

CSS

Cascading Style Sheets

O que é?

- Não é programação!
- Linguagem de estilização para web
- Baseada em seletores, propriedades e valores
- Texto puro (não depende de editores)

Para que serve?

- Define o layout de documentos HTML.
- Controla fontes, cores, margens, linhas, alturas, larguras, imagens de fundo, posicionamentos, etc.

Seletores

- É a identificação do(a) elemento(a) aos quais o estilo deve ser aplicado.
- Um seletor pode ser do tipo Tag, ID (Único) ou Class
- Podem ser combinados para facilitar a localização do elemento
- No seletor são aplicadas uma ou mais `propriedade` `:` `valor` `;`

Seletor de tag

`<div>`

Texto exemplo

`</div>`

`<p>`

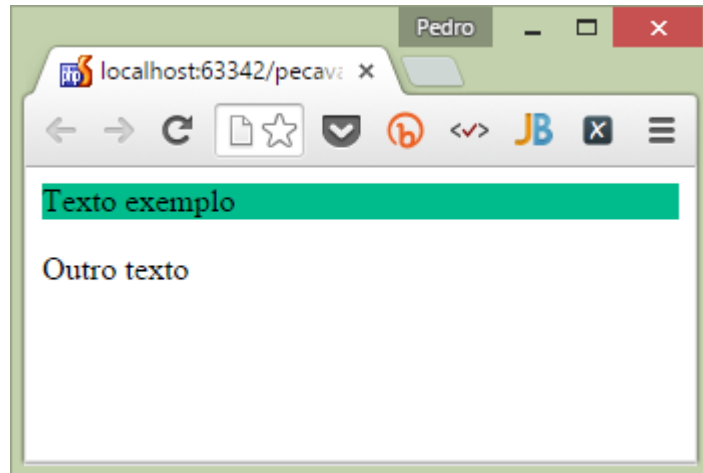
Outro texto

`</p>`

`div {`

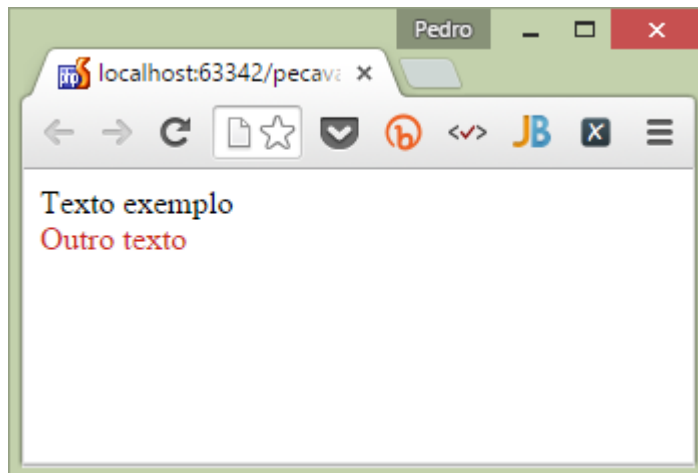
background-color: #00bc8c;

`}`



Seletor de ID (ID É ÚNICO)

```
<div>  
  Texto exemplo  
</div>  
<div id="vermelho"> DEVE SER ÚNICO  
  Outro texto  
</div>  
  
#vermelho {  
  color: #cd2a19;  
}
```



Seletor de Class

```
<div class="negrito"> PODE REPETIR
```

Primeiro texto

```
</div>
```

```
<div>
```

Segundo texto

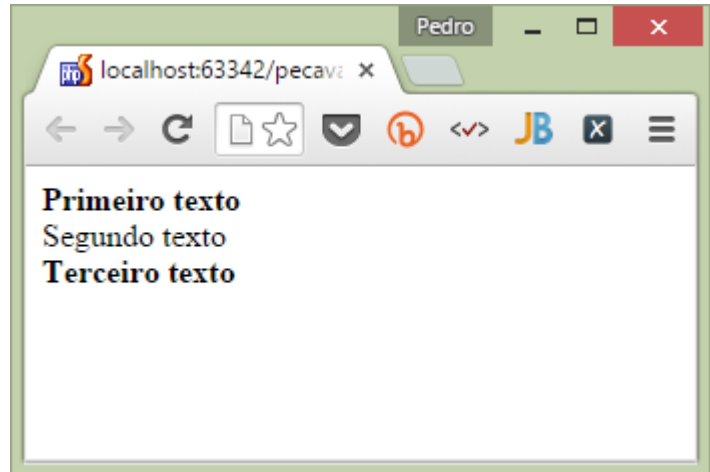
```
</div>
```

```
<div class="negrito"> PODE REPETIR
```

Terceiro texto

```
</div>
```

```
.negrito {  
    font-weight: bold;  
}
```



Combinando seletores

```
<p class="azul">
```

Primeiro texto

```
</p>
```

```
<div class="azul">
```

Primeiro texto

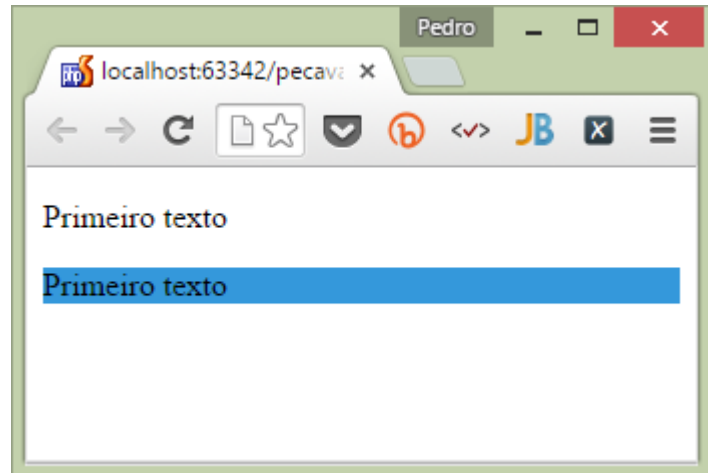
```
</div>
```

```
div.azul {
```

SEM ESPAÇOS

```
background-color: #3498db;
```

```
}
```



Hierarquia de seletores

```
<div id="header">
```

```
  <p>
```

Título

```
  </p>
```

Subtítulo

```
</div>
```

```
<p>
```

Texto comum

```
</p>
```

```
#header p {
```

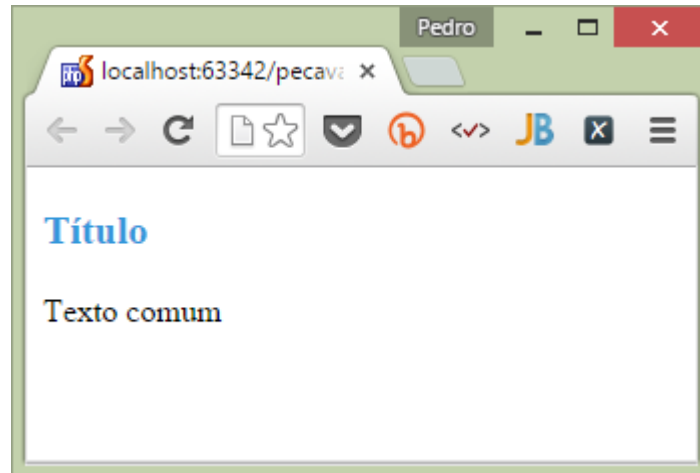
COM ESPAÇO

```
  color: #3498db;
```

```
  font-size: 20px;
```

```
  font-weight: bold;
```

```
}
```



Onde fica o CSS?

O CSS é carregado junto com o HTML, sendo aplicado a todo o documento, ajudando a renderizar corretamente a aplicação.

Existem 3 modos de incluir CSS à um documento HTML:

- **Inline**
- Internamente (via tag)
- Externamente (via folha de estilos)

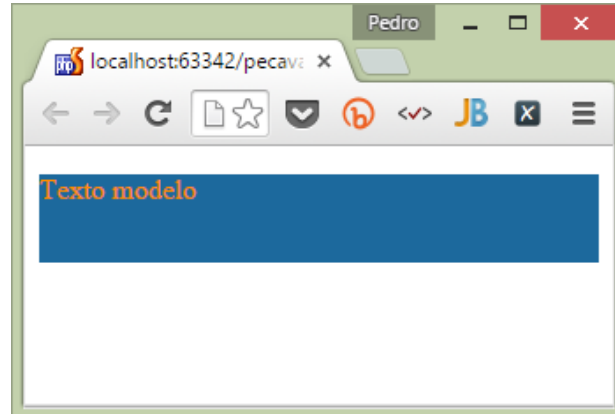
CSS Inline

A propriedade CSS é aplicada diretamente em uma tag HTML. Apesar de fácil, não deve ser utilizado, pela falta de manutenibilidade e semântica.

```
<p style="background-color: #1c699d; color: #ff8c00; height: 50px">
```

Texto modelo

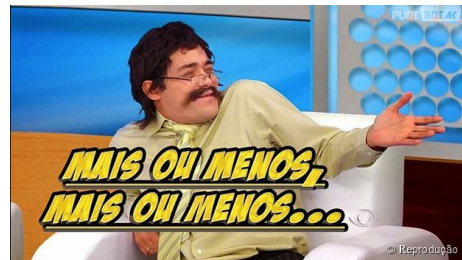
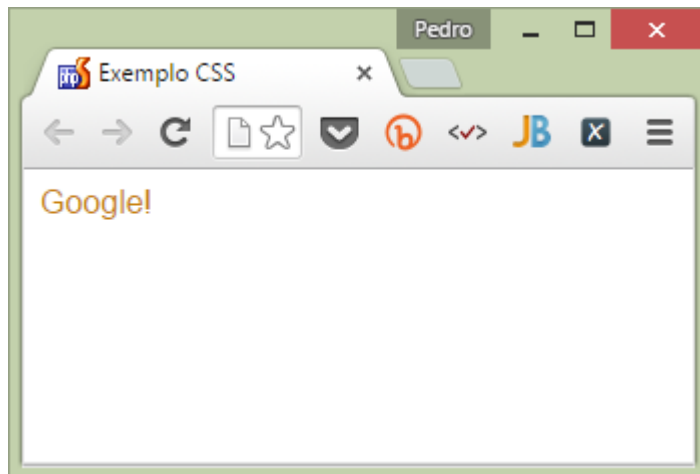
```
</p>
```



CSS Interno

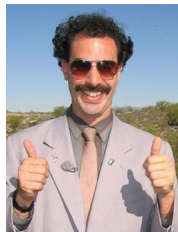
As propriedade CSS são definidas em uma tag `<style>`, dentro do documento HTML.

```
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>Exemplo CSS</title>
  <style>
    body a {
      text-decoration: none;
      font-family: Helvetica;
      color: #c87f0a;
    }
  </style>
</head>
<body>
  <a href="http://google.com">Google!</a>
</body>
</html>
```

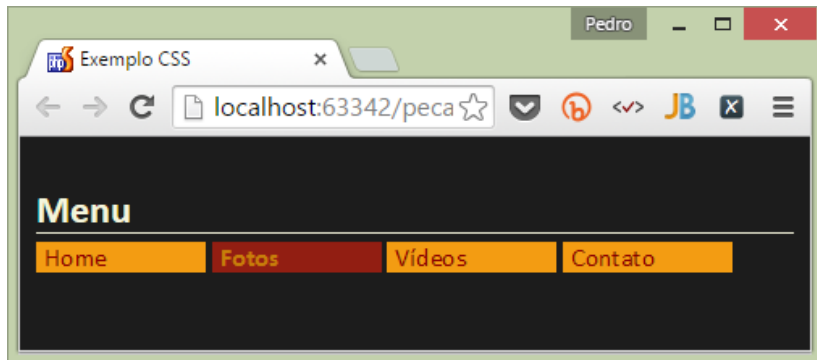


CSS Externo

Uma folha de estilos é “linkada” ao documento HTML, isolando as linguagens e centralizando os estilos da página. É o modo recomendado.



```
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>Exemplo CSS</title>
  <link rel="stylesheet" type="text/css" href="css/styles.css" />
</head>
<body>
<h2>Menu</h2>
<div>
  <ul>
    <li><a href="#">Home</a></li>
    <li class="selected"><a href="#">Fotos</a></li>
    <li><a href="#">Vídeos</a></li>
    <li><a href="#">Contato</a></li>
  </ul>
</div>
</body>
</html>
```



Propriedades CSS

- Propriedades podem ser aplicadas a qualquer elemento, mas nem todos os elementos suportam todas as propriedades
- Cada propriedade possui diversos valores, mudando seu comportamento.
- As unidades de medida mais comuns são px, % e em
- Uma propriedade pode sobrescrever a outra
- Existem muitas, muitas, muitas...

Color

Define a cor do primeiro plano de um elemento

```
h1 {  
  color: #ff0000;  
}  
  
<h1>  
  Título  
</h1>
```



Background-color

Define a cor de fundo de um elemento

```
body {  
  background-color: #deb887;  
}  
  
<body>  
  <h2>  
    Título  
  </h2>  
</body>
```

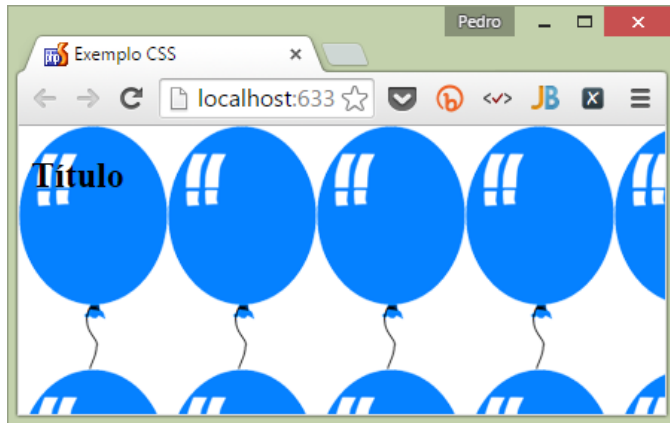


Background-image

Define a imagem de fundo de um elemento. Por padrão, repete vertical e horizontalmente

```
body {  
  background-image: url('balloon-blue.png');  
}
```

```
<body>  
  <h2>  
    Título  
  </h2>  
</body>
```



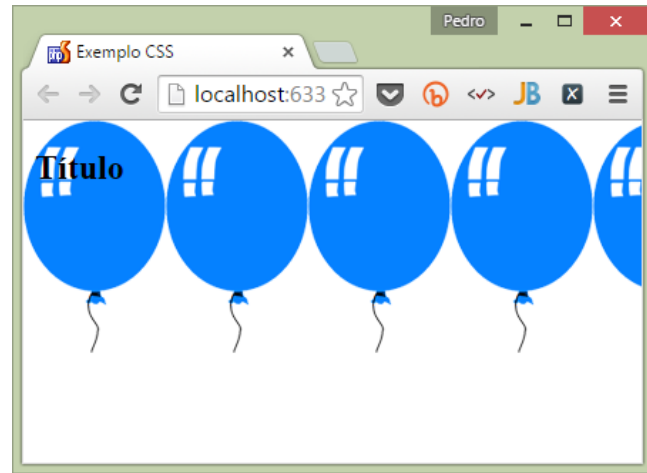
Background-repeat

Controla a repetição da imagem de fundo.

```
body {  
  background-image: url('balloon-blue.png');  
  background-repeat: repeat-x  
}
```

Valores aceitos:

- repeat-x
- repeat-y
- repeat
- no-repeat



Background-[outros]

Controlam diversas outras funcionalidades do fundo.

- `background-attachment`: controla se o fundo é fixo na página ou rola junto com o conteúdo
- `background-position`: define o posicionamento do fundo, em relação as bordas do elemento
- É possível juntar uma ou mais propriedades de background na tag principal `background`

Propriedades de texto

- **text-align**: define o alinhamento do texto, em relação ao elemento pai.
 - right, center, left, justify
- **text-decoration**: permite adicionar “efeitos” e “decorações” ao texto, como sublinhado, riscado, etc.
 - underline, line-through, overline
- **letter-spacing**: define o espaçamento entre letras (px, em)
- **text-transform**: controla a capitalização do texto
 - capitalize, uppercase, lowercase, none
- **word-spacing**: define o espaçamento entre palavras (px, em)
- +

Propriedades de fonte

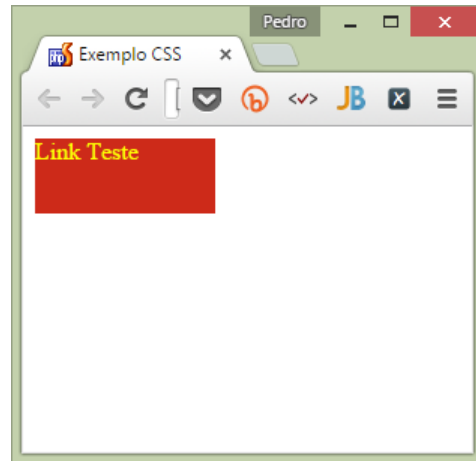
- **font-size**: define tamanho da fonte exibida (px, em)
 - 10px, 2em
- **font-weight**: define o “peso” da fonte exibida
 - normal, bold, bolder, lighter, 100-900
- **font-family**: define uma lista de fontes e sua prioridade na apresentação de um elemento
 - arial, verdana, sans-serif, “Times New Roman”, serif
- **font-style**: define se a fonte é normal, itálica ou oblíqua
 - normal, italic, oblique
- +

Dimensões

As dimensões de um elemento são definidas pelas propriedades `height` (altura) e `width` (largura)

```
div {  
  background-color: #cd2a19;  
  color: yellow;  
  width: 120px;  
  height: 50px;  
}
```

```
<a href="http://google.com">  
  <div>  
    Link Teste  
  </div>  
</a>
```



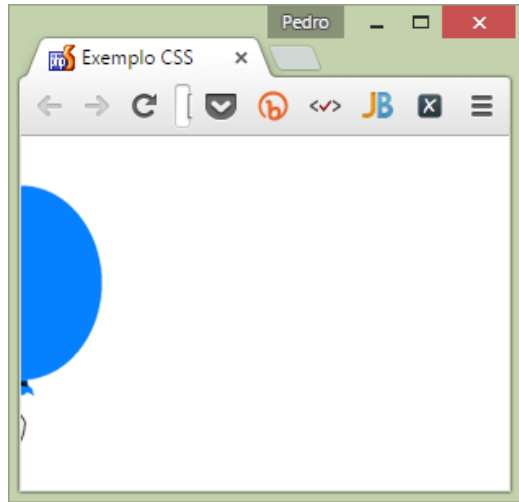
Margin

Define a margem externa (transparente) do elemento

```
img {  
  margin-left: -50px; PODE NEGATIVO  
  margin-top: 30px;  
}
```

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

Abreviações **margin:** Top Right Bottom Left, **margin:** All, etc

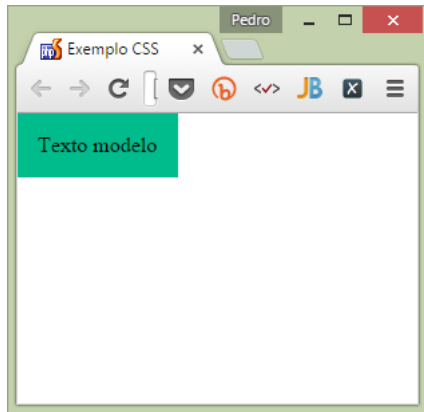


Padding

Define a margem interna do elemento

```
a{  
  background-color: #00bc8c;  
  padding: 15px;  
}  
  
<body>  
  <a>Texto modelo</a>  
</body>
```

Permite abreviações, assim como o `margin`



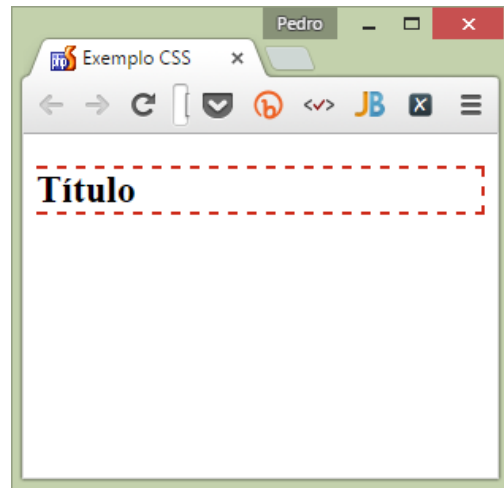
Border

Define a borda externa do elemento

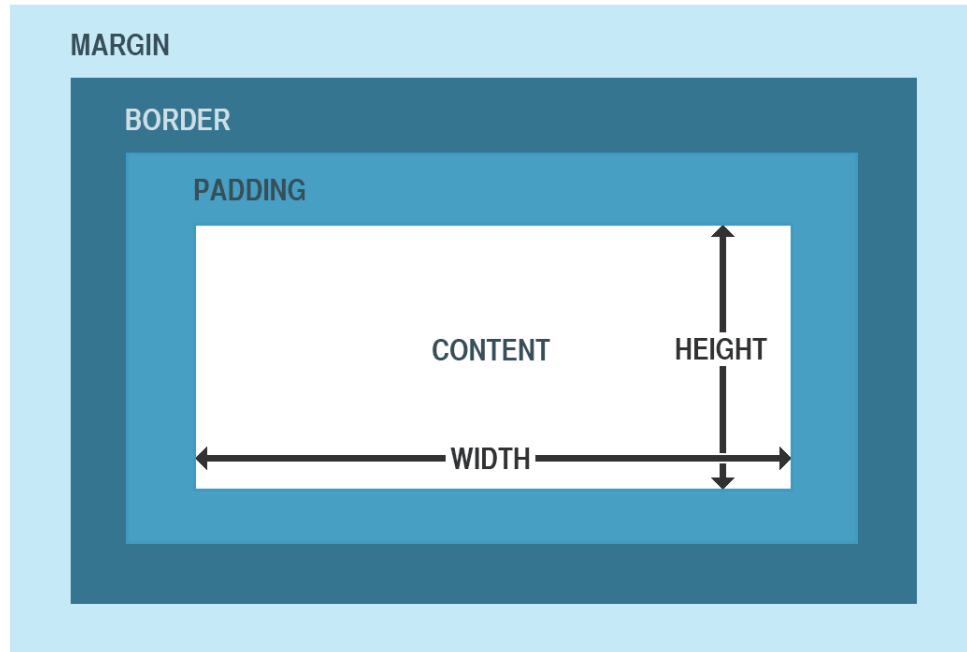
```
h2{  
  border-width: 2px;  
  border-style: dashed;  
  border-color: #cd2a19;  
  border-left: none;  
}
```

```
<body>  
  <h2>Título</h2>  
</body>
```

Permite abreviações, assim como o `margin` e `padding`



O Box Model



Float & Clear

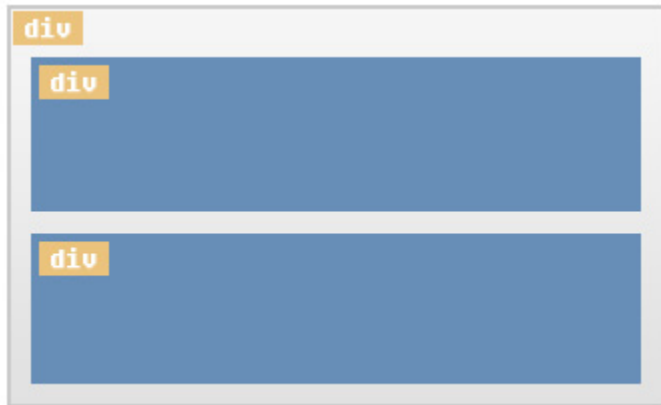
Float define se um elemento deve ou não flutuar.

Clear define o comportamento de elementos flutuantes.



Baby steps...

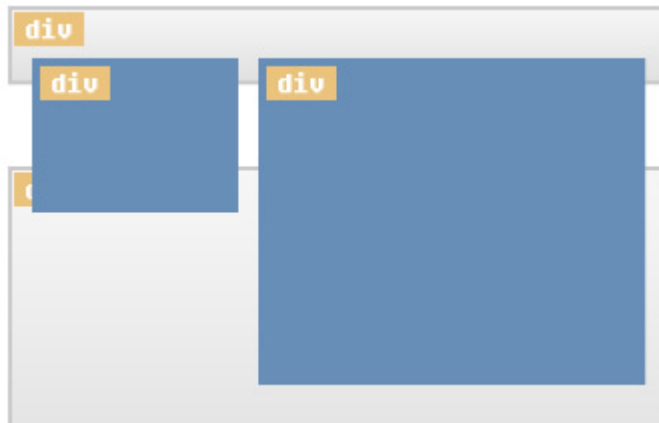
```
<div class="tudo">  
  <div class="menu">Menu</div>  
  <div class="conteudo">Conteúdo</div>  
</div>
```



Float

O Float é uma propriedade que faz o objeto flutuar à esquerda ou à direita do restante do conteúdo.

```
.menu {  
  float: left;  
  width: 30%;  
}  
  
.conteudo {  
  float: right;  
  width: 60%;  
}
```

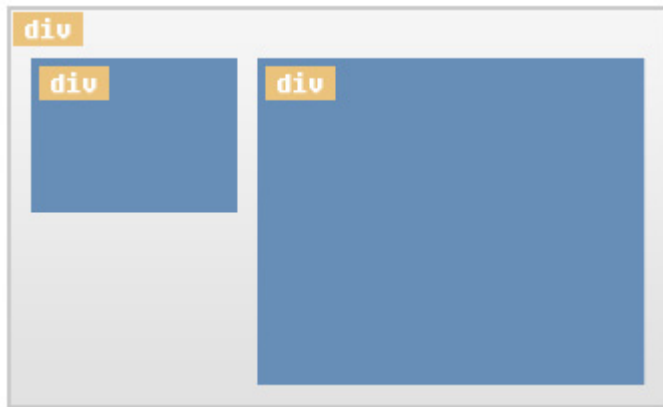


Clear

O clear especifica quais lados de um elemento não devem permitir flutuação.

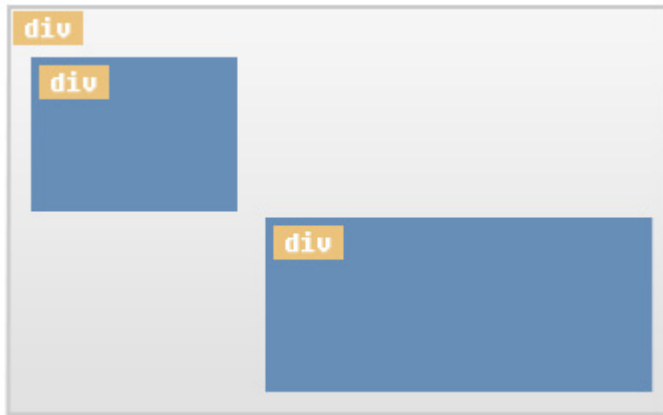
```
<div class="tudo">  
  <div class="menu">Menu</div>  
  <div class="conteudo">Conteúdo</div>  
  <div class="clear"></div>  
</div>
```

```
.clear {  
  clear: both;  
}
```



As vezes fica estranho

É muito comum, ainda mais no início, que o layout fique quebrado ao usar a propriedade float. Verificar sempre as dimensões e quais elementos estão sendo flutuados.



Pra tudo tem salvação!

Existem diversos Frameworks CSS que facilitam a nossa vida na hora de desenvolver um layout. Os mais populares são:

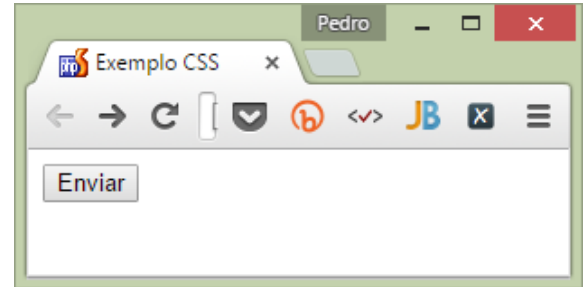
- [Bootstrap](#)
- [Foundation](#)
- [Semantic UI](#)
- [Pure](#)
- [Ulkit](#)

Bootstrap

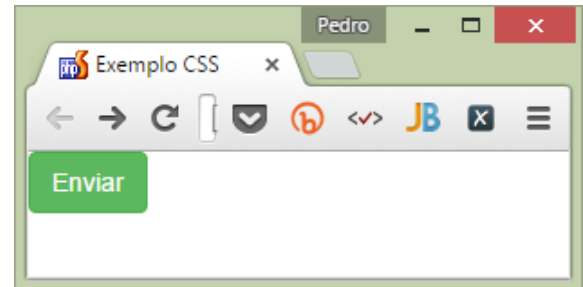
- É composto de uma folha de estilos base (.css), um tema opcional (.css), um arquivo de scripts (.js) e algumas fontes (opcionais)
- Depende do jQuery para funcionar completamente
- É constituído basicamente de inúmeras classes pré definidas, que permitem a criação de layouts bonitos e bem estruturados, com um mínimo de trabalho.

Praticidade

`<input type="button" value="Enviar">`



`<input type="button" value="Enviar" class="btn btn-success">`



Estrutura pré definida

- A principal vantagem de usar um framework CSS, é a facilidade de estruturar o conteúdo do site
- O Bootstrap, por exemplo, inclui um sistema de “grade” fluído e responsivo, que permite definir até 12 colunas distintas, que se ajustam automaticamente de acordo com o tamanho da tela.

E por fim...

