Agenda

Angular Interview Q&A

- Angular History
- Single Page App (SPA)
- Angular CLI
- Angular Configuration
- Angular Building Blocks
- Components and Decorators

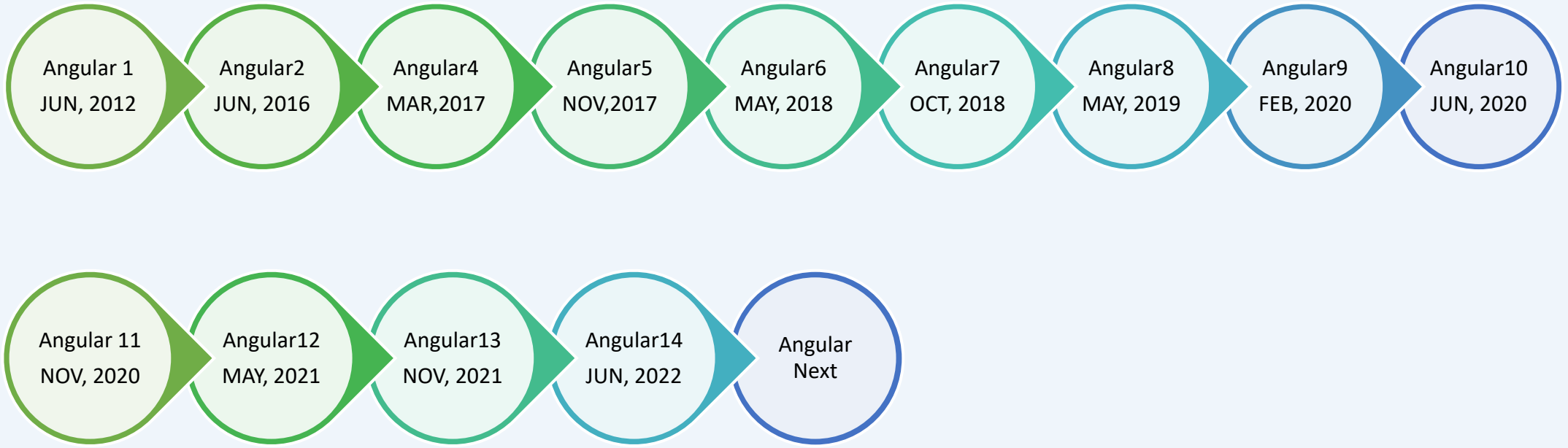
Q&A:

Angular History

Q1. What is Angular?

- An open-source framework for building single page applications (SPA) using web technologies like html, css and js.
- Angular is written in TypeScript and follow TypeScript syntax to write code.
- Empowers developers to build applications for browsers, mobiles, or desktop

Q2. How many versions of Angular have been released?



*Every Six Months Major Release Cycle

ScholarHat

Q3. What is difference between Angular2 and Angular1?

Angular2+	Angular 1.x
• Based on components	• Based on controller and scope
• Improved DI	• Supports DI
• Mobile First	• Not built with mobile first
• Supports ES5/6, TS or Dart	• Supports ES5/6 or Dart
• Angular CLI	• Doesn’t have CLI
• Class is only way to define services in Angular2	• factory, service, provider, value and constant are used for services
• Runs on client-side & server-side	• Run on only client-side
• bootstrapModule() function is used to initialize	• ng-app and angular.bootstrap() function are used to initialize

Difference between Angular2 and Angular1 Contd..

Angular2+

- Supports Pipe
- Supports camelCase & PascalCase syntaxes like ngModel, ngForm and NgForm
- Use Html DOM elements properties and events
- Use () for events and [] for attributes

Angular 1.x

- Support filters
- Supports spinal-case & camelCase syntaxes like ng-model, ng-class and ngModel
- Use it’s own directives like ng-click, ng-show and ng-src etc.
- Doesn’t support () and [] based syntax

Q4. How to Set Up Angular Dev Environment?

- Download and Install Node.js
- Download and Install IDE - Visual Studio Code or Other TypeScript IDE.
- Download and Install Angular CLI

https://nodejs.org/en/

npm install -g @angular/cli

https://www.npmjs.com/

VS Code

https://code.visualstudio.com/

Q&A:

Single Page App (SPA)

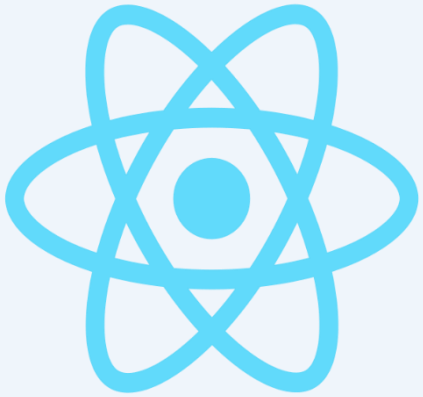
Q1. What is single page application?

- Single HTML page is loaded when the app is loaded
- Heavy emphasis on JS & UX (User Experience)
- Consumes data asynchronously from a RESTful API
- Typically, URL doesn’t change except for hash (#)
- No page reloads
- Universally accessible through a web browser

Q2. What are single page application examples?

Q3. What frameworks can be used to build SPA?

Angular

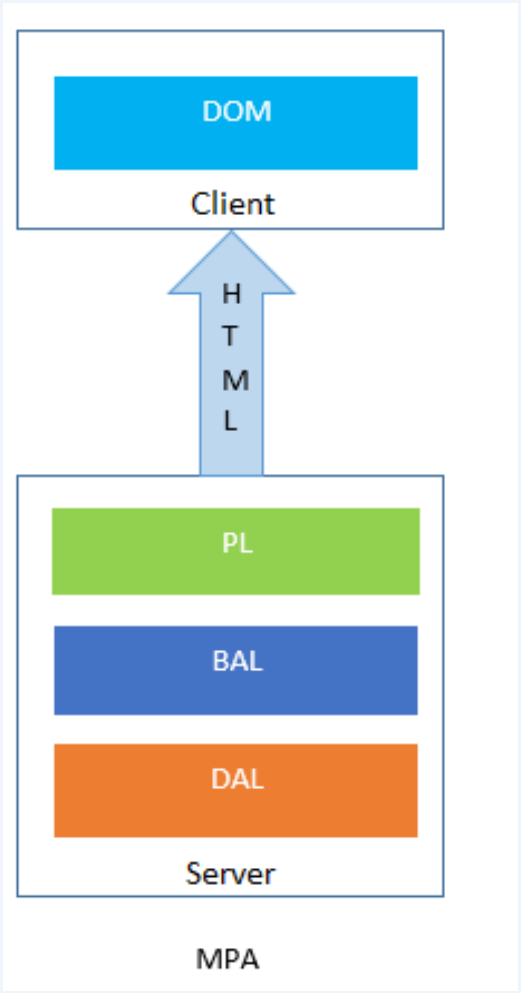


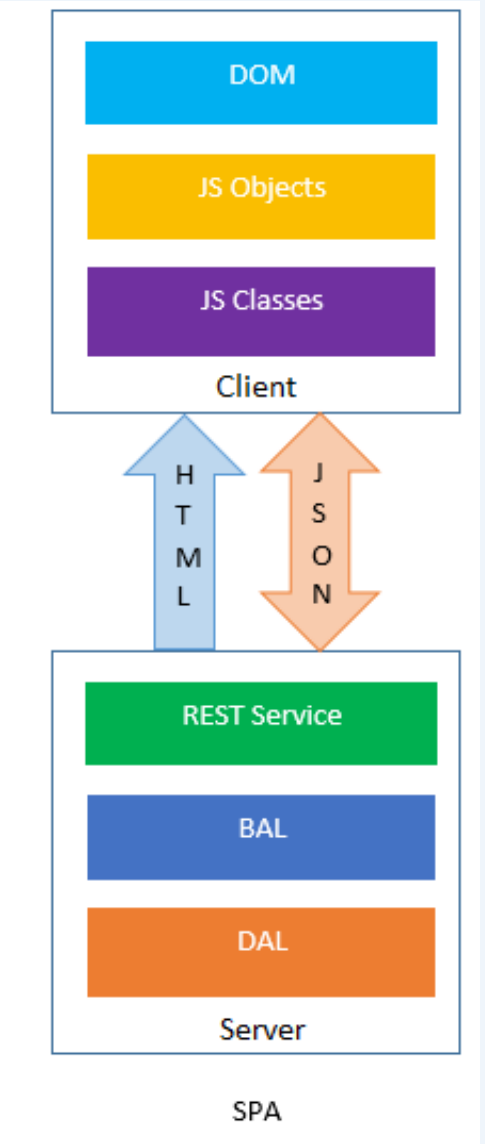
React

Vue

Meteor

Q4. How MPA are different from SPA?





Q&A:

Angular CLI

Q1. What is Angular CLI?

- A powerful to create, build, compile and serve Angular2 App
- Used to generate new components, routes, services and pipes
- Installing Angular CLI
  - *npm install -g @angular/cli*
- Generating and serving Angular app
  - *ng new proj_name --skip-install*
  - *cd proj_name*
  - *npm install*
  - *ng serve*

Q2. What are Angular CLI Options?

Options	Usage
Help	ng --help
Build	ng build --*env*
Build and Run	ng serve
Testing	ng test
End-End Testing	ng e2e

Q3. What are Angular CLI Commands?

Scaffold	Usage	In Short
Module	ng generate module *my-module*	ng g m *my-module*
Component	ng generate component *my-component*	ng g c *my-component*
Directive	ng generate directive *my-directive*	ng g d *my-directive*
Pipe	ng generate pipe *my-pipe*	ng g p *my-pipe*
Service	ng generate service *my-service*	ng g s *my-service*
Guard	ng generate guard *my-guard*	ng g g *my-guard*
Class	ng generate class *my-class*	ng g cl *my-class*
Interface	ng generate interface *my-interface*	ng g i *my-interface*
Enum	ng generate enum *my-enum*	ng g e *my-enum*

Q4. What are Angular CLI Advantages?

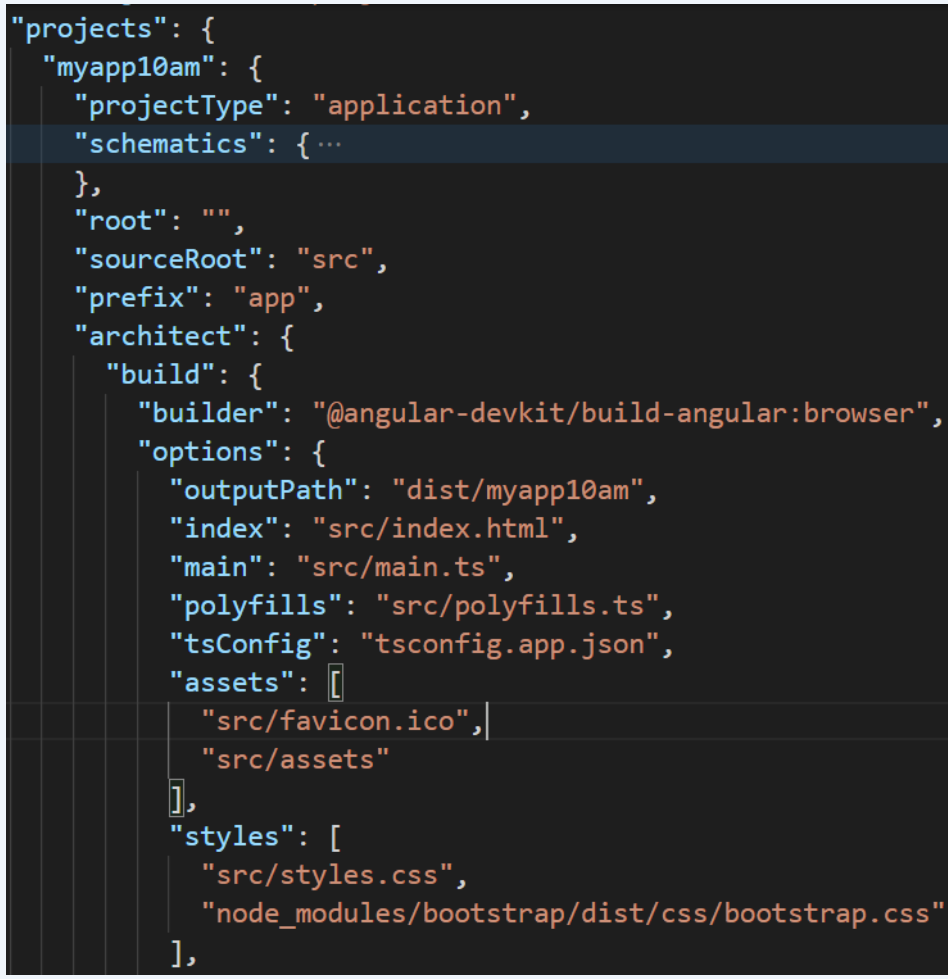
- Follow Angular Best Practices
- Configure Style Guides like css, saas
- Use Dev Server Webpack
- Handle Environments
- Build Management
- Testing using Karma and Protractor

Q&A:

Angular Configuration

Q1. What is the role of angular.json file?

- Contains workspace-wide and project-specific configuration used for build and development used by the Angular CLI.
- Defines the structure of our application and other settings.
- Path values given here are relative to the root workspace folder.



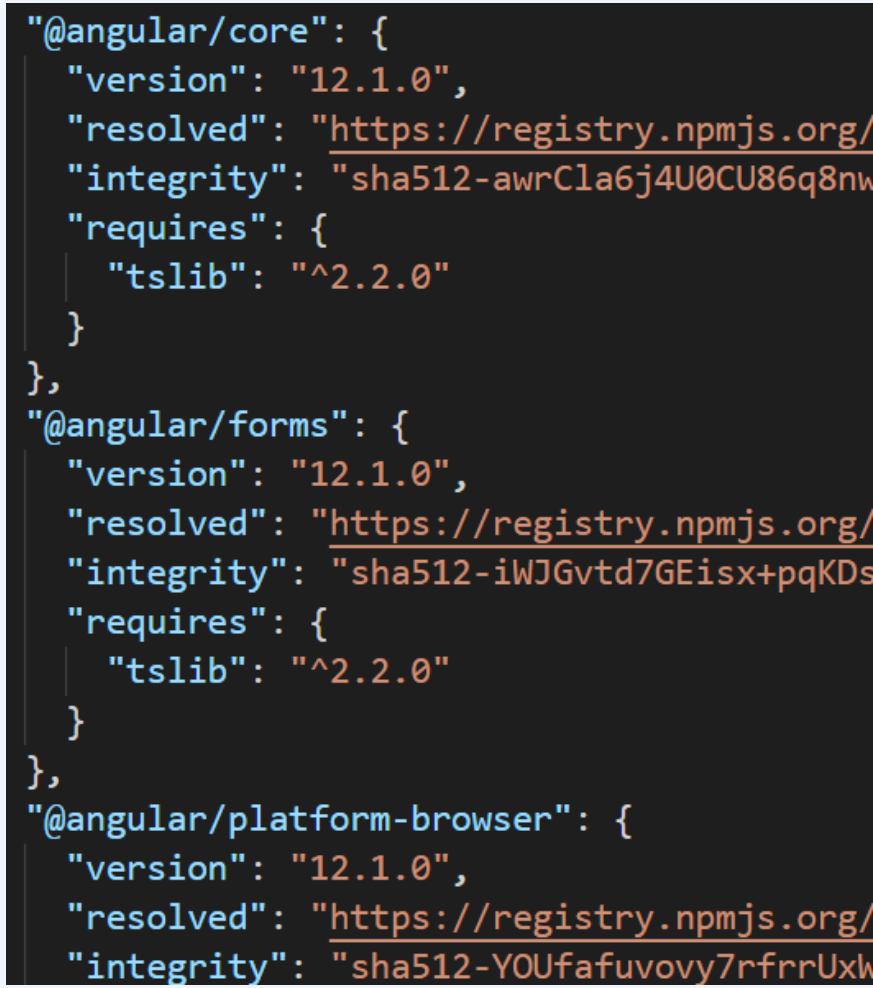
Q2. What is the role of package.json file?

- A node package manager(npm) configuration file.
- Includes details about application's package dependencies for development and production.
- **dependencies** option includes necessary packages for production deployment like @angular/core, @bootstrap etc.
- **devDependencies** option includes necessary packages for development purpose like karma, cli.
- **scripts** option includes commands/scripts for packaging our application itself.

```
"scripts": {
  "ng": "ng",
  "start": "ng serve",
  "build": "ng build",
  "test": "ng test"
},
"dependencies": {
  "@angular/common": "~12.1.0",
  "@angular/core": "~12.1.0",
  "@angular/forms": "~12.1.0",
  "@angular/router": "~12.1.0",
  "bootstrap": "^5.0.2"
},
"devDependencies": {
  "@angular/cli": "~12.1.0",
  "@angular/compiler-cli": "~12.1.0",
  "@types/jasmine": "~3.6.0",
```

Q3. What is the role of package-lock.json file?

- Used to lock dependencies to a specific version number.
- Records the exact version of each installed package which allows you to re-install them.
- While package.json records the minimum version you application needs.



Q4. What is tsconfig.json file?

- A configuration file for typescript compiler.
- Specifies compiler options required to compile the project.

ScholarHat

Q5. What is karma.conf.js file?

- Angular uses karma framework for testing your application code.
- This file includes partial testing configuration settings for karma.

```
module.exports = function (config) {
  config.set({
    basePath: '',
    frameworks: ['jasmine', '@angular-devkit/build-a
    plugins: [
      require('karma-jasmine'),
      require('karma-chrome-launcher'),
      require('karma-jasmine-html-reporter'),
      require('karma-coverage-istanbul-reporter'),
      require('@angular-devkit/build-angular/plugins
    ],
    client: {
      clearContext: false // leave Jasmine Spec Runn
    },
    coverageIstanbulReporter: {
      dir: require('path').join(__dirname, '../cover
```

Q6. What are Polyfills?

- Polyfills are scripts that ensure that your application code (which use the new browser features) do not break in the older browsers.
- Polyfills makes our application compatible for different browsers especially IE old versions IE8, IE9 and IE10.
- Polyfills.ts file is used by angular to setup everything for browser compatibility.

Q7. What is the role of Webpack in Angular?

- The Angular build process uses webpack behind the scenes to transpile TypeScript to JavaScript, transform Sass files to CSS, and many other tasks.
- Webpack is an open-source JavaScript-based build tool.
- Used to bundle various assets like JavaScript, CSS, and Images for usage in a browser.
- Webpack takes the dependencies and generates a dependency graph which allows web developers to use a modular approach for development.
- Used from the command line, or can be configured using a config file using webpack.config.js.

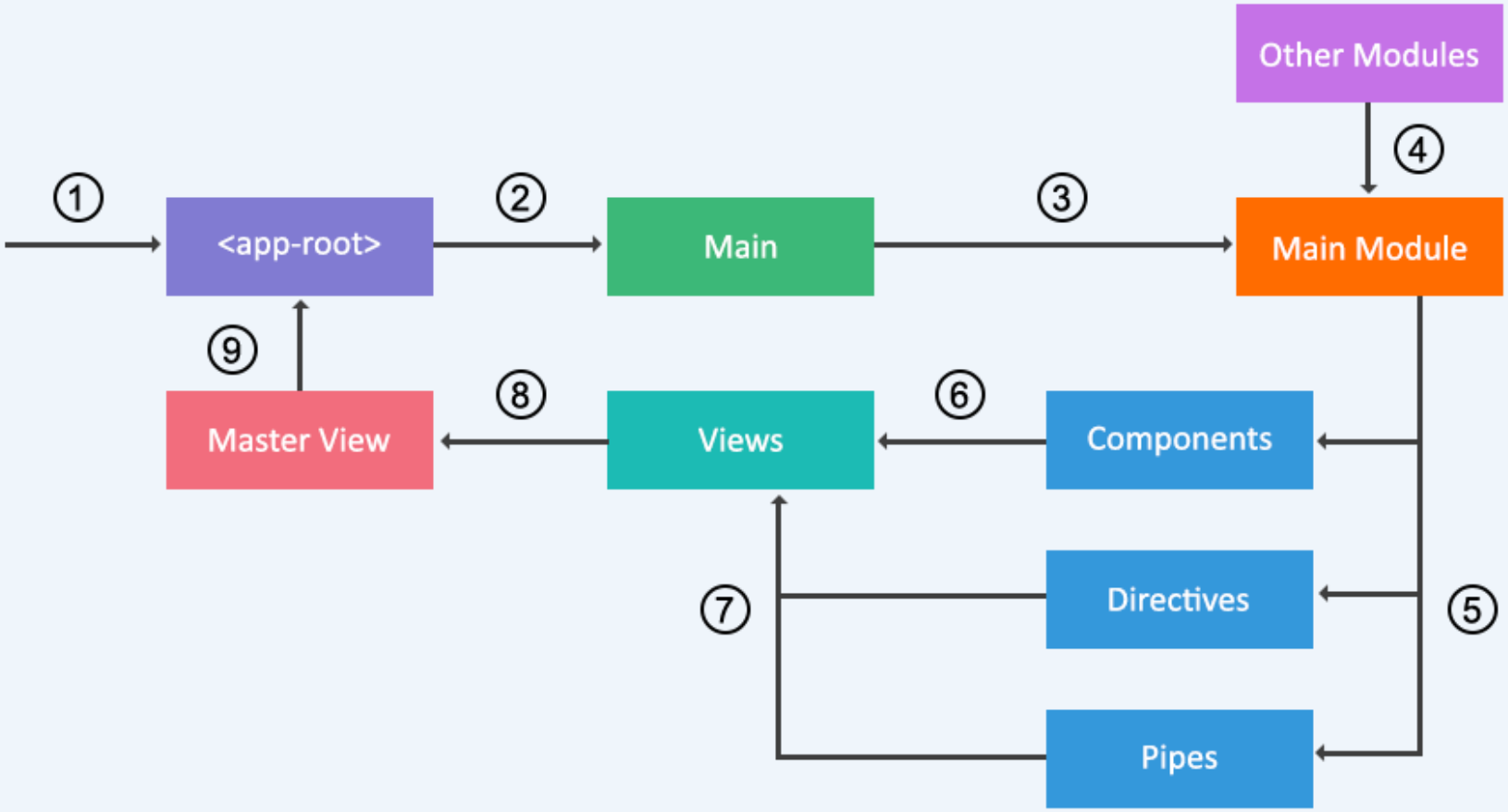
Q&A:

Angular Building Blocks

Q1. What are Angular building blocks?

- Modules
- Components
- Templates
- Metadata
- Data binding
- Directives
- Pipes
- Routing
- Forms
- Services
- Dependency injection

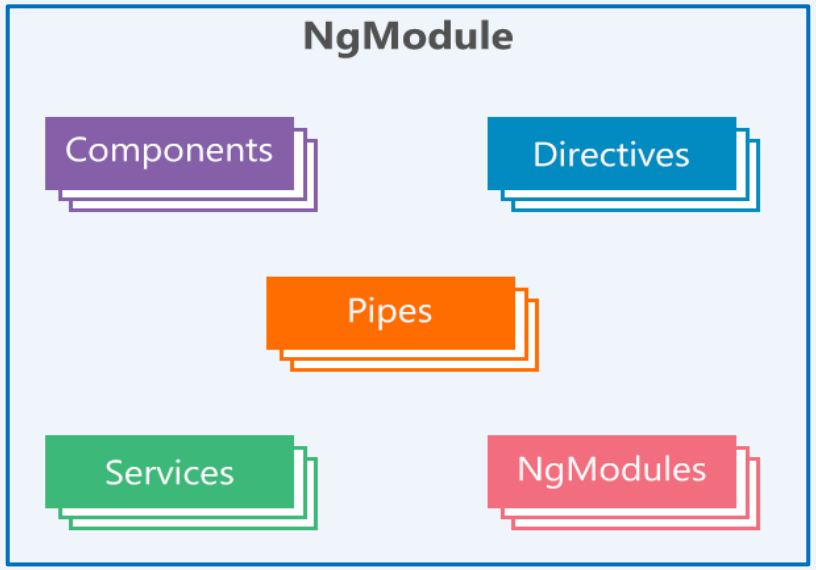
Q2. Do explain Angular initialization process?



ScholarHat

Q3. What is Module?

- A module organize an application into unified blocks of functionality
- An Angular module is a class with an *@NgModule* decorator
- Accepts a single metadata object whose properties describe the module
- Each Angular app must have at least one module, known as root module



Q4. What are module main properties?

- **imports** – Specify other dependent modules whose classes are required by the component templates declared in the module
- **declarations** – Specify the components, directives, and pipes that belong to the module
- **bootstrap** – Specify the main app view i.e root component. Only the **root module** can have this bootstrap property
- **exports** – A subset of declarations that will be visible and usable in the other modules. A root module doesn't have export option.
- **providers** – Specify the services, accessible across the app

Q5. What are Built-In Modules?

- Angular has built-In library modules starting with the @angular as prefix

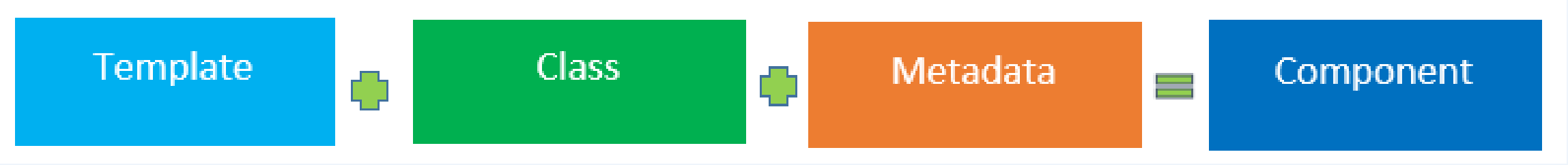
- Built-In library & third part modules can be installed using npm manager
- Built-In modules, components, services, directives etc. can be imported by using built-In library modules

Q&A:

Components and Decorators

Q1. What is a Component?

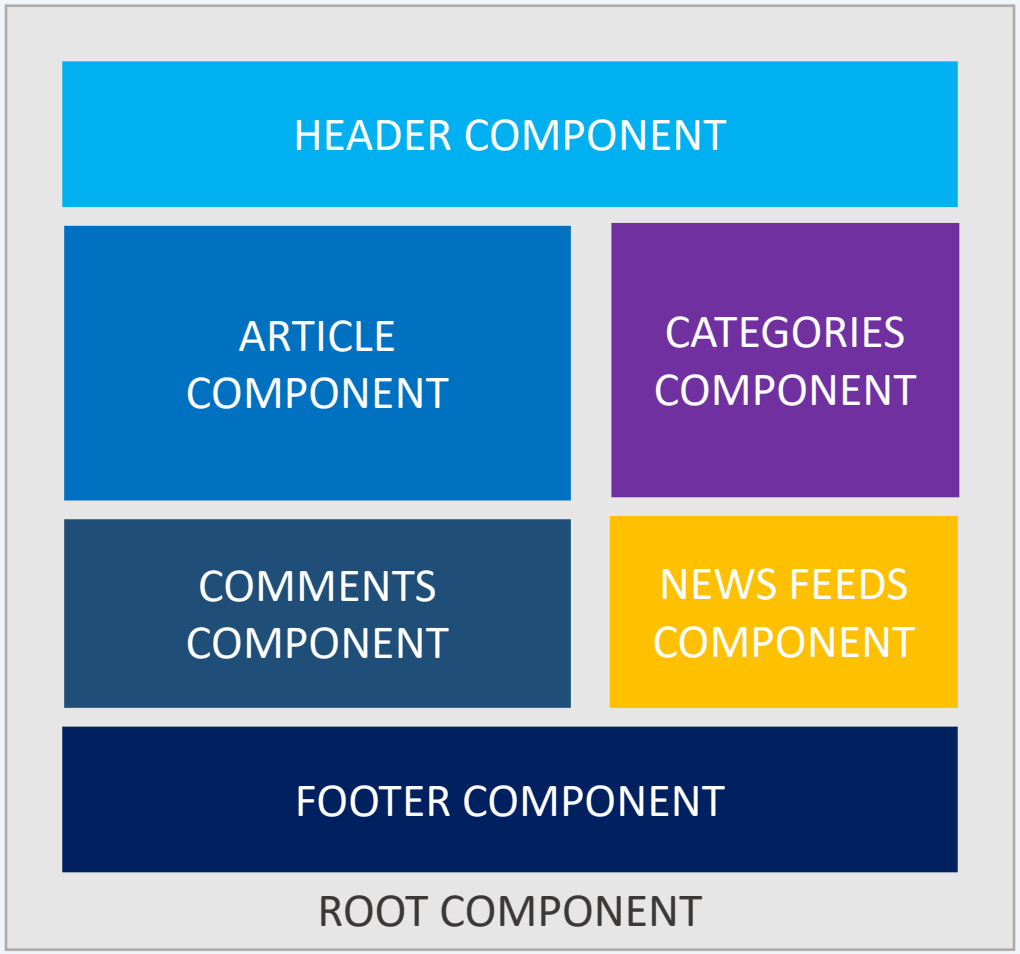
- A type of directives with template, styles and logic for user interaction
- Exported as a custom HTML tag like as:
  - <my-component></my-component>
- Initialized by Angular Dependency Injection engine



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

@Component({
  selector: 'my-component',
  template: `<h3>Interpolation</h3>
             <p>Name : {{name}}</p>
             <p><input type="text" value="{{name}}" /></p>`,
  styles: []
})
export class MyComponent {
  name: string = 'Shailendra';
  constructor() { }
}
```

Q2. How Components can help you to make Webpage?



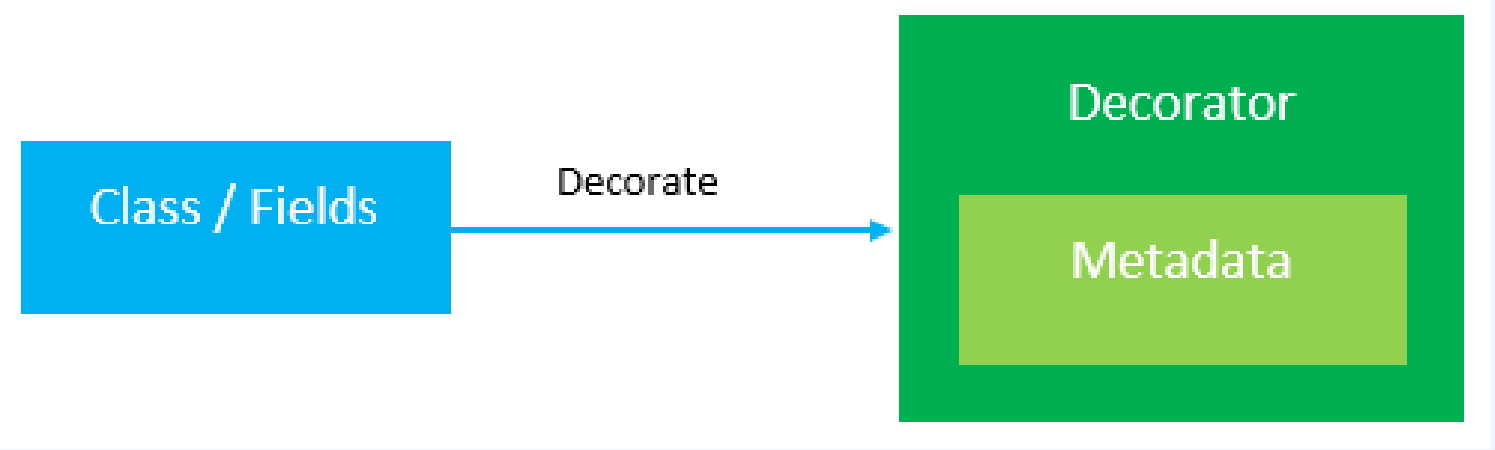
Q3. What is a Template?

- Define the view of a component
- Contains Html markup and angular directives, attributes etc.
- Describe how a component is rendered on the page

```
<h3>Interpolation</h3>
<p>Name : {{name}}</p>
<p>
  <input type="text" value="{{name}}" />
</p>
```

Q4. What is a Decorator?

- A function that adds metadata to a class, class members
- These are prefix with @ symbol
- Angular has built-In decorators like - @Component, @NgModule, @Directive, @Pipe etc.



Q5. What are different types of decorators ?

- Class decorators
  - @NgModule – Used for defining a module
  - @Component – Used for defining a component
  - @Directive – Used for defining a directive
  - @Injectable – Used for injecting dependencies
  - @Pipe – Used for defining a pipe
- Class field decorators
  - @Input – Used for receiving data (input) from parent to child component
  - @Output – Used for passing data (events) from child to parent component

Q6. What is Metadata?

- Tells Angular how to process a class
- Decorators are used to attach metadata to a class