**3+ Ways to Add Bootstrap 4 to Angular 10/9 With Example & Tutorial**

**Throughout this tutorial, we’ll learn how to use bootstrap 4 with Angular 10 (and any previous versions such as Angular 9) to make your Angular app responsive.**

**We’ll learn how to install (and uninstall) bootstrap in Angular from *npm, cdn or using the ng add schematic*, and add bootstrap to the *styles/scripts* arrays of the angular.json file. Next, we’ll learn how to build user interfaces with various components such as navigation bars, grid, date and time pickers, tooltips, carousels, modals, tabs, dropdowns, and forms.**

**We’ll also see the advantages of Angular implementations for Bootstrap  —  ng-bootstrap vs ngx-bootstrap vs mdbootstrap. And see some popular Angular Bootstrap templates that you can use with Angular to quickly create your layouts.**

**3+ Ways to Add Bootstrap 4 to Angular**

**In this section, we will see ways to integrate Angular and Bootstrap to style apps built.**

**We’ll see how to integrate Angular with Bootstrap, in various ways including using ng-bootstrap and ngx-bootstrap packages.**

**We’ll be using Angular CLI 10 for generating a brand new project.**

**These are the steps of our tutorial:**

* **Step 1 — Installing Angular CLI v10**
* **Step 2 — Installing Bootstrap 4 in Your Angular 10 Project**
* **Step 3 (Method 1) — Adding Bootstrap 4 to Angular 10 Usingangular.json**
* **Step 3 (Method 2) — Adding Bootstrap 4 to Angular 10 Usingindex.html**
* **Step 3 (Method 3) — Adding Bootstrap 4 to Angular 10 Using**
* **Alternative Step — Adding Bootstrap 4 Using ng-bootstrap and ngx-bootstrap**

***Note: You can also use Ionic UI components to create beautiful and professional Angular apps, read***[***Building Chat App Frontend UI with JWT Auth Using Ionic 5/Angular 10***](https://www.techiediaries.com/angular/ionic-chat-ui-jwt-auth/)

**What is Bootstrap**

[**Bootstrap**](https://www.techiediaries.com/bootstrap-tutorial/)**is the most popular HTML and CSS framework for building responsive layouts with ease and without having a deep knowledge of CSS (Still custom CSS is required to customize your design and make it different from the other Bootstrap-styled websites unless you are using a BS theme developed specifically for your requirements.**

**Bootstrap 4 is the latest version of Bootstrap which brings many new and powerful features to the framework most importantly Flexbox which is now the default display system for the Bootstrap grid layout (one of the most important features of Bootstrap).**

**3+ Ways to Include Bootstrap 4 In Your Angular Project**

**You can include Bootstrap in your Angular project in multiple ways:**

* **Including the Bootstrap CSS and JavaScript files in the <head> section of the index.html file of your Angular project with a <link> and <script> tags,**
* **Importing the Bootstrap CSS file in the global styles.css file of your Angular project with an @import keyword.**
* **Adding the Bootstrap CSS and JavaScript files in the styles and scripts arrays of the angular.json file of your project**

**Step 1 — Installing Angular CLI v10**

**Let’s get started by installing**[**Angular CLI v10**](https://cli.angular.io/)**if it is not yet installed on your machine.**

**Head over to a new command-line interface and run the following command to install the latest version of the Angular CLI:**

**$ npm install -g @angular/cli**

***Note: This will***[***install the Angular 10 CLI***](https://www.ahmedbouchefra.com/angular/install-angular-9-cli-and-create-project-with-routing/)***globally on your system so depending on your npm configuration you may need to add sudo (for superuser access) in macOS and Linux or use a command prompt with admin access in Windows.***

**After the installation, you’ll have at your disposal the *ng* utility. Let’s use it to generate a new Angular 10 project as follows:**

**$ ng new angular-bootstrap-examples**

**You will be prompted for a couple of questions:**

**? Would you like to add Angular routing? Yes? Which stylesheet format would you like to use? (Use arrow keys)> CSS SCSS [ https://sass-lang.com/documentation/syntax #scss ] Sass [ https://sass-lang.com/documentation/syntax#the-indented-syntax ] Less [ http://lesscss.org ] Stylus [ http://stylus-lang.com ]**

**Most importantly, choose CSS as the stylesheet format because we’ll use the CSS version of Bootstrap in our tutorial.**

**The command will generate the directory structure and necessary files for the project and will install the required dependencies.**

**Next, navigate inside the root folder of your project:**

**$ cd angular-bootstrap-examples**

**You can then serve your**[**Angular 10 application**](https://www.techiediaries.com/angular/angular-10-8-tutorial-by-example-rest-crud-apis-http-get-requests-with-httpclient/)**using the ng serve command as follows:**

**$ ng serve**

**Your app will be served from**[**http://localhost:4200/**](http://localhost:4200/)

**Step 2 — Installing Bootstrap 4 in Your Angular 10 Project**

**In this step, we’ll proceed to add Bootstrap 4 to our Angular 10 application.**

**There are various ways that you can use to install Bootstrap in your project:**

* **Installing Bootstrap from npm using the npm install command,**
* **Downloading Bootstrap files and adding them to the src/assets folder of your Angular project,**
* **Using Bootstrap from a CDN.**

**Let’s proceed with the first method. Go back to your command-line interface and install Bootstrap 4 via npm as follows:**

**$ npm install bootstrap**

**This will also add the *bootstrap* package to package.json. At the time of writing this tutorial, bootstrap v4.3.1 will be installed.**

**The Bootstrap 4 assets will be installed in the node\_modules/bootstrap folder. You'll need to tell Angular where to look for them.**

**Next, you also need to install jQuery using the following command:**

**$ npm install jquery**

**At the time of this tutorial jquery v3.4.1 will be installed.**

**Step 3 (Method 1) — Adding Bootstrap 4 to Angular 10 Using angular.json**

**Open the angular.json file of your project and include:**

* **node\_modules/bootstrap/dist/css/bootstrap.css in the projects->architect->build->styles array,**
* **node\_modules/bootstrap/dist/js/bootstrap.js in the projects->architect->build->scripts array,**
* **node\_modules/bootstrap/dist/js/bootstrap.js in the projects->architect->build->scripts array,**

**As follows:**

**{ "$schema": "./node\_modules/@angular/cli/lib/config/schema.json", "version": 1, "newProjectRoot": "projects", "projects": { "angular-bootstrap-examples": { "projectType": "application", "schematics": {}, "root": "", "sourceRoot": "src", "prefix": "app", "architect": { "build": { "builder": "@angular-devkit/build-angular:browser", "options": { "outputPath": "dist/angular-bootstrap-examples", "index": "src/index.html", "main": "src/main.ts", "polyfills": "src/polyfills.ts", "tsConfig": "tsconfig.app.json", "aot": true, "assets": [ "src/favicon.ico", "src/assets" ], "styles": [ "./node\_modules/bootstrap/dist/css/bootstrap.css", "src/styles.css" ], "scripts": [ "./node\_modules/jquery/dist/jquery.js", "./node\_modules/bootstrap/dist/js/bootstrap.js" ] },**

***Note: You also need to add the jQuery JavaScript library file.***

**Step 3 (Method 2) — Adding Bootstrap 4 to Angular 10 Using index.html**

**You can also include Bootstrap files from node\_modules/bootstrap using the index.html file.**

**Open the src/index.html file and add the following tags:**

* **A <link> tag for adding the bootstrap.css file in the <head> section,**
* **A <script> tag for adding the jquery.js file before the closing </body> tag,**
* **A <script> tag for adding the bootstrap.js file before the </body> tag.**

**This is an example:**

**<!doctype html><html lang="en">**

**<head>**

**<meta charset="utf-8">**

**<title>Angular Bootstrap 4 Examples</title> <base href="/">**

**<meta name="viewport" content="width=device-width, initial-scale=1">**

**<link rel="icon" type="image/x-icon" href="favicon.ico">**

**<link rel="stylesheet" href="../node\_modules/bootstrap/dist/css/bootstrap.css">**

**</head>**

**<body>**

**<app-root></app-root>**

**<script src="../node\_modules/jquery/dist/jquery.js"></script> <script src="../node\_modules/bootstrap/dist/js/bootstrap.js"></script>**

**</body>**

**</html>**

**Step 3 (Method 3) — Adding Bootstrap 4 to Angular 10 Using styles.css**

**We can also use the styles.css file to add the CSS file of Bootstrap to our project.**

**Open the src/styles.css file of your Angular project and import the bootstrap.css file as follows:**

**@import "~bootstrap/dist/css/bootstrap.css"**

**This replaces the previous method(s), so you don’t need to add the file to the styles array of the angular.json file or to the index.html file.**

***Note: The JavaScript file(s) can be added using the scripts array or the <script> tag like the previous methods.***

**Alternative Step — Adding Bootstrap 4 Using ng-bootstrap and ngx-bootstrap**

**Bootstrap depends on the jQuery and popper.js libraries, and if you don’t include them in your project, any Bootstrap components that rely on JavaScript will not work.**

**Why not include those libs? For Angular, it’s better to avoid using libraries that make direct manipulation of the DOM (like jQuery) and let Angular handle that.**

**Now, what if you need the complete features of Bootstrap 4 without the JavaScript libraries?**

**A better way is to use component libraries created for the sake of making Bootstrap work seamlessly with Angular such as ng-bootstrap andngx-bootstrap**

***Should I add bootstrap.js or bootstrap.min.js to my project? No, the goal of ng-bootstrap is to completely replace JavaScript implementation for components. Nor should you include other dependencies like jQuery or popper.js. It is not necessary and might interfere with ng-bootstrap code***[***Source***](https://ng-bootstrap.github.io/#/getting-started)

**So first you’ll need to install the library from npm using the following command:**

**$ npm install @ng-bootstrap/ng-bootstrap**

**Once you finish the installation, you’ll need to import the main module:**

**import {NgbModule} from '@ng-bootstrap/ng-bootstrap';**

**Next, you’ll need to add the module you imported in your app root module as follows:**

**import {NgbModule} from '@ng-bootstrap/ng-bootstrap';**

**@NgModule({ declarations: [/\*...\*/], imports: [/\*...\*/, NgbModule.forRoot()], /\*...\*/})export class AppModule {}**

**Please note that ng-bootstrap requires the Bootstrap 4 CSS file to be present.**

**You can add it in the styles array of the angular.json file as follows:**

**"styles": [ "styles.css", "../node\_modules/bootstrap/dist/css/bootstrap.css"],**

**Now, you can use Bootstrap 4 in your Angular application.**

**You can find all the available components via this**[**link**](https://ng-bootstrap.github.io/#/components/accordion/examples)**.**

**You can also use the ngx-bootstrap library. Simply head back to your terminal, make sure you are inside your Angular project then run the following command to install ngx-bootstrap:**

**$ npm install ngx-bootstrap**

**You also need the Bootstrap 4 CSS files. Add the following line in the <head> of your Angular app which includes Bootstrap from a CDN:**

**<link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">**

**You can also install bootstrap from npm and use the previous way to include the CSS file (via the styles array in the angular-cli.json file):**

**"styles": [**

**"../node\_modules/bootstrap/dist/css/bootstrap.min.css",**

**"styles.css" ],**

**Next, open the src/app/app.module.ts file and update is as follows:**

**import { BsDropdownModule} from 'ngx-bootstrap/dropdown';import { AlertModule } from 'ngx-bootstrap';**

**/\*...\*/**

**@NgModule({ /\*...\*/ imports: [BsDropdownModule.forRoot(),AlertModule.forRoot(), /\*...\*/ ], /\*...\*/ })**

**This an example of importing two components *BsDropdownModule* and *AlertModule*.**

**You need to import the module for each component you want to use in the same way.**

**ngx-bootstrap provides each Bootstrap component in each own module so you only import the components you need. In this way, your app will be smaller since it bundles only the components you are actually using.**

**You can find all the available components that you can use from the**[**docs**](https://valor-software.com/ngx-bootstrap/#/getting-started)**.**

**Adding Bootstrap 4 to Angular 10 Using Schematics**

**Thanks to the new ng add command added on Angular 7+, you have a new simpler and easier way to add Bootstrap without using the npm install command for installing the required dependencies or adding any configurations.**

**You can simply run the following command to add ng-bootstrap:**

**$ ng add @ng-bootstrap/schematics**

**That’s it. You now have support for Bootstrap components and styles without any extra configurations. You also don’t need jQuery since we are using ng-bootstrap.**

**Summary**

**In this section, we’ve seen different ways of including Bootstrap 4 in Angular 10 apps, such as:**

* **Using original Bootstrap 4 assets from npm,**
* **Using the ng-bootstrap library,**
* **And finally using the ngx-bootstrap**

**The most important difference between ng-bootstrap vs. ngx-bootstrapis that ngx-bootstrap uses separate modules for components to reduce the final app size.**

**Styling an Angular 10 Example App with Bootstrap 4 Navbar, Jumbotron, Tables, Forms, and Cards**

**In this section, we’ll learn how to integrate and use bootstrap 4 with Angular 10 by building an example application step by step.**

**We’ll see how to initialize an Angular 10 project and integrate it with Bootstrap 4. Next, we’ll use the various Bootstrap 4 CSS utilities to create a responsive layout with navbars, tables, forms, buttons, cards, and jumbotrons.**

**Bootstrap is a free and open-source CSS framework for creating responsive layouts, it’s mobile-first and contains ready CSS utilities for typography, forms, buttons, and navigation, etc.**

***This tutorial works with all recent versions of Angular i.e version 7, 8, 9 and 10.***

**There are various ways of integrating Angular with Bootstrap that we have seen in the previous section. Let’s see a possible solution by example. These are the steps of this section:**

* **Step 1 — Installing Angular CLI 10**
* **Step 2 — Initializing your Angular 10 Project**
* **Step 3 — Installing Bootstrap 4**
* **Step 4 — Creating Angular Components and Setting up Routing**
* **Step 5 — Adding a Bootstrap 4 Jumbotron**
* **Step 6 — Creating an Angular Bootstrap 4 Table**
* **Step 7 — Adding a Bootstrap 4 Form Component**

**Step 1 — Installing Angular CLI 10**

**Let’s start by installing the latest version of Angular CLI. In your terminal, run the following command:**

**$ npm install -g @angular/cli**

**Step 2 — Initializing your Angular 10 Project**

**After installing Angular CLI, let’s initialize an Angular 10 project by running the following command:**

**$ ng new angular10bootstrapexample**

**The CLI will then ask you:**

**Would you like to add Angular routing?**

**Press Y.**

**Next, it will ask you:**

**Which stylesheet format would you like to use?**

**Choose “CSS”.**

**Next, we need to set up Angular 10 forms.**

**Go to the src/app/app.module.ts file, import FormsModule from @angular/forms, and include it in the imports array as follows:**

**import { BrowserModule } from '@angular/platform-browser';import { NgModule } from '@angular/core';import { AppRoutingModule } from './app-routing.module';import { FormsModule } from '@angular/forms';**

**@NgModule({ declarations: [ /\* ... \*/ ], imports: [ BrowserModule, AppRoutingModule, FormsModule ], providers: [], bootstrap: [AppComponent]})export class AppModule { }**

**Step 3 — Installing Bootstrap 4**

**After initializing your Angular 10 project, let’s proceed to install Bootstrap 4 and integrate it with Angular.**

**Go to your project’s folder:**

**$ cd angular10bootstrapexample**

**Next, install Bootstrap 4 and jQuery from npm using the following command:**

**$ npm install bootstrap jquery**

**Next, go the angular.json file and add the paths of Bootstrap CSS and JavaScript files as well as jQuery to the styles and scripts arrays under the build target as follows:**

**"architect": { "build": { [...], "styles": [ "src/styles.css", "node\_modules/bootstrap/dist/css/bootstrap.min.css" ], "scripts": [ "node\_modules/jquery/dist/jquery.min.js", "node\_modules/bootstrap/dist/js/bootstrap.min.js" ] },**

**Step 4 — Creating Angular Components and Setting up Routing**

**After installing and integrating Bootstrap 4 with your Angular 10 project, let’s create some components to test various Bootstrap styles.**

**Go to your command-line interface and run the following commands:**

**$ ng generate component jumbotron**

**$ ng generate component bootstrap-form**

**$ ng generate component bootstrap-table**

**Next, we need to include these components in the routing module to enable multiple views.**

**Go to the src/app/app-routing.module.ts file and update it as follows:**

**import { NgModule } from '@angular/core';**

**import { Routes, RouterModule } from '@angular/router';**

**import { BootstrapTableComponent } from './bootstrap-table/bootstrap-table.component';**

**import { BootstrapFormComponent } from './bootstrap-form/bootstrap-form.component';**

**import { JumbotronComponent } from './jumbotron/jumbotron.component';**

**const routes: Routes = [**

**{path: "", pathMatch: "full",redirectTo: "home"},**

**{path: "jumbotron", component: JumbotronComponent},**

**{path: "bootstrap-form", component: BootstrapFormComponent},**

**{path: "bootstrap-table", component: BootstrapTableComponent} ];**

**@NgModule({**

**imports: [RouterModule.forRoot(routes)], exports: [RouterModule]})**

**export class AppRoutingModule { }**

**Step 5 — Adding A Bootstrap 4 Jumbotron**

[**A Bootstrap Jumbotron**](https://getbootstrap.com/docs/4.4/components/jumbotron/)**is a lightweight, flexible component that can optionally extend the entire viewport to showcase key marketing messages on your site**

**Let’s add a Bootstrap Jumbotron component to our jumbotron page.**

**Head to the src/app/jumbotron/jumbotron.component.html file and add the following HTML markup:**

**<div class="jumbotron" style="height: calc(95vh);">**

**<h1>Angular 10 Bootstrap 4 Demo</h1>**

**<p class="lead">**

**This tutorial teaches you how to integrate Bootstrap 4 with Angular 10**

**</p>**

**</div>**

**We use the built-in .jumbotron class to create a Bootstrap Jumbotron.**

**Step 6 — Creating an Angular 10 and Bootstrap 4 Table**

**Let’s now see how to use a Bootstrap 4 table to display tabular data.**

**Go the src/app/bootstrap-table/bootstrap-table.component.ts file and add some data that we can display:**

**import { Component, OnInit } from '@angular/core';**

**@Component({ selector: 'app-bootstrap-table', templateUrl: './bootstrap-table.component.html', styleUrls: ['./bootstrap-table.component.css']})**

**export class BootstrapTableComponent implements OnInit {**

**employees = [ {id: 1, name: "E 001", description: "E 001 des", email: "e001@email.com"}, {id: 2, name: "E 002", description: "E 002 des", email: "e002@email.com"}, {id: 3, name: "E 003", description: "E 003 des", email: "e003@email.com"}, {id: 4, name: "E 004", description: "E 004 des", email: "e004@email.com"} ]; selectedEmployee;**

**constructor() { }**

**ngOnInit() { }**

**public createEmployee(e: {id, name, description, email}){ this.employees.push(e); }**

**public selectEmployee(e){ this.selectedEmployee = e; }}**

**We simply defined two variables employees and selectedEmployee for holding the set of employees and the selected employee. And a selectEmployee() method which assigns the selected employee to the selectedEmployee variable.**

**Next, go to the src/app/bootstrap-table/bootstrap-table.component.html file and update it as follows:**

**<div class="container" style="margin-top: 70px;"> <table class="table table-hover"> <thead> <tr> <th>#</th> <th>Name</th> <th>Email</th> <th>Actions</th> </tr> </thead> <tbody> <tr \*ngFor="let employee of employees">**

**<td></td> <td> </td> <td> </td> <td> <button class="btn btn-primary" (click)="selectEmployee(employee)"> Select</button> </td> </tr> </tbody> </table> <div class="card text-center" \*ngIf="selectedEmployee"> <div class="card-header"> # </div> <div class="card-block"> <h4 class="card-title"></h4> <p class="card-text">**

**</p> </div>**

**</div></div>**

[**A Bootstrap 4 Card**](https://getbootstrap.com/docs/4.4/components/card/)**is a flexible and extensible content container. It includes options for headers and footers, a wide variety of content, contextual background colors, and powerful display options. If you’re familiar with Bootstrap 3, cards replace our old panels, wells, and thumbnails. Similar functionality to those components is available as modifier classes for cards.**

**We use the built-in .table and .table-hover classes to create Bootstrap tables, the .card, .card-block, .card-title and .card-text classes to create cards.**

**Step 7 — Adding A Bootstrap 4 Form Component to your Angular 10 App**

**Let’s proceed by adding a Bootstrap-styled form to the bootstrap-form component.**

**Next, go to the src/app/bootstrap-form/bootstrap-form.component.ts file and update it as follows:**

**import { Component, OnInit } from '@angular/core';**

**@Component({ selector: 'bootstrap-form/-create', templateUrl: './bootstrap-form/.component.html', styleUrls: ['./bootstrap-form/.component.css']})export class BootstrapForm/Component implements OnInit {**

**employee : {id, name, description, email} = {id: null, name: "", description: "", email: ""};**

**constructor() { }**

**ngOnInit() { }**

**createEmployee(){ console.log("Employee created: ", this.employee); this.employee = {id: null, name: "", description: "", email: ""};**

**}}**

**Next, go to the src/app/bootstrap-form/bootstrap-form.component.html file and update it as follows:**

**<div class="container" style="margin-top: 70px;">**

**<div class="row">**

**<div class="col-sm-8 offset-sm-2">**

**<div> <form> <div class="form-group"> <label for="id">ID</label> <input [(ngModel)]="employee.id" type="text" name="id" class="form-control" id="id" aria-describedby="idHelp" placeholder="Employee ID"> <small id="idHelp" class="form-text text-muted">Enter your employee’s ID</small>**

**<label for="name">Employee Name</label> <input [(ngModel)]="employee.name" type="text" name="name" class="form-control" id="name" aria-describedby="nameHelp" placeholder="Enter your employee name"> <small id="nameHelp" class="form-text text-muted">Enter your employee’s name</small>**

**<label for="email">Employee Email</label> <input [(ngModel)]="contact.email" type="text" name="email" class="form-control" id="email" aria-describedby="emailHelp" placeholder="Enter your employee email"> <small id="nameHelp" class="form-text text-muted">Enter your employee’s email</small>**

**<label for="description">Employee Description</label> <textarea [(ngModel)]="employee.description" name="description" class="form-control" id="description" aria-describedby="descHelp"> </textarea> <small id="descHelp" class="form-text text-muted">Enter your employee’s description</small>**

**</div> </form> <button class="btn btn-primary" (click)="createEmployee()">Create employee</button> </div> </div> </div></div>**

**We make use of the .form-group and .form-control classes to create a Bootstrap form.**

**Step 8 — Serving your Angular 10 Application**

**Head over to your command-line interface, and run the following command from the folder of your project:**

**$ ng serve**

**A development server will be started at the http://localhost:4200 address.**

**Summary**

**As a recap, we have seen how to initialize an Angular 10 project and integrate it with Bootstrap 4. Next, we used various Bootstrap CSS utilities to create a responsive layout with tables, forms, buttons, cards and jumbotrons.**

**Angular 10 Carousel Example with Bootstrap 4**

**In this tutorial, we'll learn how to create a carousel with the latest Angular 10 version and Bootstrap 4.**

***The carousel is a slideshow for cycling through a series of content, built with CSS 3D transforms and a bit of JavaScript. It works with a series of images, text, or custom markup. It also includes support for previous/next controls and indicators.***[***Source***](https://getbootstrap.com/docs/4.4/components/carousel/)

**Before getting started you need a few prerequisites:**

* **Basic knowledge of TypeScript. Particularly the familiarity with Object Oriented concepts such as TypeScript classes and decorators.**
* **A local development machine with Node 10+, together with NPM 6+ installed. Node is required by the Angular CLI like the most frontend tools nowadays. You can simply go to the downloads page of**[**the official website**](https://nodejs.org/en/download/)**and download the binaries for your operating system. You can also refer to your specific system instructions for how to install Node using a package manager. The recommended way though is using**[**NVM**](https://github.com/nvm-sh/nvm)**— Node Version Manager — a POSIX-compliant bash script to manage multiple active Node.js versions.**

**Note: If you don't want to install a local environment for Angular development but still want to try the code in this tutorial, you can use**[**Stackblitz**](https://stackblitz.com/)**, an online IDE for frontend development that you can use to create an Angular project compatible with Angular CLI.**

**Step 1 — Installing Angular CLI 10**

**Let's start by installing the latest Angular CLI 10 version.**

[**Angular CLI**](https://cli.angular.io/)**is the official tool for initializing and working with Angular projects. Head over to a new command-line interface and run the following command:**

**$ npm install -g @angular/cli**

**At the time of writing this tutorial, angular/cli v10 will be installed on your system.**

**Step 2 — Creating a New Angular 10 App**

**In the second step, let's create our project. Head back to your command-line interface and run the following commands:**

**$ cd ~**

**$ ng new angular10carousel**

**The CLI will ask you a couple of questions — If Would you like to add Angular routing? Type y for Yes and Which stylesheet format would you like to use? Choose CSS.**

**Next, navigate to you project’s folder and run the local development server using the following commands:**

**$ cd angular10carousel**

**$ ng serve**

**Open your web browser and navigate to the http://localhost:4200/ address to see your app running.**

**Step 3 — Installing Ng-Bootstrap**

**Next, we need to install ng-bootstrap using the following command:**

**$ ng add @ng-bootstrap/ng-bootstrap**

**This library provides an Angular implementation for Bootstrap 4 so you don't need to use jQuery.**

**Open the src/app/app.component.ts file and update it as follows:**

**import { Component } from '@angular/core';**

**import { NgbCarouselConfig } from '@ng-bootstrap/ng-bootstrap';**

**@Component({**

**selector: 'my-app',**

**templateUrl: './app.component.html',**

**styleUrls: [ './app.component.css' ],**

**providers: [NgbCarouselConfig] // add NgbCarouselConfig to the component providers**

**})**

**export class AppComponent {**

**images = [700, 800, 807].map((n) => `https://picsum.photos/id/${n}/900/500`);**

**constructor(config: NgbCarouselConfig) {**

**//**

**config.interval = 2000;**

**config.keyboard = true;**

**config.pauseOnHover = true;**

**}**

**}**

**We import NgbCarouselConfig and add it to the providers array of the component, next we inject it via the constructor and use it to customize default values of carousels used by this component and its children. We set the interval between slides to two seconds, enabled the keyboard to move slides, and pause on hover on each slide.**

**We also defined an array of images to use for slides.**

**Next, open the src/app/app.component.html file and update it as follows:**

**<div class="container-fluid">**

**<h1>**

**Angular 10 Carousel Example**

**</h1>**

**<h2>Full tutorial in Techiediaries</h2>**

**<ngb-carousel \*ngIf="images">**

**<ng-template ngbSlide>**

**<div class="wrapper">**

**<img [src]="images[0]" alt="Random first slide">**

**</div>**

**<div class="carousel-caption">**

**<h3>First Slide</h3>**

**<p> Angular 10 Carousel Example </p>**

**</div>**

**</ng-template>**

**<ng-template ngbSlide>**

**<div class="wrapper">**

**<img [src]="images[1]" alt="Random second slide">**

**</div>**

**<div class="carousel-caption">**

**<h3>Second Slide</h3>**

**<p> Check out Techiediaries</p>**

**</div>**

**</ng-template>**

**<ng-template ngbSlide>**

**<div class="wrapper">**

**<img [src]="images[2]" alt="Random third slide">**

**</div>**

**<div class="carousel-caption">**

**<h3>Third Slide</h3>**

**<p>for full tutorial...</p>**

**</div>**

**</ng-template>**

**</ngb-carousel>**

**</div>**

**We use ng-template with the ngbSlide directive for add a carousel slide and we use HTML to define the content for the slide.**

**Next, open the src/app/app.component.html file and add the following styles:**

**ngb-carousel .wrapper {**

**position: relative;**

**height: 0;**

**padding-top: 55%; /\* Keep ratio for 900x500 images \*/**

**}**

**ngb-carousel .wrapper>img {**

**position: absolute;**

**top: 0;**

**left: 0;**

**bottom: 0;**

**right: 0;**

**}**

**You can play with this example from**[**https://stackblitz.com/edit/angular-10-carousel-example**](https://stackblitz.com/edit/angular-10-carousel-example)

**Summary**

**In this tutorial, we've seen how to create a carousel with rich slides in Angular 10 using ng-bootstrap.**

**Angular 10 Rating Example With Ng-Bootstrap**

**In this section, we'll see by example how to create a rating component with Bootstrap 4, HTML Select and Angular 10 Forms. We'll be using the ngb-rating component from ng-bootstrap. We'll also see how to use the HTML select control with the ngFor directive inside a reactive form. How to bind select element to a TypeScript object or string literal using [ngValue] and value properties respectively, and how to assign a default value to select from an array of elements.**

[***NgBoostrap***](https://ng-bootstrap.github.io/#/home)***is the Angular adapted version of Bootstrap UI components. Using ng-bootstrap we can easily integrate bootstrap library to our Angular project and use it’s awesome UI components very easily. Bootstrap is tried and tested and fully responsive for multiple platforms and screen sizes. Moreover, it is now an industrial standard adopted almost everywhere.***

**HTML forms are necessary in most web applications. Selects in forms can be used when you have multiple options and want users to select one of them before submitting the form. In Angular, you can use objects for option values and not only strings.**

**This is an example of a component template with the select control:**

**<form [formGroup]="myForm">**

**<select formControlName="myControl">**

**<option [value]="city" \*ngFor="let city of cities">**

**</option>**

**</select>**

**</form>**

**In our Angular component, we need to have a cities array with some cities.**

**We use the value property to bind the city to select but you can also use ngValue instead. The value property is used with string literals only, whereas ngValue can be used with objects. We'll see next another example of using select with objects instead of strings.**

**This comes handy if we have a drop-down where we need to show the names of the objects from a TypeScript array. But when selecting the element from the drop-down you need to select the id of the array element for querying the database for example. In this case, we need to use ngValue as it works with TypeScript objects and not just strings.**

**In the previous tutorial, you've seen how you can install Angular 10 CLI from npm, so let's start by creating a new project.**

**Creating a New Angular 10 Project**

**Let's get started by creating a new Angular 10 project using the following command:**

**$ ng new AngularRatingExample**

**? Would you like to add Angular routing? Yes**

**? Which stylesheet format would you like to use? CSS**

**Adding Ng-Bootstrap**

**We’ll see how to set up Bootstrap in our project for styling our UI using ng-bootstrap. For that purpose, we’ll first need to add bootstrap and ng-bootstrap from npm in our project using the following command:**

**$ cd AngularRatingExample**

**$ ng add @ng-bootstrap/ng-bootstrap**

**This will install ng-bootstrap for the default application specified in your angular.json file.**

**Since ng-bootstrap has a dependency on i18n, we’ll also need add the package to our project using the following command:**

**$ ng add @angular/localize**

**Next, open the src/app/app.module.ts file and add NgbModule and ReactiveFormsModule in the imports array of AppModule as follows:**

**// [...]**

**import { ReactiveFormsModule } from '@angular/forms';**

**import { NgbModule } from '@ng-bootstrap/ng-bootstrap';**

**@NgModule({**

**imports: [**

**// [...]**

**ReactiveFormsModule,**

**NgbModule**

**],**

**})**

**Next, open the src/app/app.component.html file and update it as follows:**

**<div class="container">**

**<div class="row">**

**<div class="col-12">**

**<router-outlet></router-outlet>**

**</div>**

**</div>**

**</div>**

**We simply wrap the router outlet with some HTML markup styled with Bootstrap 4.**

**Creating an Angular 10 Rating Component**

**Next, let's create an Angular 10 component that will encapsulate our form. Head back to your terminal and run the following command:**

**$ ng generate component rating**

**Using the HTML Select Control with the ngFor Directive with a Reactive Form**

**Next, open the src/app/rating/rating.component.html file and add the following form:**

**<form [formGroup]="form" (ngSubmit)="submit()">**

**<div class="form-group">**

**<label>Book</label>**

**<select formControlName="book">**

**<option \*ngFor="let book of books" [ngValue]="book"></option>**

**</select>**

**</div>**

**<div class="form-group">**

**<ngb-rating [max]="5" formControlName="rating"></ngb-rating>**

**</div>**

**<button [disabled]="form.invalid || form.disabled" class="btn btn-primary">Rate the book!</button>**

**</form>**

**We used the HTML select element with ngValue, and the ngFor directive inside our reactive form.**

[**\*ngFor**](https://angular.io/guide/template-syntax#ngFor)**is the Angular repeater directive. It simply repeats the host element for each element in a list.**

**The syntax in this example is as follows:**

* **<option> is the host element.**
* **books holds the books list from the RatingComponent class.**
* **book holds the current book object for each iteration through the list.**

**Next, update the src/app/rating/rating.component.ts file as follows:**

**// [...]**

**export class RatingComponent implements OnInit {**

**books = [**

**{ name: 'Book 1' },**

**{ name: 'Book 2' },**

**{ name: 'Book 3' },**

**{ name: 'Book 4' },**

**{ name: 'Book 5' }**

**];**

**form = new FormGroup({**

**book: new FormControl(this.books[0], Validators.required),**

**rating: new FormControl('', Validators.required),**

**});**

**submit() {**

**console.log(JSON.stringify(this.form.value));**

**this.form.reset();**

**}**

**}**

**We also provided a default value from the array books for select using book: new FormControl(this.books[0], Validators.required). The first parameter is the default value.**

**Next, we need to add our rating component to the router configuration. Open the src/app/app-routing.module.ts file and update it as follows:**

**import { NgModule } from '@angular/core';**

**import { Routes, RouterModule } from '@angular/router';**

**import { RatingComponent } from '../rating/rating.component.ts';**

**const routes: Routes = [**

**{path: '', component: RatingComponent}**

**];**

**@NgModule({**

**imports: [RouterModule.forRoot(routes)],**

**exports: [RouterModule]**

**})**

**export class AppRoutingModule { }**

**Finally, you can start your development server using the following command:**

**$ ng serve**

**Summary**

**In this Angular 10 tutorial, we’ve covered building a basic form with a rating component based on the ngb-rating component from ng-bootstrap. We have seen how to use the HTML select control with the ngFor directive inside a reactive form. How to bind select to a TypeScript object instead of string and how to assign a default value to select.**

**Angular 10 Star Rating Example**

**In this section, we'll build a star rating component with the latest Angular 10 version and Bootstrap.**

**Star rating is a common feature in product recommendation and eCommerce websites**

**We commonly rate something between zero and five stars.**

**Before getting started you need a few prerequisites:**

* **Basic knowledge of TypeScript. Particularly the familiarity with Object Oriented concepts such as TypeScript classes and decorators.**
* **A local development machine with Node 10+, together with NPM 6+ installed. Node is required by the Angular CLI like the most frontend tools nowadays. You can simply go to the downloads page of**[**the official website**](https://nodejs.org/en/download/)**and download the binaries for your operating system. You can also refer to your specific system instructions for how to install Node using a package manager. The recommended way though is using**[**NVM**](https://github.com/nvm-sh/nvm)**— Node Version Manager — a POSIX-compliant bash script to manage multiple active Node.js versions.**

**Note: If you don't want to install a local environment for Angular development but still want to try the code in this tutorial, you can use**[**Stackblitz**](https://stackblitz.com/)**, an online IDE for frontend development that you can use to create an Angular project compatible with Angular CLI.**

**Step 1 — Installing Angular CLI 10**

**Let's begin by installing the latest Angular CLI 10 version (at the time of writing this tutorial).**

[**Angular CLI**](https://cli.angular.io/)**is the official tool for initializing and working with Angular projects. To install it, open a new command-line interface and run the following command:**

**$ npm install -g @angular/cli**

**At the time of writing this tutorial, angular/cli v10 will be installed on your system.**

**Step 2 — Creating a New Angular 10 App**

**In the second step, let's create our project. Head back to your command-line interface and run the following commands:**

**$ cd ~**

**$ ng new angular10star-rating**

**The CLI will ask you a couple of questions — If Would you like to add Angular routing? Type y for Yes and Which stylesheet format would you like to use? Choose CSS.**

**Next, navigate to you project’s folder and run the local development server using the following commands:**

**$ cd angular10star-rating**

**$ ng serve**

**Open your web browser and navigate to the http://localhost:4200/ address to see your app running.**

**Step 3 — Installing Ng-Bootstrap**

**Next, we need to install ng-bootstrap using the following command:**

**$ ng add @ng-bootstrap/ng-bootstrap**

**This library provides an Angular implementation for Bootstrap 4 and also provides some useful components such as NgbRating -- a directive that allows you to display star rating bar.**

**Open the src/app/app.component.ts file and update it as follows:**

**import { Component } from '@angular/core';**

**@Component({**

**selector: 'app-root',**

**templateUrl: './app.component.html',**

**styleUrls: ['./app.component.css']**

**})**

**export class AppComponent {**

**currentRate = 0;**

**}**

**Next, open the src/app/app.component.html file and update it as follows:**

**<h1 class="text-primary">**

**Angular 10 Start Rating Example**

**</h1>**

**<ngb-rating [max]="5" [(rate)]="currentRate" [readonly]="false"></ngb-rating>**

**<p>Rate: </p>**

**Next, open the src/app/app.component.css file and add the following CSS styles:**

**ngb-rating {**

**font-size: 100px;**

**color:brown;**

**background: rgba(23, 221, 16, 0.815);**

**}**

**You can find this example in**[**https://stackblitz.com/edit/angular-10-star-rating-example**](https://stackblitz.com/edit/angular-10-star-rating-example)

**Summary**

**In this short article, we've seen how to create a star rating component with Angular 10 and ng-bootstrap. Read the**[**official docs**](https://ng-bootstrap.github.io/#/components/rating/examples)**for more details.**

**Creating an Angular Calendar with ngx-bootstrap Datepicker**

**In this tutorial, we'll see how to create an Angular 9 calendar UI with the datepicker component available from ngx-bootstrap.**

**We'll learn to build a calendar component using bootstrap and ngx-bootstrap datepicker component.**

**Date pickers are commonly used in web apps for choosing dates.**

**Dates can be selected by navigating between various dates using UI.**

**The ngx-bootstrap library provides a datepicker component with many configuration options that you can use to customize the calendar view in your Angular 9 web app.**

**Throughout this tutorial, we will learn with steps to build a calendar component.**

**Let's start bu initializing a new Angular 9 app for our calendar example using Angular CLI.**

**We need to have the following prerequisites:**

* **Node.js and NPM installed on your development machine,**
* **Angular CLI v9 installed on your machine.**

**Generating your Angular Project**

**Open a new command-line interface and run the following command:**

**ng new angular-calendar-example**

**Navigate to your project's folder using:**

**cd angular-calendar-example**

**Adding Bootstrap to Angular**

**Next, we need to integrate bootstrap and ngx-bootstrap with your Angular 9 project.**

**First, run the following command to install bootstrap 4 from npm:**

**npm install bootstrap**

**Next, install the ngx-bootstrap package:**

**$ npm install ngx-bootstrap --save**

**Importing and Configuring the Datepicker Component**

**Open the angular.json file and add the following styling:**

**"styles": [**

**"node\_modules/bootstrap/dist/css/bootstrap.min.css",**

**"node\_modules/ngx-bootstrap/datepicker/bs-datepicker.css",**

**"src/styles.css"**

**]**

**Next, we need to import the date picker component. Open src/app/app.module.ts file and update it as follows:**

**import { BrowserModule } from '@angular/platform-browser';**

**import { NgModule } from '@angular/core';**

**import { AppComponent } from './app.component';**

**import { BrowserAnimationsModule } from '@angular/platform-browser/animations';**

**import { BsDatepickerModule } from 'ngx-bootstrap/datepicker';**

**@NgModule({**

**declarations: [**

**AppComponent**

**],**

**imports: [**

**BrowserModule,**

**BrowserAnimationsModule,**

**BsDatepickerModule.forRoot()**

**],**

**providers: [],**

**bootstrap: [AppComponent]**

**})**

**export class AppModule { }**

**Adding a Calendar UI in your Angular 9 Template**

**Next, let's add a calendar component. Open the src/app/app.component.html file and add the following HTML code:**

**<div class="container">**

**<div class="row">**

**<div class="col-xs-12">**

**<input type="text"**

**placeholder="Choose date"**

**class="form-control"**

**bsDatepicker>**

**</div>**

**<div class="col-xs-12">**

**<input**

**type="text"**

**placeholder="Choose date"**

**class="form-control"**

**bsDatepicker**

**bsDaterangepicker**

**[bsConfig]="{ isAnimated: true }">**

**</div>**

**</div>**

**</div>**

**That's it, save your file and run the following command in your terminal:**

**ng serve**

**Summary**

**In this tutorial, we have a simple example of a calendar with the datepicker component of ngx-bootstrap.**

**Appendix 1: Bootstrap 4 Card Explained**

**In this section, we’ll learn about the new Card component introduced in Bootstrap 4 which replaces many old components in Bootstrap 3, and can be used to build modern style web layouts without re-inventing the well or being a CSS designer with deep CSS knowledge. Now and thanks to Bootstrap 4 you can, as a developer with small CSS knowledge, build modern and professional card-based layouts.**

**Bootstrap cards provide flexible containers for displaying a mix of content (text, images, links, etc.) in a grouped way. They can be created by only adding a bunch of CSS classes to HTML markup and they can be easily styled and customized to present a new and distinguished look.**

**Cards are used to create advanced layouts such as Masonry. You’ll find cards in many big web platforms such as e-commerce websites, analytics dashboards, and blogs etc.**

**For Bootstrap users, Cards are a replacement for the popular components : panels, thumbnails and wells (all these can now be created with cards)**

**Before we start seeing how to create different card types. These are a summary of information about Bootstrap 4 Cards**

* **Cards are used to present a mix of content in an elegant and compact way**
* **Cards are built on top of Flexbox**
* **Cards have no margin by default**
* **Cards can be easily organized in groups, decks or Masonry columns**
* **Cards have a header, footer, and one or more bodies**
* **Cards can be easily aligned and well mixed with the other Bootstrap components etc.**

**In this tutorial you will get introduced to the new card component then you will see different elements of a card and finally you will build a Masonry-like layout using Cards and Card Columns**

**First, start by creating an HTML file andcopy the following**[**starter template from the Bootstrap docs**](https://getbootstrap.com/docs/4.0/getting-started/introduction/)**in the new file:**

**<!doctype html><html lang="en"> <head> <!-- Required meta tags --> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">**

**<!-- Bootstrap CSS --> <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">**

**<title>Bootstrap 4 Card Example</title> </head> <body> <div class="container"> <!-- Our Simple Card Goes Here --> </div>**

**<!-- Optional JavaScript --> <!-- jQuery first, then Popper.js, then Bootstrap JS --> <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script> <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script> <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script> </body></html>**

**Now let’s start by creating a basic card. Add a <div> with the class .card to create the outer container of your card component.**

**Next inside the outer container, add a header, card body, and footer.**

**You can add a header by either creating a <div> or <h\*> heading element then add the .card-header class.**

**You can create a card body by adding a .card-body class to a <div>.**

**For the footer, you need to add a <div> with the class .card-footer:**

**<div class="card">**

**<div class="card-header">**

**Card Header**

**</div>**

**<div class="card-body">**

**Card Body**

**</div>**

**<div class="card-footer">**

**Card Footer**

**</div>**

**</div>**

**Please note that you can add one or many card bodies to your card and you can also wrap the content inside the body in other tags such as <p>:**

**<div class="card-body">**

**<p>**

**You can also add a second body </p>**

**</div>**

**Use a card body when you want to create a padded section within a card.**

* **Using .card-img-top and card-img-bottom you can create**[**image caps**](https://getbootstrap.com/docs/4.0/components/card/#image-caps)**which are images that exist on the top and bottom of the card body/bodies**

**<img class="card-img-top" src="https://source.unsplash.com/daily" alt="Card image top"><div class="card-body"></div>**

**<img class="card-img-bottom" src="https://source.unsplash.com/daily" alt="Card image bottom">**

**You can see that the card component is taking the whole width of its container division but you can control the width with different means:**

* **using the width and max-width CSS properties**

**<div class="card" style="width:20rem;">...</div>**

* **using the Bootstrap 4 Grid**

**You can also set the height but in most cases the height needs to fit the vertical alignment of the card content so you’ll rarely need to set it.**

**For adding content inside the card body you also have a bunch of classes (.card-title, .card-subtitle, .card-text) that make it easy to style different content types such as the card title, subtitle and text.**

**You can use standard HTML tags to style text within .card-text class**

**<div class="card-body">**

**<h3 class="card-title">Card title</h3> <h4 class="card-subtitle">Card subtitle</h4> <p class="card-text">This is a card body 1</p>**

**</div>**

**Want to use the cap image as an overlay? you can simply switch the .card-body class with the .card-img-overlay class**

**<div class="card">**

**<img class="card-img-top" src="https://source.unsplash.com/daily?nature" alt="Card image top"> <div class="card-img-overlay"> <h3 class="card-title">Card 2 title</h3> <h4 class="card-subtitle">Card 2 subtitle</h4> <p class="card-text">This is a card 2 body </p> </div> </div>**

**You can add navigation tabs and pills to the header of your card using nav, nav-tabs|nav-pills, card-header-tabs| card-header-pill classes with <ul> tags**

**<div class="card-header">**

**<ul class="nav nav-tabs card-header-tabs"> <li class="nav-item"> <a class="nav-link active" href="#">Tab 1</a> </li> <li class="nav-item"> <a class="nav-link" href="#">Tab 2</a> </li> </ul> </div>**

**You can add links to card content using the .card-link class with an <a> element**

**<div class="card">**

**<img class="card-img-top" src="https://source.unsplash.com/daily?rock" alt="Card image top"> <div class="card-img-overlay white"> <h3 class="card-title">Card title</h3> <h4 class="card-subtitle">Card subtitle</h4> <p class="card-text">You can add links to card body</p> <a href="#" class="card-link">Link 1</a> <a href="#" class="card-link">Link 2</a> </div> </div>**

**You can use list as body content. For example this is how you can create two stylish lists without any extra custom CSS**

**<div class="card text-primary border-primary"> <ul class="list-group"> <li class="list-group-item">Item 1</li> <li class="list-group-item">Item 2</li> <li class="list-group-item">Item 3</li> </ul>**

**</div> <br> <div class="card text-primary border-primary"> <div class="card-body"> <h2 class="card-title"> My List</h2> <ul class="list-group list-group-flush"> <li class="list-group-item">Item 1</li> <li class="list-group-item">Item 2</li> <li class="list-group-item">Item 3</li> </ul> </div>**

**How to Create Complex Card Based Layouts?**

**Singe cards are great way to display a mix of content but usually you need to display a collection of cards to build complex layouts such as Masonry like layouts.**

**Bootstrap Card Groups**

**You can create a single unit of multiple cards using Cards. The columns of the unit will take the same width and height (this is achieved using Flexbox).**

**This is an example of a group of 3 cards**

**Bootstrap Card Decks**

**Card Decks are similar to Card Groups except that individual cards inside a deck are not attached**

**Bootstrap Card Columns**

**Cards Columns provides a grouping that organizes cards into a masonry like (Pinterest style) layout. You don’t need to provide extra CSS or JavaScript/jQuery code just wrap your cards inside a <div> with .card-columns and Bootstrap 4 will present you with a Masonry layout. But keep in mind this is just CSS not JavaScript code involved so the masonry layout is simple, cards are ordered from top-left, bottom-right and there is no easy way to chage this behavior unless you are using to a Mosonry plugin.**

**See the following pen for an example**

**Summary**

**In this section, we’ve introduced the Card component.**

**Thanks to Bootstrap 4 you will no more be intimidated when you want to create modern style and card-based layouts even if you are just starting with HTML and CSS or you are a developer with no extensive knowledge of CSS.**

**Appendix 2: Bootstrap 4 Grid/Flexbox Explained**

**In this section, we’ll learn about the Grid layout based on Flexbox in Bootstrap 4.**

**This section is an introduction of the Bootstrap Grid system with taking Flexbox into consideration as Bootstrap 4 is now using Flexbox as the default display system for the grid layout that brings many new and powerful features to how you can build your website layouts using the Bootstrap 4 grid.**

**Flexbox is a CSS 3 display system that aims to make it easy and straightforward to create layouts for dynamic or unknown screen sizes by allowing the container to have more control over the size of elements and then adapt to different viewports.**

**Without further introductions let’s create and decipher a simple page layout using Bootstrap 4 grid system.**

**First, you need to create an HTML page and link the Bootstrap 4 assets. You can simply use this template from the official Bootstrap 4**[**docs**](https://getbootstrap.com/docs/4.0/getting-started/introduction/)

**<!doctype html><html lang="en"> <head> <!-- Required meta tags --> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">**

**<!-- Bootstrap CSS --> <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">**

**<title>BS 4 Flexbox Examples</title> </head> <body> Add Content Here**

**<!-- Optional JavaScript --> <!-- jQuery first, then Popper.js, then Bootstrap JS --> <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script> <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script> <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script> </body></html>**

**In order to create a grid, you’ll have to use some predefined BootStrap classes (.container, .row, col-\*-\*)**

**The grid should have a container, rows and columns.**

**The container is simply a <div> with the .container or .container-fluid classes. Why do you need a container and what is the difference between those two classes?**

**The container simply provides a width or a maximal width for all the other elements of the grid. For the difference .container allows you to create fixed-width container and .container-fluid allows you to create a full *100%* container.**

**Rows provide home for columns. The number of rows can theoretically be unlimited.**

**Columns are what make the cells for the grid system. Each row can only take up to 12 columns (each column spans *1/12* of the available width). Practically a column spans more than one unit but there is one important rule ALL COLUMNS SHOULD ADD UP TO 12**

**To create a column you simply use a <div> with the the class(es) col-\*-\***

**The first star for the screen breakpoint (xs, sm, md, lg, xl) and the second star for column size (*1 .. 12*).**

**<div class="row"> <div class="col-sm-1 first-column"> <p>.col-sm-1</p> </div> ... <div class="col-sm-1 second-column"> <p>.col-sm-1</p> </div> </div>**

**See this**[**code pen**](https://codepen.io/techiediaries/pen/BYJOrL)

**Now let’s create a second row with two columns**

**<div class="row"> <div class="col-sm-6 first-column"> <p>.col-sm-6</p> </div> <div class="col-sm-6 second-column"> <p>.col-sm-6</p> </div> </div>**

**Note that .col-sm-6 + .col-sm-6 === .col-sm-12**

**Also note that you don’t need to explicitly add upper breakpoint classes i.e col-md-6 to col-xl-6 because .col-sm-6 means the size of the column should be 6 units from the small breakpoint and up**

**Next let’s add a third row with three columns**

**The first column spans two units, the second spans three units and the third spans the remaining 7 units. All colmuns sized add up to 12**

**.col-sm-2 + .col-sm-3 + .col-sm-7 === .col-sm-12**

**If you have previously used Bootstrap before version 4 you should have be familiar with all these concepts so what the new features Bootstrap 4 bring?**

**If you need to create a layout with automatic width columns you simply need to use .col-\* or just .col without specifying the size and Bootstrap will take care of equally dividing the available width among those columns**

**<div class="row"> <div class="col first-column"></div> <div class="col second-column"></div> <div class="col first-column"></div> </div>**

**Now if you add a bunch of text to each column in order to chaine their heights you are going to get something like**

**As can be seen, unlike Bootstrap 3 all columns have the same height disregarding their contents which gives a better sense of a grid system.**

**Bootstrap 4 Flex Utilities**

**Bootstrap 4 by default uses Flexbox but it also provides a set of class utilities that allow you to work with Flexbox without resorting to custom CSS.**

**If you want to make a div element a flex container you can simply add the .d-flex class to a <div>**

**<div class="d-flex">This is now a flex container</div>**

**Or you can instead use d-inline-flex to create a inline flex container.**

**These two classes have also responsive versions i.e you can use d-d-\*-flex and d-\*-inline-flex where the start can be one of these breakpoints values xs, sm, md, lg and xl.**

**One important aspect of Flexbox is the ability to easily set the direction of items inside a flex container.**

**By default the direction is horizontal from left to right but can be set to be horizontal but from right to left with .flex-row-reverse or set it to be from left to right again with .flex-row (the default).**

**You can also use the .flex-column class to set a vertical direction from top to bottom, or the .flex-column-reverse class to change the vertical direction from bottom to top.**

**See the other available classes to control flex properties from the**[**docs**](https://getbootstrap.com/docs/4.0/utilities/flex/)**.**

**Bootstrap 4 Display Utilities**

**Bootstrap 4 provides a set of utilities that make it easy to set the display property of elements responsively i.e you can change the display property for specific breakpoints or screen sizes. See the docs for more**[**details**](https://getbootstrap.com/docs/4.0/utilities/display/)

***Quickly and responsively toggle the display value of components and more with our display utilities. Includes support for some of the more common values, as well as some extras for controlling display when printing.***

**Simply use .d-{type} for the xs breakpoint or .d-{breakpoint}-{type} for sm, md, lg, and xl where type can be one of these display types**

* **none**
* **inline**
* **inline-block**
* **block**
* **table**
* **table-cell**
* **table-row**
* **flex**
* **inline-flex**

**Summary**

**In this section, we’ve seen the grid layout in Bootstrap 4. Bootstrap has always been a powerful CSS framework for developers building their websites without a CSS designer in the team and now with the new features such as the default support for Flexbox you have great tools to build responsive layouts without being a CSS expert.**