

### ***RESUMO:***

O presente documento faz referencia ao curso básico de CSS. No primeiro momento temos uma apostila referente ao curso, logo depois apresentamos o modelo ADDIE que serviu de embasamento para produzir o curso e suas ferramentas. Temos ainda os slides das três aulas que foram ministradas mais suas avaliações e o relatório da avaliação com gráficos.

Bruno Santos de Miranda

bilnp29@gmail.com



## Sumário

1.	História .....	4
2.	Definição .....	5
3	Sintaxe e inclusão do CSS .....	6
3.1	Atributo Style .....	6
3.2	A tag style .....	7
3.3	Arquivo Externo .....	8
4	Seletores .....	9
5	Propriedades tipográficas e fontes .....	11
5.1	Imagens de fundo .....	13
5.2	Bordas .....	14
5.2.1	Border-style .....	14
5.2.2	Border-width .....	15
5.2.3	Border-color .....	15
5.3	Espaçamento e margens .....	16
5.3.1	Padding .....	16
5.3.2	Margin .....	17
6	Dimensões .....	18
7	Seletores de ID e Filho .....	18
7.1	Seletores de Classe .....	19
8	Posicionamento .....	20



8.1	Posicionamento (Static).	20
8.2	Posicionamento (Relative).	20
8.3	Posicionamento (Fixed).	22
8.4	Posicionamento (absolute).	22
9	CSS Reset	22
10	CSS3	22
10.1	Rounded Corners.	23
10.2	CSS3 Backgrounds.	23
10.2.1	Background-Size	24
10.2.2	background-origin.	25
10.2.3	background-clip	26
10.3	Object-fit.	28
11	Buttons	29
12	Framework.	29
13	CSS em prática.	30
14	Referencias.	37



## 1. História

A internet, que conhecemos hoje, é fruto de uma evolução constante e gradativa desde sua invenção, dos seus protocolos até suas linguagens de desenvolvimento sofreram tais mudanças e com certeza, um dos principais componentes dessa grande estrutura mundial é o HTML. Mas espera aí, não íamos falar sobre CSS?

Pois bem, vejamos então o começo de tudo, a linguagem de marcação surgindo para solucionar o problema de um cientista que queria uma forma de divulgar seus artigos na rede, para isso, ele criou algo simples, restrito a comunidade científica e usada apenas para estrutura o conteúdo e exibir um texto de forma coerente. Mas a linguagem usada pelos cientistas se tornou padrão da Internet, e com o tempo, passou a ser mais testada. Todos queriam imagens, cores e designer avançado. E o HTML antigo não suportava isso, então foram adicionadas novas tags. Isso solucionou o problema por um tempo, mas começou a surgir outro grande problema. Com os novos recursos, designers e desenvolvedores de web sites, abusavam da criatividade e enchiam suas paginas de fontes e estilos visuais, mas para alterar uma linha até que era mais simples, porém quando se queria mudar a cor de um link, por exemplo, e esse link estava em 300 paginas diferentes, era um trabalho manual absurdo, um por um, tag por tag.

Misturar estilo e estrutura não era mais interessante, e foi assim que em 1995, Håkon Wium Lie e Bert Bos apresentaram a proposta do CSS(Cascading Style Sheets) que logo foi apoiada pela W3C. A idéia geral era, utilizar HTML somente para estruturar o website e a tarefa de apresentação fica com o CSS disposto em um arquivo separado (.css) ou no próprio HTML demarcado pelas tags . Desde sua criação o CSS teve algumas versões, atualmente ele se encontra na versão 3(três). W3C é a empresa que mais apoia a linguagem, visando tornar o desenvolvimento padrão e também exigir dos navegadores uma interpretação adequada e compatível.



## 2. Definição

O CSS formata a informação entregue pelo HTML. Essa informação pode ser qualquer coisa: imagem, texto, vídeo, áudio ou qualquer outro elemento criado. Quando escrevemos o HTML, marcamos o conteúdo da página com tags que melhor representam o significado daquele conteúdo. Aí quando abrimos a página no navegador é possível perceber que o navegador mostra as informações com estilos diferentes. Um h1, por exemplo, fica em negrito numa fonte maior. Parágrafos de texto são espaçados entre si, e assim por diante. Isso quer dizer que o navegador tem um estilo padrão para as tags que usamos. Mas, claro, pra fazer sites bonitões vamos querer customizar o design dos elementos da página.

Antigamente, isso era feito no próprio HTML. Se quisesse um título em vermelho, era só fazer:

```
<h1><font color="red">Curso Básico de CSS</font></h1>
```

Além da tag font, várias outras tags de estilo existiam. Mas isso é passado. **Tags HTML para estilo são má prática** hoje em dia. Em seu lugar, surgiu o **CSS**, que é outra linguagem, separada do HTML, com objetivo único de cuidar da estilização da página. A vantagem é que o CSS é bem mais robusto que o HTML para estilização, como veremos. Mas, principalmente, escrever formatação visual misturado com conteúdo de texto no HTML se mostrou algo bem impraticável. O CSS resolve isso separando as coisas; regras de estilo não aparecem mais no HTML, apenas no CSS. Definindo o CSS formalmente temos: *Cascading Style Sheets (CSS) é uma "folha de estilo" composta por "camadas" e utilizada para definir a apresentação (aparência) em páginas da internet que adotam para o seu desenvolvimento linguagens de marcação (como XML, HTML e XHTML).*

### 3 Sintaxe e inclusão do CSS

A sintaxe do CSS tem estrutura simples: Teremos um seletor, onde este indicará qual tag será customizado no site, o escopo será dimensionado entre chaves, dentro delas teremos uma declaração de propriedades e valores separados por um sinal de dois pontos“:”, e cada propriedade é separada por um sinal de ponto e vírgula“;” da seguinte maneira:

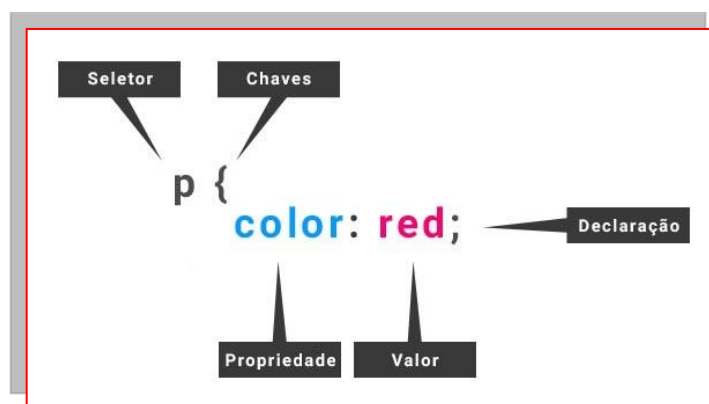


Figura 1- Sintaxe do CSS

A inclusão do CSS pode ser feita de 3 maneiras diferentes.

#### 3.1 Atributo Style

A primeira delas é como um atributo style no próprio elemento:

```
<p style="color: blue; background-color: yellow;">
```

O conteúdo desta tag será exibido em azul com fundo amarelo no navegador!

Este tipo de customização é feito dentro do próprio HTML, Mas tínhamos acabado de discutir que uma das grandes vantagens do CSS era manter as regras de estilo fora do HTML. Justamente por isso não se recomenda esse tipo de uso na prática, mas sim os que veremos a seguir.



### 3.2 A tag *style*

A outra maneira de se utilizar o CSS é declarando suas propriedades dentro de uma tag `<style>`. Como estamos declarando as propriedades visuais de um elemento em outro lugar do nosso documento, precisamos indicar de alguma maneira a qual elemento nos referimos. Fazemos isso utilizando um seletor CSS. É basicamente uma forma de buscar certos elementos dentro da página que receberão as regras visuais que queremos. No exemplo a seguir, usaremos o seletor que pega todas as tags `p` e altera sua cor e background:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Css Curso Básico</title>
    <style>
      p {
        background-color: yellow;
        color: blue;
      }
    </style>
  </head>
  <body>
    <p>
      O conteúdo desta tag será exibido em azul com fundo amarelo!
```

O código anterior da tag `<style>` indica que estamos alterando a cor e o fundo de todos os elementos com tag `p`. Dizemos que selecionamos esses elementos pelo nome de sua tag, e aplicamos certas propriedades CSS apenas neles.



### 3.3 *Arquivo Externo*

A terceira maneira de declararmos os estilos do nosso documento é com um arquivo externo, geralmente com a extensão (.css). Para que seja possível declarar nosso CSS em um arquivo à parte, precisamos indicar em nosso documento HTML uma ligação entre ele e a folha de estilo. Além da melhor organização do projeto, a folha de estilo externa traz ainda as vantagens de manter nosso HTML mais limpo e do reaproveitamento de uma mesma folha de estilos para diversos documentos. A indicação de uso de uma folha de estilos externa deve ser feita dentro da tag <head> do nosso documento HTML:

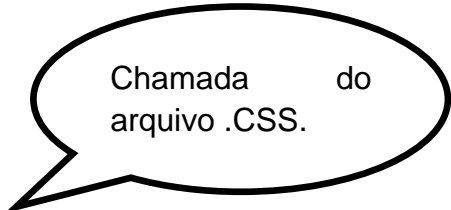
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Sobre a Mirror Fashion</title>

    <link rel="stylesheet" href="estilos.css">

  </head>
  <body>
    <p>
      O conteúdo desta tag será exibido em azul com fundo amarelo!
    </p>

    <p>
      <strong>Também</strong> será exibido em azul com fundo amarelo!
    </p>

  </body>
</html>
```



E dentro do arquivo **estilos.css** colocamos apenas o conteúdo do CSS:

```
p {
  color: blue;
  background-color: yellow;
}
```





## 4 Seletores

Os seletores são a alma do CSS e você precisa dominá-los. São com os seletores que você irá escolher um determinado elemento dentro todos os outros elementos do site para formatá-lo. Boa parte da inteligência do CSS está em saber utilizar os seletores de uma maneira eficaz, escalável e inteligente. Ele representa uma estrutura, Essa estrutura é usada como uma condição para determinar quais elementos de um grupo de elementos serão formatados. Seletores encadeados e seletores agrupados são à base do CSS. Você os aprende por osmose durante o dia a dia. Para você lembrar o que são seletores encadeados e agrupados segue um exemplo abaixo:

Exemplo de seletor encadeado:

```
div p strong a {  
  color: red;  
}
```

Este seletor formata o link (a), que está dentro de um strong, que está dentro de P e que por sua vez está dentro de um DIV.

Exemplo de seletor agrupado:

```
strong, em, span {  
  color: red;  
}
```

Você agrupa elementos separados por vírgula para que herdem a mesma formatação.



Estes seletores são para cobrir suas necessidades básicas de formatação de elementos. Eles fazem o simples. Mais adiante falaremos sobre os seletores de CLASS e ID.



## 5 Propriedades tipográficas e fontes

Da mesma maneira que alteramos cores, podemos alterar o texto. Podemos definir fontes com o uso da propriedade `font-family`. A propriedade `font-family` pode receber seu valor com ou sem aspas. No primeiro caso, passaremos o nome do arquivo de fonte a ser utilizado, no último, passaremos a família da fonte.

Por padrão, os navegadores mais conhecidos exibem texto em um tipo que conhecemos como “serif”. As fontes mais conhecidas (e comumente utilizadas como padrão) são “Times” e “Times New Roman”, dependendo do sistema operacional. Elas são chamadas de **fontes serifadas** pelos pequenos ornamentos em suas terminações.

Podemos alterar a família de fontes que queremos utilizar em nosso documento para a família “sans-serif” (sem serifas), que contém, por exemplo, as fontes “Arial” e “Helvetica”. Podemos também declarar que queremos utilizar uma família de fontes “monospace” como, por exemplo, a fonte “Courier”.

```
h1 {  
    font-family: serif;  
}  
  
h2 {  
    font-family: sans-serif;  
}  
  
p {  
    font-family: monospace;  
}
```

Outro exemplo de utilização de fontes nos textos:



```
body {  
  font-family: "Arial", "Helvetica", sans-serif;  
}
```

Figura 2- Exemplo de utilização de fontes.

Nesse caso, o navegador verificará se a fonte “Arial” está disponível e a utilizará para renderizar os textos de todos os elementos do nosso documento que, por cascata, herdarão essa propriedade do elemento body. Caso as fontes “Arial” e “Helvetica” não estejam disponíveis o navegador solicita qualquer fonte que pertença à família “sans-serif”, declarada logo a seguir, e a utiliza para exibir o texto, não importa qual seja ela.

Temos outras propriedades para manipular a fonte, como a propriedade **font-style**, que define o estilo da fonte que pode ser: normal (normal na vertical), italic (inclinada) e oblique (oblíqua).

A propriedade **font-weight** define quão negrito ou pesado deve ser a fonte. Uma fonte pode ser **normal** ou **bold**. Para definirmos o tamanho da fonte utilizaremos a propriedade **font-size**. Podemos utilizar o pixel ou a porcentagem. Uma das propriedades mais simples, porém muito utilizada, é a que diz respeito ao alinhamento de texto: a **propriedade text-align**.

```
p {  
  font-size:12px;  
  font-weight:bold;  
  text-align: right;  
}
```



Podemos ter ainda alinhamento no Center (centro), Left (esquerda), e Justify (justificado). É possível configurar também uma série de espaçamentos de texto com o CSS:

```
p {  
    line-height: 3px; /* tamanho da altura de cada linha */  
    letter-spacing: 3px; /* tamanho do espaço entre cada letra */  
    word-spacing: 5px; /* tamanho do espaço entre cada palavra */  
    text-indent: 30px; /* tamanho da margem da primeira linha do texto */  
}
```

## 5.1 *Imagens de fundo*

A propriedade background-image permite indicar um arquivo de imagem para ser exibido ao fundo do elemento.

```
h1 {  
    background-image: url(sobre-background.jpg);  
}
```

Com essa declaração, o navegador vai requisitar um arquivo sobre-background.jpg, que deve estar na mesma pasta do arquivo CSS onde consta essa declaração.

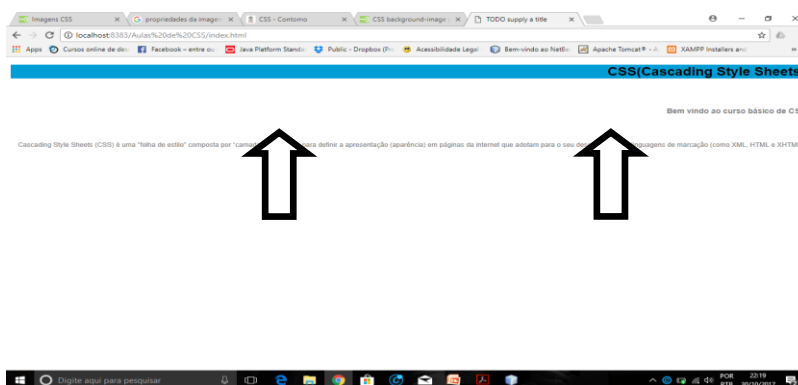


Figura 3 - Utilizando a propriedade background-image.

## 5.2 Bordas

O CSS apresenta varias opções de definição de bordas para um elemento, podemos determinar sua cor, seu estilo de exibição e sua largura. Utilizando as seguintes propriedades.

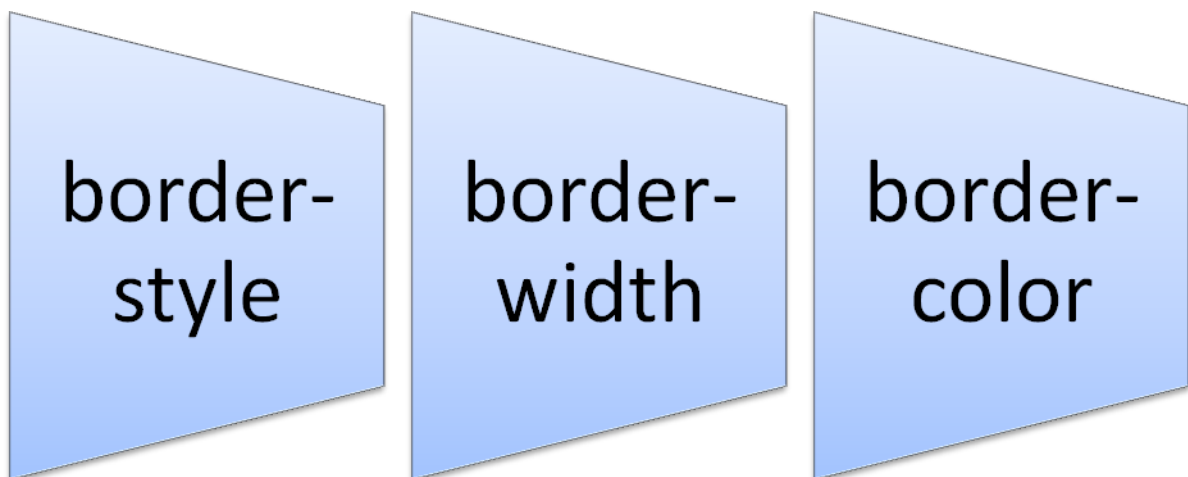


Figura 4 - Propriedades para determinar estilos de bordas.

### 5.2.1 Border-style

A propriedade border-style especifica o tipo de borda a ser exibida.

Os seguintes valores são permitidos:

- **dotted** - Define uma borda pontilhada
- **dashed** - Define uma borda tracejada
- **solid** - Define uma borda sólida
- **double** - Define uma borda dupla
- **groove**- Define uma borda 3D ranhurada. O efeito depende do valor da cor da borda



- **ridge**- Define uma borda 3D ridged. O efeito depende do valor da cor da borda
- **inset**- Define uma borda de inserção 3D. O efeito depende do valor da cor da borda
- **outset**- Define uma borda de início 3D. O efeito depende do valor da cor da borda
- **none** - Define nenhuma borda
- **hidden** - Define uma borda oculta

A border-style propriedade pode ter de um a quatro valores (para a borda superior, borda direita, borda inferior e a margem esquerda).

### 5.2.2 Border-width

A border-width propriedade especifica a largura das quatro bordas, podendo ser de tamanho específico, (px, pt, cm) ou valores pré-definidos: thin, medium, or thick.

```
body {  
    border-style: solid;  
    border-width: 1px;  
}
```

Esta propriedade também suporta de um a quatro valores (para a borda superior, borda direita, borda inferior e a margem esquerda).

### 5.2.3 Border-color

Propriedade é usada para definir a cor das quatro bordas.

```
body {  
    borde-color: red;  
    borde-color: #ff0000;  
    borde-color: (255,0,0);  
    border-style: solid;  
}
```



Especifique um nome de cor, como "vermelho"
Especifique um valor hexadecimal, como "# ff0000"
Especifique um valor RGB, como "rgb (255,0,0)"
Transparente

Figura 5 - Especificação de cores no CSS.

### 5.3 *Espaçamento e margens*

Utilizamos a propriedade padding para espaçamento e margin para margem.

#### 5.3.1 **Padding**

A propriedade **padding** é utilizada para definir uma margem interna em alguns elementos. E tem as subpropriedades listadas a seguir:

- padding-top
- padding-right
- padding-bottom
- padding-left.

A propriedade funciona da seguinte forma: Se colocarmos só um valor, exemplo.

```
p {  
padding: 10px;  
}
```

Este valor se aplica a todas as direções.

Caso utilizemos 2 valores, o primeiro será aplicado acima e abaixo (equivalente a passar o mesmo valor para padding-top e padding-bottom) e o segundo será





aplicado à direita e à esquerda (equivalente ao mesmo valor para padding-right e padding-left).

```
p {  
padding: 10px, 15px;  
}
```

Se passados três valores, o primeiro será aplicado acima (equivalente a padding-top), o segundo será aplicado à direita e à esquerda (equivalente a passar o mesmo valor para padding-right e padding-left) e o terceiro valor será aplicado abaixo do elemento (equivalente a padding-bottom)

```
p {  
padding: 10px, 15px, 10px;  
}
```

Se passados 4 valores serão direcionados as subpropriedades mencionadas acima.

### 5.3.2 Margin

A propriedade margin é utilizada para definir uma margem externa do elemento. E sua utilização é igual à propriedade padding, apresentado as mesma subpropriedade.

- margin-top
- margin-right
- margin-bottom
- margin-left.



## 6 Dimensões

É possível determinar as dimensões de um elemento, por exemplo:

```
p {  
  background-color: red;  
  height: 300px;  
  width: 300px;  
}
```

Todos os parágrafos do nosso HTML ocuparão 300 pixels de largura e de altura, com cor de fundo vermelha.

## 7 Seletores de ID e Filho.

Já vimos como selecionar elementos no CSS usando simplesmente o nome da tag:

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

Simples, porém limitada. Em determinados momentos não queremos pegar todos os parágrafos de uma página. É possível aplicar propriedades visuais **a um elemento selecionado pelo valor de seu atributo id**. Para isso, o seletor deve iniciar com o caractere “#” seguido do valor correspondente.

```
#cabecalho {  
  color: white;  
  text-align: center;  
}
```



O seletor acima fará com que o elemento do nosso HTML que tem o atributo id com valor “cabeçalho” tenha seu texto renderizado na cor branca e centralizado.

Note que não há nenhuma indicação para qual tag a propriedade será aplicada. Pode ser tanto uma <div> quanto um <p>, até mesmo tags sem conteúdo como uma <img>, desde que essa tenha o atributo id com o valor “cabeçalho”. Como o atributo id deve ter valor único no documento, o seletor deve aplicar suas propriedades declaradas somente àquele único elemento e, por cascata, a todos os seus elementos • filhos.

Podemos ainda utilizar um seletor hierárquico que permite aplicar estilos aos elementos • lhos de um elemento pai:

```
#rodape img {  
  
    margin-right: 35px;  
    vertical-align: middle;  
    width: 94px;  
  
}
```

No exemplo anterior, o elemento pai rodape é selecionado pelo seu id. O estilo será aplicado apenas nos elementos img • lhos do elemento com id=rodape.

## 7.1 Seletores de Classe

Uma terceira forma é a utilização de classes. O código é semelhante, mas usa o atributo **class no HTML** e o **ponto no CSS**.

```
<div class = "menu-opcoes"> </div >
```

Figura 6 - Utilizando a atributo class no html

e, no CSS.



```
.menu-opcoes {  
    ...  
}
```

Figura 7 - Estilizando o atributo class de valor .menu-opcoes.

## 8 Posicionamento

Existe um conjunto de propriedades que podemos utilizar para posicionar um elemento na página, que são top, left, bottom e right. Porém essas propriedades, por padrão, não são obedecidas por nenhum elemento, pois elas dependem de outra propriedade, a position.

A propriedade **position** determina qual é o modo de posicionamento de um elemento, e ela pode receber como valor **static**, **relative**, **absolute** ou **fixed**.

### 8.1 Posicionamento (Static).

Sua posição é dada automaticamente pelo fluxo da página: por padrão ele é renderizado logo após seus irmãos. Não aceita um posicionamento manual (left, right, top, bottom).

### 8.2 Posicionamento (Relative).

Utilizando o position Relative o elemento passa a aceitar as propriedades Top, Bottom, Left e Right. Com elas você pode alterar o posicionamento do elemento.

```
.logotipo {  
    position: relative;  
    top: 20px;  
    left: 50px;  
}
```



Os elementos em nosso documento que receberem o valor “logotipo” em seu atributo class terão 20px adicionados ao seu topo e 50px adicionados à sua esquerda em relação à sua posição original. Note que, ao definirmos coordenadas, estamos adicionando pixels de distância naquela direção, então o elemento será renderizado mais abaixo e à direita em comparação à sua posição original.



### 8.3 *Posicionamento (Fixed).*

Uma configuração de posicionamento vertical (left. ou right) e uma horizontal (top ou bottom) é obrigatória o elemento será renderizado na página na posição indicada: mesmo que ocorra uma rolagem o elemento permanecerá no mesmo lugar.

### 8.4 *Posicionamento (absolute).*

Uma configuração de posicionamento vertical (left. ou right) e uma horizontal (top ou bottom) é obrigatória. O elemento será renderizado na posição indicada, porém relativa ao primeiro elemento pai cujo position seja diferente de static ou, se não existir este pai, relativa à página.

## 9 CSS Reset

Quando não especificamos nenhum estilo para nossos elementos do HTML, o navegador utiliza uma série de estilos padrão, que são diferentes em cada um dos navegadores. Em um momento mais avançado dos nossos projetos, poderemos enfrentar problemas com coisas que não tínhamos previsto; por exemplo, o espaçamento entre caracteres utilizado em determinado navegador pode fazer com que um texto que, pela nossa definição deveria aparecer em 4 linhas, apareça com 5, quebrando todo o nosso layout.

Para evitar esse tipo de interferência, alguns desenvolvedores e empresas criaram alguns estilos que chamamos de **CSS Reset**. A intenção é setar um valor básico para todas as características do CSS, sobrescrevendo totalmente os estilos padrão do navegador. No site <https://cssreset.com/> temos alguns arquivos já definidos, basta verificar qual melhor se adéqua a sua implementação.

## 10 CSS3

**CSS3** é a versão mais recente, onde se define estilos para páginas web com efeitos de transição, imagem, e outros, que dão um estilo novo às páginas [Web 2.0](#) em todos os aspectos de design do layout.



A principal função do CSS3 é abolir as imagens de plano de fundo, bordas arredondadas, apresentar transições e efeitos para criar animações de vários tipos, como um simples relógio de ponteiros.

## 10.1 *Rounded Corners*

Com propriedade *border-radius* podemos obter qualquer elemento com “Cantos Arredondados.” Se você especificar apenas um valor para a *border-radius* propriedade, esse raio será aplicado a todos os 4 cantos. No entanto, você pode especificar cada canto separadamente, se desejar. Aqui estão as regras:

- **Quatro valores:** o primeiro valor se aplica ao superior esquerdo, o segundo valor se aplica ao superior direito, o terceiro valor se aplica ao canto inferior direito e o quarto valor se aplica ao canto inferior esquerdo
- **Três valores:** o primeiro valor aplica-se ao superior esquerdo, o segundo valor aplica-se ao canto superior direito e inferior esquerdo e o terceiro valor se aplica ao canto inferior direito.
- **Dois valores:** o primeiro valor aplica-se ao canto superior esquerdo e inferior direito e o segundo valor aplica-se ao canto superior direito e inferior esquerdo
- **Um valor:** os quatro cantos são arredondados igualmente

```
#rcorners1 {  
    border-radius: 25px;  
    background: #73AD21;  
    padding: 20px;  
    width: 200px;  
    height: 150px;  
}
```

Figura 8 - Exemplo utilizando *border-radius*.

## 10.2 *CSS3 Backgrounds*

Contém algumas novas propriedades de fundo, que permitem um maior controle do elemento de fundo.

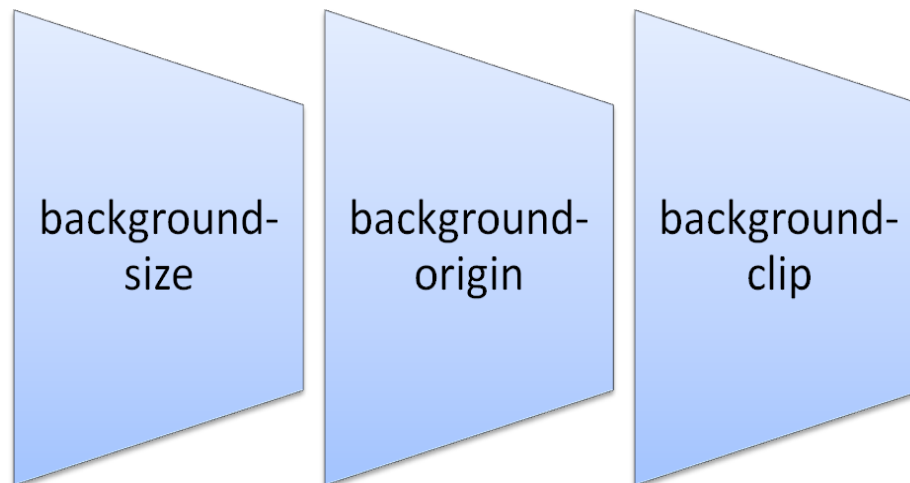


Figura 9 - background CSS3

### 10.2.1 Background-Size

A **background-size** propriedade CSS3 permite que você especifique o tamanho das imagens em segundo plano. O CSS3 nos permite reutilizar imagens em segundo plano em diferentes contextos. O tamanho pode ser especificado em comprimentos, porcentagens ou usando uma das duas palavras-chave: *contain* ou *cover*.

```
<!DOCTYPE html>
<html>
<head>
<style>
#example1 {
border: 1px solid black;
background:url(img_flwr.gif);
background-repeat: no-repeat;
padding:15px;
}

#example2 {
border: 1px solid black;
background:url(img_flwr.gif);
background-size: 100px 80px;
background-repeat: no-repeat;
padding:15px;
}
</style>
</head>
<body>

<p>Original background-image:</p>
<div id="example1">
<h2>Lorem Ipsum Dolor</h2>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
<p>Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
```



Figura 10 - Redimensionando Imagem.





### 10.2.2 background-origin

A **background-origin** propriedade CSS3 especifica onde a imagem de fundo está posicionada.

**border-box:** A imagem de fundo começa a partir do canto superior esquerdo da borda

**padding-box:** (Padrão) a imagem de fundo começa a partir do canto superior esquerdo da borda de preenchimento

**content-box:** A imagem de fundo começa a partir do canto superior esquerdo do conteúdo.

No background-origin (padding-box is default):



Figura 11 - Imagem no limite da borda



`background-origin: border-box;`

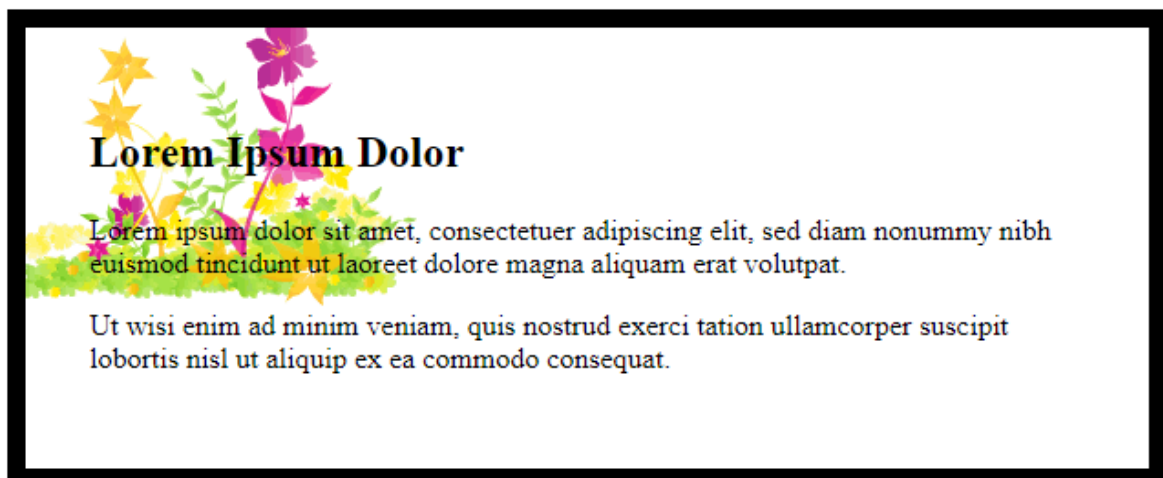


Figura 12 - Imagem ultrapassando a borda.

`background-origin: content-box;`



Figura 13 - Imagem por traz do conteúdo.

### 10.2.3 background-clip

A **background-clip** propriedade CSS3 especifica a área de pintura do plano de fundo.

**border-box:** (Padrão) o fundo é pintado para a borda externa da borda.



**padding-box:** O fundo é pintado para a borda externa do preenchimento.

**content-box:** O fundo é pintado dentro da caixa de conteúdo.

No background-clip (border-box is default):



Figura 14 - Ultrapassando a borda

background-clip: padding-box:



Figura 15 - No limite da borda.



`background-clip: content-box;`



Figura 16 - Sobre o conteúdo.

### 10.3 *Object-fit*

A *object-fit* propriedade CSS é usada para especificar como uma `<img>` ou `<video>` deve ser redimensionado para se ajustar ao seu contêiner.

Propriedades:

*fill*- Isso é padrão. O conteúdo é dimensionado para preencher a caixa de conteúdo do elemento. Se necessário, o objeto será esticado ou esmagado para se ajustar;

*contain* - O conteúdo é dimensionado para manter sua relação de aspecto enquanto se encaixa na caixa de conteúdo do elemento;

*cover*- O conteúdo é dimensionado para manter sua relação de aspecto ao preencher a caixa de conteúdo inteiro do elemento. O objeto será cortado para caber;

*none* - O conteúdo não é redimensionado;

*scale-down* - O conteúdo é dimensionado como se nenhum ou conteúdo fosse especificado .

## 11 Buttons

Podemos definir alguns aspectos com o CSS3 para um botão.

**background-color** – Cor de fundo;

**font-size** –Alterar tamanho da fonte do botão;

**Padding** –Alterar o preenchimento de um botão ;

**border-radius** –Adicionar cantos arredondados para um botão;

**hover – alterar** –O estilo de um botão quando você move o mouse sobre.



Figura 17 - Exemplo de botões.

## 12 Framework

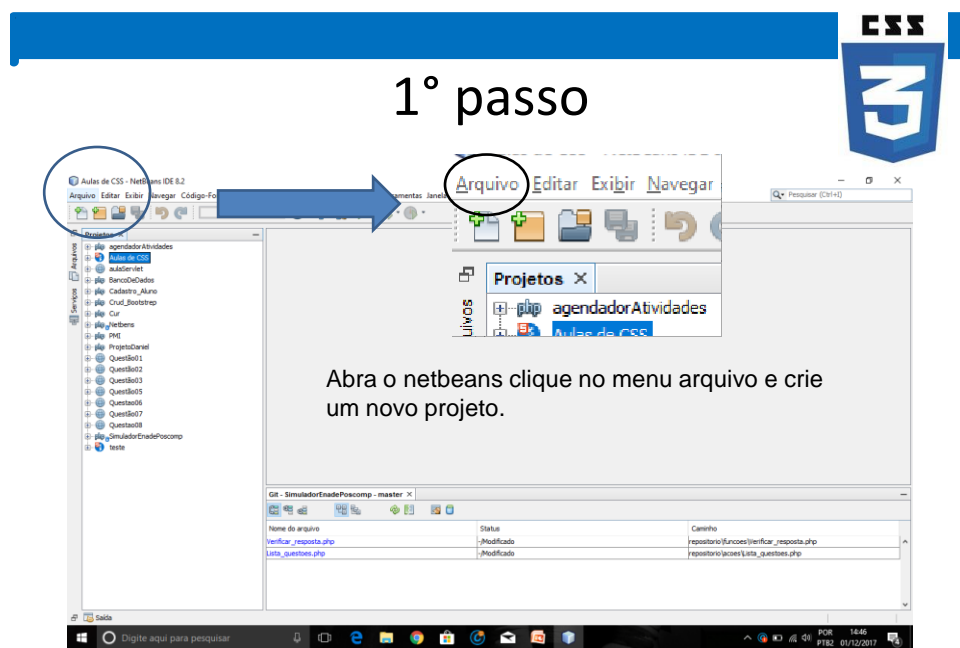
Framework é um conjunto de componentes que provêm uma estrutura básica de elementos reutilizáveis, tendo uma arquitetura consistente de funcionalidade genérica sob a qual a aplicação será construída.

Alguns tipos de **Framework** para o desenvolvimento **Front-end**. LESS Framework, PhoneGap, Twitter Bootstrap, HTML5 Boilerplate

## 13 CSS em prática.

Iremos agora estilizar um tela de login, para isso, vamos utilizar o Netbeans.

- 1º passo: Abra o netbeans e crie um novo projeto.



Curso Básico de CSS - Bruno Miranda  
- bilnp29@gmail.com

1

Figura 18 - CSS em Prática. (Passo I)

- 2º passo: Escolher projeto – Categoria: HTML5/JAVASCRIPT.  
Projeto: aplicação HTML5/ JS em seguida aperte próximo

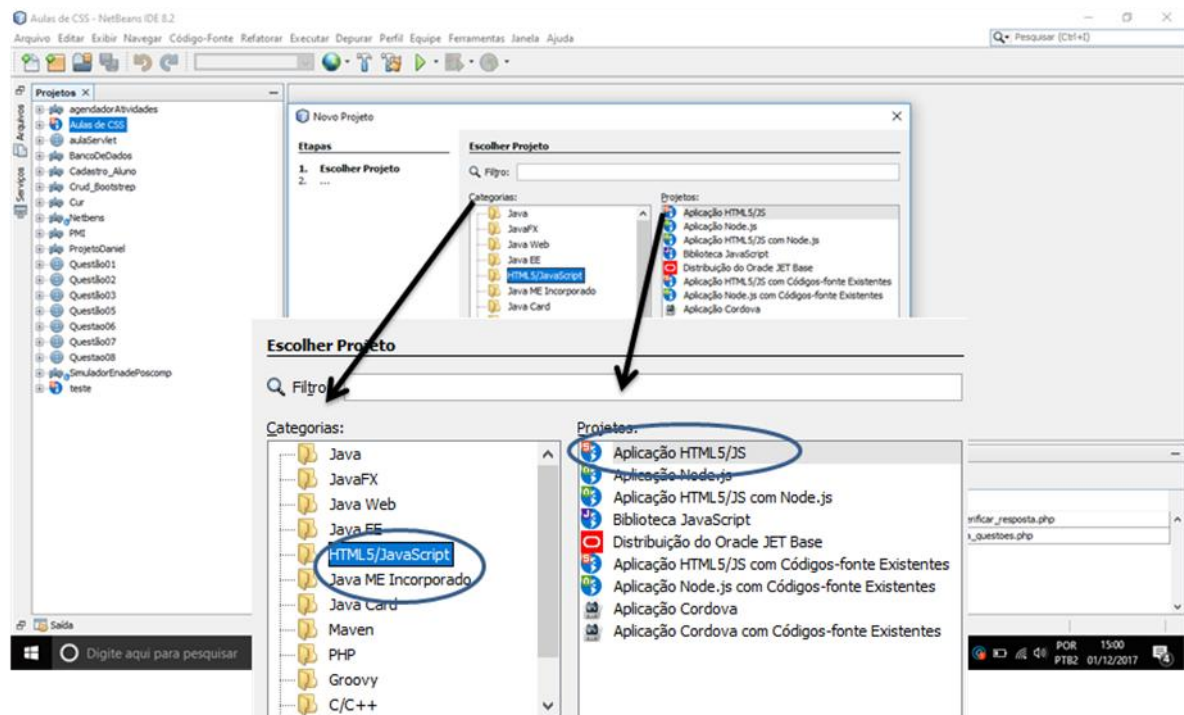


Figura 19 - CSS em prática - (Passo II)



- 3º passo: Nomear projeto.

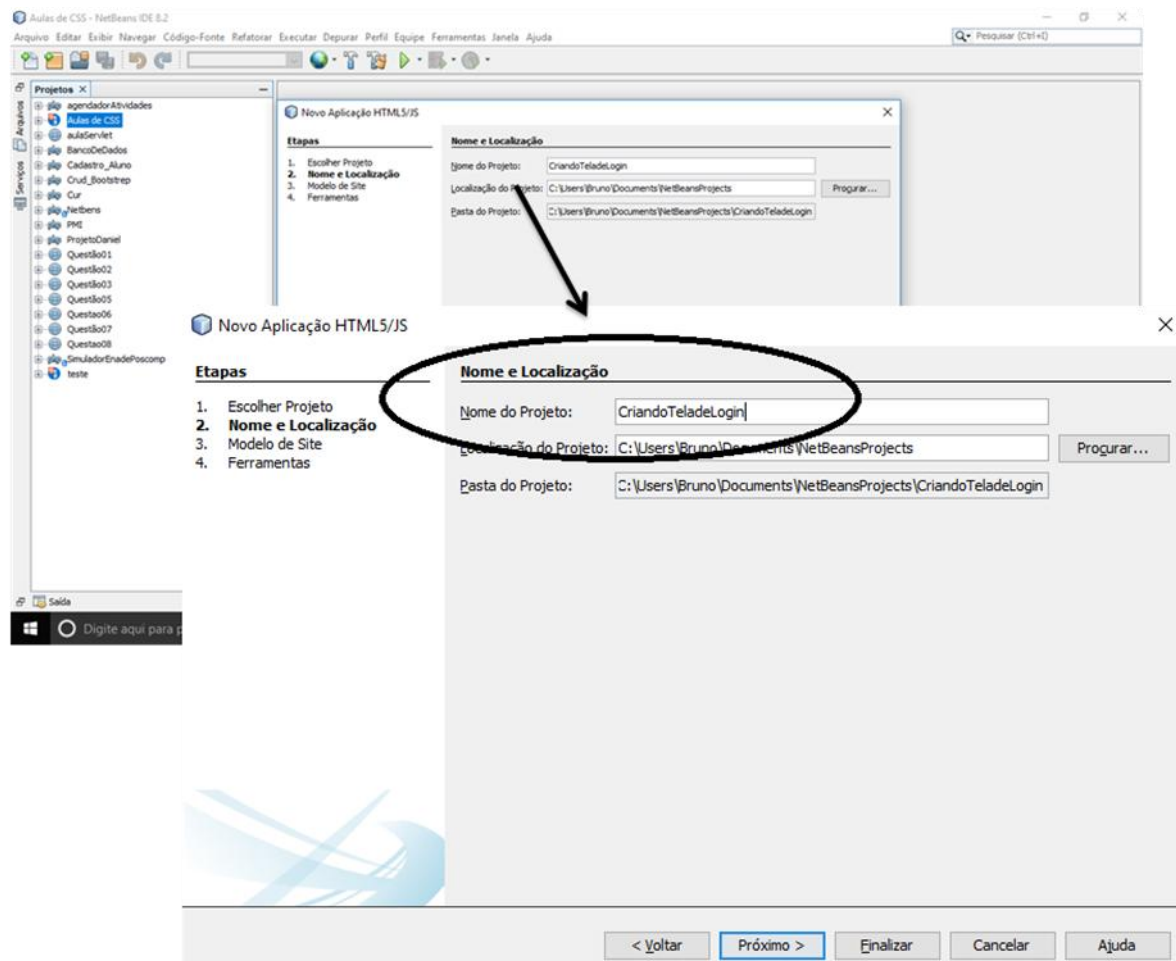
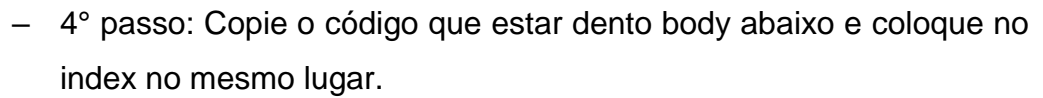


Figura 20 - CSS em prática - (Passo III)





- 5º passo: Identificar as <div> e seletores <h2>, com os atributos class e id.

**Figura 22 - CSS em Prática - (Passo V)**



- 6º Passo: Criar uma folha de estilo e inserir no arquivo .html.



- Na pasta raiz do projeto, crie uma nova pasta com o nome CSS, dentro dela crie uma folha de estilo em cascata.

Figura 23 - CSS em prática - (Passo VI)

- 7º Passo: Agora é só implementar a folha de estilo, estilizando cada tag.



```
body{
  background-image: url(imagen.jpg);
}

h2{
  text-align: center;
  font-size: 30px;
  font-family: sans-serif, serif;
  color: black;
}

.container {
  height: 350px;
  width: 280px;
  padding: 10px 20px;
  margin-left: 500px;
  margin-top: 100px;
  background: transparent;
}
```

```
Label {
  text-align: left;
  text-transform: uppercase;
  font-family: sans-serif, serif;
  font-style: normal;
  font-size: 16px;
  font-weight: bold;
  border: 1px;
}
```

```
#login{
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
  font-family: sans-serif, serif;
  font-size: 16px;
  background: transparent;
  border-radius: 5px;
}
```

**Figura 24 - CSS em prática - Passo VII**

```
#senha{

width: 100%;
padding: 12px 20px;
margin: 8px 0;
box-sizing: border-box;
font-family: sans-serif, serif;
font-size: 16px;
background: transparent;
border-radius: 5px;

}
```

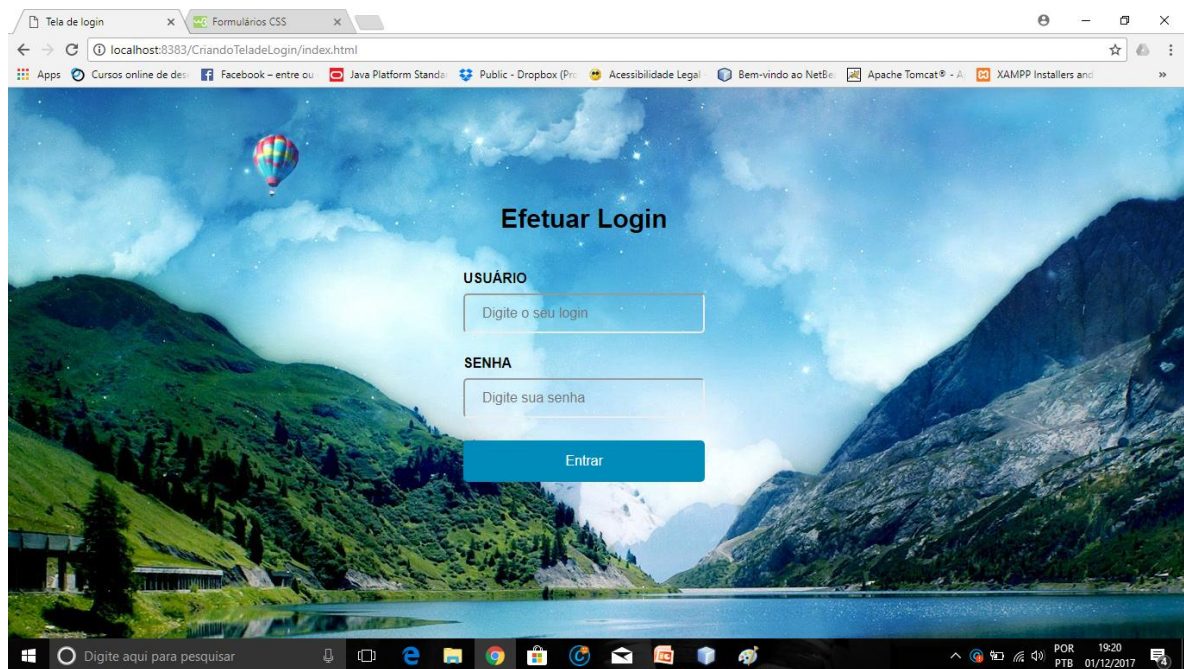
```
#botaoLogin{

background-color: #008CBA; /* Azul */
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
width: 100%;
border-radius: 5px;

}
```

**Figura 25 - CSS em prática - (Passo VII)**

– 8º Passo: Ao final teremos a seguinte tela.



**Figura 26 - CSS em prática - (Passo VIII)**



## 14 Referencias.

- CAELUM.Desenvolvimento web com html, css e javascript , São Paulo: banco de dados. Disponível em: < <https://www.caelum.com.br/apostila-html-css-javascript/> >.Acesso em: 30/11/2017.
- W3SCHOOLS .COM. Disponível em: < <https://www.w3schools.com/css/default.asp>>. Acessado em: 30/11/2017.
- Cssreset.com. Disponível em <:<https://cssreset.com/> > Acessado em: 30/11/2017.
- Devmedia.com. Disponível em:< <https://www.devmedia.com.br/como-utilizar-a-tecnica-css-reset/26797> >. Acessado em: 30/11/2017.