

# A Comparison of the Frontend JavaScript GUI Frameworks Angular, React, and Vue

# History of Frontend GUIs

# History of Frontend GUIs

- JavaScript Introduced in 1995
- Standardized in 1997 by ECMA
- DOM Manipulations
- Plugins
- jQuery in 2006 by John Resig

# Component-Based Web Design

# My ToDos



Enter ToDo, eg. Finish this survey...

Water Plants



Go shopping



Do laundry



Mow lawn



Bring out the trash



Filter items: [All](#) [Active](#) [Done](#)

[clear completed items \(2\)](#)

Layout: Page

Head

# Title My Todos



New Item Form

Enter ToDo, eg. Finish this survey...

ToDo Items

ToDo Item

Water Plants



Text shopping

Remove Button

Check Do laundry



ToDo Items, done

Mow lawn



ToDo Item, done

Bring out the trash



Actions

Filters:

# Component-Based Web Design

- Brad Frost in 2013: „Atomic Design“
- Improved Stability
- Separation of Concerns
- Reusability
- Cleaner Design
- Predictability

# Framework Overview





# Angular

- Miško Hevery in 2009
- „Google Feedback“
- Angular 2 in 2016
- TypeScript
- Two Separate Documentations
- High Modularity



# Angular

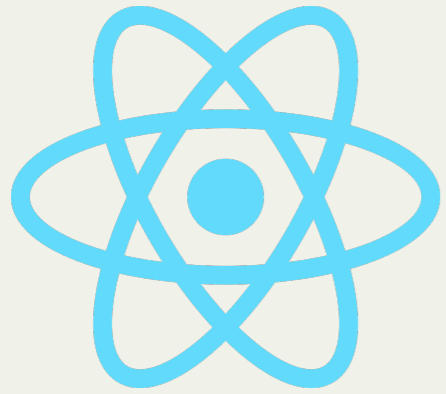
```
<!-- greeter.component.html -->
<h1 class="greeting">{{ this.name }}</h1>

// greeter.component.ts
import { Component, Input } from '@angular/core';

import template from './greeter.component.html'

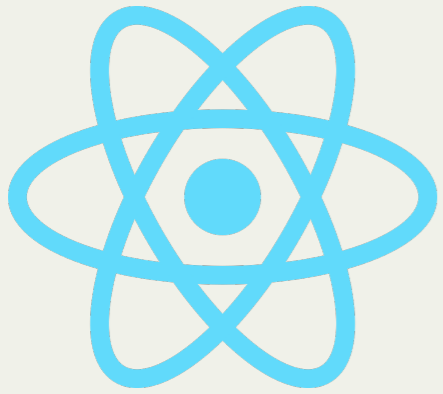
@Component({
  selector: 'greeter',
  template: template
})

export class Greeter {
  @Input() name: string;
}
```



# React

- Jordan Walke in 2011
- Facebook Newsfeed
- Instagram
- Open Sourced in 2013
- JSX
- Data Management



# React

```
// greeter.jsx
import React from 'react';
import ReactDOM from 'react-dom';

class Greeter extends React.Component {
  render() {
    return (
      <h1 className="greeting">
        Hello, {this.props.name}
      </h1>
    )
  }
}

export default Greeter;
```



# Vue

- Evan You in 2014
- „Improved Angular“
- Single-File Components
- Proprietary Templating Language
- Scoped CSS
- Progressive Frontend Framework



# Vue

```
<!-- greeter.vue -->
<template>
  <h1 class='greeting'>Hello, {{ name }}</h1>
</template>

<script>
export default {
  props: ['name']
}
</script>

<style scoped>
h1.greeting {
  text-decoration: underline;
}
</style>
```

# Virtual DOM

- DOM Manipulations Expensive
- Additional Data Structure
- Minimal Change-Sets
- Not Necessarily More Performant
- Implemented by React and Vue

# ToDo List Application



# There is much to do!

+ Add

---

My ToDos Active Items: 3, Total Items: 5

---

Repairs Active Items: 1, Total Items: 3

---

Shopping Active Items: 2, Total Items: 3

---

← Select other Framework

# Yay! A new List!

---

\* Title

Create List

# My ToDos



Enter ToDo, eg. Finish this survey...

Water Plants



Go shopping



Do laundry



Mow lawn



Bring out the trash



Filter items: [All](#) [Active](#) [Done](#)

[clear completed items \(2\)](#)

# Edit "My ToDos"

---

\* Title

My ToDos

Update List

***DEMO***

# ToDo List Application

- Getting to Know Each Framework
- Test Frameworks under „Real-World“ Conditions
- Consists of 4 „Pages“
- Reference Implementation
- Implementation Order: Vue, React, Angular

# Results

# Rendering

	Angular	React	Vue
<b>Component-Based</b>	✓	✓	✓
<b>Rendering Technique</b>	Service Worker	Virtual DOM	Virtual DOM
<b>Container Components</b>	✓	✓	✓
<b>Directives</b>	✓	✗	✓



# Dynamic Content

	Angular	React	Vue
Data Binding	✓	✓	✓
Two-Way Bindings	✓	✗	✓
Conditional Rendering	Directives	JavaScript	Directives
Event Handlers	Directives	JavaScript	Directives
Message Passing	Bottom-Up	Top-Down	Bottom-Up

# Routing

	Angular	React	Vue
Routing Module	✓	Third Party	✓
Hash-Based	✓	✓	✓
History pushState	✓	✓	✓
Dynamic Segments	✓	✓	✓
Nested Routes	✓	✓	✓

# Documentation

	Angular	React	Vue
Getting Started Guide	✓	✓	✓
User Guides	Extensive	Basics	Basics
API Documentation	✓	✓	✓
User Experience	Needs Improvement	Good	Great

# Overall Comparison

	Angular	React	Vue
<b>Initial Startup Time (ToDo List Application)</b>	6 h	3 h	2 h
<b>Total Time Until Finished (ToDo List Application)</b>	17 h	12 h	10 h
<b>Beginner Experience</b>	Poor	Ok	Great
<b>Target Audience</b>	Medium-Large Teams	Small-Medium Teams	Small-Medium Teams
<b>Intended Use-Case</b>	SPAs	SPAs, Small Components	SPAs, Small Components

# Questions?