Front-end

Bootstrap

- Get started
- Grid System
- Containers
- Columns
- Display

by Mohammad Amin H.B. Tehrani - Reza Yazdani

www.maktabsharif.ir

Get started

Get started

Intro

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Bootstrap 4 is completely free to download and use!

Versions:

- **Bootstrap 4:** Bootstrap 4 is the newest version of Bootstrap; with new components, faster stylesheet and more responsiveness. Bootstrap 4 supports the latest, stable releases of all major browsers and platforms. However, Internet Explorer 9 and down is not supported.
- **Bootstrap 3:** If you require IE8-9 support, use Bootstrap 3. It is the most stable version of Bootstrap, and it is still supported by the team for critical bugfixes and documentation changes. However, new features will NOT be added to it.

Get started

Installation

1. **Download Bootstrap 4 CSS source code, and link it into your html page.**

   Download Bootstrap 4 from getbootstrap.com

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

2. **Use Bootstrap CDN:**

   MaxCDN provides CDN support for Bootstrap's CSS and JavaScript. You must also include jQuery

   ```
   <link rel="stylesheet"
   href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
   ```

JS Requirements

Many of our components require the use of JavaScript to function. Specifically, they require jQuery, Popper.js, and our own JavaScript plugins. Place the following `<script>`s near the end of your pages, right before the closing `</body>` tag, to enable them. jQuery must come first, then Popper.js, and then our JavaScript plugins.

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
```

Get started

Initialize your web page

1. **Add the HTML5 doctype:**
   Bootstrap 4 uses HTML elements and CSS properties that require the HTML5 doctype.
   Always **include the HTML5 doctype** at the beginning of the page, along with the **lang attribute** and the **correct character set.**

2. **Bootstrap 4 is mobile-first:**
   Bootstrap 4 is designed to be responsive to mobile devices.
   To ensure proper rendering and touch zooming, add the following **<meta>** tag inside the **<head>** element:

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
</html>
```

Grid system

Grid system

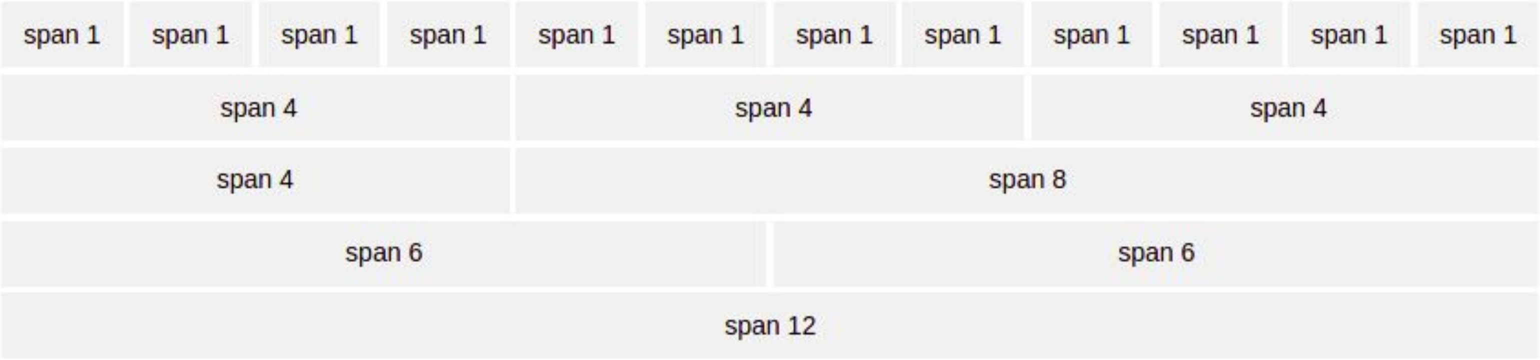
How it works

Bootstrap’s grid system uses a series of containers, rows, and columns to layout and align content. It’s built with flexbox and is fully responsive. Below is an example and an in-depth look at how the grid comes together.

CSS-Tricks: Flexbox

Bootstrap's grid system allows up to 12 columns across the page.

If you do not want to use all 12 column individually, you can group the columns together to create wider columns:



span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1 | span 1

span 4 | span 4 | span 4

span 4 | span 8

span 6 | span 6

span 12

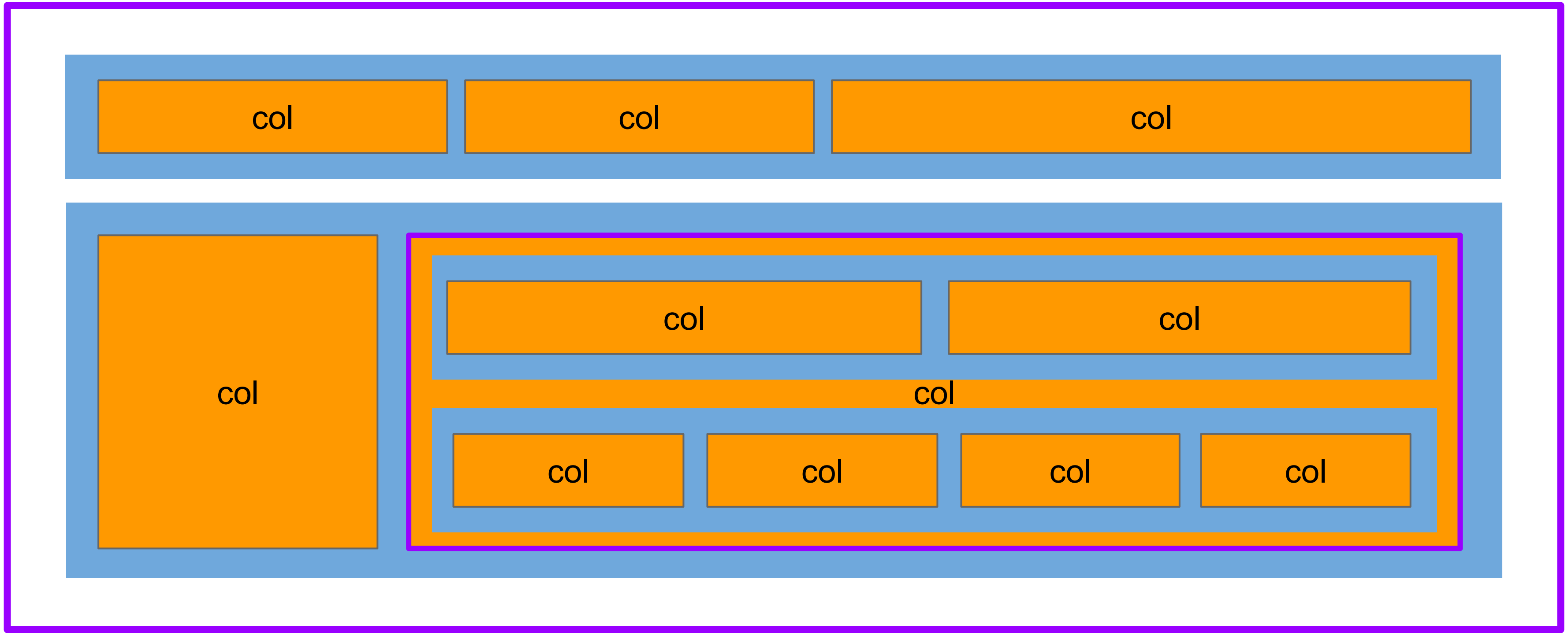
Grid system

Grid classes

Grid classes	Extra small <576px	Small (sm) ≥576px	Medium (md) ≥768px	Large (lg) ≥992px	Extra large (xl) ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	.col-	.col-sm-	.col-md-	`.col-lg-`	`.col-xl-`
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

Grid system

Grid structure



Containers

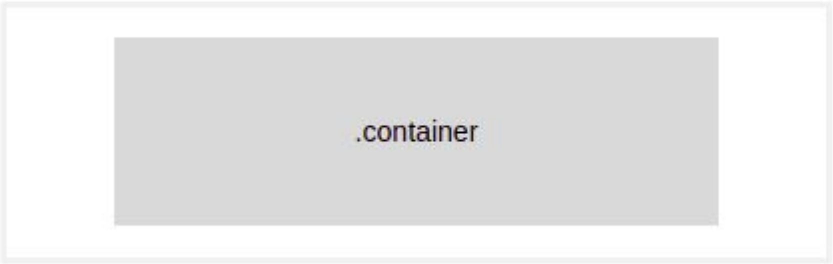
Containers

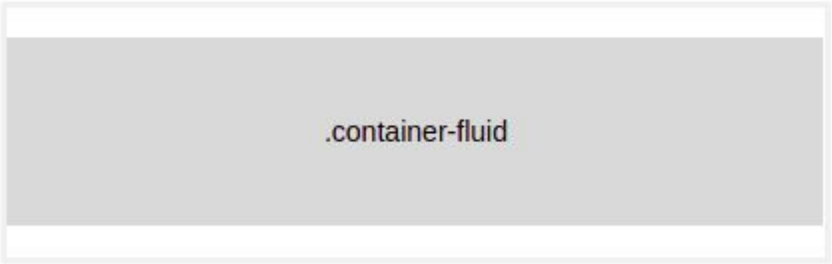
Intro

You learned from the previous chapter that Bootstrap requires a containing element to wrap site contents.

Containers are used to pad the content inside of them, and there are two container classes available:

1. The .container class provides a responsive fixed width container
2. The .container-fluid class provides a full width container, spanning the entire width of the viewport





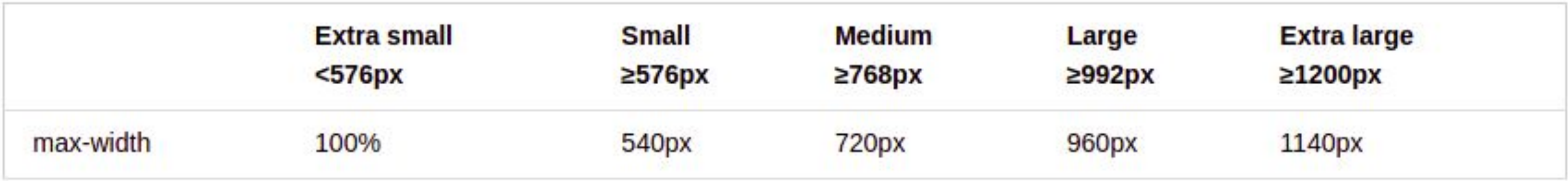
Containers

.container (Fixed container)

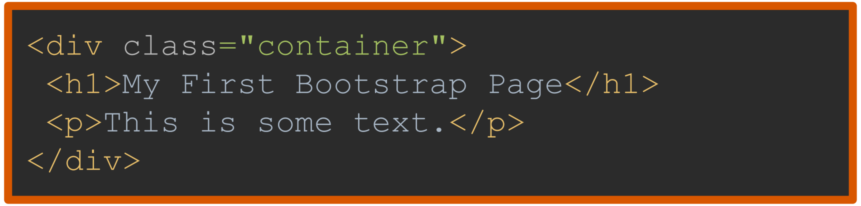
Use the .container class to create a responsive, fixed-width container.

Try it!

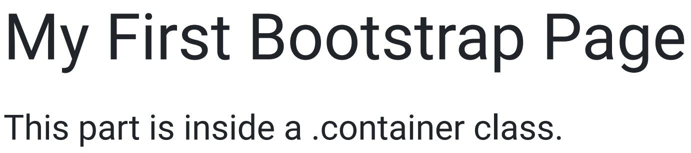
Note that its width (max-width) will change on different screen sizes:



	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
max-width	100%	540px	720px	960px	1140px



```
<div class="container">
 <h1>My First Bootstrap Page</h1>
 <p>This is some text.</p>
</div>
```



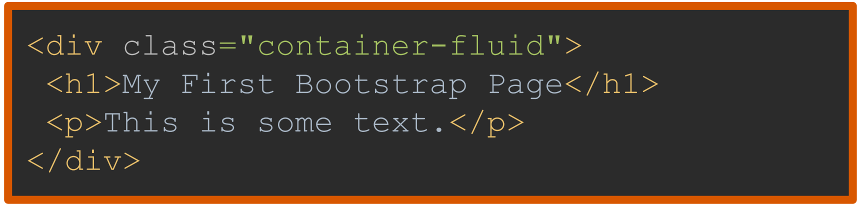
My First Bootstrap Page

This part is inside a .container class.

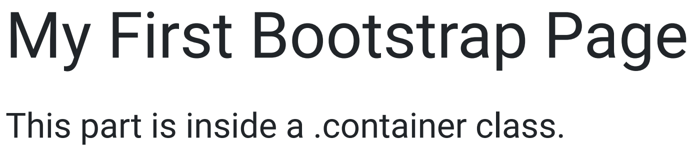
Containers

.container-fluid (Fluid Container)

Use the .container-fluid class to create a full width container, that will always span the entire width of the screen (width is always 100%):



```
<div class="container-fluid">
 <h1>My First Bootstrap Page</h1>
 <p>This is some text.</p>
</div>
```



My First Bootstrap Page

This part is inside a .container class.

Try it!

Columns

Columns

Syntax

```
col-{breakpoint_class}-{column_width}
```

Breakpoint classes:

- Auto
- Xsmall : (empty)
- Small: sm
- Medium: md
- Large: lg
- XLarge: xl

Column width:

- 1
- 2
- 3
- …
- 11
- 12

Columns

Xsmall columns: col-*

Width <576px

	Extra small	Small	Medium	Large	Extra Large
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
Screen width	<576px	>=576px	>=768px	>=992px	>=1200px

Assume we have a simple layout with two columns. We want the columns to split 25%/75% for ALL devices.

We will add the following classes to our two columns.

Try it!

Columns

Small columns: col-sm-*

576px < Width < 767px

	Extra small	Small	Medium	Large	Extra Large
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
Screen width	<576px	>=576px	>=768px	>=992px	>=1200px

The following example will result in a 25%/75% split on small (and medium, large and xlarge) devices. On extra small devices, it will automatically stack (100%):

Try it!

Columns

Medium columns: col-md-*

768px < Width < 991px

	Extra small	Small	Medium	Large	Extra Large
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
Screen width	<576px	>=576px	>=768px	>=992px	>=1200px

The following example will result in a 25%/75% split on small devices and a 50%/50% split on medium (and large and xlarge) devices. On extra small devices, it will automatically stack (100%):

Try it!

Columns

Large columns: col-lg-*

992px < Width < 1199px

	Extra small	Small	Medium	Large	Extra Large
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
Screen width	<576px	>=576px	>=768px	>=992px	>=1200px

The following example will result in a 25%/75% split on small devices, a 50%/50% split on medium devices, and a 33%/66% split on large and xlarge devices. On extra small devices, it will automatically stack (100%):

Try it!

Columns

XLarge columns: col-xl-*

1200px < Width

	Extra small	Small	Medium	Large	Extra Large
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
Screen width	<576px	>=576px	>=768px	>=992px	>=1200px

The following example will result in a 25%/75% split on small devices, a 50%/50% split on medium devices, and a 33%/66% split on large and a 20%/80% split on xlarge devices. On extra small devices, it will automatically stack (100%):

Try it!

Display

Display

Intro

```
d-{breakpoint_class}-{display_type}
```

Breakpoint classes:

- **Auto**
- Xsmall : (empty)
- Small: sm
- Medium: md
- Large: lg
- XLarge: xl

Display type:

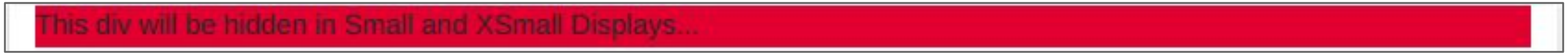
- none
- inline
- block
- …
- flex
- table

display

Example

```
<div class="d-none d-md-flex bg-danger">
 This div will be hidden in Small and XSmall Displays...
</div>
```

Medium display:



This div will be hidden in Small and XSmall Displays...

XSmall display: