



Base do desenvolvimento

O Bootstrap é construído em um grid responsivo de 12 colunas, com layouts e componentes

Estilos globais

Sistema de Grid

Sistema de grid fluido

Layouts

Design Responsivo

configurações globais

Requer o doctype de HTML5

O Bootstrap faz o uso de certos elementos HTML e propriedades CSS que requer o uso do doctype HTML5. Inclua no início dos seus projetos

```
1. <!DOCTYPE html>
2. <html lang="en">
3.   ...
4. </html>
```

Tipografia e links

O Bootstrap usa estilos básicos de tela, tipografia, e estilos de link. Especificadamente, nós temos:

- Remove `margin` do corpo
- Escolhemos o `background-color: white;` no `body`
- Use os atributos `@baseFontFamily`, `@baseFontSize`, and `@baseLineHeight` para nossa base tipográfica
- Escolhemos cores globais de link via `@linkColor` e aplicamos linha abaixo do link com `:hover`

Estes estilos podem ser encontrados em **scaffolding.less**.

40px
▼

Reset via Normalize

Com o Bootstrap 2, os velhos blocos de reset foram retirados em favor do Normalize.css (<http://necolas.github.com/normalize.css/>), um projeto do Nicolas Gallagher (<http://twitter.com/necolas>) que também tem o poder do HTML5 Boilerplate (<http://html5boilerplate.com>). Enquanto nós usamos muito do Normalize em nosso **reset.less**, nós removemos alguns elementos especificadamente para o Bootstrap

Sistema de grid padrão

Exemplo de grid real

O sistema padrão do grid do Bootstrap utiliza **12 columns**, com um container de 940px sem funcionalidades de design responsivo (`./scaffolding.html#responsive`) habilitadas. Com o arquivo responsive adicionado, o grid se adapta de 724px e 1170px dependendo do nosso viewport. Abaixo de 767px, as colunas se tornam fluidas e ficam estagnadas verticalmente

1

1

1

1

1

1

1

1

1

2

3

4

4

5

9

HTML do grid básico

Para um simples layout de duas colunas, crie um `.row` e adicione um número apropriado de `.span*`. Como o grid é de 12 colunas, cada `.span*` tem um número que somam 12 colunas, e deve sempre ter 12 colunas para cada linha (ou o número de colunas do pai).

```
1. <div class="row">
2.   <div class="span4">...</div>
3.   <div class="span8">...</div>
4. </div>
```

No exemplo dado, nós temos `.span4` e `.span8`, fazendo 12 o total de colunas de uma linha completa

Dando um offset nas colunas

Mova colunas para a direita usando as classes `.offset*`. Cada classe aumenta a margem esquerda de uma coluna por toda colunagem. Por exemplo, `.offset4` move `.span4` por 4 colunas.

4

3 offset 2

3 offset 1

3 offset 2

6 offset 3

```
1. <div class="row">
2.   <div class="span4">...</div>
3.   <div class="span3 offset2">...</div>
4. </div>
```

Aninhando colunas

Para aninhar seu conteúdo com o grid padrão, adicione um novo `.row` e escolha um `.span*` em * colunas com uma coluna existente de `.span*`. Linhas aninhadas devem incluir um set de colunas que adicionam ao número de colunas do seu pai

Coluna nível 1

Nível 2

Nível 2

```
1. <div class="row">
2.   <div class="span9">
3.     Coluna nível 1
4.     <div class="row">
5.       <div class="span6">Nível 2</div>
6.       <div class="span3">Nível 2</div>
7.     </div>
8.   </div>
9. </div>
```

Sistema de grid fluido

Exemplo de grid fluido

O sistema de grid fluido usa porcentagem ao invés de pixels para colunagem nas larguras. Ele tem as mesmas funcionalidades responsivas como nosso grid fixo, assegurando as proporções adequadas para tela chave nas resoluções e dispositivos

1

1

1

1

1

1

1

1

1

1

1

1

4

4

4

4

8

6

6

12

HTML básico do grid fluido

Para tornar qualquer linha "fluid" mudando `.row` para `.row-fluid`.

```
1. <div class="row-fluid">
2.   <div class="span4">...</div>
3.   <div class="span8">...</div>
4. </div>
```

offset do fluido

Opera da mesma forma como o grid fixo com um offset: adicione `.offset*` para qualquer coluna para dar um offset para qualquer coluna

4

4 offset 4

3 offset 3

3 offset 3

6 offset 6

```
1. <div class="row-fluid">
2.   <div class="span4">...</div>
3.   <div class="span4 offset2">...</div>
4. </div>
```

Teste de fluido

Grids fluidos utilizam aninhamento diferente: cada nível aninhado de coluna deve adicionar mais de 12 colunas. Isto é por que o grid fluido usa porcentagens, não pixels, para atribuir as larguras

Fluid 12

Fluid 6

Fluid 6

Fluid 6

Fluid 6

```
1. <div class="row-fluid">
2.   <div class="span12">
3.     Fluid 12
4.   <div class="row-fluid">
5.     <div class="span6">
6.       Fluid 6
7.     <div class="row-fluid">
8.       <div class="span6">Fluid 6</div>
9.       <div class="span6">Fluid 6</div>
10.    </div>
11.   </div>
12.   <div class="span6">Fluid 6</div>
13. </div>
14. </div>
15. </div>
```

Layouts

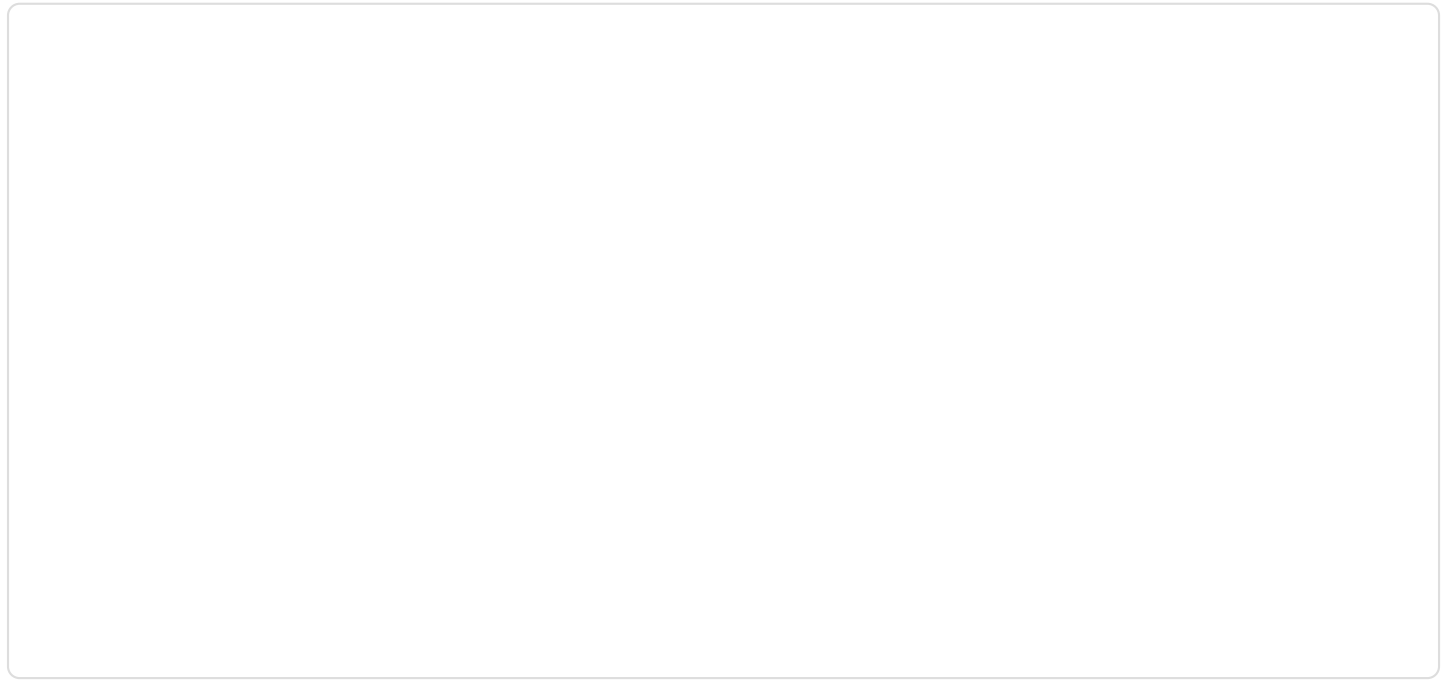
Layout fixo

Provê uma largura fixa (e opcionalmente responsiva) com qualquer `<div class="container">` requerido.

```
1. <body>
2.   <div class="container">
3.     ...
4.   </div>
5. </body>
```

Layout fluido

Criar uma página de duas colunas fluida com `<div class="container-fluid">` —ótimo para aplicação e documentação



```
1. <div class="container-fluid">
2.   <div class="row-fluid">
3.     <div class="span2">
4.       <!--conteúdo da lateral-->
5.     </div>
6.     <div class="span10">
7.       <!--conteúdo do corpo-->
8.     </div>
9.   </div>
10. </div>
```

Design Responsivo

Habilitando funcionalidades responsivas

Ligue o CSS responsivo no nosso projeto incluindo a meta tag e folha de estilos adicionais dentro do `<head>` do documento. Se você está com o Bootstrap da página customizada, você irá precisar incluir esta meta tag.

```
1. <meta name="viewport" content="width=device-width, initial-scale=1.0">
2. <link href="assets/css/bootstrap-responsive.css" rel="stylesheet">
```


Atenção! Bootstrap não inclui funcionalidade responsivas por padrão neste tempo por que nem tudo precisa ser responsiva. Ao invés disto nós encorajamos os desenvolvedores a remover esta funcionalidade, nós fizemos de uma forma que é possível habilitar quando necessário.

Sobre o Bootstrap responsivo

Os Media queries permitem ter CSS customizado baseado em um número de condições—aspecto, larguras, tipo de exibição, etc—mas geralmente focado em `min-width` and `max-width`.



- Modificar a largura da coluna no seu grid
- Elementos estagnados ao invés de float quando for necessário
- Redimensionando cabeçalhos e textos para ser mais apropriados para dispositivos

Usar media queries responsivamente e somente como princípio para a audiência de dispositivos móveis. Para projetos maiores, considere dedicar bases de código e não em camadas de media queries

Dispositivos suportados

O Bootstrap suporta uma utilidade de media queries em um único arquivo para ajudar a fazer seus projetos mais apropriados para diferentes dispositivos e resoluções de tela. Aqui está o que é incluso:

Nome	Largura do layout	largura da coluna	Largura do espaçamento
Resolução de tela grande	1200px and up	70px	30px
Padrão	980px and up	60px	20px
Tablets no formato na orientação horizontal	768px and above	42px	20px
Phones até tablets	767px and below	Colunas fluidas, larguras não fixas	
Phones	480px and below	Colunas fluidas, larguras não fixas	

```
1.  /* Desktop maiores */
2.  @media (min-width: 1200px) { ... }
3.
4.  /* tables em formato porta retrato até os deskts no formato paisagem */
5.  @media (min-width: 768px) and (max-width: 979px) { ... }
6.
7.  /* dispositivo em paisagem até os tablets em formato porta-retrato */
8.  @media (max-width: 767px) { ... }
9.
10. /* dispositivo no formato paisagem e abaixos */
11. @media (max-width: 480px) { ... }
```

Classes utilitárias responsivas

Para um desenvolvimento rápido e amigável em mobile, use estas classes utilitárias para mostrar e esconder conteúdo dependendo do dispositivo. Abaixo é uma tabela de classes disponíveis e seus efeitos num layout específico(rotulado por dispositivo). Eles podem ser encontrados em `responsive.less`.

Classe	Phones 767px e abaixo	Tablets 979px até 768px	Desktops Default
<code>.visible-phone</code>	Visível	Escondido	Escondido
<code>.visible-tablet</code>	Escondido	Visível	Escondido
<code>.visible-desktop</code>	Escondido	Escondido	Visível
<code>.hidden-phone</code>	Escondido	Visível	Visível
<code>.hidden-tablet</code>	Visível	Escondido	Visível
<code>.hidden-desktop</code>	Visível	Visível	Escondido

Quando usar

Use de forma básica e evite de criar versões completamente diferentes do mesmo site. Ao invés disto, use-os para complementar cada apresentação do dispositivo. Utilitários responsivos não devem ser usados com tabelas, e isto não é suportado.

Caso de teste responsivo

Redimensione seu browser ou carregar dispositivos diferentes para testar as classes acima

Visível em...

Marcações verdes indica que a classe é visível na nossa atual resolução

✓Phone

Tablet

Desktop

Escondido em...

Aqui, marcas verdes indica que a classe é escondido na nossa resolução atual

Phone

✓~~Tablet~~

✓~~Desktop~~

Desenhado e construído com todo amor do mundo por @mdo (<http://twitter.com/mdo>) e @fat (<http://twitter.com/fat>). [Voltar ao topo](#)

Este projeto é uma versão do Twitter Bootstrap da globo.com mantido por Alexandre Magno (<http://blog.alexandremagno.net>).

Código licenciado sob Apache License v2.0 (<http://www.apache.org/licenses/LICENSE-2.0>), documentação sob CC 3.0 (<http://creativecommons.org/licenses/by/3.0/>).

Glyphicons grátis (<http://glyphicons.com>) licenciado sob CC 3.0 (<http://creativecommons.org/licenses/by/3.0/>).

[Blog \(http://blog.getbootstrap.com\)](http://blog.getbootstrap.com) · [Issues \(https://github.com/twitter/bootstrap/issues?state=open\)](https://github.com/twitter/bootstrap/issues?state=open) · [Roadmap e changelog \(https://github.com/twitter/bootstrap/wiki\)](https://github.com/twitter/bootstrap/wiki)