

# ANGULAR FRAMEWORK

Angular is a framework for building client applications in HTML and either JavaScript or a language like TypeScript that compiles to JavaScript. The framework consists of several libraries, some of them core and some optional.

Angular is a JavaScript Framework which allows you to create reactive **Single Page Applications (SPAs)**.

Angular is a TypeScript-based open-source front-end web application platform led by the Angular Team at Google and by a community of individuals and corporations. Angular is a complete rewrite from the same team that built AngularJS. But Angular is completely different from AngularJS.

## DIFFERENCES BETWEEN ANGULAR AND ANGULARJS

- The architecture of an Angular application is different from AngularJS. The main building blocks for Angular are modules, components, templates, metadata, data binding, directives, services and dependency injection.
- Angular was a complete rewrite of AngularJS.
- Angular does not have a concept of “scope” or controllers instead, it uses a hierarchy of components as its main architectural concept.
- Angular has a simpler expression syntax, focusing on “[ ]” for property binding, and “( )” for event binding
- **Mobile development** – Desktop development is much easier when mobile performance issues are handled first. Thus, Angular first handles mobile development.
- **Modularity** – Angular follows modularity. Similar functionalities are kept together in same modules. This gives Angular a lighter & faster core.

# ANGULAR FEATURES

## CROSS PLATFORM

- **Progressive web apps**  
It uses modern web platform capabilities to deliver an app-like experience. It gives high performance, offline, and zero-step installation. So, working with Angular is pretty much easy.
- **Native**  
You can build native mobile apps with strategies using Ionic Framework, NativeScript, and React Native.
- **Desktop**  
Create desktop-installed apps across Mac, Windows, and Linux using the same Angular methods you’ve learned for the web plus.

## SPEED AND PERFORMANCE

- **Code generation**  
Angular turns your templates into code that’s highly optimized for JavaScript virtual machines, giving

you all the benefits of hand-written code with the productivity of a framework.

- **Universal**  
You can use any technology with Angular for serving the application like node.js, .NET, PHP and other servers.
- **Code splitting**  
Angular apps load quickly with the new Component Router, which delivers automatic code-splitting, so users only load code required to render the view they request.

## PRODUCTIVITY

- **Templates**  
Quickly create UI views with simple and powerful template syntax.
- **Angular CLI**  
Command line tools: You can easily and quickly start building components, adding components, testing them, and then, instantly deploy them using Angular CLI.
- **IDEs**  
Get intelligent code completion, instant errors, and other feedback in popular editors and IDEs like Microsoft's VS Code.

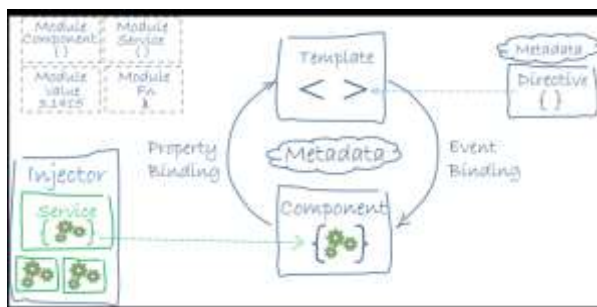
## FULL DEVELOPMENT STORY

- **Testing**  
With Karma for unit tests, you can identify your mistake on the fly and Protractor makes your scenario tests run faster and in a stable manner.

# BUILDING BLOCKS OF ANGULAR APPLICATIONS

You write Angular applications by composing HTML templates with Angularized markup, writing component classes to manage those templates, adding application logic in services, and boxing components and services in modules.

Then you launch the app by bootstrapping the root module. Angular takes over, presenting your application content in a browser and responding to user interactions according to the instructions you've provided.





## MODULES

Angular apps are modular and Angular has its own modularity system called `NgModules`. Every Angular app has at least one `NgModule` class, the root module, conventionally named `AppModule`.

## ANGULAR LIBRARIES

Angular ships as a collection of JavaScript modules. You can think of them as library modules. Each Angular library name begins with the `@angular` prefix. You install them with the **npm** package manager and import parts of them with JavaScript `import` statements.

- `Component`                      `@angular/core`
- `BrowserModule`                `@angular/platform-browser`
- `FormsModule`                  `@angular/forms`

## COMPONENT

A component controls a patch of screen called a view and used to manage templates. You define a component's application logic—what it does to support the view—inside a class. The class interacts with the view through an API of properties and methods.

## TEMPLATE

A template is a form of HTML that tells Angular how to render the component

## METADATA

Metadata tells Angular how to process a class. We can attach metadata to a class by using a **decorator**. `@Component`, `@Injectable`, `@Input`, and `@Output` are a few of the more popular decorators

# ENVIRONMENT SETUP

## NODE.JS AND NPM

To get started with Angular, you'll need to have Node.js installed. There are a couple of different ways you can install Node.js, so please refer to [the Node.js website](https://nodejs.org/download/)<sup>1</sup> for detailed information.

Make sure you install Node **6.9.0** or higher.

The **Node Package Manager** (**npm** for short) is installed as a part of Node.js. To check if `npm` is available as a part of our development environment, we can open a terminal window and type:

```
> npm -v
```

If a version number is not printed out and you receive an error, make sure to download a Node.js installer that includes `npm`.

## TYPESCRIPT

---

<sup>1</sup> <https://nodejs.org/download/>

Once you have Node.js setup, the next step is to install TypeScript. Make sure you install at least version 2.1 or greater. To install it, run the following `npm` command:

```
> npm install -g typescript
```

## BROWSER

We highly recommend using the Google Chrome Web Browser to develop Angular apps.

## ANGULAR CLI

CLI – Command Line Interface – This is simply a tool set, which is used for creating, managing and building angular applications quickly. It quickly creates new angular projects, and then you can use some commands to build that project for production and so on.

CLI is also a great tool for big and very big Angular projects. It allows you to focus on Angular code instead of workflow tasks.

Angular is a very futuristic framework and therefore it uses features like TypeScript – which is a superset of JavaScript – which needs to be compiled before it runs in the browser.

Angular provides a utility to allow users to create and manage projects from the command line. It automates tasks like creating projects, adding new controllers, etc. It's generally a good idea to use Angular CLI as it will help create and maintain common patterns across our application.

To install Angular CLI, just run the following command

```
> npm install -g @angular/cli
```

Once it's installed you'll be able to run it from the command line using the `ng` command. When you do, you'll see a lot of output, but if you scroll back, you should be able to see the following:

```
> ng --version
```

If everything installed correctly, you should see the current version output to your terminal. Congratulations!



```
C:\Users\VISHAL>ng --version

Angular CLI
Angular CLI: 1.6.5
Node: 8.0.0
OS: win32 x64
Angular: ...

C:\Users\VISHAL>
```