

Bootstrap: Template de Layout Simples



Uma das primeiras coisas que você deve saber fazer com o Bootstrap é criar layouts com ele.

Para este tutorial, você vai precisar fazer o download do Bootstrap, e criar a estrutura inicial de arquivos.

Caso você não tenha feito, é só seguir os passos do [tutorial Bootstrap: Como Começar](#).

Vou assumir que você já conhece os conceitos iniciais, e assim poderemos começar a escrever a marcação de layouts com o Bootstrap.

Passo 1: Crie a Marcação Básica

Na pasta do projeto, crie um arquivo index.htm, e coloque a marcação a seguir:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="width=device-
width, initial-scale=1">
  <title>Bootstrap 3: Layouts</title>

  <link href="css/bootstrap.min.css"
rel="stylesheet">
  <link href="css/style.css" rel="stylesheet">

  <!--[if lt IE 9]>
  <script src="https://oss.maxcdn.com
/libs/html5shiv/3.7.0/html5shiv.js"></script>
  <script src="https://oss.maxcdn.com
/libs/respond.js/1.4.2/respond.min.js"></script>
  <![endif]-->
</head>
<body>
  <div class="container">
    </div>

    <script src="https://ajax.googleapis.com
/ajax/libs/jquery/1.11.0/jquery.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

Esse HTML abaixo contém uma estrutura inicial, os metadados, e um espaço para conteúdo. Vamos trabalhar com essa marcação a partir de agora.

Passo 2: Crie o CSS

O CSS abaixo é apenas para fins didáticos. Para que você consiga visualizar no browser, como que a estrutura se comporta no navegador.

Crie o arquivo `style.css` e coloque os seguintes estilos:

```
/*
 * style.css
 */
.container, .container-fluid {
  padding-right: 15px;
  padding-left: 15px;
  background-color: #f9f9f9;
}
.row {
  margin-bottom: 20px;
}
[class*="col-"], header, footer, nav, aside,
section, div[role="main"] {
  padding-top: 15px;
  padding-bottom: 15px;
  background-color: #f5f5f5;
  background-color: rgba(86,61,124,.15);
  border: 1px solid #ddd;
  border: 1px solid rgba(86,61,124,.2);
  text-align: center;
}
div[role="main"] {
  background-color: #fff;
}
div[role="main"], nav, aside {
  height: 600px;
}
```

Agora com essa estrutura inicial, já podemos trabalhar apenas dentro do container para criar os nossos layouts.

Passo 3: Crie o Layout de Largura Fixa

O primeiro tipo de layout mais comum é o layout fixo (em inglês, Fixed Width Layout). Esse é, também, um dos mais simples.

Neste layout, você pode usar três rows: uma para o header (se houver), uma para o conteúdo principal, e uma para o footer (se houver). Cada row vai ter uma função específica na página. E, num primeiro momento, você terá apenas uma coluna de conteúdo.

A marcação para este layout é a seguinte:

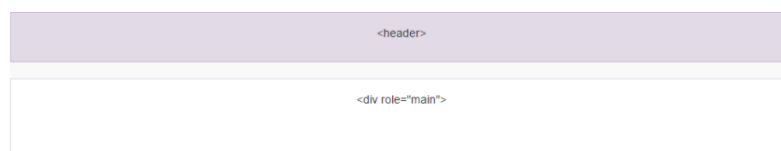
```
<div class="container">
  <header class="row">

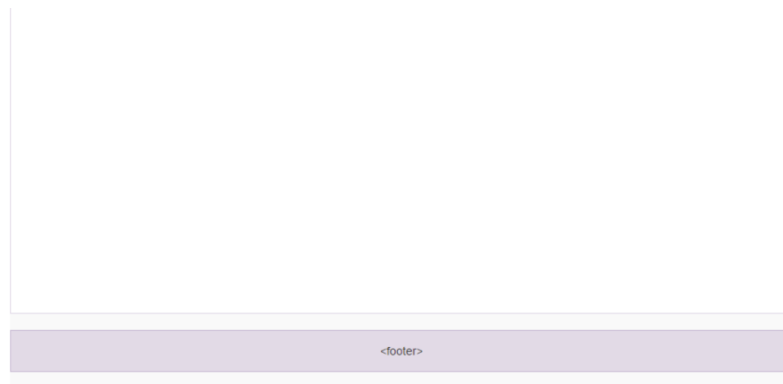
    </header>
  <div class="row">
    <div role="main">
      <!-- conteudo principal -->
    </div>
  </div>
  <footer class="row">

    </footer>
</div>
```

A maioria dos sites e aplicações web segue esse padrão, com três linhas principais de conteúdo.

Vamos ver, agora, um preview de como ficaria este layout:





Primeiramente, toda essa marcação deve estar dentro do container (a área em cinza claro) que nós criamos na estrutura inicial.

Depois, é colocado um header com a classe row. Assim, o Bootstrap irá tratar este item como uma linha que conterá outros itens. E, você pode adicionar nele os itens do cabeçalho da página, como um logo ou o título do site.

Logo após o header, você vê uma div com a classe row. Essa div é, apenas, um outro espaço para os itens da página que deverão ficar entre o cabeçalho e o rodapé. Dentro dela, você encontra a div main (a área com fundo branco), e é essa área que terá o conteúdo principal da página.

E, por fim, temos o footer, também com a classe row, que serve como área para os itens do rodapé.

Observação: Você pode trocar a tag header e a tag footer por div's com a classe row. O resultado seria o mesmo. Mas como queremos uma página semântica (HTML5) devemos seguir as boas práticas e utilizar essas tags.

A partir dessa marcação, o Bootstrap pode cuidar dos tamanhos e posicionamentos dos itens da página, mesmo se a tela, ou a orientação, mudar de tamanho. (experimente redimensionar seu navegador para ver a mudança)

Vamos ver uma variação deste layout...

Passo 4: Deixe o Layout com Full Width (Opcional)

Outro layout muito comum é o layout fluído, ou layout 100% (em inglês, Fluid Layout ou Full Width Layout).

Com o Bootstrap, é bastante simples de fazê-lo:

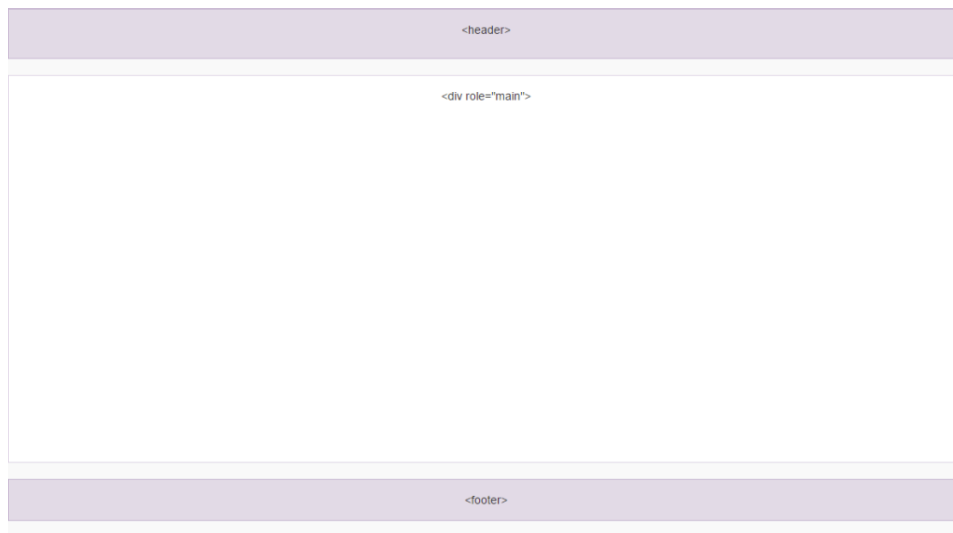
```
<div class="container-fluid">
  <header class="row">

    </header>
    <div class="row">
      <div role="main">
        <!-- conteudo principal -->
      </div>
    </div>
    <footer class="row">

  </footer>
</div>
```

A marcação é, praticamente, a mesma do layout fixo. A única diferença é a utilização da classe `.container-fluid`, no container principal.

Vamos ver o preview de como ficaria este layout:



Com essa marcação, o Bootstrap cria um layout que ocupa a largura inteira da tela. Além de cuidar, também, dos tamanhos e posicionamentos dos itens da página. Esse layout é muito bom para aplicações web e documentos online.

Observe que, tanto neste layout como no fixo, cada uma das áreas ocupam as doze colunas do grid do Bootstrap, apesar de isso não ser visualmente fácil de assimilar. Isso acontece porque não definimos nenhuma coluna na marcação, mas é o jeito correto de se fazer.



Publicado por Ruan Carvalho

Ruan é especialista em Front-end e trabalha com Desenvolvimento Web desde 2007.

Tutoriais, User Experience

Bootstrap



[Ver todos os posts por Ruan Carvalho](#)

1 COMENTÁRIO

Web Dev Academy

 Iniciar sessão ▾

 Recomendar

 Partilhar

Mostrar primeiro os mais votados ▾

Escreva o seu comentário...

INICIE SESSÃO COM O



OU REGISTE-SE NO DISQUS 

Nome



alnmaker • há 4 meses

Gostei muito do seu tutorial, eu que estou iniciando para um projeto de tcc de outra area estou me adequando muito bem.

^ | ▾ • Responder • Partilhar ›

TAGS

Bootstrap CRUD Drupal HTML & CSS JavaScript jQuery PHP SQL

Tooling WordPress **MAIS ACESSADOS**

- » [CRUD com Bootstrap, PHP & MySQL - Parte I](#)
- » [Bootstrap 3: Como Começar](#)
- » [Bootstrap: Criando Layouts e Grids](#)
- » [CRUD com Bootstrap, PHP & MySQL – Parte II](#)
- » [Aprenda a Fazer um CRUD com Bootstrap, PHP & MySQL](#)

**ORGULHOSAMENTE FEITO COM WORDPRESS | TEMA: BASKERVILLE 2 POR
ANDERS NOREN**

ACIMA ↑