

Why Angular?

The DOM API has issues

- A lot of JavaScript code involves updating the DOM, thus interweaving business logic and UI logic
- Code gets bloated really fast. These bits of code do not make most developers happy:

```
var td = document.getElementsByTagName('td');  
var textnode = document.createTextNode('Hi');  
td.appendChild(textnode);
```

```
var elem = document.createElement('div');  
elem.setAttribute('class', 'special');
```

```
var elem = document.getElementById('main-menu');  
elem.setAttribute('style', 'display: none;');
```

JavaScript libraries have issues



- A fantastic tool for cross-browser compatible DOM interaction and AJAX requests
- No help for structuring or testing your code
- And, aren't we almost done with the era of "cross-browser compatible"-ness?

React, Vue.js and LitElement

This is actually quite a hot discussion and it seems to come down to a matter of taste.

- React and Vue.js are essentially **view libraries**. You will have to add additional libraries for other architectural responsibilities.
- LitElement provides a small layer of data binding on top of **web component standards**.
- Angular is a full framework, an **all-in-one solution**.

Angular to the rescue!



- **Suitable for large applications** in JavaScript
- **Component based** structure for your application
- Extends HTML
- Embraces modern web standards
- Focuses on **speed and performance**

Angular features

- **Two-way databinding** through a declarative syntax
- **Advanced dependency injection** for working together in a loosely-coupled manner
- Extends HTML
- **Testable** from the ground up
- Runs everywhere
 - browser, NativeScript, server side rendering, web workers, ..
- **Strongly typed** (using TypeScript) to improve maintainability