

Bootstrap

Framework front-end para
aplicações web e mobile

versão 3.0
leanpub.com/livro-bootstrap

```
<div class="container">
<div class="navbar-header">
  <button type="button" class="navbar-toggle"
    data-toggle="collapse" data-target=".navbar-collapse">
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
  </button>
  <a class="navbar-brand" href="#">Project name</a>
</div>
</div>
```

Daniel Schmitz

Bootstrap 3

Framework front-end para desenvolvimento web e mobile

Daniel Schmitz

This book is for sale at <http://leanpub.com/livro-bootstrap>

This version was published on 2014-01-21



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

©2013 - 2014 Daniel Schmitz

Also By **Daniel Schmitz**

AngularJS na prática

Conteúdo

Introdução	1
O que é Bootstrap	1
O que podemos fazer com Bootstrap	1
O que não é Bootstrap	1
O que você precisa para testar o Bootstrap	2
Exemplos do livro	2
Conceitos iniciais	3
Instalação	3
Tudo compilado	3
CDN	3
Não esqueça do jQuery	4
Template básico	4
Características do Bootstrap	6
Doctype	6
Viewport	6
Desenvolvimento web responsivo	6
A classe container	7
Grid System	8
Introdução	8
Diferenciando o tamanho dos dispositivos	8
Trabalhando com offsets	12
Colunas aninhadas	13
Tipografia	16
Alinhamento	16
Criando ênfase ao texto com cores	16
Abreviação	17
Blockquotes (citações)	17
Listas	18
Tabelas	20
Tabela simples	20

CONTEÚDO

Listras em zebra	22
Bordas	22
Destacando linhas	23
Tabelas menores (condensadas)	23
Linhas contextuais	24
Mesclando classes	24
Formulários	26
Compreendo o form-group	26
Formulário in-line	27
Formulários horizontais	28
Componentes suportados	30
Checkbox e radio na horizontal	30
Controles estáticos	30
Campos desabilitados	31
Validações	31
Tamanho dos campos	32
Texto para ajuda	33
Botões	33
Tamanho dos botões	34
Botões em grupo	34
Ícones	35
Botões dropdown (menu)	35
Inputs diferenciados com group	36

Introdução

O que é Bootstrap

O Bootstrap é, basicamente, um kit básico contendo diversos componentes web prontos para que você possa desenvolver a sua aplicação web/mobile de forma mais fácil e objetiva, sem necessitar um sólido conhecimento em Javascript e CSS para isso.

Este framework está estritamente ligado ao front-end e a camada View do padrão MVC. Ele não interfere em nada outras tecnologias ou linguagens, pois o seu foco é o “desenho” da sua aplicação, sendo “montado” exclusivamente com html puro.

O Bootstrap nasceu como uma biblioteca ligada ao Twitter (o nome era “Twitter Bootstrap”), no qual a equipe de desenvolvimento usa para o desenvolvimento desta aplicação, e como o resultado ficou muito bom, resolveram disponibilizar de forma gratuita para a comunidade. Com o avanço do produto, tendo ajuda de diversos desenvolvedores, o framework front-end evolui para esta terceira geração, na qual chamamos apenas de Bootstrap.

O que podemos fazer com Bootstrap

O Bootstrap é utilizado para desenhar telas em html, que serão acessadas via navegador web ou dispositivo mobile. Tudo que você precisa saber sobre Bootstrap é, na verdade, html. Com ele, podemos criar sites inteiros e estruturas complexas, mas que podem ser acessadas facilmente em diferentes dispositivos. Também podemos criar telas com tabelas, formulários, janelas e controles complexos. Pode-se facilmente criar um sistema web, apenas utilizando este framework, tomando as devidas precauções.

Também é possível criar um site inteiro, apenas com o Bootstrap, como iremos rever nos diversos exemplos desta obra.

O que não é Bootstrap

Qualquer tarefa que não envolva o desenho da tela, não está relacionado com Bootstrap. Isso envolve, por exemplo, realizar o bind entre variáveis javascript e seus respectivos campos input, ou calcular datas ou fornecer formas de validar algum processo. Tudo que não é visual ou não é ligado ao html, o Bootstrap não tem domínio.

O que você precisa para testar o Bootstrap

Como todo o framework foi criado a partir de css e javascript, não há necessidade nenhuma de utilizar um servidor web para executar as páginas html

Exemplos do livro

Todos os exemplos do livro estão no github: <https://github.com/danielps/livro-bootstrap>

Conceitos iniciais

Instalação

A instalação do framework está diretamente ligada a incluir no documento html que você precisa criar algumas bibliotecas Javascript e css. Existem diversas formas para que você possa incluir estas bibliotecas, use a forma que achar mais conveniente para o seu projeto.

Tudo compilado

Javascript é uma linguagem interpretada, mas quando dizemos “compilado” estamos fazendo uma referência a forma como o código está organizado, isto é, todo o código javascript está em uma única linha, sem espaços em branco e com variáveis ilegíveis para nós humanos. Assim, temos um arquivo javascript/css menor e um carregamento mais rápido da biblioteca.

O primeiro passo para instalar Bootstrap é baixar a biblioteca acessando a url <http://getbootstrap.com/getting-started/> e clicar no botão Download Bootstrap. Nesta obra, estaremos utilizando a versão 3.0.0, mas talvez na época que esteja baixando a biblioteca, pode-se estar em uma versão superior.

Após baixar o arquivo `bootstrap-3.0.0-dist.zip` você pode extraí-lo para o seu projeto e começar a usar o framework, conforme veremos logo adiante.

CDN

Outra forma de incluir as bibliotecas do Bootstrap em seu projeto é a utilização de um serviço chamado CDN, que é uma forma de distribuir conteúdo livre através de servidores espalhados pelo mundo. Ao invés de utilizar um servidor central para o download das bibliotecas, o CDN funciona como uma forma de obter esses arquivos do servidor mais próximo ao cliente. Para utilizar o CDN no Bootstrap, você pode utilizar o seguinte template:


```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootst\
rap.min.css">

<!-- Optional theme -->
<link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootst\
rap-theme.min.css">

<!-- Latest compiled and minified JavaScript -->
<script src="//netdna.bootstrapcdn.com/bootstrap/3.0.0/js/bootstrap.min.js"></scr\
ipt>
```

Não esqueça do jQuery

O Bootstrap está intimamente ligado ao jQuery, que é um framework Javascript para manipulação da DOM do documento HTML. O Bootstrap precisa do jQuery, então você deve incluí-lo no seu projeto. O jQuery pode ser baixado pelo site <http://jquery.com>, sendo utilizada a versão 1.10.2.

Template básico

Em todos os nossos exemplos, estaremos utilizando o template a seguir. A forma como os arquivos javascript e css estão organizados pode ser alterada de acordo com as necessidades do seu projeto, mas lembre-se de alterar o caminho no template para que tudo funcione perfeitamente.

Os arquivos do template obedecem a seguinte estrutura:

```
bootstrap/
├─ template.html
├─ css/
│   ├─ bootstrap.css
│   ├─ bootstrap.min.css
│   ├─ bootstrap-theme.css
│   └─ bootstrap-theme.min.css
├─ js/
│   ├─ bootstrap.js
│   ├─ bootstrap.min.js
│   └─ jquery-1.10.2.min
└─ fonts/
    ├─ glyphsicons-halflings-regular.eot
    ├─ glyphsicons-halflings-regular.svg
    ├─ glyphsicons-halflings-regular.ttf
    └─ glyphsicons-halflings-regular.woff
```

Veja que existem duas versões para cada arquivo css e js. Isso acontece para que você possa escolher qual versão deseja. A minificada (.min.js por exemplo) é bem menor e carrega mais rápido, mas é ilegível. A versão normal é maior e está legível.

Esta estrutura é a mesma criada quando extraímos o arquivo bootstrap-3.0.0-dist.zip, sendo que incluímos a biblioteca jquery na pasta js e criamos o arquivo template.html com o seguinte código html:

template.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Bootstrap Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="css/bootstrap.min.css" rel="stylesheet" media="screen">
  </head>
  <body>
    <h1>Hello, world!</h1>

    <script src="js/jquery-1.10.2.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

Este é um template básico que podemos utilizar para a maioria dos projetos. Todo o conteúdo HTML virá após a tag <body> e antes da inclusão do script jquery. Não inclua o arquivo de script jquery e bootstrap na seção <head> do seu documento html, pois isso pode degradar o tempo de carregamento da página.

Com o template pronto, podemos iniciar o nosso estudo sobre as diversas características que o Bootstrap possui.

Sempre deixe o projeto template vazio, copiando e colando os arquivos em um novo projeto.

Características do Bootstrap

Este framework possui um conjunto extenso de funcionalidades e particularidades que devem ser compreendidas para que possamos tirar utilizar todos os recursos com eficiência. Nos tópicos a seguir, iremos ilustrar cada característica com exemplos, para que possamos compreender melhor o seu funcionamento.

Doctype

Para que o Bootstrap funcione corretamente, é obrigatório o uso da tag doctype no documento html.

Viewport

Nesta versão, o Bootstrap foi remodelado para se adequar perfeitamente ao desenvolvimento mobile. Com isso, é necessário informar ao navegador a metatag viewport, que indica a área onde o navegador o site é carregado. Isso é necessário porque, sem ele, o navegador do dispositivo mobile iria deixar o tamanho de todo o site maior que a área disponível do dispositivo.

O Viewport é configurado na tag <head> do documento html, da seguinte forma:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Como podemos ver, o viewport acima irá renderizar toda a página html no tamanho do dispositivo, em uma escala 1:1.

Desenvolvimento web responsivo

O termo responsivo vem sendo constantemente empregado no desenvolvimento de websites e sistemas web. Ele sugere que todo o conteúdo de um site deve se adequar a largura do dispositivo, que pode ser um celular, um tablet ou um navegador.

O bootstrap tem como uma de suas metas tornar o desenvolvimento de interfaces responsivas algo mais simples, que demanda apenas um pouco de configuração.

Por exemplo, em uma imagem, é possível torná-la responsiva através da atribuição da seguinte classe a tag :

```

```

Existem diversas outras técnicas que iremos abordar ao longo desta obra.

A classe container

No Bootstrap, podemos criar diversos elementos `<div>` e assim formar uma estrutura para o desenho da página. Para que o conteúdo possa ser centralizado corretamente, usamos a classe `container`, que em geral é a primeira classe que usamos para definir o conteúdo da página.

O uso do `container` é [exemplificado a seguir](https://github.com/danielps/livro-bootstrap/blob/master/exemplos/classe-container.html)¹. Lembre-se de testá-lo no navegador:

Classe Container

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Bootstrap Template</title>
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <link href="css/bootstrap.min.css" rel="stylesheet" media="screen">
```

```
  </head>
```

```
  <body>
```

```
    <h1>Hello, world!</h1>
```

```
    <div class="container">
```

Aqui entra o conteúdo. A classe `container` adiciona um box centralizado no elemento atual.

```
      <br/><br/>
```

```
    <div class="container">
```

Aqui entra outro conteúdo. Ele está centralizado em relação ao `container` pai. Você pode usar o chrome/firefox e investigar a margem que é criada dinamicamente pelo Bootstrap, de forma a deixar tudo em ordem.

```
      <br/><br/>
```

```
    <div class="container">
```

Faça isso e diminua gradativamente a largura do navegador. Você irá reparar que, a medida que a largura diminui, o conteúdo é reposicionado

```
      </div>
```

```
    </div>
```

```
  </div>
```

```
  <script src="js/jquery-1.10.2.min.js"></script>
```

```
  <script src="js/bootstrap.min.js"></script>
```

```
</body>
```

```
</html>
```

¹<https://github.com/danielps/livro-bootstrap/blob/master/exemplos/classe-container.html>

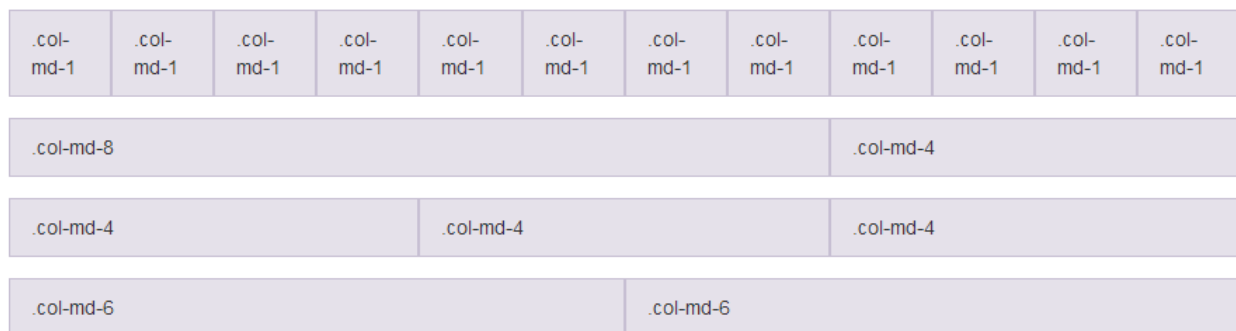
Grid System

Introdução

O Bootstrap trabalha com um enquadramento de divs na tela chamado de grid, no qual é possível criar blocos horizontais de conteúdo. A largura destes blocos não é dimensionada através de porcentagens ou pixels, mas sim através de um sistema de 12 colunas.

Neste sistema, uma “linha” da tela pode ser separada em 12 blocos, ou colunas, e através do Bootstrap é possível configurar estes blocos para se adequarem ao tamanho que desejarem.

A figura a seguir ilustra este processo.



Grid System

O sistema de grid é a base para que você possa desenhar uma tela que seja compatível tanto com dispositivos mobile quanto desktop. O uso do grid system é importante porque não será você que irá definir os tamanhos e margens de cada bloco, mas sim o framework, de acordo com a largura atual da tela. Isso significa que o Bootstrap pode até definir que um bloco irá ficar abaixo do outro, mesmo tendo a configuração horizontal.

Diferenciando o tamanho dos dispositivos

Para que possamos ver um pouco de código em ação, é preciso ainda conhecer outro conceito importante neste sistema. Nesta versão do Bootstrap, você pode configurar o tamanho de cada bloco de acordo com o tamanho do dispositivo em questão.

O tamanho da tela é determinado através de 4 tipos

- Muito Pequeno: Definido através da classe `.col-xs-X`, onde X é um número de 1 a 12

- Pequeno: Definido através da classe `.col-sm-X`, onde X é um número de 1 a 12
- Médio: Definido através da classe `.col-md-X`, onde X é um número de 1 a 12
- Grande: Definido através da classe `.col-lg-X`, onde X é um número de 1 a 12

Cada tipo é definido através da largura do dispositivo, da seguinte forma:

- Muito Pequeno: Até 750 pixels
- Pequeno: De 750 à 970 pixels
- Médio: De 970 à 1170 pixels
- Grande: Acima de 1170 pixels

Vamos codificar um exemplo simples, criando 12 blocos de texto para uma página web. Neste exemplo, usamos a configuração `devices` muitos pequenos e outra para `devices` Grandes.

Grid System

[illegible]

```
<div class="col-lg-1">.col-lg-1</div>
<div class="col-lg-1">.col-lg-1</div>
<div class="col-lg-1">.col-lg-1</div>
</div>
</div>
```

Você pode ver a página completa deste exemplo [neste link](#)² e abrir o arquivo no seu navegador. Para testar como o framework se comporta, redimensione a largura do navegador de forma a diminuí-la gradativamente. Em um certo momento, você verá que os blocos criados com `.col-lg` serão quebrados, ficando um abaixo do outro.

Mas porque isso aconteceu? Como os blocos estavam configurados para serem exibidos em uma tela “grande”, ao menor sinal de que o conteúdo não poderá mais ser exibido, o Bootstrap se encarrega de garantir que os blocos de texto devam ser exibidos, colocando-os um abaixo do outro.

Já nas divs configuradas através do `col-xs-1` não são quebradas, porque o framework espera que elas sejam visualizadas através de um dispositivo pequeno.

Para melhorar o entendimento ao processo, vamos supor a seguinte situação. Imagine que, em uma tela maior (desktop) você deseja criar dois blocos de texto, sendo que o primeiro bloco é maior que o segundo. Agora, caso a mesma página seja vista em um tablet, o ideal seria exibir os conteúdos em cada metade da tela.

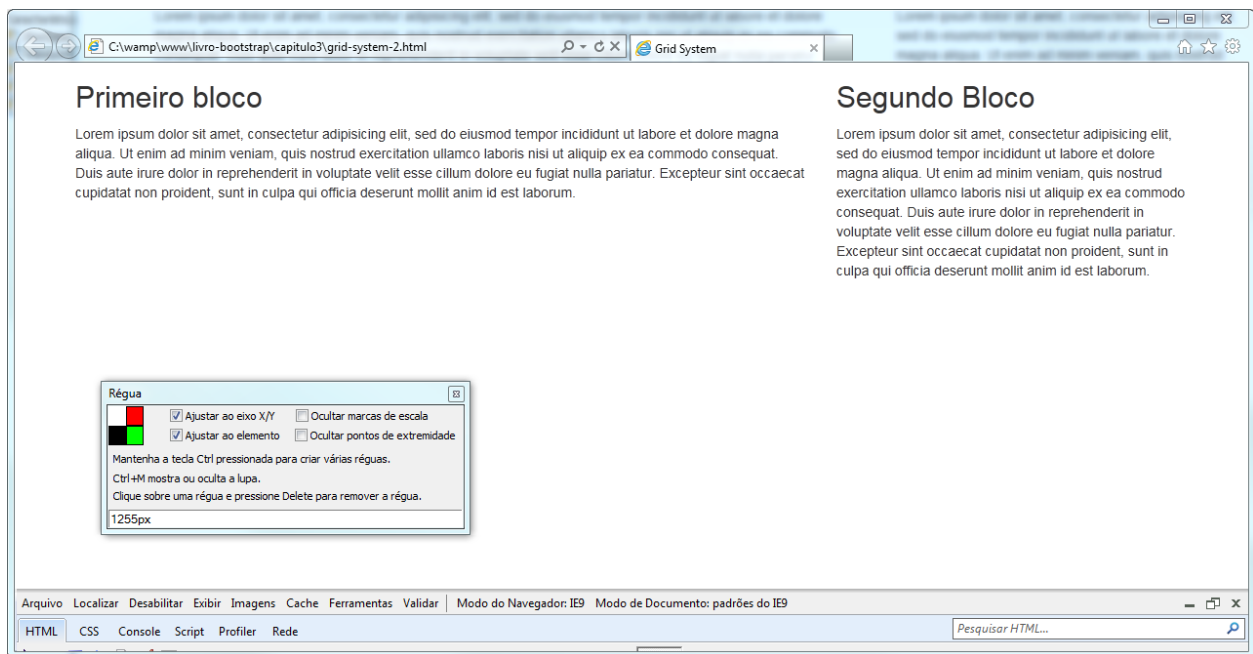
Para configurar esses blocos de texto, precisamos usar `col-lg` e `col-md` na mesma div, de forma que o framework possa assumir a quantidade de blocos de acordo com o tamanho da tela. Resumindo, a configuração das divs deve ser a seguinte:

```
<div class="col-lg-8 col-sm-6">
  ...
</div>
<div class="col-lg-4 col-sm-6">
  ...
</div>
```

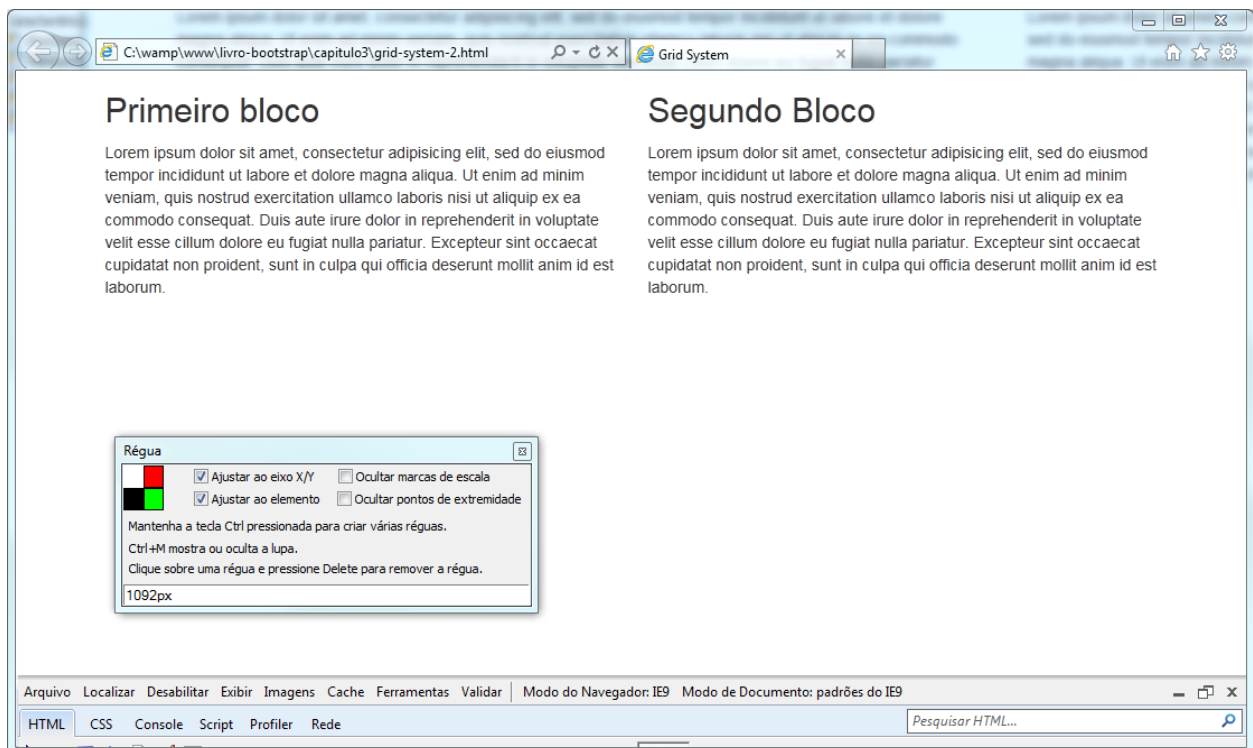
Neste código, dizemos para que a primeira div assuma 8 blocos (dos 12 disponíveis) em uma tela maior (desktop), mas se a tela for média (tablet), a mesma div irá possuir 6 blocos. As imagens a seguir mostram as diferentes situações de acordo com o código html criado anteriormente. O código completo está [neste link](#)³

²<https://github.com/danielps/livro-bootstrap/blob/master/exemplos/grid-system-1.html>

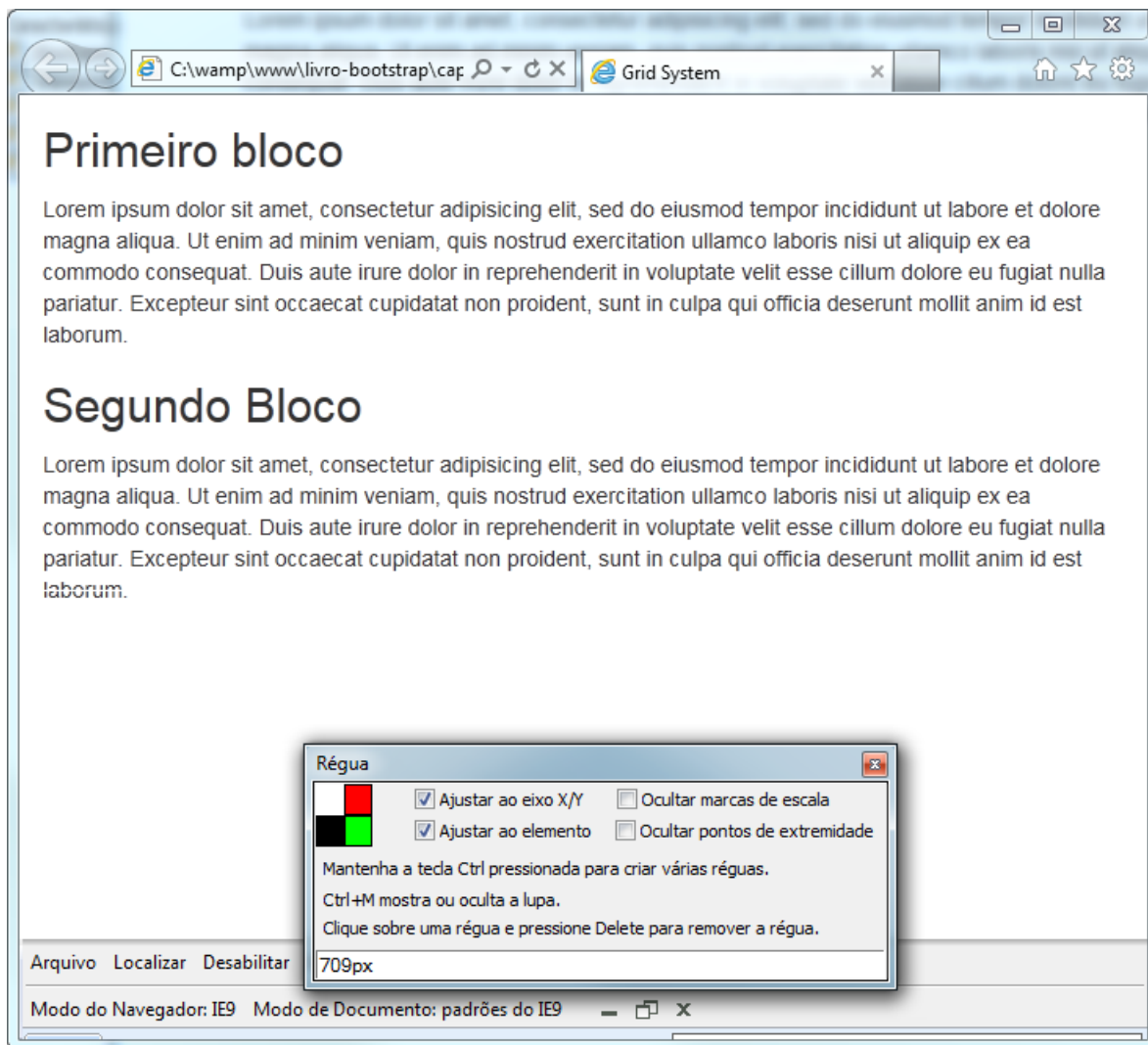
³<https://github.com/danielps/livro-bootstrap/blob/master/exemplos/grid-system-2.html>



Divs com tela grande



Divs com tela média



Divs com tela pequena

Trabalhando com offsets

Um offset é, resumidamente, um espaço em branco adicionado à esquerda do bloco de conteúdo. Por exemplo, se você deseja adicionar conteúdo somente no segundo bloco dos 12 disponíveis, você pode usar a seguinte classe: `.com-md-offset-1`. Nesta classe, usamos o `md` para definir o padrão para tamanhos de tela médio, e o `offset-1` para deslocar o bloco de conteúdo em 1 bloco.

No exemplo a seguir, criamos um bloco de conteúdo que omite os dois primeiros blocos, e os dois últimos, deixando a interface do texto mais centralizada.

Exemplo com Offset

```
<div class="col-md-8 col-md-offset-2">
```

Você poderá verificar o exemplo completo [neste link](#)⁴.

Colunas aninhadas

Em determinadas situações é necessário inserir blocos de conteúdo dentro de outros blocos. Desta forma, quando usamos as colunas aninhadas uma à outra, o bloco interno assume o sistema de 12 blocos do bloco anterior.

Por exemplo, se temos um bloco com 8 colunas, e dentro deste bloco queremos criar dois blocos centralizados, criamos então o seguinte código:

Colunas aninhadas

```
<div class="row">
  <div class="col-md-8">
    Nível 1: .col-md-8
    <div class="row">
      <div class="col-md-6">
        Nível 2: .col-md-6
      </div>
      <div class="col-md-6">
        Nível 2: .col-md-6
      </div>
    </div>
  </div>
</div>
```

Veja que os dois blocos internos possuem 6 colunas cada, e não 8, que é o tamanho do bloco pai. No [exemplo a seguir](#)⁵, criamos uma estrutura bem conhecida para sites com muito conteúdo. Visualize este código html no navegador para perceber como as colunas se comportam com o redimensionamento do navegador.

⁴<https://github.com/danielps/livro-bootstrap/blob/master/exemplos/grid-system-3.html>

⁵<https://github.com/danielps/livro-bootstrap/blob/master/exemplos/grid-system-4.html>

Estrutura complexa envolvendo blocos aninhados

```
<!DOCTYPE html>
<html>
<head>
  <title>Grid System</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="css/bootstrap.min.css" rel="stylesheet" media="screen">
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col-md-8">
        <h2>bloco 1</h2>
        <p>
          Lorem ipsum
        </p>
        <div class="row">
          <div class="col-md-6">
            <h2>bloco 1.1</h2>
            <p>
              Lorem ipsum
            </p>
          </div>
          <div class="col-md-6">
            <h2>bloco 1.2</h2>
            <p>
              Lorem ipsum
            </p>
          </div>
        </div>
      </div>
      <div class="row">
        <div class="col-md-6 col-md-offset-3">
          <h2>bloco 1.3</h2>
          <p>
            Lorem ipsum
          </p>
        </div>
      </div>
    </div>
    <div class="col-md-4">
      <h2>bloco 2</h2>
    </div>
  </div>
</body>
</html>
```

```
<p>
  Lorem ipsum
</p>
<h2>bloco 3</h2>
<p>
  Lorem ipsum
</p>
</div>
</div>
</div>
```

```
</div> </div>
```

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

Outro fator importante a ser considerado no Grid System é que o próprio framework determina as dimensões de cada bloco, cada elemento possui margens determinadas para “caber” no dispositivo o máximo possível. Alterar estas margens através de CSS não é recomendado.

Tipografia

Neste capítulo veremos que o Bootstrap altera toda a tipografia das tags HTML para que possam ser utilizadas na criação de suas páginas. Inicialmente, a tag <body> recebe um tamanho de 14 pixels, além de outros atributos. O parágrafo <p> recebe também uma margem inferior, além de uma classe especial chamada lead, que destaca melhor o parágrafo em relação aos outros.

```
<p class="lead">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
</p>
```

Alinhamento

Pode-se também alinhar o texto de um parágrafo utilizando as classes text-left, text-right e text-center, veja:

```
<p class="text-left">Texto alinhado à esquerda</p>
<p class="text-center">Texto alinhado no centro</p>
<p class="text-right">Texto alinhado à direita</p>
```

Texto alinhado à esquerda

Texto alinhado no centro

Texto alinhado à direita

Resultado do código HTML acima

Criando ênfase ao texto com cores

É possível dar uma certa destaque a algum texto utilizando cores, ao invés do negrito () e itálico (<i>) tradicional. Claro que é possível alterar a cor de um texto com , mas isso nunca deverá ser feito.

Para que tenhamos um padrão lógico de cores, que são aplicados não somente a textos, mas a botões, links e caixas de mensagens, usamos uma referência a 6 determinadas situações. São elas: mute, primary, success, info, warning, danger.

Cada uma delas possui uma cor específica que é padrão dentro do bootstrap. Para testar o destaque em texto, vamos ao seguinte código:

```
<p class="text-muted">Texto "mudo" ou sem foco</p>
<p class="text-primary">Texto com um pouco de destaque</p>
<p class="text-success">Texto com alguma mensagem boa</p>
<p class="text-info">Texto de informação</p>
<p class="text-warning">Texto de aviso</p>
<p class="text-danger">Texto de perigo, erro</p>
```

Texto "mudo" ou sem foco

Texto com um pouco de destaque

Texto com alguma mensagem boa

Texto de informação

Texto de aviso

Texto de perigo, erro

Resultado do código HTML acima

Abreviação

A abreviação é um novo elemento no html 5 que utiliza a tag `<abbr>` em conjunto com a propriedade `title`. Veja o exemplo a seguir:

```
<p><abbr title="HyperText Markup Language">HTML</abbr> é a melhor linguagem de ma\
rcação de texto existente.</p>
```

HTML é a melhor linguagem de marcação de texto existente.

HyperText Markup Language

Resultado do código HTML acima

Blockquotes (citações)

Esta tag é usada para citar um texto com um pouco mais de ênfase. A tag utilizada é `<blockquote>`, e o bootstrap adicionou algumas classes extras para dar mais sentido ao blockquote. Você pode usar `<small>` dentro do bloco, incluindo a tag `<cite>` que define o nome da pessoa citada no bloco. Também pode-se utilizar a classe `.pull-right` para flutuar o texto para a direita.

```
<blockquote class="pull-right">
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a ante.</p>
  <small>Someone famous in <cite title="Source Title">Source Title</cite></small>
</blockquote>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

 esquerda

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

direita  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a ante.

Someone famous in Source Title —

Resultado do código HTML acima

Listas

As listas do html são criadas através das tags ``, `` e `<dl>`. O bootstrap adiciona algumas funcionalidades extras para que possamos trabalhar melhor com estas listas. Vamos ver cada opção a seguir.

Unstyled

Utilizando a classe `list-unstyled`, remove a marca da lista que pode ser o ponto para a lista formada com `` ou a numeração de uma lista ``.

```
<ul class="list-unstyled">
  <li>...</li>
</ul>
```

Lado a lado (inline)

Caso deseje agrupar os itens de uma lista lado a lado, ao invés de um abaixo do outro, utilize a classe `list-inline`, da seguinte forma:

```
<ul class="list-inline">
  <li>Lorem</li>
  <li>Ipsum</li>
  <li>dolor</li>
</ul>
```

Listas com descrição na horizontal

A tag <dl> cria listas através do uso do <dt> e <dd>. Cada lista possui um item que é o título (<dt>) e outra que é a descrição (<dd>).

No exemplo a seguir, usamos a lista no formato original, veja:

```
<dl>
  <dt>...</dt>
  <dd>...</dd>
</dl>
```

E com da lista no formato horizontal:

Sem dl-horizontal

Lorem
ipsum dolor sit amet
consectetur
adipiscing elit. t enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Integer
posuere erat a ante. Lorem ipsum dolor sit amet, consectetur adipiscing elit

Com dl-horizontal

Lorem	ipsum dolor sit amet
consectetur	adipiscing elit. t enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Integer	posuere erat a ante. Lorem ipsum dolor sit amet, consectetur adipiscing elit

Resultado do código HTML acima

Os exemplos deste capítulo estão localizados [neste link].(<https://github.com/danielps/livro-bootstrap/blob/master/exemplos/textos.html>).

Tabelas

Tabelas são a forma mais comum de organizar dados em listas, principalmente se houverem muitos campos a serem exibidos. As tabelas foram implementadas desde o início do HTML, através da tag `<table>` e necessitam de uma grande quantidade de otimização visual para que fiquem com uma boa apresentação.

HTML Table Example:

First Name	Last Name	Points
Jill	Smith	50
Eve	Jackson	94
John	Doe	80
Adam	Johnson	67

Uma tabela simples sem formatação

Com o Bootstrap, o uso de tabelas se torna algo simples e sem nenhuma customização complexa, bastando apenas definir a classe `table`, além de algumas propriedades extras

Em todos os nossos exemplos, usamos também as tags `<thead>` para definir o cabeçalho da tabela e `<tbody>` para definir o corpo da tabela.

Tabela simples

Para criar uma tabela no Bootstrap, use a tag `<table>` e a classe `table`, na seguinte forma:

```
<table class="table">  
  ...  
</table>
```

No exemplo anterior, ao aplicar a classe `table`, temos o seguinte resultado:

```
<table class="table">
  <thead>
    <tr>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Points</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Jill</td>
      <td>Smith</td>
      <td>50</td>
    </tr>
    <tr>
      <td>Eve</td>
      <td>Jackson</td>
      <td>94</td>
    </tr>
    <tr>
      <td>John</td>
      <td>Doe</td>
      <td>80</td>
    </tr>
    <tr>
      <td>Adam</td>
      <td>Johnson</td>
      <td>67</td>
    </tr>
  </tbody>
</table>
```

Tabela Bootstrap

class='table'

First Name	Last Name	Points
Jill	Smith	50
Eve	Jackson	94
John	Doe	80
Adam	Johnson	67
Jill	Smith	50
Eve	Jackson	94
John	Doe	80
Adam	Johnson	67

Tabela com a classe 'table'

Listras em zebra

Uma forma fácil de deixar a tabela no formato zebra, com uma linha de fundo branco e outra linha no fundo cinza, de forma alternada, é utilizar a classe `table-striped`, da seguinte forma:

```
<table class="table table-striped">  
  ...  
</table>
```

Tabela em zebra

class='table table-striped'

First Name	Last Name	Points
Jill	Smith	50
Eve	Jackson	94
John	Doe	80
Adam	Johnson	67
Jill	Smith	50
Eve	Jackson	94
John	Doe	80
Adam	Johnson	67

Tabela com a classe 'table-striped'

Bordas

Não inclua bordas usando `border='1'`, use a classe `table-bordered` para que o Bootstrap possa incluir corretamente as bordas na tabela.

```
<table class="table table-bordered">
  ...
</table>
```

Tabela com bordas

class='table table-bordered'

First Name	Last Name	Points
Jill	Smith	50
Eve	Jackson	94
John	Doe	80
Adam	Johnson	67
Jill	Smith	50
Eve	Jackson	94
John	Doe	80
Adam	Johnson	67

Tabela com a classe 'table-bordered'

Destacando linhas

O efeito hover é bem conhecido por nós desenvolvedores web. Ele muda uma cor ou formato quando o ponteiro do mouse está ativado naquele ponto. Nas tabelas, é possível adicionar esse efeito destacando a linha que o mouse passa. Para fazer isso, você pode usar a classe table-hover

```
<table class="table table-hover">
  ...
</table>
```

Tabelas menores (condensadas)

Caso o espaço entre cada célula (padding e margin) da tabela estejam atrapalhando a visualização de um item, você pode retirá-los através da classe table-condensed, conforme o exemplo a seguir:

```
<table class="table table-condensed">
  ...
</table>
```

Linhas contextuais

É possível adicionar um contexto a uma linha de uma tabela, através das classes `active`, `success`, `warning` ou `danger`. As classes devem ser aplicadas na tag `<tr>` que define a linha da tabela.

```
<table class="table">
  <tbody>
    <tr class="danger">...</tr>
  </tbody>
</table>
```

Tabela com linhas contextuais

`class='table'`

First Name	Last Name	Points
Jill	Smith	50
Eve	Jackson	94
John	Doe	80
Adam	Johnson	67
Jill	Smith	50
Eve	Jackson	94
John	Doe	80
Adam	Johnson	67

Tabelas com linhas contextuais

Mesclando classes

É possível mesclar todas as classes disponíveis para formatação da tabela, de forma a obter um melhor formatação da mesma. Por exemplo, você pode usar as classes `table-striped` e `table-bordered` juntas.

Tabela mesclada

`class="table table-bordered table-striped"`

First Name	Last Name	Points
Jill	Smith	50
Eve	Jackson	94
John	Doe	80
Adam	Johnson	67
Jill	Smith	50
Eve	Jackson	94
John	Doe	80
Adam	Johnson	67

Tabela mesclada

Os exemplos deste capítulo estão localizados [neste link](#)^a.

^a<https://github.com/danielps/livro-bootstrap/blob/master/exemplos/tabelas.html>

Formulários

Os formulários, cuja tag principal é `<form>`, constituem o principal meio de entrada de dados em páginas web. Como existem diversas formas de se criar um formulário, podendo inclusive mesclar formas variadas, vamos exibir a seguir as principais funcionalidades que o Bootstrap possui.

Formulário simples, sem nada

First name:

Last name:

Um simples formulário sem a formatação do Bootstrap

Compreendo o form-group

A classe `form-group` é usada para delimitar um grupo único de um formulário, que na maioria das vezes é definida como um campo. Para criar um formulário com muitos campos, é necessário criar um formulário com vários elementos com a classe `form-group`. No exemplo a seguir, criamos um formulário com dois elementos, nome e email, no qual cada um é definido por uma `div` com a classe `form-group`.

```
<form>
  <div class="form-group">
    <label for="nome">Nome</label>
    <input type="text" class="form-control" id="nome">
  </div>
  <div class="form-group">
    <label for="email">Email</label>
    <input type="text" class="form-control" id="email">
  </div>
</form>
```

Formulário simples com form-group

Nome

Email

Formulário formatado com Bootstrap

Neste exemplo, podemos perceber que a largura de cada campo `input` foi alterada para 100%, e que o formulário está devidamente formatado para atender aos padrões web. Esta formatação é obtida através da classe `form-control` que está presente na tag `<input>`. Obviamente, todo controle html de formulário deve possuir esta classe.

As diferenças entre um formulário simples e com o bootstrap são configuradas principalmente através da criação da `<div>` com a classe `form-group` e o campo `<input>` com a classe `form-control`.

Formulário in-line

Um formulário in-line é aquele em que os campos ficam lado a lado. Para estes campos, é necessário informar a largura de cada um deles. Além disso, mesmo que os cabeçalhos de cada campo não apareçam, eles devem ser informados, para fins de semântica, e podem ser escondidos através da classe `sr-only`.

No exemplo a seguir, adicionamos apenas a classe `form-inline` na tag `<form>`.

```
<form class="form-inline">
  <div class="form-group">
    <label for="nome">Nome</label>
    <input type="text" class="form-control" id="nome">
  </div>
  <div class="form-group">
    <label for="email">Email</label>
    <input type="text" class="form-control" id="email">
  </div>
</form>
```


Formulário inline

Nome Email

Formulário inline

Podemos refinar o formulário e incluir a classe `sr-only` com o objetivo de esconder os labels, deixando-os disponíveis apenas para a semântica da página. Também usamos `placeholder` para definir o label do campo nele mesmo.

```
<form class="form-inline">
  <div class="form-group">
    <label for="nome" class="sr-only">Nome</label>
    <input type="text" class="form-control" id="nome" placeholder="Nome">
  </div>
  <div class="form-group">
    <label for="email" class="sr-only">Email</label>
    <input type="text" class="form-control" id="email" placeholder="Email">
  </div>
</form>
```

Formulário inline com sr-only

Nome Email

Formulário inline sem labels

Formulários horizontais

Estes são os tipos mais usados para a entrada de dados. Um formulário horizontal é definido pela classe `form-horizontal` na tag `<form>` e deve ter o tamanho de cada label de cada campo definido. Os tamanhos são definidos da mesma forma vista no capítulo 3, pelo sistema de grids.

No exemplo a seguir, criamos um formulário disposto na forma horizontal, veja:

```
1 <form class="form-horizontal">
2   <div class="form-group">
3     <label for="nome" class="col-sm-2 control-label">Nome</label>
4     <div class="col-sm-5">
5       <input type="text" class="form-control" id="nome" placeholder="Digite o s\
6 eu nome"/>
7     </div>
8   </div>
9   <div class="form-group">
10    <label for="email" class="col-sm-2 control-label">Email</label>
11    <div class="col-sm-5">
12      <input type="text" class="form-control" id="email" placeholder="Digite o \
13 seu email"/>
14    </div>
15  </div>
16  <div class="form-group">
17    <div class="col-sm-offset-2 col-sm-4">
18      <button type="submit" class="btn btn-default">Enviar</button>
19    </div>
20  </form>
```

Este exemplo contém detalhes importantes sobre o Bootstrap. Inicialmente, na linha 1, criamos o `<form>` com a classe `form-horizontal` e logo em seguida, criamos o primeiro `form-group`.

Na linha 3, adicionamos a classe `col-sm-2`, relativa ao grid system fazendo que o `label` ocupe 2 espaços dos 12 disponíveis. Também usamos a classe `control-label` que irá formatar o `label` de acordo com o formulário horizontal.

Na linha 4 incluímos uma `div` que possui a classe `col-sm-5`, que será utilizada para incluirmos a caixa de texto. Na linha 6 inserimos a caixa de texto com a tag `input`. O processo se repete para o email, na linha 9.

Na linha 16 criamos uma `div` para inserir o botão de `submit` do formulário. Veja que, na linha 17, usamos a classe `col-sm-offset-2` que irá gerar um `offset` com 2 espaços na `div`, fazendo com que o botão fique alinhado a caixa de texto. Este formulário é representado na imagem a seguir:

Formulário horizontal

Nome	<input type="text" value="Digite o seu nome"/>
Email	<input type="text" value="Digite o seu email"/>
<input type="button" value="Enviar"/>	

Formulário horizontal

Componentes suportados

O Bootstrap dá suporte a todos os controles comuns do formulário, tais como: text, password, datetime, datetime-local, date, month, time, week, number, email, url, search, tel, e color. Alguns deles possuem opções extras, das quais veremos a seguir.

Checkbox e radio na horizontal

O Bootstrap oferece uma opção extra a estes controles de forma a disponibilizar cada item na forma horizontal, conforme o código a seguir:

```
<label class="checkbox-inline">
  <input type="checkbox" id="inlineCheckbox1" value="option1"> 1
</label>
<label class="checkbox-inline">
  <input type="checkbox" id="inlineCheckbox2" value="option2"> 2
</label>
<label class="checkbox-inline">
  <input type="checkbox" id="inlineCheckbox3" value="option3"> 3
</label>
```

Controles estáticos

É possível adicionar um controle estático, na maioria das vezes, um label, em formulários cujo o objeto é apenas informar algum campo que já foi preenchido. Esta configuração é realizada através da classe `form-control-static`.

```
<div class="form-group">
  <label class="col-sm-2 control-label">Email</label>
  <div class="col-sm-10">
    <p class="form-control-static">email@example.com</p>
  </div>
</div>
```

Campos estáticos

Email	email@example.com
Senha	<input type="password" value="Password"/>

Formulário com campos estáticos

Campos desabilitados

Para desabilitar um campo, basta adicionar o atributo `disabled` a ele, conforme o exemplo a seguir:

```
<input class="form-control" id="disabledInput" type="text" placeholder="Disabled \
input here..." disabled>
```

Você pode desabilitar um fieldset também: `<fieldset disabled>`

Validações

O bootstrap não realiza validações em si, claro, ele apenas oferece meios de se visualizar a validação dos campos. A validação deve ser feita através de javascript, e é possível adicionar classes a classe `form-group`, da seguinte forma:

```
<div class="form-group has-success">
  <label class="control-label" for="inputSuccess">Input com success</label>
  <input type="text" class="form-control" id="inputSuccess">
</div>
<div class="form-group has-warning">
  <label class="control-label" for="inputWarning">Input com warning</label>
  <input type="text" class="form-control" id="inputWarning">
</div>
<div class="form-group has-error">
  <label class="control-label" for="inputError">Input com error</label>
  <input type="text" class="form-control" id="inputError">
</div>
```

Validação de campos

Input with success

Input with warning

Input with error

Formulário com validação

Neste exemplo, usamos as classes `has-success`, `has-warning` e `has-error` para definir as validações

Tamanho dos campos

Pode-se criar diversos tamanhos pré definidos para os campos de um formulário. Os tamanhos mais comuns são definidos na própria tag `input`, através das classes `input-lg`, para um campo maior e `input-sm` para um campo menor.

Já a largura do campo deve ser configurada através do grid system, usando, por exemplo, `col-xs`, `col-sm`, entre outros, conforme visto no Capítulo 3, por exemplo:

```
<div class="row">
  <div class="col-xs-2">
    <input type="text" class="form-control" placeholder=".col-xs-2">
  </div>
  <div class="col-xs-3">
    <input type="text" class="form-control" placeholder=".col-xs-3">
  </div>
  <div class="col-xs-4">
    <input type="text" class="form-control" placeholder=".col-xs-4">
  </div>
</div>
```

Texto para ajuda

É muito comum em formulários inserir um texto de help abaixo do campo, como forma de explicar algo relativo ao campo em questão. Esse texto deve ser inserido em uma tag `` logo a seguir do `<input>`, conforme o exemplo a seguir:

```
<div class="form-group">
  <label for="email" class="col-sm-2 control-label">Email</label>
  <div class="col-sm-5">
    <input type="text" class="form-control" id="email" placeholder="Digite o \
seu email"/>
    <span class="help-block">Informe um email válido, pois será verificado ap\
ós a conclusão do cadastro</span>
  </div>
</div>
```

Botões

Botões fazem parte de qualquer formulário. Com o Bootstrap você pode criar botões para finalidades diferentes, sendo elas: `default`, `primary`, `success`, `info`, `warning`, `danger`. Cada um desses botões são definidos pela sua classe, da seguinte forma:

```
<!-- Botão padrão -->
<button type="button" class="btn btn-default">Default</button>
<!-- Possui uma cor diferenciada e é definido como um botão que identifica a ação\
principal de um conjunto de botões -->
<button type="button" class="btn btn-primary">Primary</button>
<!-- Botão que indica uma ação de sucesso -->
<button type="button" class="btn btn-success">Success</button>
<!-- Botão que indica uma mensagem de informação -->
<button type="button" class="btn btn-info">Info</button>
<!-- Botão que indica atenção, perigo -->
<button type="button" class="btn btn-warning">Warning</button>
<!-- Botão que indica algo potencialmente negativo, ou um erro -->
<button type="button" class="btn btn-danger">Danger</button>
<!-- Este botão assume o comportamento de um link -->
<button type="button" class="btn btn-link">Link</button>
```

Como podemos perceber no código acima, os botões necessitam basicamente de duas classes. A primeira é a padrão `btn` que informa ao Bootstrap que aquele elemento é um botão. A segunda é justamente a forma como o botão será, podendo indicar sucesso ou erro, por exemplo.

Tamanho dos botões

Você pode usar as seguintes classes para definir o tamanho de um botão:

`btn-lg` : Large, botão grande

`btn-sm` : Small, botão pequeno

`btn-xs` : Extra Small, botão muito pequeno

Botões em grupo

Caso deseje adicionar um conjunto de botões unidos um ao outro, pode-se utilizar a classe `btn-group` para isso, conforme o código a seguir:

```
<div class="btn-group">
  <button type="button" class="btn btn-default">Left</button>
  <button type="button" class="btn btn-default">Middle</button>
  <button type="button" class="btn btn-default">Right</button>
</div>
```

Ícones

Existe uma funcionalidade no Bootstrap que usa uma biblioteca de ícones prontos, que são os Glyphicons. Para usá-los, deve-se adicionar a classe do ícone ao elemento.

Por exemplo, se você deseja adicionar um ícone com um sinal de busca, você pode adicioná-lo com o seguinte código:

```
<span class="glyphicon glyphicon-search"></span>
```

Os ícones podem ser usados em conjunto com o botão, bastando incluí-lo da seguinte forma:

```
<button type="button" class="btn btn-default btn-lg">  
  <span class="glyphicon glyphicon-star"></span> Star  
</button>
```

Em nossa página de exemplos (formularios.html) existe uma lista completa de ícones.

Botões dropdown (menu)

Para criar um botão com opções, deve-se um conjunto de tags html e classes conforme o código a seguir:

```
<div class="btn-group">  
  <button type="button" class="btn btn-default dropdown-toggle" data-toggle="drop\  
down">  
    Arquivo <span class="caret"></span>  
  </button>  
  <ul class="dropdown-menu" role="menu">  
    <li><a href="#">Abrir</a></li>  
    <li><a href="#">Salvar</a></li>  
    <li><a href="#">Salvar como</a></li>  
    <li class="divider"></li>  
    <li><a href="#">Fechar</a></li>  
  </ul>  
</div>
```


Inicialmente criamos uma div que possui a classe `btn-group`, que irá agrupar tanto o botão com o texto quanto o dropdown que abre o menu com os itens. Repare que após a palavra Arquivo incluímos um ícone com a classe `caret` que é uma seta para baixo.

O menu dropdown é formado pelo conjunto de tags `ul li`, seguido da classe `dropdown-menu`. O que faz o menu abrir com o clique do botão é a propriedade `data-toggle="dropdown"` do botão, que abre o `dropdown-menu` interno a div contida no `btn-group`.

Validação de campos

Input with success

Input with warning

Input with error

Menu Dropdown

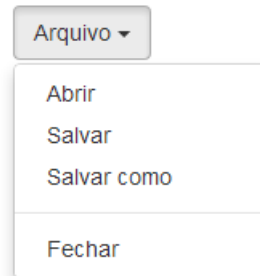
Inputs diferenciados com group

Uma das opções que o Bootstrap oferece na criação dos campos é formar um grupo entre um input e uma classe ou botão. Os exemplos a seguir ilustram este processo, de forma que usamos a classe `input-group` para unir um texto qualquer a uma caixa de texto.

```
<div class="input-group">
  <span class="input-group-addon">@</span>
  <input type="text" class="form-control" placeholder="Username">
</div>
<div class="input-group">
  <input type="text" class="form-control">
  <span class="input-group-addon">.00</span>
</div>
<div class="input-group">
  <span class="input-group-addon">$</span>
  <input type="text" class="form-control">
  <span class="input-group-addon">.00</span>
</div>
```

O resultado deste código é visto na imagem a seguir.

Botão dropdown



Input Group

Caso deseje adicionar um botão à um input group, faça da seguinte forma:

```
<div class="input-group">
  <input type="text" class="form-control">
  <span class="input-group-btn">
    <button class="btn btn-default" type="button">Ir! </button>
  </span>
</div>
```

Os exemplos deste capítulo estão localizados [neste link](https://github.com/danielps/livro-bootstrap/blob/master/exemplos/formularios.html)^a.

^a<https://github.com/danielps/livro-bootstrap/blob/master/exemplos/formularios.html>