplique estilos em certas partes de um documento. Por exemplo, o pseudo-elemento **::first-line** aplica o estilo apenas na primeira linha do elemento especificado pelo seletor.

Pseudo-elementos

Sintaxe

```
seletor::pseudo-elemento {  
    propriedade: valor;  
}
```

Pseudo-elementos

O pseudo-elemento **first-letter** é usado para obter um efeito especial na primeira letra de um texto. Exemplo:

```
<style>
  p {
    font-size: 12px;
  }

  p:first-letter {
    font-size: 300%;
  }
</style>

<body>

  <p>Este texto destina-se a demonstrar o
  pseudo-elemento first-letter, bla...bla...bla...
  bla... bla...bla...bla...bla...bla... bla...bla...
  bla... bla...bla...bla...bla...bla... bla...bla...</p>
```

[exe6_0_0](#)

Pseudo-elementos

Todos os pseudo-elementos:

::after

::before

::first-letter

::first-line

::selection

Pseudo-elementos

Atividade

Crie **exemplos** com os pseudo-elementos listados anteriormente e descreva sua funcionalidade.

Pseudo-elementos

Observação: os pseudo-elementos, como `::before` e `::after`, são escritos com dois "dois-pontos", enquanto pseudo-classes, como `:hover` e `:link` são escritos com um "dois-pontos".

A principal diferença é que pseudo-elementos são, como o nome diz, elementos que podem ser criados via CSS, enquanto as pseudo-classes são classes especiais que respondem a ações ou a estados específicos de um elemento (por exemplo: quando o mouse passa em cima ou quando um link já foi visitado).

CSS FLOAT

LME - CSS

```
<style>
  .dv01{
    border: 5px red solid; /* borda: largura cor tipo */

    /* ordem: top right bottom left */
    padding: 15px 10px 15px 10px; /* distancia interna à borda */
    margin: 100px 50px 100px 50px; /* distancia externa à borda*/
    width:100px; /* largura da caixa */
    height: 100px; /* altura da caixa */
    Float: left;
  }
  .dv02{
    border: 5px blue solid; /* borda: largura cor tipo */

    /* ordem: top right bottom left */
    padding: 15px 10px 15px 10px; /* distancia interna à borda */
    margin: 100px 50px 100px 50px; /* distancia externa à borda*/
    width:100px; /* largura da caixa */
    height: 100px; /* altura da caixa */
    float:left;
  }
</style>
</head>
<body>
  <div class="dv01">Esta é uma caixa 1</div>
  <div class="dv02">Esta é uma caixa 2</div>
```

Flutuação dos Elementos FLOAT

A propriedade Float serve para dizer como um elemento deve comportar-se em relação a seus vizinhos no HTML;

Basicamente cada elemento pode flutuar à esquerda (left) ou à direita (right) em relação aos outros;

É possível também definir a flutuação como none (nenhuma).

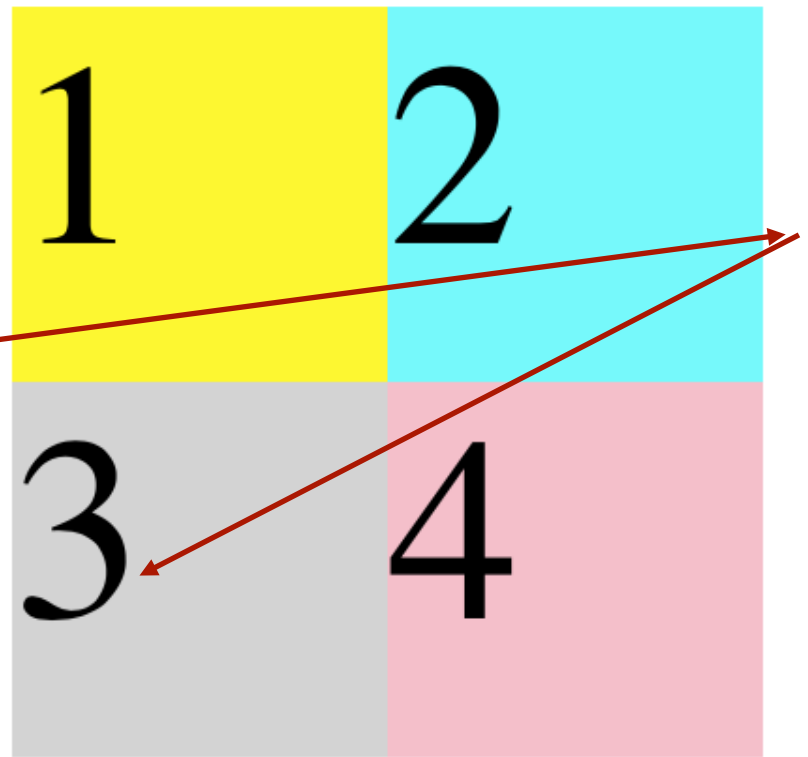
Flutuação dos Elementos FLOAT

A propriedade clear serve para dizer que o próximo elemento não flutuará em relação ao atual.

Pense na propriedade clear como uma quebra de linha para as divs.

```
6 <style>
7   body { font-size:70px;}
8
9   #dv01{
10      width:100px;
11      height: 100px;
12      float:left;
13      background-color:yellow;
14   }
15   #dv02{
16      width:100px;
17      height: 100px;
18      background-color: cyan;
19      float:left;
20   }
21   #dv03{
22      width:100px;
23      height: 100px;
24      background-color: lightgray;
25      float:left;
26      clear:both;
27   }
28   #dv04{
29      width: 100px;
30      height: 100px;
31      background-color: pink;
32      float:left;
33   }
34 }
35 </style>
36 </head>
37 <body>
38   <div id="dv01">1</div>
39   <div id="dv02">2</div>
40   <div id="dv03">3</div>
41   <div id="dv04">4</div>
```

CLEAR



CSS

Elementos Bloco e Em Linha

Elementos Bloco e Em Linha

- Todos os elementos HTML podem ser separados em dois grupos, em termos de comportamento visual CSS:
 - **Elementos bloco (*block*);**
 - **Elementos em linha (*inline*).**

Elementos Bloco e Em Linha

- **Elementos em Bloco**

- Ocupam todo o espaço horizontal disponível e iniciam uma nova linha no documento. Novos elementos irão começar na próxima linha livre.

Elementos Bloco e Em Linha

- **Exemplos de elementos bloco:**
 - div (*division*, ou divisão, tag bloco genérica)
 - h1 até h6 (títulos)
 - p (parágrafo)
 - blockquote (citação em bloco)
 - ul (lista não ordenada)
 - ol (lista ordenada)
 - form (formulários)

Elementos Bloco e Em Linha

- **Elementos em linha**

- Ocupam apenas o espaço necessário e não iniciam uma nova linha. São chamados elementos em linha justamente por aparecer na mesma linha que outros elementos, caso seja possível.

Elementos Bloco e Em Linha

- **Exemplos de elementos em linha:**
 - span (tag em linha genérica)
 - strong (destaca importância)
 - em (ênfase)
 - a (âncora, usada para links)
 - img (imagem)

Aninhamento de Elementos

- Uma página é feita de blocos empilhados.
- E comum inserir um elemento dentro de outro para conseguirmos construir um layout adequado e com a hierarquia de informações que desejamos.
- Inserir um elemento dentro de outro chama-se **aninhamento** (ou *nesting*) e em documentos HTML há algumas regras principais a serem adotadas:

Aninhamento de Elementos

- Elementos bloco podem conter qualquer elemento em linha;
- Elementos bloco podem conter outros elementos bloco, observadas algumas restrições (por exemplo, não podemos colocar uma div dentro de um parágrafo mas o contrário é válido);
- Elementos em linha **nunca** poderão conter elementos bloco (ex: colocar um parágrafo dentro de um link ou uma div dentro de uma span). A HTML5 apresenta algumas mudanças no caso do que links (tags a) podem englobar, exceto isso esta regra é geral;
- Elementos em linha podem conter outros elementos em linha (por exemplo, colocar span dentro de um link ou o contrário).

Aplicação de Propriedades CSS

Com as diferenças já vistas entre elementos bloco ou em linha. Também diferem nas propriedades CSS que podem ser aplicadas.

Elementos em linha podem ter os seguintes tipos de propriedades modificadas:

- Cores de texto e de fundo;
- Propriedades de texto, como font-family, font-size e outras, como text-decoration;
- Bordas.

Aplicação de Propriedades CSS

Observação:

Propriedades (width, height, margin e padding) podem surtir efeito no elemento em questão, mas geralmente não irão afetar os outros elementos.

Assim sendo, para fins de layout e posicionamento, quando se usam muitas definições de dimensões e margens, usar elementos em linha não é aconselhável.

Aplicação de Propriedades CSS

Os Elementos em bloco, além de todas as propriedades dos elementos em linha, também é possível modificar:

- Largura (width) e altura (height);
- Margens internas (padding) e externas (margin).

Quando queremos usar tags para compor um menu, muitas vezes a transformamos em elemento bloco. Podemos fazer isso usando a propriedade display CSS.

Propriedades Display CSS

A propriedade display serve para mudarmos o comportamento visual de um elemento, como mudar uma tag **a** (em linha) para bloco ou uma tag **p** para elemento em linha. Ela aceita vários valores, dos quais os principais são:

- block: elemento aparece como se fosse bloco;
- inline: elemento aparece como se fosse em linha;
- none: elemento não aparece, útil quando utilizado junto com JavaScript para fazê-lo aparecer com algum efeito visual, entre outras aplicações.

Propriedades Display CSS

```
<title>Exemplo - Elementos Bloco e Em Linha</title>
<style>
    .link-normal {
        background: #00F;
        color: #FFF;
        margin: 10px;
        padding: 20px;
        border: 1px solid #000;
    }

    .link-bloco {
        display: block;
        background: #00F;
        color: #FFF;
        margin: 10px;
        padding: 10px;
        border: 1px solid #000;
    }
</style>
```

Propriedades Display CSS

No exemplo anterior perceba a diferença entre os dois elementos: enquanto o link em linha ocupa apenas o espaço necessário para o texto, o link bloco ocupa toda a largura da página.

Apesar das propriedades margin e padding terem sido aplicadas, elas não afetam os outros elementos da mesma forma.

A margem interna do link em linha, por exemplo, está fazendo efeito porém não afeta os elementos ao seu redor (perceba que a área do link está sobrepondo a do link bloco).

Propriedades Display CSS

Ao montarmos layouts, normalmente iremos converter os elementos para bloco (ou então utilizaremos elementos bloco) já que permite um maior controle sobre as dimensões, posicionamento e margens.

CSS

Estrutura do Web Site

Estrutura do Web Site

O bloco de construção básico de um site é a tag div. Essa tag é uma das mais utilizadas e será a base do nosso projeto de site.

Importante lembrar do arquivo "index.html" e o arquivo "estilos.css" que estão dentro da pasta "website".

Vamos ver o layout que temos na página a seguir:



BRASIL

Serviços

Participe

Acesso à informação

Legislação

Canais



[Ir para o conteúdo](#) [Ir para o menu](#) [Ir para a busca](#) [Ir para o rodapé](#)

[ACESSIBILIDADE](#) [ALTO CONTRASTE](#) [MAPA DO SITE](#)

Instituto Federal Sul-rio-grandense

Câmpus Pelotas

MINISTÉRIO DA EDUCAÇÃO

Buscar no portal



[Acesso à Informação](#) | [Fale Conosco](#) | [Fale com o Diretor](#) | [Webmail](#) | [Help Desk](#) | [Telefones](#)



 Espaço do
ESTUDANTE

 Espaço do
SERVIDOR

[Processo Seletivo](#)

[Concursos](#)

[Editais](#)



Notícias



MONITORIA

Câmpus divulga resultado final do processo seletivo para bolsas de monitoria

Estrutura do Web Site

Podemos verificar que há um cabeçalho e, dentro dele, o menu principal.

Abaixo, temos a área de conteúdo, com uma área de conteúdo principal e outra de conteúdo secundário.

Finalizando, temos o rodapé com informações complementares.

Traduzimos a descrição acima para HTML criando uma div para cada área.

Estrutura do Web Site

```
<div id="cabecalho">
  <!-- aqui colocaremos a logo -->
  <div id="menu-principal">

    </div>
  </div>
<div id="conteudo">
  <div id="conteudo-principal">

    </div>
    <div id="conteudo-secundario">

    </div>
  </div>
<div id="rodape">

</div>
```

index.html


```
/*  
-- Estilos CSS - Website da Empresa  
*/
```

```
/*  
-- Padrões e Reset  
*/
```

```
/*  
-- Tipografia e Cores  
*/
```

```
/*  
-- Estrutura  
*/
```

```
/*  
-- Estilos específicos de Seção  
*/
```

```
/*  
-- Formulários  
*/
```

```
/*  
-- Tabelas  
*/
```

Sugestão
estilos.css

Estrutura do Web Site

- **Padrões e Reset:** nesta seção usa-se estilos que padronizam o visual da página entre navegadores, além de outras preferências pessoais.
- **Tipografia e Cores:** nesta seção define-se as fontes utilizadas no site, além de tamanhos, cores e margens para elementos de texto e links
- **Estrutura:** coloca-se os estilos referentes à estrutura e layout do site.
- **Estilos específicos de seção:** algumas seções possuem estilos próprios apenas para elas. Quando for sites pequenos compensa colocar os estilos de cada página.
- **Formulários e Tabelas:** nestas seções iremos inserir estilos gerais para formulários e tabelas, respectivamente.

CSS

TAGS DE TEXTO

Principais Tags de Texto

- Tag p (parágrafo);
- Tag em (ênfase);
- Tag strong (forte ênfase);
- Tags h1 a h6 (títulos);
- Tag a (âncora);
- Tag ul (listas não ordenadas);
- Tag ol (listas ordenadas);
- Etc...

Entidades HTML

Quando precisamos escrever símbolos como $>$, $<$ e ©, utilizamos entidades HTML para isto. Uma entidade começa com "&" e termina com ";" exemplos:

- Para escrever $>$, no código escrevemos `>`; (**gt** significa “maior que”)
- Para escrever $<$, no código escrevemos `<`; (**lt** significa “menor que”)
- Para escrever $&$, no código escrevemos `&`; (**amp** significa “é comercial”)
- Para escrever © no código escrevemos `©`; (**copy** significa “copyright”)

Atividade

Pesquise como escrevemos essas entidades em HTML?

1. ® (marca registrada)
2. € (euros)
3. ½ (uma metade)
4. ≤ (menor ou igual)
5. ≠ (diferente)

Importantes Propriedades de Texto

As folhas de estilo são fundamentais para estilização do texto em uma página. Com as regras CSS podemos mudar sua cor, tamanho, fonte, entre outras.

Vejamos:

- **color:** muda a cor do texto. Ex: `color: #FF0000;`
- **font-family:** define uma lista de fontes. Ex: Tahoma, Ariel
- **font-size:** muda o tamanho do texto. Podemos utilizar unidades como px (pixels), em (unidade relativa), %, pt (pontos), entre outras. Ex: `font-size: 20px.`



Importantes Propriedades de Texto

- **font-weight:** muda o peso da fonte. Valores incluem bold e normal. Ex: font-weight: bold;
- **font-style:** muda o estilo da fonte. Valores incluem italic e normal. Ex: font-style: italic;
- **text-decoration:** muda a decoração do texto. Valores incluem underline e none (por ex: para retirar o sublinhado de links). Ex: text-decoration: underline;
- **line-height:** muda a altura de cada linha, ou *entrelinha*. É comum utilizar a unidade em, pois assim deixamos a entrelinha proporcional ao tamanho da fonte. Ex: line-height: 1.5em; (1.5 vezes o tamanho da fonte).

Importantes Propriedades de Texto

- **text-indent:** muda a indentação da primeira linha de um bloco de texto. Ex: text-indent: 2em; (indentação de 2 vezes o tamanho da fonte na primeira linha do texto).
- **text-transform:** muda a "caixa" do texto. Valores incluem uppercase (caixa-alta), lowercase (caixa-baixa) e capitalize (primeira letra maiúscula e o restante minúsculas). Ex: text-transform: uppercase;
- **font-variant:** muda a variante da fonte. Valores incluem normal e small-caps. Ex: font-variant: small-caps;
- **word-spacing:** determina o espaçamento entre palavras. Ex: word-spacing: 5px;
- **letter-spacing:** determina o espaçamento entre letras. Ex: letter-spacing: 2px;

CORES E CSS

Podemos escrever as cores de várias formas no código CSS. Os principais tipos são:

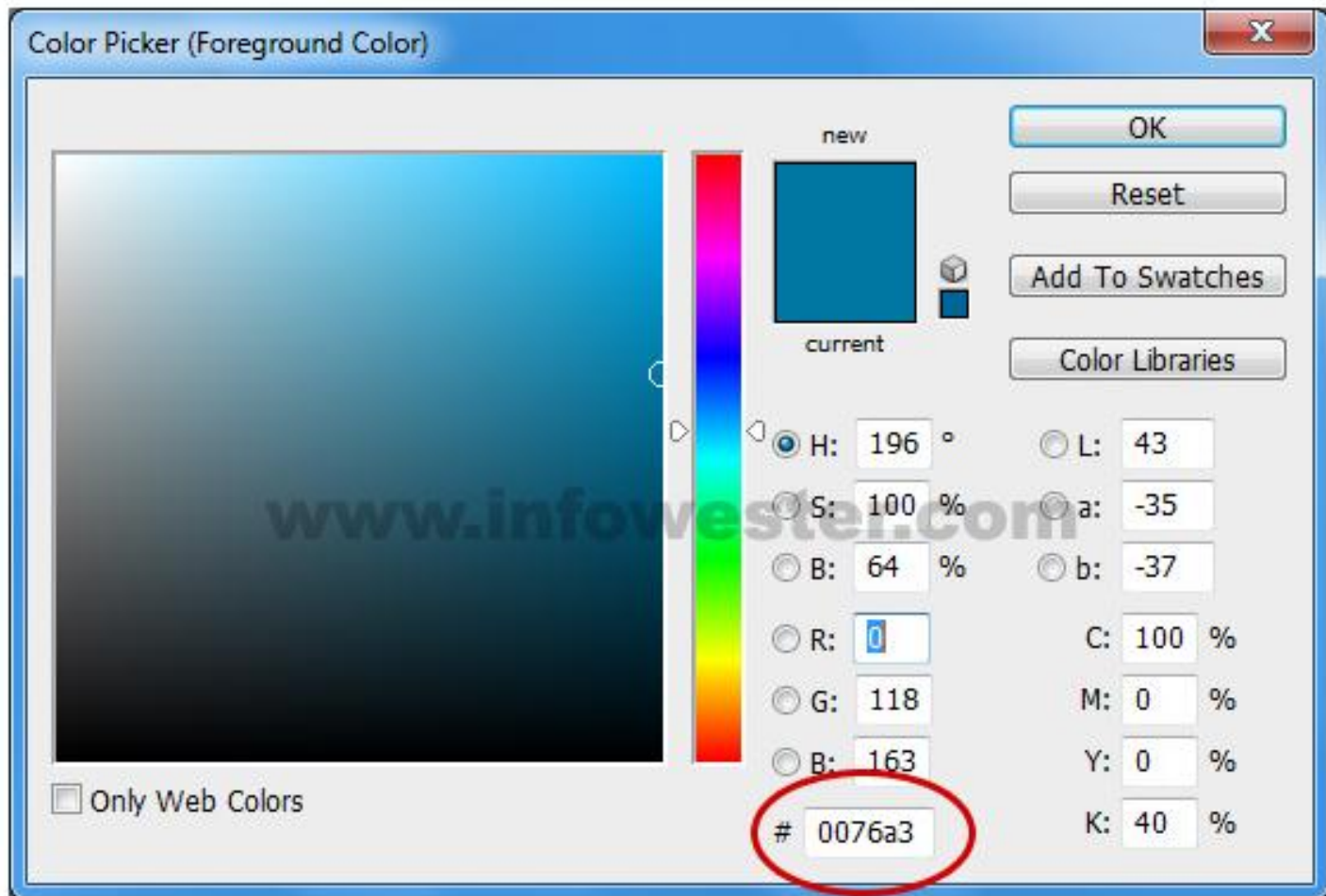
- **Nome:** podemos escrever o nome da cor, dentre alguns pré-definidos, como `black`, `white`, `blue`, `green` e entre outras;
- **RGB** (*red, green, blue*): neste modo colocamos os valores (de 0 a 255), para R, G e B.
Ex: `rgb(0,0,0)` para preto e `rgb(255,255,255)` para branco;

CORES E CSS

- **Número Hexadecimal:** neste modo colocamos os valores para R, G e B com números hexadecimais (de 00 a FF, ou de 0 a 255). Por exemplo: #000000 é preto e #FFFFFF é branco. Cores que possuem valores repetidos, como as duas exemplificadas, também podem ser escritas de forma mais curta como #000 e #FFF.
- **RGBA** (*red, green, blue, alpha*): recurso novo no CSS3. Praticamente igual ao modo RGB, porém com um quarto valor, alpha, que permite controlar a opacidade (transparência) da cor, de 0 (transparente) a 1 (opaca). Por exemplo, `rgba(0,0,0,0.5)`, cor preta com 50% de opacidade.

CORES E CSS

Para descobrir o valor RGB ou Hexadecimal usar programas como Adobe Photoshop ou Corel Draw.



ATIVIDADE

Pesquise outras novas formas de cores em CSS?

- HSL Colors
- HSLA Colors

Exercícios

