ANGULAR INTERVIEW QUESTIONS

PART - 1

1

What is a Angular Framework ?

Angular is a comprehensive open-source front-end web development framework developed and maintained by Google. It is specifically designed for building dynamic and interactive web applications, particularly single-page applications (SPAs).

Angular uses **TypeScript** as its primary programming language, offering strong typing and improved error detection during development.

As a platform, Angular includes:

- A component-based framework for building scalable web applications.
- A collection of well-integrated libraries that cover a wide variety of features, including routing, forms management, client-server communication, and more.
- A suite of developer tools to help you develop, build, test, and update your code.
- With Angular, you're taking advantage of a platform that can scale from single-developer projects to enterprise-level applications.

2

Explain Pros and Cons of Angular ?

Pros

- Two-Way Data Binding
- Modularity
- Powerful Templating
- Dependency Injection
- Rich Ecosystem
- Strong Community and Documentation
- Cross-Platform Development

Cons

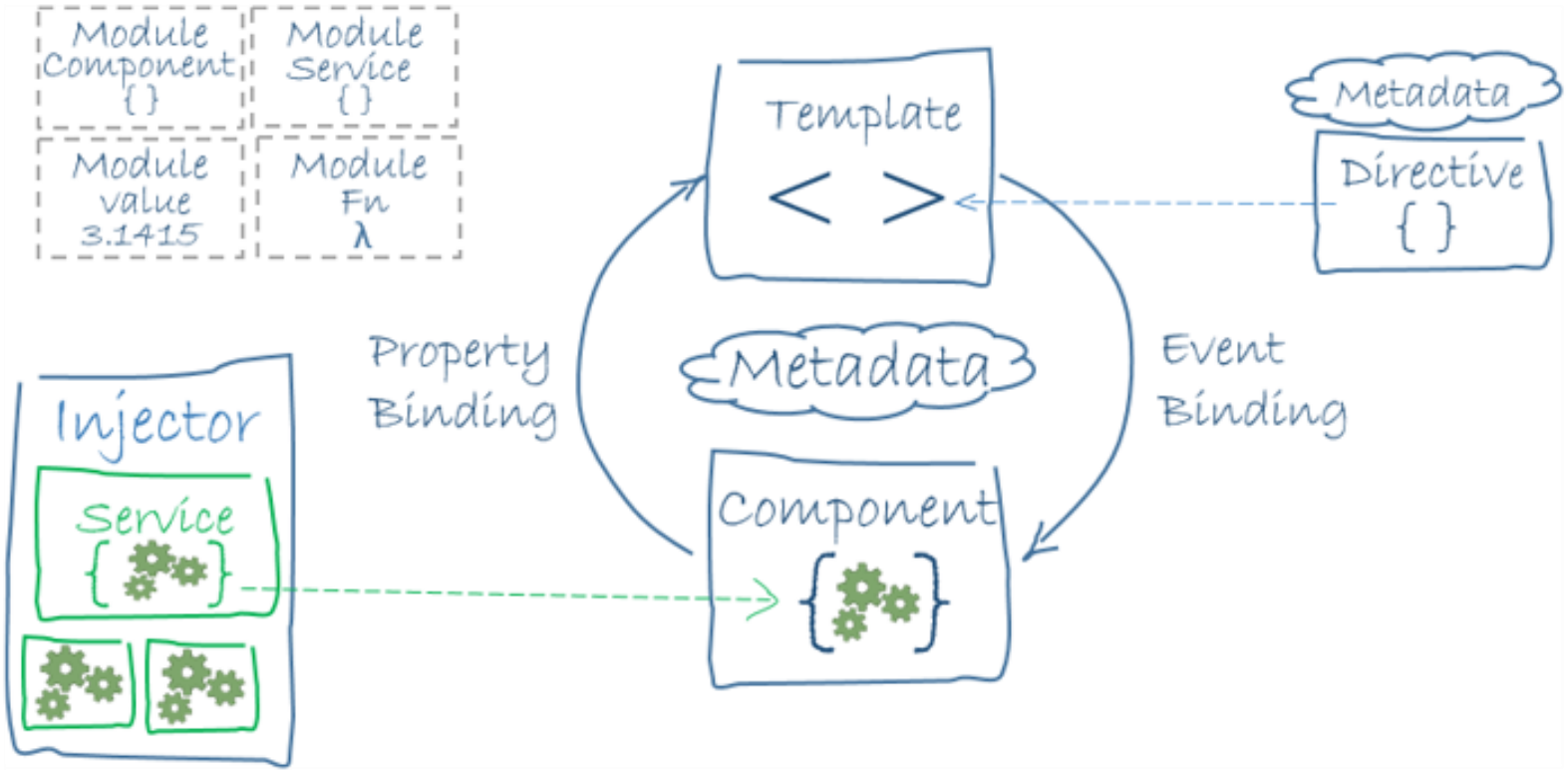
- Learning Curve
- Verbose Code
- Performance
- Complexity
- Version Migration
- Bundle Size

In summary, Angular is a powerful and versatile framework with many advantages, but it may not be the best choice for every project. Its suitability depends on the specific requirements, team expertise, and project complexity. Weighing the pros and cons and considering alternative options like React or Vue.js can help you make an informed decision.

3

Draw a visualization of the Angular architecture ?

The main building blocks of an Angular application are shown in the diagram below:-



What is TypeScript ?

Building on JavaScript, TypeScript is a tightly typed, object-oriented, and compiled programming language. It is a superset of the JavaScript language and was created to provide you with better tooling at any size.

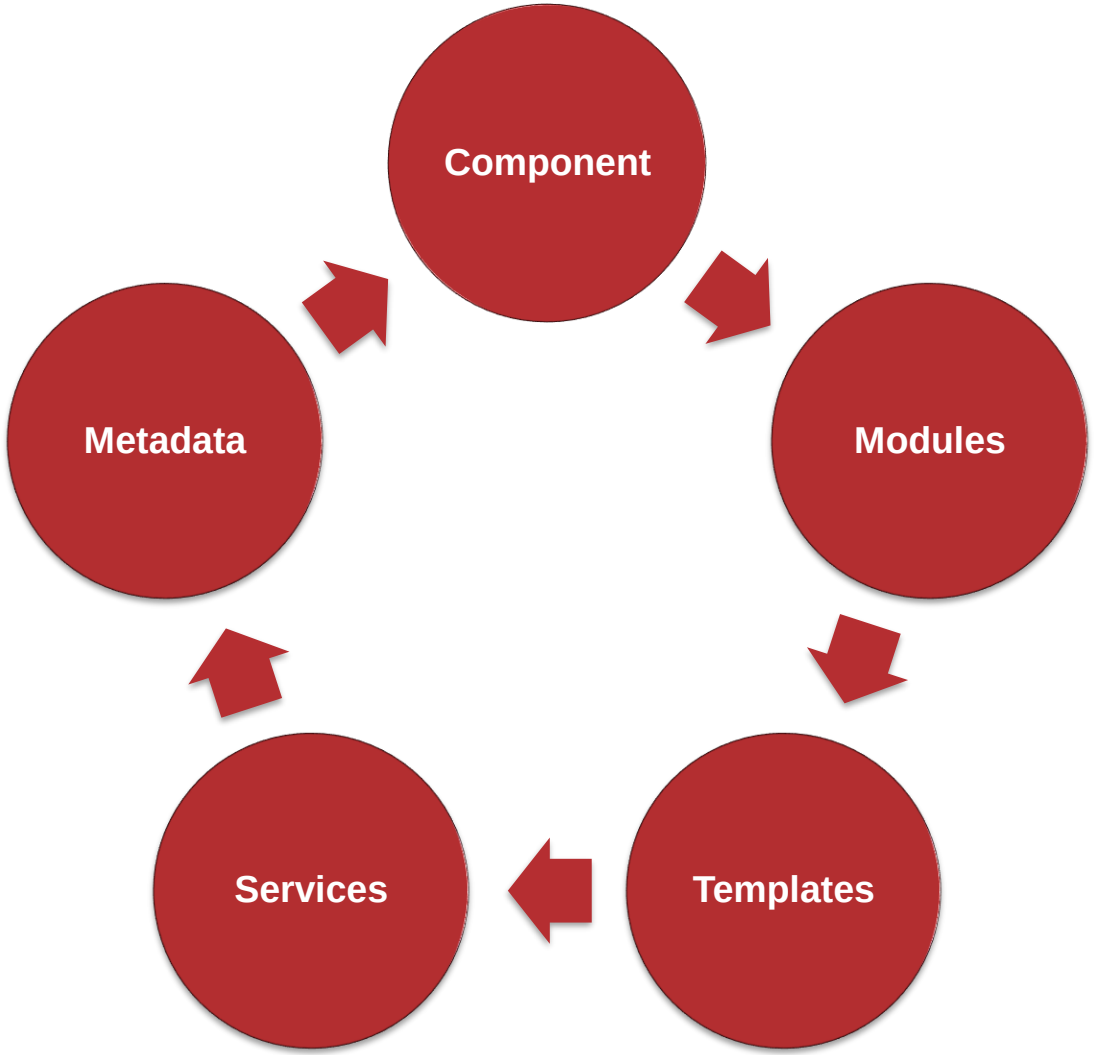
TypeScript calls itself **JavaScript with syntax for types**. In short, it is JavaScript with some additional features.

Four main goal of TypeScript:

- Make JavaScript development more efficient.
- Introduce optional types to JavaScript.
- Help catch mistakes earlier.
- Implement planned features of future JavaScript.

5

What are the key components of Angular ?



What are Directives ?

Directives are defined as classes that can add new behavior to the elements in the template or modify existing behavior.

```
import { Directive, ElementRef, Input } from '@angular/core';

@Directive({ selector: '[highlight]' })
export class HighlightDirective {
    constructor(el: ElementRef) {
       el.nativeElement.style.backgroundColor = 'red';
    }
}
```

Now this directive extends HTML element behavior with a red background as below

```
<p myHighlight>Highlight me!</p>
```

What are Components ?

The simplest UI building unit of an Angular project is a component, which together make up a tree of Angular components. These parts are a subset of directives. Contrary to directives, components always have a template, and only one component can be instantiated for each element in a template. Here is a straightforward Angular component example.

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

@Component ({
  selector: 'my-app',
  template: ` <div>
    <h1>{{title}}</h1>
    <div>Learn Angular with examples</div>
  </div> `,
})

export class AppComponent {
  title: string = 'Welcome to Angular world';
}
```

8

What is Template ?

An Angular component's properties can be bound to controls in a template to display data in an HTML view. One of two locations can be used to store the template for your component. You can create the template directly within the component using the template property, or you can define it in a separate HTML file and link to it in the component metadata using the templateUrl property of the @Component decorator.

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

@Component ({
   selector: 'my-app',
   templateUrl: 'app/app.component.html'
})

export class AppComponent {
   title: string = 'Hello World';
}
```

What is Module ?

Applications are organized into various modules to isolate their functionality, and modules act as logical boundaries inside an application. Let's look at an example where the root module in app.module.ts has the @NgModule decorator.

```
import { NgModule }      from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent }  from './app.component';

@NgModule ({
  imports:      [ BrowserModule ],
  declarations: [ AppComponent ],
  bootstrap:    [ AppComponent ],
  providers: []
})
export class AppModule { }
```

What is Metadata ?

Metadata is used to decorate a class so that it can configure the expected behavior of the class. The metadata is represented by decorators.