AngularJS

DEEPER LIFE HIGH SCHOOL

Introduction to Angular

Angular is a popular open-source front-end web application framework maintained by Google and a community of developers

Also, AngularJS is a **JavaScript framework**. It can be added to an HTML page with a <script> tag.
AngularJS extends HTML attributes with **Directives**, and binds data to HTML with **Expressions**.

AngularJS Example

```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.
js"></script>
<body>

<div ng-app="">
  <p>Name: <input type="text" ng-model="name"></p>
  <p ng-bind="name"></p>
</div>

</body>
</html>
```

HISTORY AND BACKGROUND

Prior to its release, a Google employee by the name of Miško Hevery, was developing a side project. This side project was to help make building web applications easier for a couple internal projects he was working on.

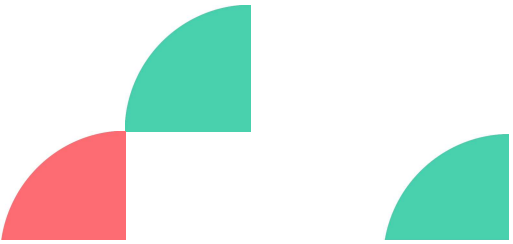
This side project later became known as AngularJS (Angular because of the < > in HTML).
Misko and a few others started to create a few more internal applications with AngularJS, eventually releasing it as an open source project in 2010.

Why use Angular JS:

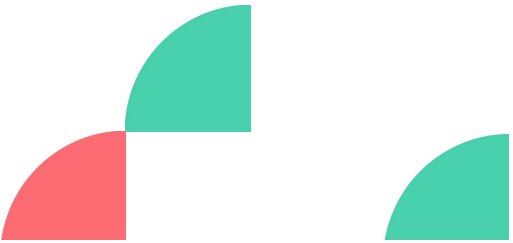
1. Component-Based Architecture: Angular applications are built around the concept of components, which are reusable, self-contained building blocks that encapsulate a part of the user interface and its functionality. Components encourage modularity and reusability in your codebase.

2. Templates: Angular uses HTML templates combined with Angular-specific syntax to define the user interface of your application. Templates are used to display data and create dynamic views.

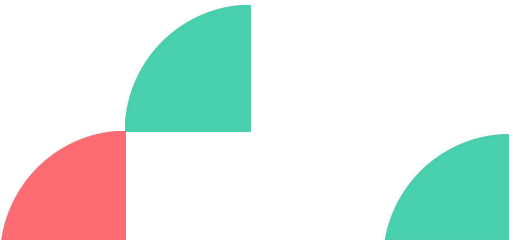
3. Data Binding: Angular provides powerful two-way data binding, allowing data changes in the model (JavaScript code) to automatically update the view (HTML) and vice versa. This simplifies the synchronization of data between the user interface and application logic.



4. Dependency Injection: Angular's dependency injection system helps manage the dependencies of your application by providing services and injecting them into components when needed. This promotes separation of concerns and testability.



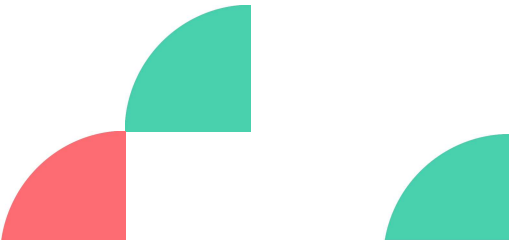
5. Directives: Directives are special HTML attributes or elements in Angular that allow you to extend HTML's behavior and appearance. Examples include *ngFor for looping and *ngIf for conditional rendering



6. Services: Services in Angular are used to encapsulate and share application logic, data retrieval, and other functionalities across components. They promote reusability and separation of concerns.

7. Routing: Angular provides a robust routing mechanism that allows you to create navigation paths and views for your SPA. You can define routes, route parameters, and route guards to control access to routes

8. Forms: Angular offers powerful form handling with features like template-driven forms and reactive forms (also known as model-driven forms). It simplifies form validation, data binding, and form submission.



9. Observables and RxJS: Angular leverages Observables from the RxJS library for handling asynchronous operations, such as HTTP requests. This enables efficient handling of data streams and events in your application.

10. Internationalization and Accessibility:

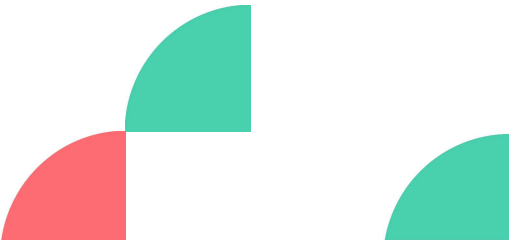
Angular supports internationalization (i18n) for building multilingual applications. It also emphasizes accessibility features to ensure your apps are usable by people with disabilities.

11. CLI (Command Line Interface): **Angular CLI is a command-line tool that simplifies project creation, code generation, testing, and deployment tasks, making it easier to get started with Angular development**

AngularJS vs. Other JavaScript framework

Learning Curve AngularJS has a steep learning curve due to its extensive set of features and unique terminology. It requires a solid understanding of concepts like dependency injection, directives, and modules.

ReactJS is relatively easy to learn, particularly for those already comfortable with JavaScript. However, it requires knowledge of additional libraries for routing and state management (like Redux), which can add complexity.



Vue.js has the easiest learning curve. Its simple and intuitive API, combined with extensive documentation, make it a favorite among beginners.

EVALUATION

1. What is AngularJS
2. Briefly narrate the history of AngularJS
3. Why in your opinion you think AngularJS is better compared to other JavaScript framework.

THANK YOU

www.deeperlifehighschool.org