

Wrapper

Wrapper significa envelopar ou envelope, esse é o nosso objetivo ao se fazer uma classe wrapper, que nesse exemplo é a container.

Existem pessoas que acreditam que wrapper e container são coisas distintas, outras, acreditam que é a mesma coisa.

Nesse momento, isso é indiferente, então vamos aprender wrapper como sendo a mesma coisa que container.

```
24 <div class="container">
25   <h3 class="titulo">HOUSE OF CARDS</h3>
26   <p class="descricao">Nada pode impedir o pol
27   <div class="botoes">
28     <button role="button" class="botao">
29       <i class="fas fa-play"></i>
30       ASSISTIR AGORA
31     </button>
32     <button role="button" class="botao">
33       <i class="fas fa-info-circle"></i>
34       MAIS INFORMAÇÕES
35     </button>
36   </div>
37 </div>
```

Essa classe container (linha 24) tem como objetivo envelopar todo esse conteúdo dentro dela, podemos ver que ela realmente faz isto ao minimizar.

```
23 <div class="filme-principal">
24   <div class="container"> ...
37 </div>
38 </div>
39 </main>
```

Note o salto de linha da 24 para 37.

Logo, uma classe wrapper por padrão é uma div que guarda vários outros elementos dentro.

Utilizamos ela por vários motivos, como:

- É mais fácil manipular os elementos filhos (diretos, de primeiro grau), pois ao invés de alterar todos os elementos filhos, altera-se uma vez o pai que já servirá para os outros, assim trabalha-se o conteúdo como um todo
- É possível movimentar ou alterar todos os elementos, alterando ou movendo apenas a caixa, no caso o container.
- Podemos ter vários containers, dependentes ou independentes.
- Tendo uma classe genérica (container), podemos usá-lo em qualquer parte do site.



```
97  .container{  
98      margin-left: 30px;  
99  }
```

Com apenas esse comando estamos alterando o título (principal e secundário), descrição e os dois botões.

The screenshot shows a web browser window with the URL `getbootstrap.com/docs/3.4/css/`. The browser's address bar and tabs are visible at the top. The main content area displays the 'Normalize.css' section, which explains its purpose for improved cross-browser rendering. Below this, the 'Containers' section is introduced, stating that Bootstrap requires a containing element for the grid system. Two types of containers are described: a responsive fixed width container (`.container`) and a full width container (`.container-fluid`), each with a corresponding HTML code snippet. The 'Grid system' section is partially visible at the bottom of the page.

Normalize.css

For improved cross-browser rendering, we use [Normalize.css](#), a project by [Nicolas Gallagher](#) and [Jonathan Neal](#).

Containers

Bootstrap requires a containing element to wrap site contents and house our grid system. You may choose one of two containers to use in your projects. Note that, due to `padding` and more, neither container is nestable.

Use `.container` for a responsive fixed width container.

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

Use `.container-fluid` for a full width container, spanning the entire width of your viewport.

```
<div class="container-fluid">
  ...
</div>
```

Grid system

O Bootstrap (framework de CSS3) utiliza e recomenda a utilização de container, para trabalhar com elementos em massa.

Container é semelhante a este pensamento simples, vamos supor que uma criança foi na sua casa e bagunçou a sala com os brinquedos, todos estão esparramados pelo chão. Você pode levar de um em um para o quarto ou colocar todos em uma caixa e levar uma única vez.

Perceba que o resultado será o mesmo mas o trabalho exigido é bem menor no segundo caso.