Difference between Angular, React and Vue

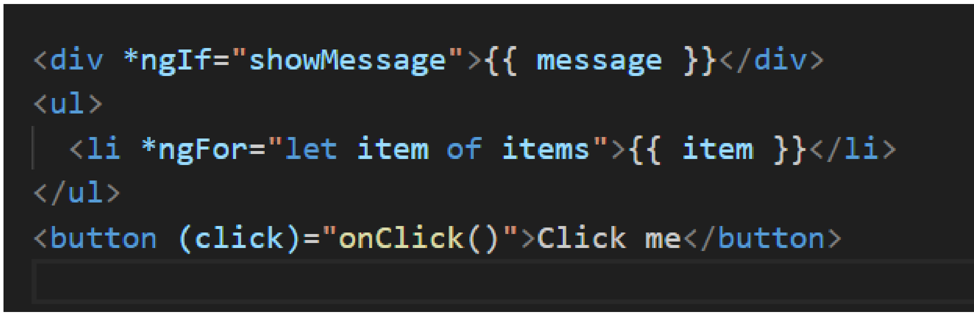
Angular: Developed and maintained by Google, Angular is a complete and opinionated framework written in TypeScript. It follows the Model-View-Controller (MVC) architecture and comes with a lot of built-in features and tooling, including dependency injection, forms handling, and routing.

Architecture:

- **Angular** follows the Model-View-Controller (MVC) architecture. It provides a complete and opinionated framework with a set of guidelines and best practices for building large-scale applications.
- **Components and Templates**: Angular's architecture is component-based. Each component encapsulates its own logic, template (HTML), and styles (CSS). Templates use Angular's template syntax that allows for two-way data binding between the model and the view.
- **Dependency Injection:** Angular has a built-in dependency injection system, which allows for managing the application's services and components more efficiently.
- **Routing:** Angular provides a robust router module for handling client-side routing and navigation within the application.

Syntax:

- Templates: Angular templates use a specific syntax that combines HTML with Angular directives and expressions. For example, you'll find directives like *ngIf, *ngFor, and property bindings using [property]="value". Event bindings use (event)="handler()".



```
<div *ngIf="showMessage">{{ message }}</div>
<ul>
  <li *ngFor="let item of items">{{ item }}</li>
</ul>
<button (click)="onClick()">Click me</button>
```

- Angular also supports template reference variables using #varName.
- Component Class: Angular components are defined using TypeScript classes with properties, methods, and lifecycle hooks.

Dom:

- DOM Manipulation: Angular uses a two-way data binding approach to automatically keep the view (DOM) in sync with the model (data).

React: Developed and maintained by Facebook, React is a JavaScript library for building user interfaces. It follows a component-based approach, where the UI is divided into reusable components. React is written in JavaScript (JSX) and allows developers to integrate it with other libraries and tools more easily.

Architecture:

- **React** is primarily a view library and doesn't enforce a specific architecture. It allows developers to choose how they want to structure their applications.

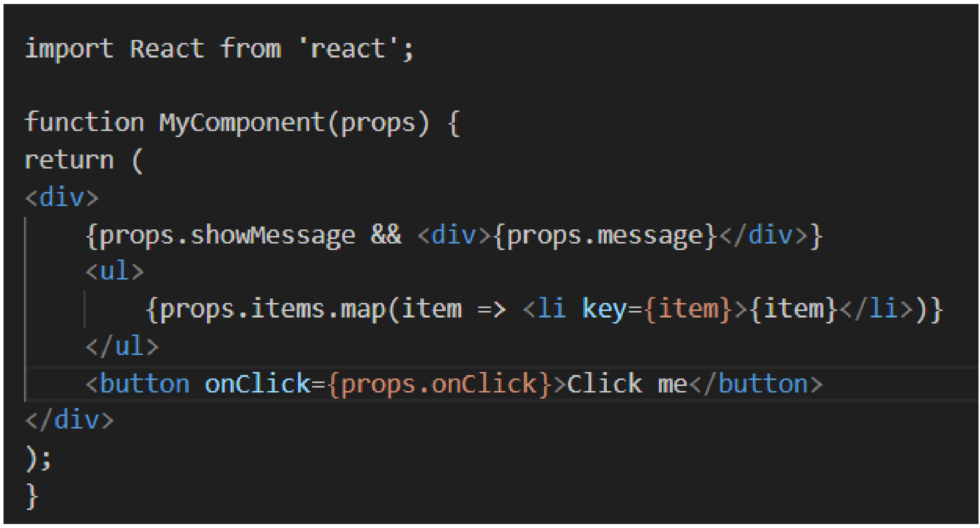
- **Components and JSX:** React's architecture is also component-based, where each component manages its own logic and state. React components use JSX (JavaScript XML) to define their templates, which are more similar to JavaScript code.

- **State Management:** React provides state management libraries like Redux or React's built-in Context API, which help manage application state and data flow in a predictable way.

- **Routing:** React itself doesn't include a built-in routing solution, but developers can use third-party libraries like React Router to handle routing and navigation.

Syntax:

- JSX: React uses JSX (JavaScript XML) as a syntax extension that allows you to write component templates using a syntax that closely resembles HTML. JSX is transpiled to JavaScript using tools like Babel. In JSX, you can include JavaScript expressions within curly braces {} and use components as if they were HTML elements.



```
import React from 'react';

function MyComponent(props) {
return (
<div>
    {props.showMessage && <div>{props.message}</div>}
    <ul>
        {props.items.map(item => <li key={item}>{item}</li>)}
    </ul>
    <button onClick={props.onClick}>Click me</button>
</div>
);
}
```

- Component Class or Function: React components can be defined using ES6 classes with methods and lifecycle hooks or as functional components using JavaScript functions.

Dom:

- Virtual DOM: React uses a virtual DOM to efficiently update the actual DOM. When a component's state or props change, React creates a virtual representation of the UI (a lightweight copy of the actual DOM)

Vue: Developed and maintained by a group of independent developers, Vue is also a JavaScript library for building user interfaces. It follows a similar component-based approach like React, but it's more lightweight and approachable for developers who are new to frontend frameworks.

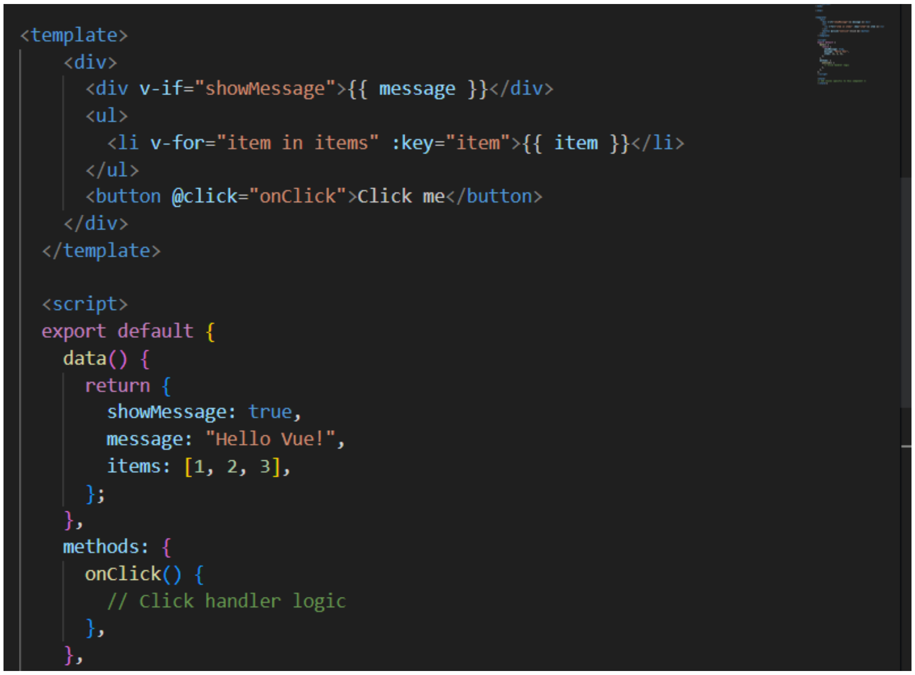
Architecture:

- follows an architecture similar to React's, but it comes with more predefined structure and opinions. It promotes a component-based approach but also provides a framework with guidelines for building scalable applications.
- **Components and Templates:** Like Angular and React, Vue's architecture is also component-based. Vue components are self-contained units with their logic, template (HTML with Vue's template syntax), and styles.
- **Reactivity:** Vue introduces reactivity using its "reactive" system, which automatically updates the DOM when data changes.
- **Routing:** Similar to React, Vue doesn't include a built-in routing solution, but developers can use Vue Router for handling routing and navigation.

Syntax:

- Templates: Vue templates are written using HTML-like syntax with additional Vue-specific directives. Similar to Angular, Vue uses directives like v-if, v-for, and attribute bindings using :property="value". Event bindings use @event="handler()". Vue templates also support template reference variables using ref="varName".

- Single File Components (SFC): Vue encourages using Single File Components, where the template, JavaScript, and styles are all defined within the same file. The template is enclosed in <template></template>, JavaScript in <script></script>, and styles in <style></style> blocks.



```
<template>
    <div>
      <div v-if="showMessage">{{ message }}</div>
      <ul>
        <li v-for="item in items" :key="item">{{ item }}</li>
      </ul>
      <button @click="onClick">Click me</button>
    </div>
  </template>

  <script>
  export default {
    data() {
      return {
        showMessage: true,
        message: "Hello Vue!",
        items: [1, 2, 3],
      };
    },
    methods: {
      onClick() {
        // Click handler logic
      },
    },
```

DOM :

- Virtual DOM: Similar to React, Vue also uses a virtual DOM to efficiently update the actual DOM.Vue's virtual DOM operates similarly to React's, by comparing the previous virtual representation with the updated one to calculate and apply minimal changes to the real DOM.