**CREATING YOUR FIRST ANGULAR APPLICATION**

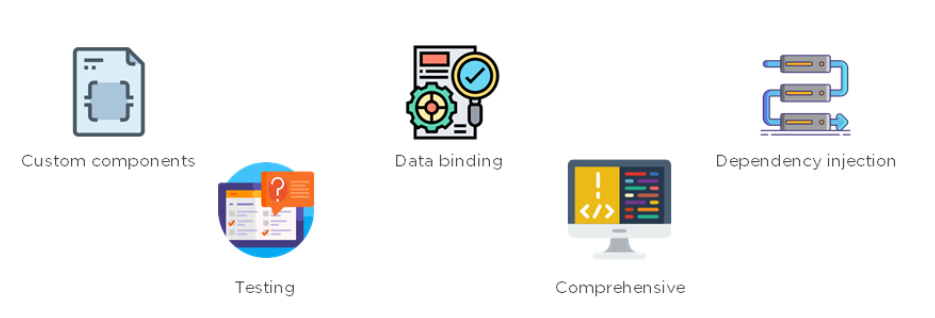
## **What is Angular?**

Angular is a [JavaScript](https://www.simplilearn.com/tutorials/javascript-tutorial/introduction-to-javascript) framework for building single-page client applications using [HTML](https://www.simplilearn.com/tutorials/html-tutorial/what-is-html) and TypeScript. It is primarily written in TypeScript and implements core functionality as a set of TypeScript libraries that you import into your apps.



Angular is maintained by Google and has clear advantages while also providing a standard structure for developers on a team to work with. It enables users to develop large applications in a maintainable manner.

Here are some of the salient features of Angular:



### **Custom Components**

Angular enables users to build their [components](https://www.simplilearn.com/tutorials/angular-tutorial/angular-components) that can pack functionality along with rendering logic into reusable pieces.

### [**Data Binding**](https://www.simplilearn.com/tutorials/angular-tutorial/angular-data-binding)

Angular enables users to effortlessly move data from JavaScript code to the view, and react to user events without having to write any code manually.

### [**Dependency Injection**](https://www.simplilearn.com/tutorials/angular-tutorial/angular-dependency-injection)

Angular enables users to write modular services and inject them wherever they are needed. This improves the testability and reusability of the same services.

### **Testing**

Tests are first-class tools, and Angular has been built from the ground up with testability in mind. You will have the ability to test every part of your application—which is highly recommended.

### **Comprehensive**

Angular is a full-fledged JavaScript framework and provides out-of-the-box solutions for server communication, routing within your application, and more.

### **Browser Compatibility**

Angular works cross-platform and compatible with multiple browsers. An Angular application can typically run on all browsers (Eg: Chrome, Firefox) and operating systems, such as Windows, macOS, and Linux.

Now that you’re familiar with Angular, in the next segment, you’ll learn about the prerequisites for creating an Angular HelloWorld Application.

## **Angular Prerequisites**

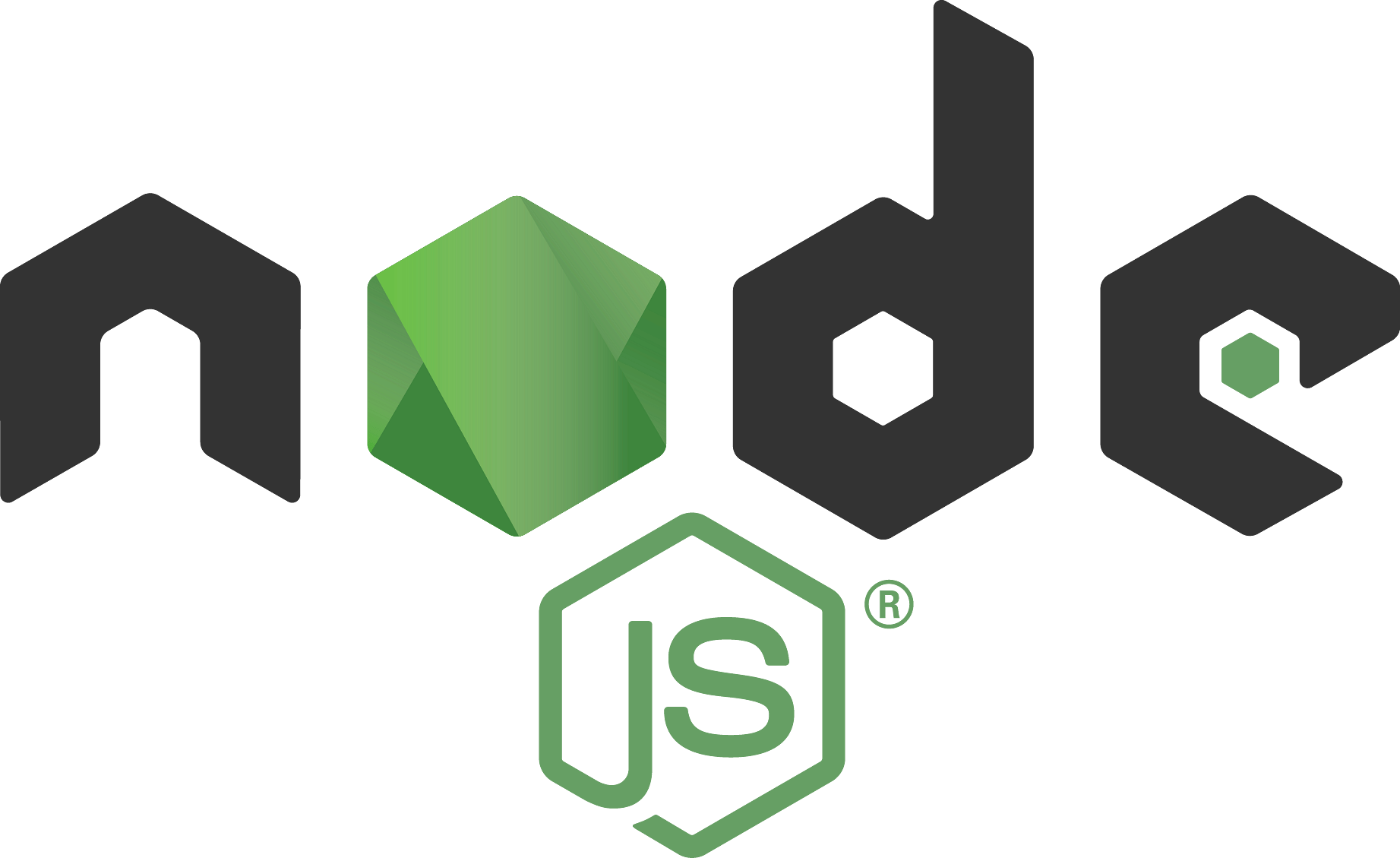
There are three main prerequisites.

### **NodeJS**

Angular uses Node.js for a large part of its build environment. As a result, to get started with Angular, you will need to have Node.js installed on your system. You can head to the NodeJS official website to download the software. Install the latest version and confirm them on you command prompt by running the following commands:

**node--version**

**npm--v**



### **Angular CLI**

The Angular team has created a command-line interface (CLI) tool to make it easier to bootstrap and develop your Angular applications. As it significantly helps to make the process of development easier, we highly recommend using it for your initial [angular projects](https://www.simplilearn.com/tutorials/angular-tutorial/angular-project) at the least.

To install the CLI, in the command prompt, type the following commands

Installation:

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

Confirmation -

**ng--version**



### **Text Editor**

You need a text editor to write and run your code. The most popularly used integrated development environment (IDE) is Visual Studio Code (VS Code). It is a powerful source code editor that is available on Windows, macOS, and Linux platforms.



Now, Let’s create our first Angular HelloWorld Application.

## **Creating the Angular Application**

### **Step 1**

Create a folder for your application in the desired location on your system and open it on VSCode. Open a new terminal and type in the following command to create your app folder.

A screenshot of a computer

Description automatically generated

When the command is run, Angular creates a skeleton application under the folder. It also includes a bunch of files and other important necessities for the application.

### **Step 2**

To run the application, change the directory to the folder created, and use the ng command.

A screenshot of a computer

Description automatically generated

Once run, open your browser and navigate to localhost:4200. If another application is running on that address, you can simply run the command.  
**ng serve--port**

It will generate a port for you to navigate to. Typically, the browser looks something like this

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface

Description automatically generated

You can leave the ng serve command running in the terminal, and continue making changes. If you have the application opened in your browser, it will automatically refresh each time you save your changes. This makes the development quick and iterative.

## **Basics of an Angular App**

At its core, any Angular application is still a Single-Page Application (SPA), and thus its loading is triggered by a main request to the server. When we open any URL in our browser, the very first request is made to our server. This initial request is satisfied by an HTML page, which then loads the necessary JavaScript files to load both Angular as well as our application code and templates.

## **Root HTML - index.html**

Graphical user interface, text, application

Description automatically generated

The root component looks very pristine and neat, with barely any references or dependencies. The only main thing in this file is the **<app-root>** element. This is the marker for loading the application. All the application code, styles, and inline templates are dynamically injected into the index.html file at run time by the **ng serve** command.

## **The Entry Point - main.ts**

The second important part of our bootstrapping piece is the main.ts file. The index.html file is responsible for deciding which files are to be loaded. The main.ts file, on the other hand, identifies which Angular module is to be loaded when the application starts.

Graphical user interface, text, application

Description automatically generated

Most of the code in the main.ts file is generic, and you will rarely have to touch or change this entry point file. Its main aim is to point the Angular framework at the core module of your application and let it trigger the rest of your application source code from that point.

## **Main Module - app.module.ts**

This is where your application-specific source code starts from. The application module file can be thought of as the core configuration of your application, from loading all the relevant and necessary dependencies, declaring which components will be used within your application, to marking which is the main entry point component of your application.

## Graphical user interface, text, application Description automatically generated**Root Component - AppComponent**

The app.component.ts is the actual Angular code that drives the functionality of the application.

Graphical user interface, text, application

Description automatically generatedA component in Angular is nothing but a TypeScript class, decorated with some attributes and metadata. The class encapsulates all the data and functionality of the component, while the decorator, @Component, specifies how it translates into the HTML.