



Vue.js

AN INTRODUCTION TO VUE.JS

at **achilles** | अकिलीज़
technosolutions
technologize!

-WITH MR. SUMEET VIJAY WAJPE

-MCT

ABOUT ME

Microsoft
CERTIFIED
Professional
Developer

Microsoft
CERTIFIED
Trainer



Microsoft
CERTIFIED
Technology
Specialist

- Has around **14 years** of experience in software development and Corporate IT training.
- Preferred trainer for **extensive latest Web Technologies & Responsive Web Design**
- **Full Stack Expert**



+91 98812 42590



sumeet.wajpe@hotmail.com

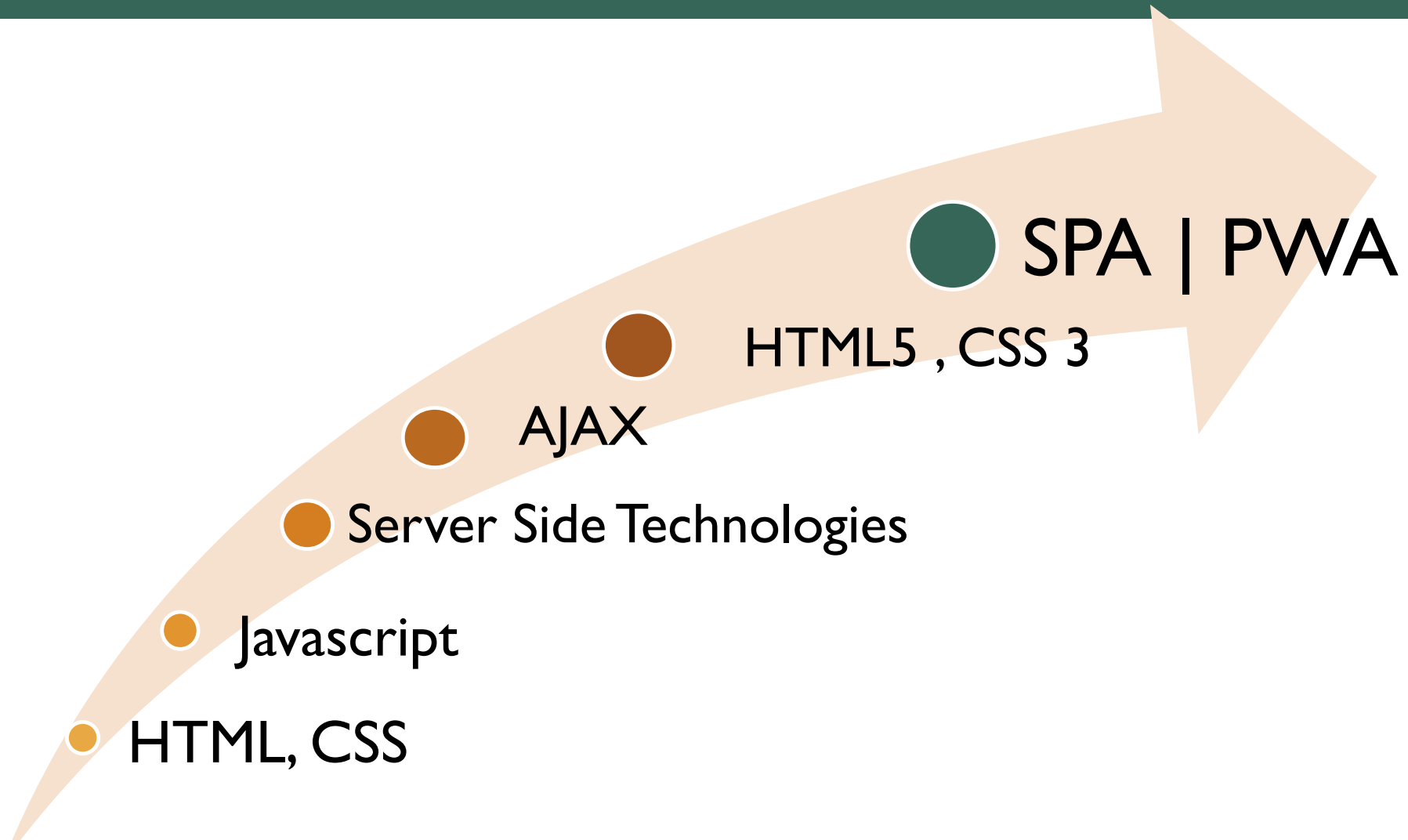


linkedin.com/in/sumeet-wajpe-the-trainer

AGENDA

- Evolution of the Web
- Why Vue.js ?
- Comparison of Vue with Angular & React
- New Vue Features
- Vue Ecosystem
- Creating a simple Application with Vue CLI

EVOLUTION OF WEB



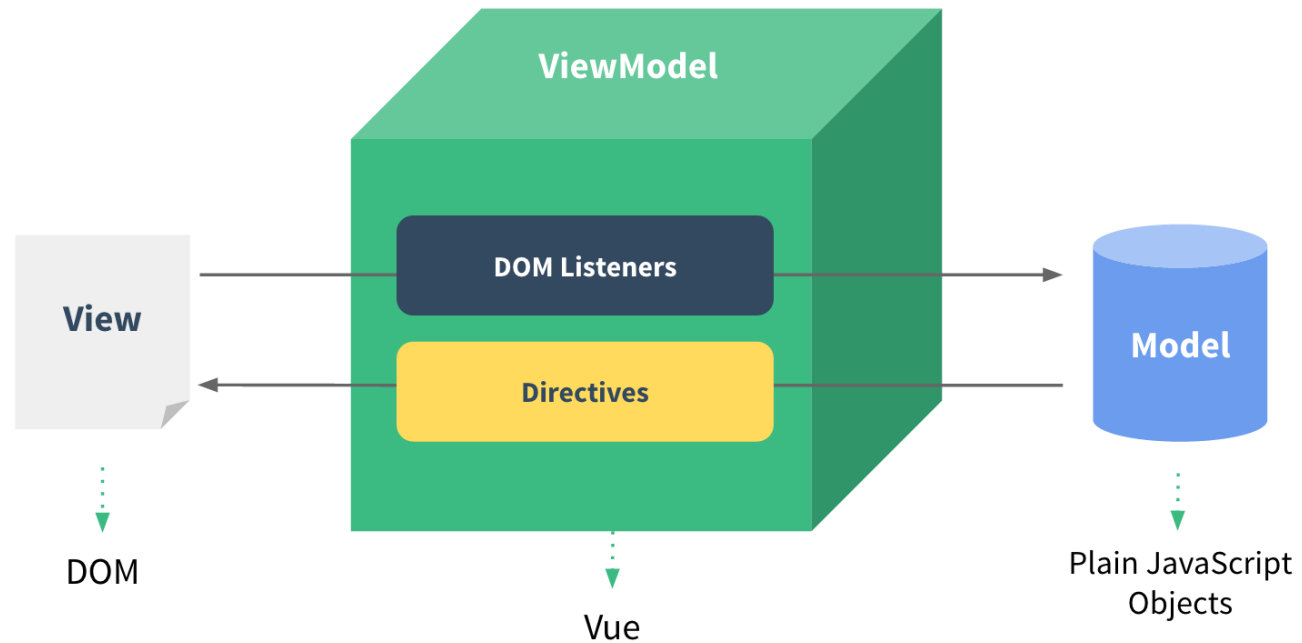
WHAT IS VUE.JS?

- Vue (pronounced /vjuː/, like **view**)
- A **progressive framework** for building user interfaces
- Created by **Evan You**
- Incrementally adoptable
- Launched in 2013

WHY VUE ?

- Very Small Size ~ 20kb (minimized and gzipped)
- Easy to Understand and Develop Applications
- Simple Integration
- Flexibility
- Reactivity
- Vue components
- Typescript Support
- MVVM inspired
- Fast
- Great documentation

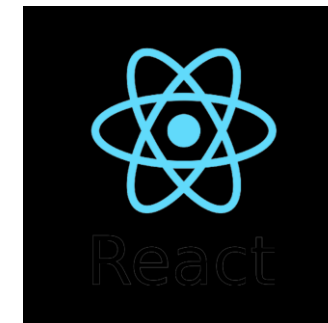
VUE ARCHITECTURE



VUE : BEST OF BOTH THE WORLDS



Vue.js



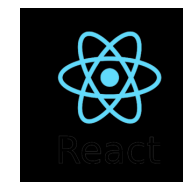
VUE : BEST OF BOTH THE WORLDS



Vue.js



- Directives
- Filters
- Component Driven
- Interpolation /
template expressions

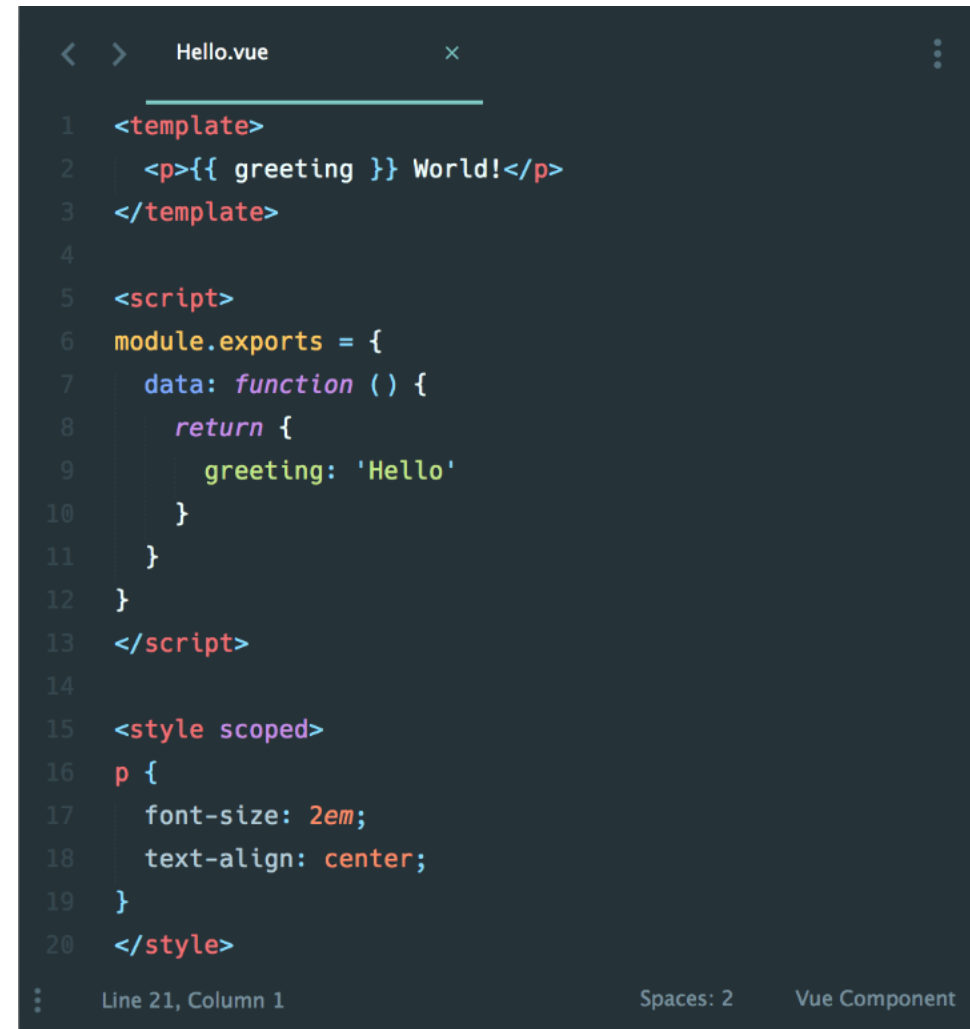


- Virtual DOM
- JSX & Key
- Props

NEW FEATURES

Single File Components

- single-file components with a .vue extension, made possible with build tools such as Webpack or Browserify.



```
< > Hello.vue x
1 <template>
2   <p>{{ greeting }} World!</p>
3 </template>
4
5 <script>
6 module.exports = {
7   data: function () {
8     return {
9       greeting: 'Hello'
10    }
11  }
12 }
13 </script>
14
15 <style scoped>
16 p {
17   font-size: 2em;
18   text-align: center;
19 }
20 </style>
⋮ Line 21, Column 1 Spaces: 2 Vue Component
```

NEW FEATURES CONTD..

Slots

- Vue implements a content distribution API inspired by the Web Components spec draft, using the `<slot>` element to serve as distribution outlets for content.

When the component renders, `<slot></slot>` will be replaced by “Your Profile”. Slots can contain any template code, including HTML:

```
<navigation-link url="/profile">
  <!-- Add a Font Awesome icon -->
  <span class="fa fa-user"></span>
  Your Profile
</navigation-link>
```

HTML



Then in the template for `<navigation-link>`, you might have:

```
<a
  v-bind:href="url"
  class="nav-link"
>
  <slot></slot>
</a>
```

```
<navigation-link url="/profile">
  Your Profile
</navigation-link>
```



NEW FEATURES CONTD..

Computed Properties

- any complex logic in template expressions, you should use a **computed property**.

```
<div id="example">
  <p>Original message: "{{ message }}"</p>
  <p>Computed reversed message: "{{ reversedMessage }}"</p>
</div>
```

```
var vm = new Vue({
  el: '#example',
  data: {
    message: 'Hello'
  },
  computed: {
    // a computed getter
    reversedMessage: function () {
      // `this` points to the vm instance
      return this.message.split('').reverse().join('')
    }
  }
})
```

Async Components



- only load a component from the server when it's needed
- Vue allows you to define your component as a factory function that asynchronously resolves your component definition.
- Vue will only trigger the factory function when the component needs to be rendered and will cache the result for future re-renders

```
Vue.component('async-example', function (resolve, reject) {
  setTimeout(function () {
    // Pass the component definition to the resolve callback
    resolve({
      template: '<div>I am async!</div>'
    })
  }, 1000)
})
```

NEW FEATURES CONTD..

Plugins

- Plugins usually add global-level functionality to Vue.
 - Add some global methods or properties.
 - Add one or more global assets: directives/filters/transitions etc
 - Add some component options by global mixin

```
MyPlugin.install = function (Vue, options) {  
  // 1. add global method or property  
  Vue.myGlobalMethod = function () {  
    // some logic ...  
  }  
  
  // 2. add a global asset  
  Vue.directive('my-directive', {  
    bind (el, binding, vnode, oldVnode) {  
      // some logic ...  
    }  
    ...  
  })  
  
  // 3. inject some component options  
  Vue.mixin({  
    created: function () {  
      // some logic ...  
    }  
    ...  
  })  
  
  // 4. add an instance method  
  Vue.prototype.$myMethod = function (methodOptions) {  
    // some logic ...  
  }  
}
```

FEATURE-RICH



A word cloud of AngularJS features. The words are in various sizes and orientations, with 'Components' and 'Directives' being the largest. The background is white with a dark green header bar at the top. The words are arranged in a way that they overlap and are scattered across the page.

Computed Properties

Syncing

Events

Outputting Data

Developer Tools

Filters

Watchers

Lists

Forms

Conditionals

Mixins

Reactivity

Dynamic Styles

Components

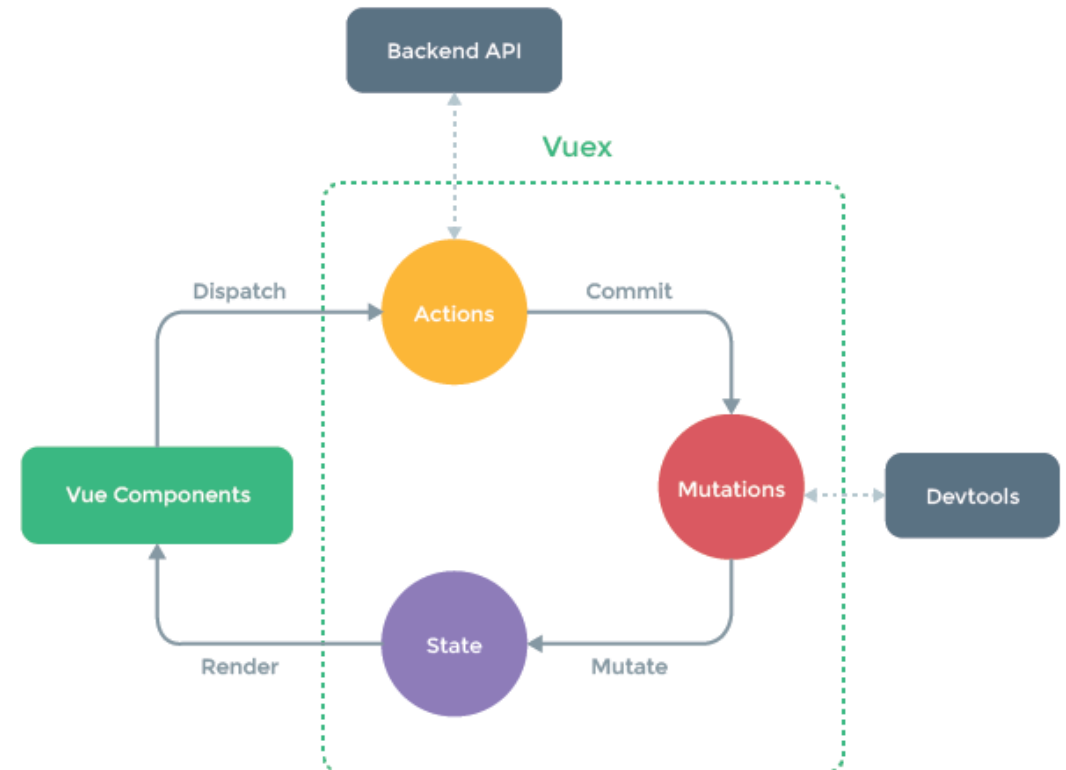
Directives

WHAT IS VUEX?

- Vuex is a **state management pattern + library** for Vue.js applications
- serves as a centralized store for all the components
- state can only be mutated in a predictable fashion
- It is a self-contained app with the following parts:
 - ❑ The **state**, the source of truth that drives our app;
 - ❑ The **view**, a declarative mapping of the **state**;
 - ❑ The **actions**, the possible ways the state could change in reaction to user inputs from the **view**.

DO I NEED VUEX?

- When we have **multiple components that share a common state**
- Multiple views may depend on the same piece of state.
- Actions from different views may need to mutate the same piece of state.



WHAT IS NUXT.JS?

- *Nuxt.js is a framework for creating Vue.js applications*
- *you can choose between Universal, Static Generated or Single Page application.*
- *Nuxt.js includes the following to create a rich web application development:*
 - Vue 2
 - Vue Router
 - Vuex (included only when using the store option)
 - Vue Server Renderer (excluded when using mode: 'spa')
 - vue-meta

VUE IN ACTION

- Demos