HANYANG UNIVERCITY

Web System Programming

한양대학교
HANYANG UNIVERSITY

Contents

Web System Programming : JAVA Script

Ⅰ. Introduction

Ⅱ. SPA

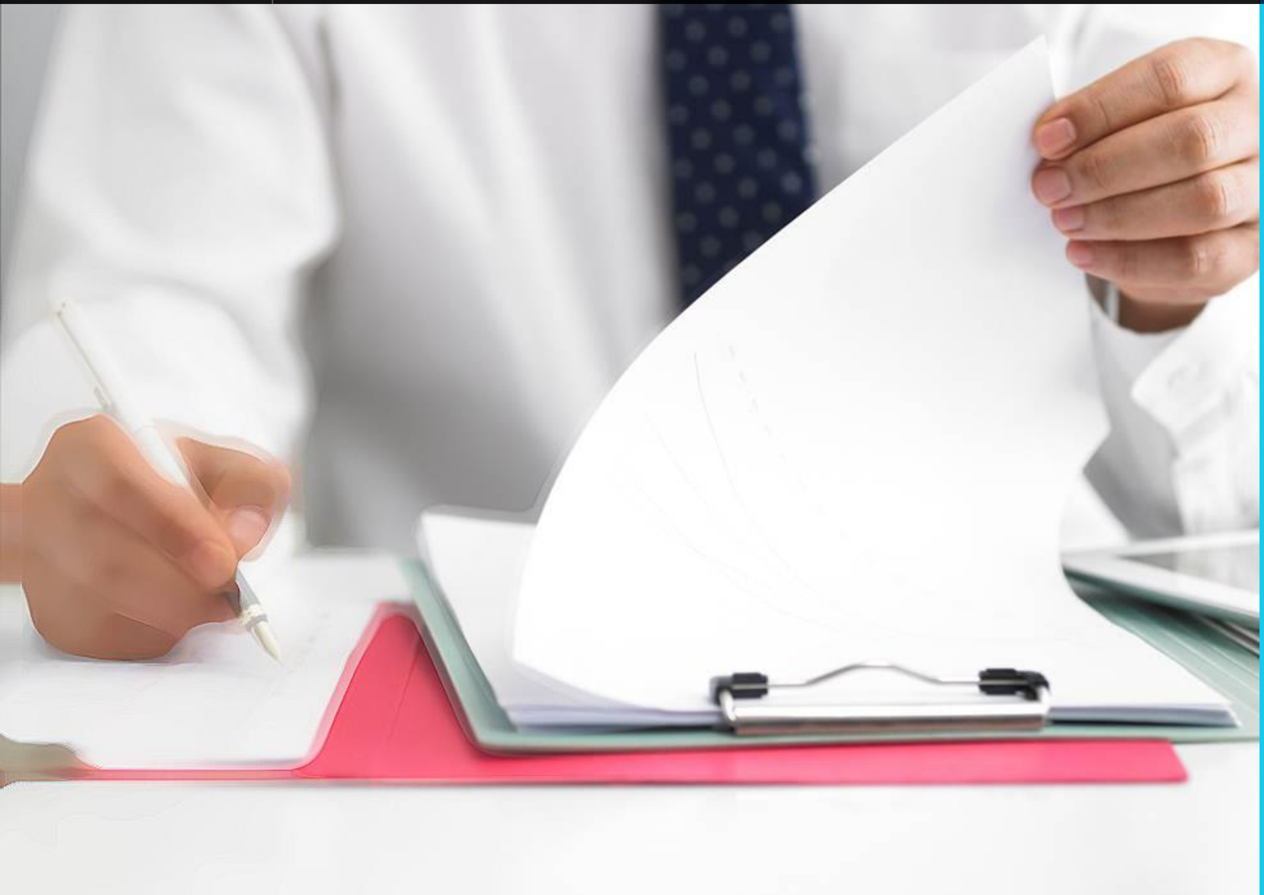
Ⅲ. PWA

Ⅰ

Web System Programming

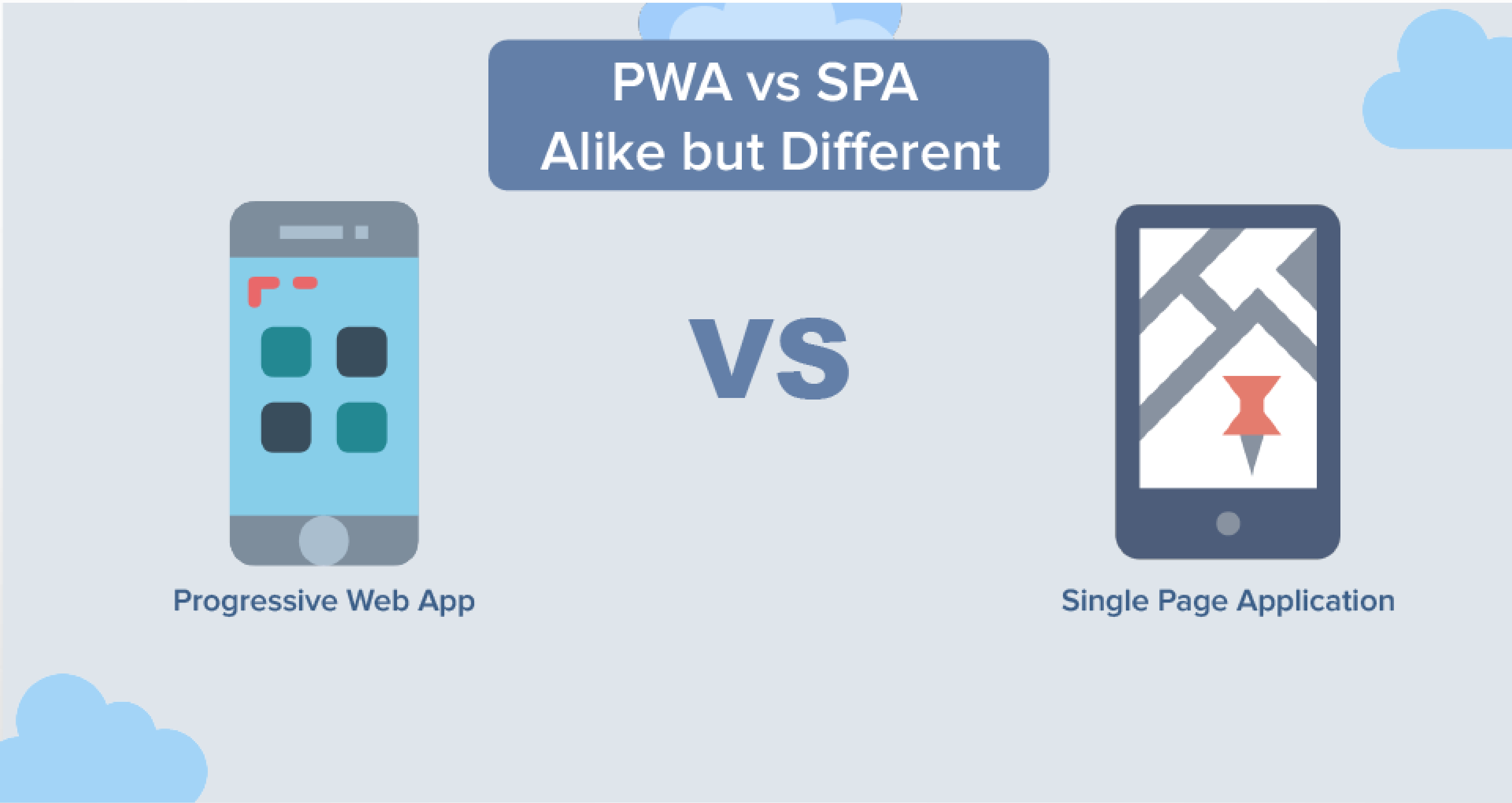
Introduction

HANYANG UNIVERSITY



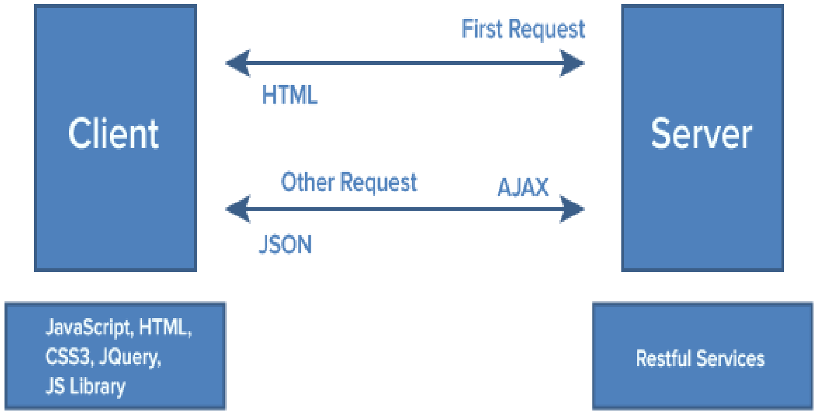
1. The class information
2. The goal and score high point
3. What is the java script?

1. THE CLASS INFORMATION : WEB SYSTEM PROGRAMMING II

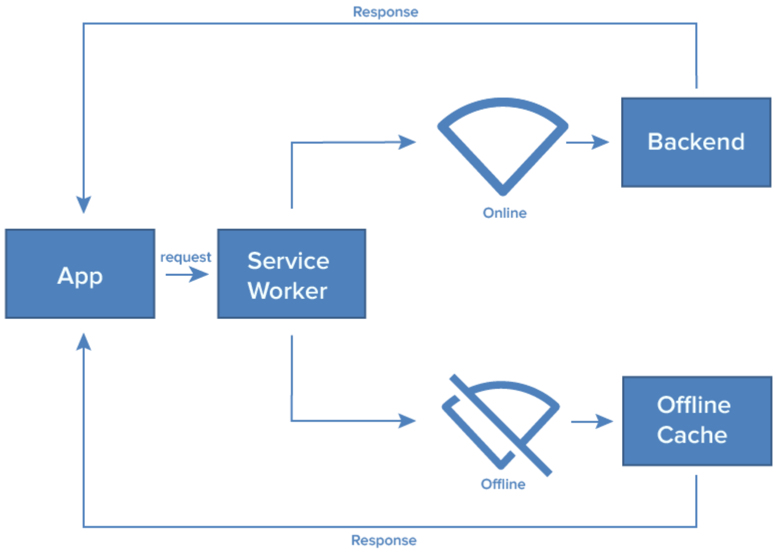


1. THE CLASS INFORMATION : WEB SYSTEM PROGRAMMING II

Single Pas Application



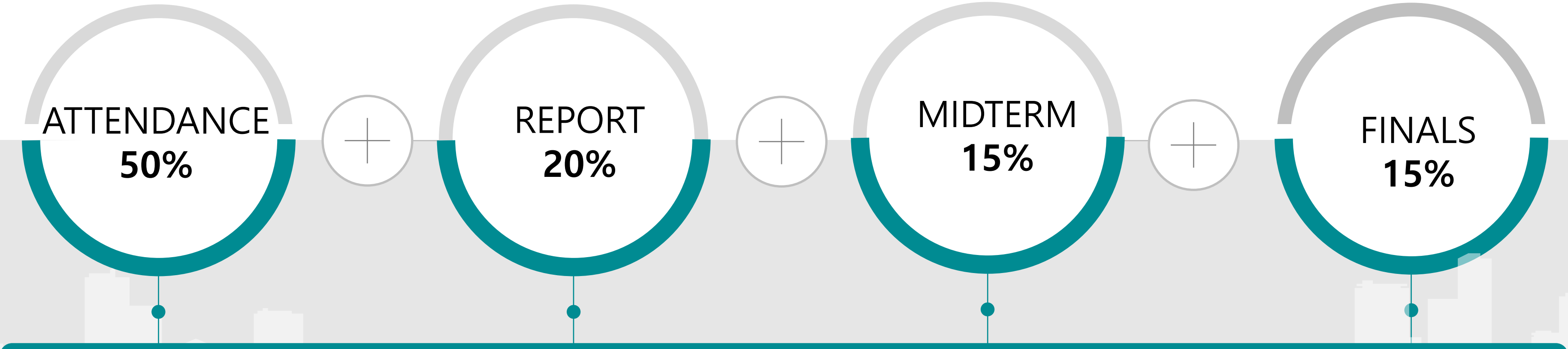
Progressive Web App



2. THE GOAL AND HIGH SCORE POINT

" The most important thing in class is attendance. The 5 absence is an F grade. "

In order to evaluate the grades, I will give final grade
by calculating a total of 4 reports, midterms, and final scores other than attendance.


ATTENDANCE **50%** + REPORT **20%** + MIDTERM **15%** + FINALS **15%**

THE GOAL OF THE CLASS IS TO GET AN WEB/APP DEVELPOMNET SKIIL USING JAVA SCRIPT.

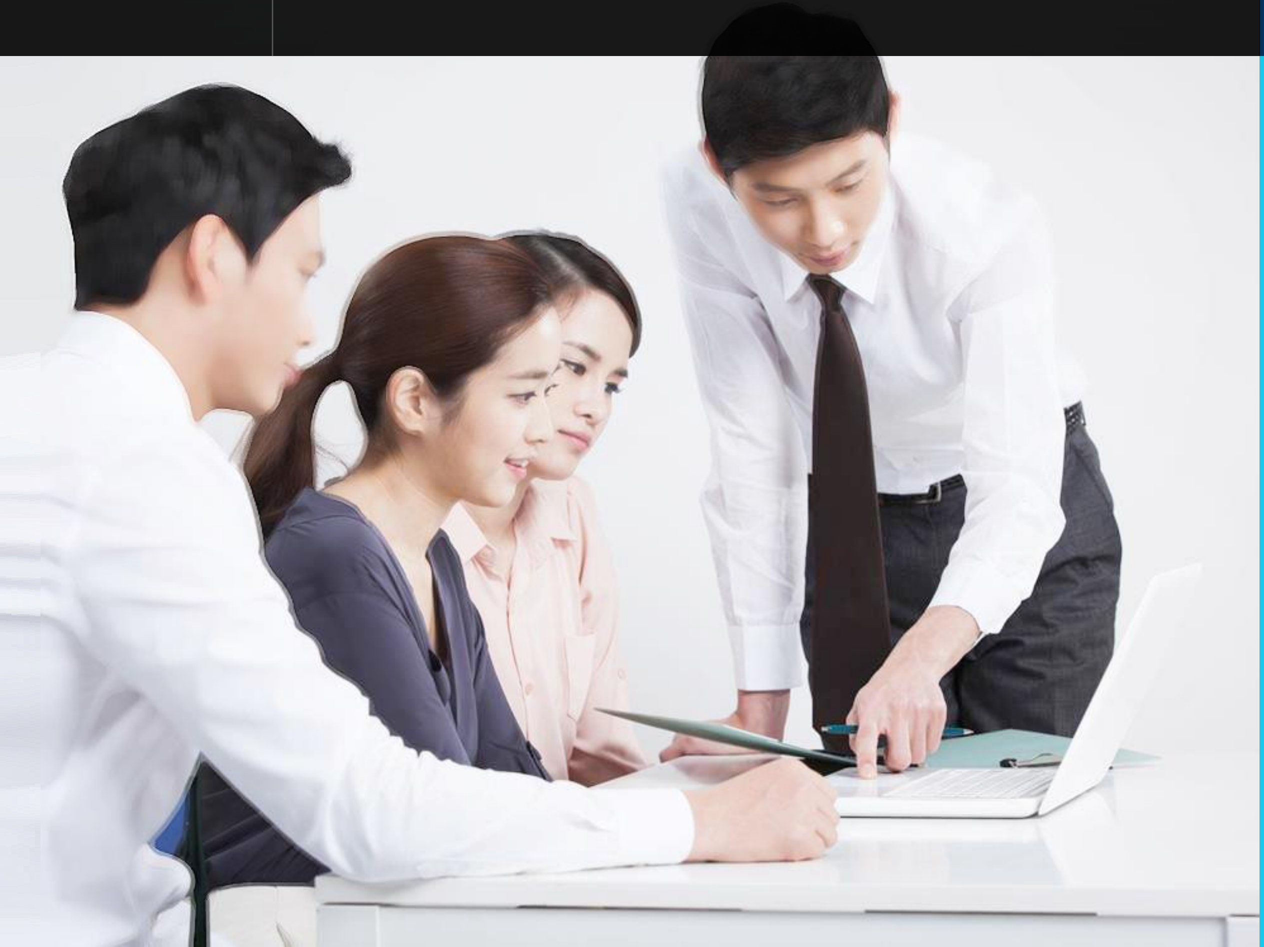
PESONAL CAPABILITIES	TEAM PROJECT
• **(REPORT)** Except for the exam week, I will give my homework once every two weeks. It may be a data survey or a coding-related materials. • **(MIDTERM)** The middle exam will be conducted with a total of 25 questions, a mixture of multiple-choice questions, single-answer and write-out answer	• **(FINALS)** Final exam is a team project that develops and submits web/apps using SPA or PWA. Organize your team members freely, but do not exceed 5 people. • Each team submits a score table that evaluates not only the final output but also the contribution of each team member by converting it into a perfect score of 100.

II

Single Page Application

SPA

HANYANG UNIVERSITY



1. Java script and framework
2. Main framework of SPA
3. Web component framework
4. Full stack framework
5. Mobile native framework

1. JAVA SCRIPT AND FRAMEWORK : What is the JavaScript?

- JavaScript was designed to add interactivity to HTML pages
- JavaScript is a scripting language
- A scripting language is a lightweight programming language
- A JavaScript consists of lines of executable computer code
- A JavaScript is usually embedded directly into HTML pages
- JavaScript is an interpreted language

  (means that scripts execute without preliminary compilation)
- Everyone can use JavaScript without purchasing a license

1. JAVA SCRIPT AND FRAMEWORK : What Can a JavaScript Do?

- **JavaScript can be used to validate data**

  - A JavaScript can be used to validate form data before it is submitted to a server. This saves the server from extra processing

- **JavaScript can be used to detect the visitor's browser**

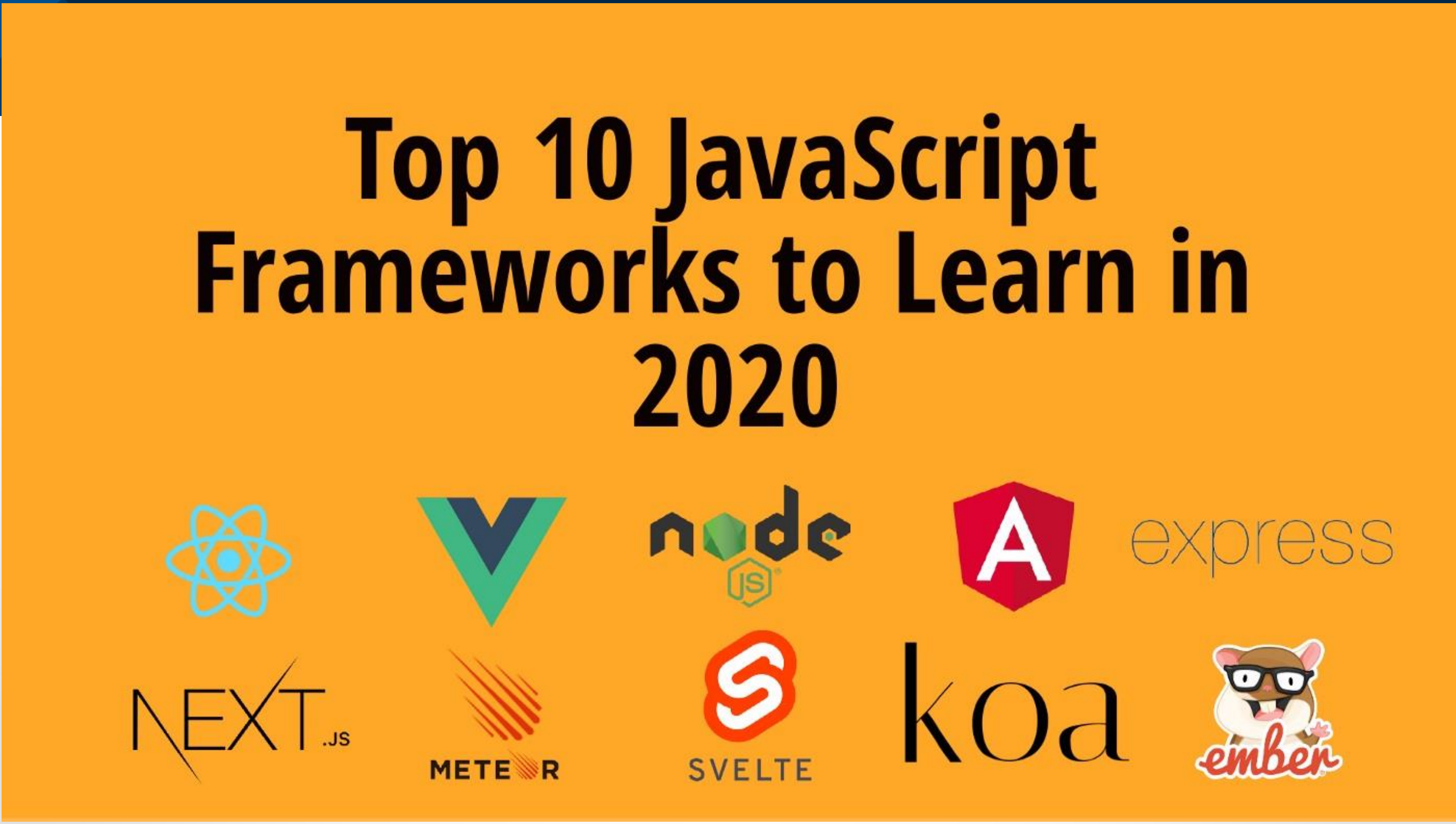
  - A JavaScript can be used to detect the visitor's browser, and – depending on the browser –
  load another page specifically designed for that browser

- **JavaScript can be used to create cookies**

  - A JavaScript can be used to store and retrieve information on the visitor's computer

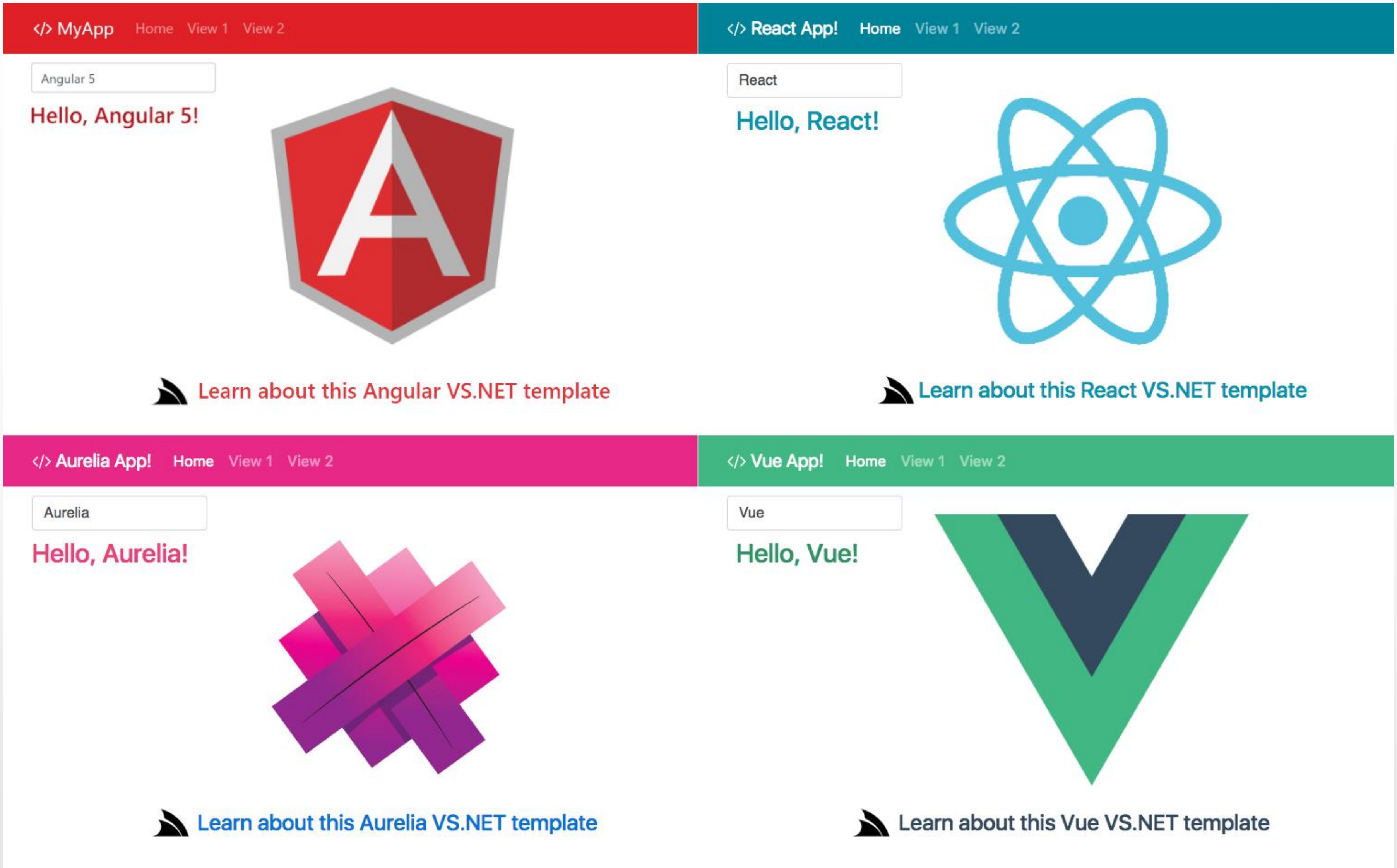
1. JAVA SCRIPT AND FRAMEWORK : What is the Framework?

- The JavaScript front-end framework defines or policies the form of all or part of the application.
- The framework stipulates how to write the contents of the design, which makes it easier to develop or to grasp the entire contents even when viewing the code later.
- In other words, the framework is to focus on **"the form to have as an application"** and provide the necessary functions to realize it.



Top 10 JavaScript Frameworks to Learn in 2020

2. MAIN FRAMEWORK OF SPA



Angular 5

Hello, Angular 5!

Learn about this Angular VS.NET template

React

Hello, React!

Learn about this React VS.NET template

Aurelia

Hello, Aurelia!

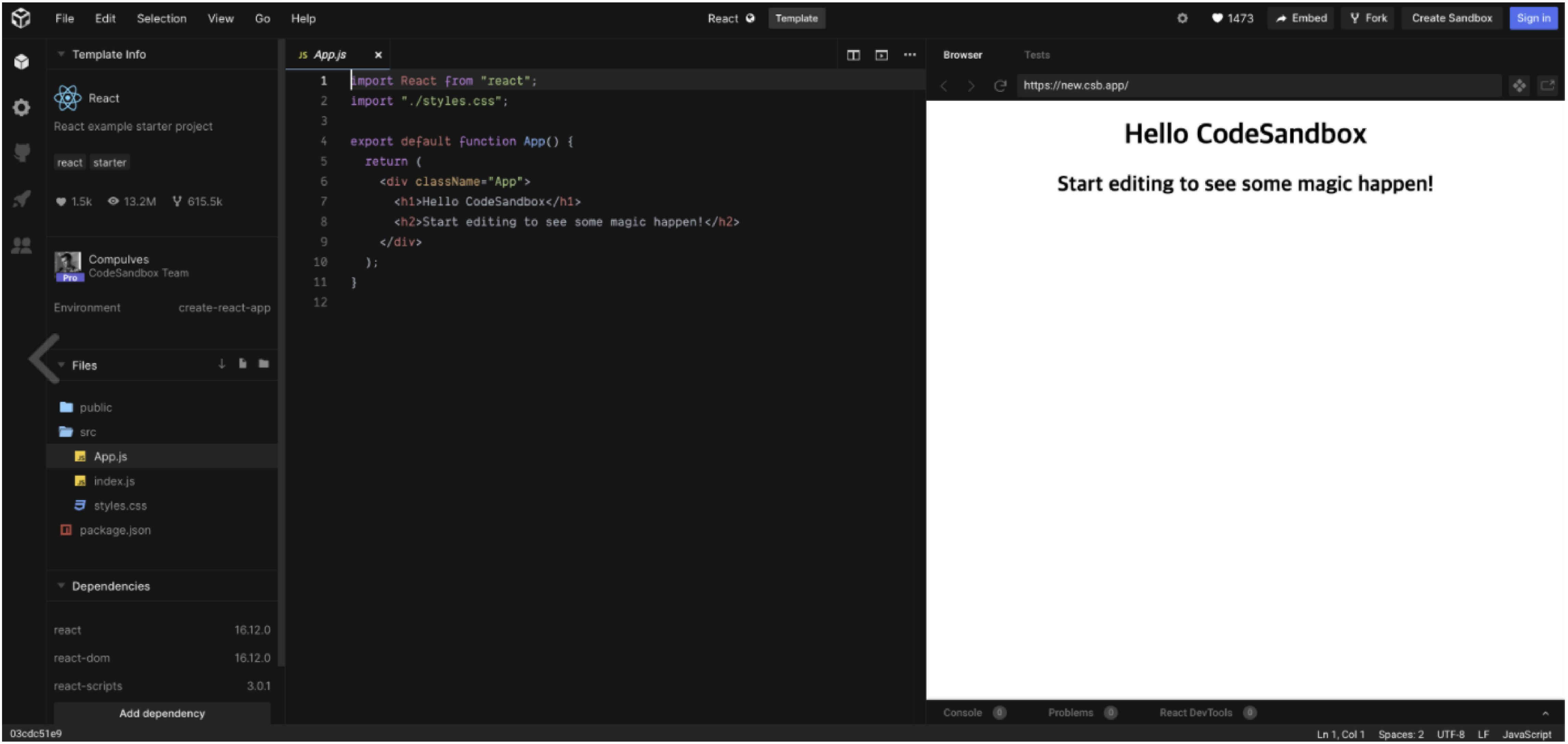
Learn about this Aurelia VS.NET template

Vue

Hello, Vue!

Learn about this Vue VS.NET template

2. MAIN FRAMEWORK OF SPA : React



Environment create-react-app

react	16.12.0
react-dom	16.12.0
react-scripts	3.0.1

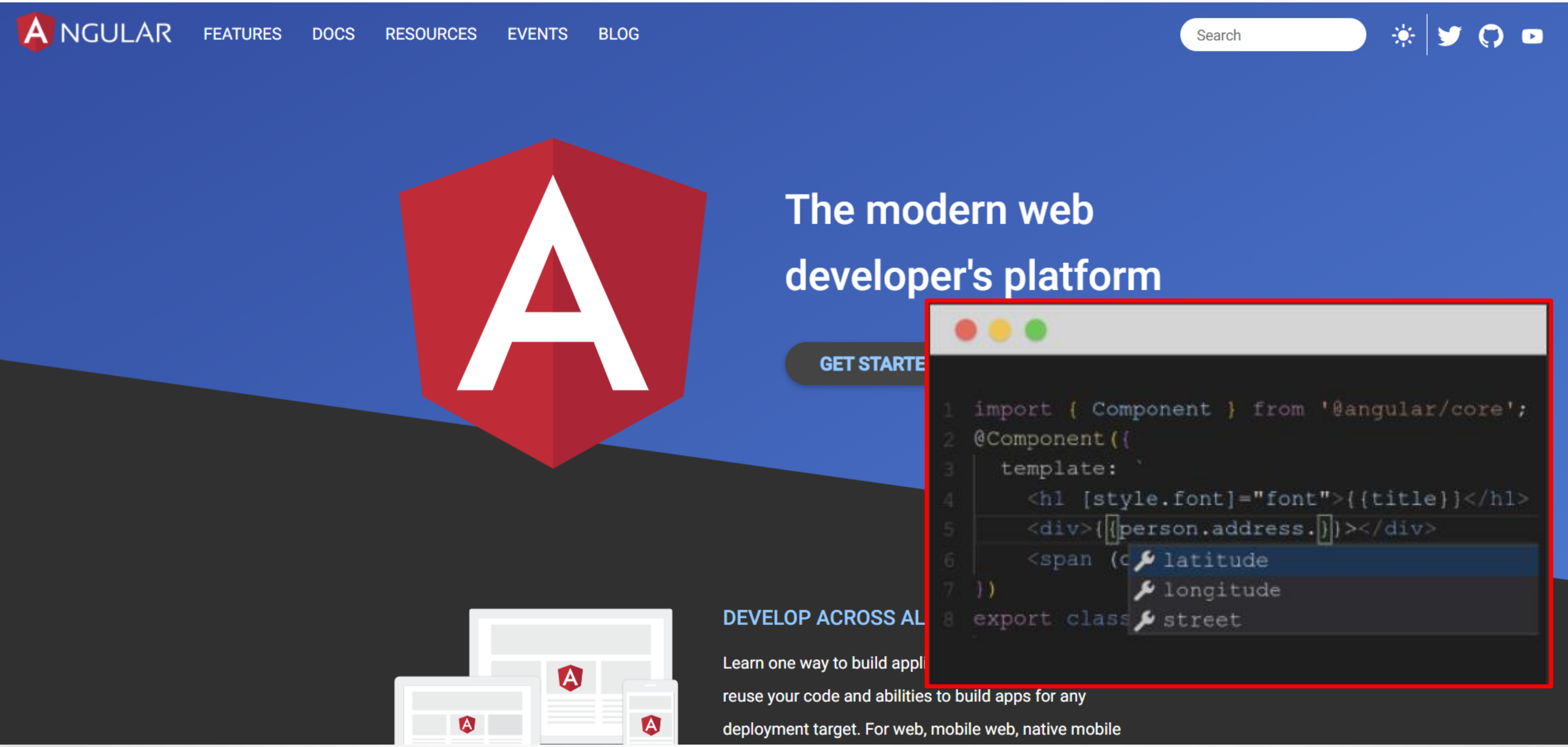
```
import React from "react";
import "./styles.css";

export default function App() {
  return (
    <div className="App">
      <h1>Hello CodeSandbox</h1>
      <h2>Start editing to see some magic happen!</h2>
    </div>
  );
}

```

https://new.csb.app/

Hello CodeSandbox

Start editing to see some magic happen!

2. MAIN FRAMEWORK OF SPA : Angular



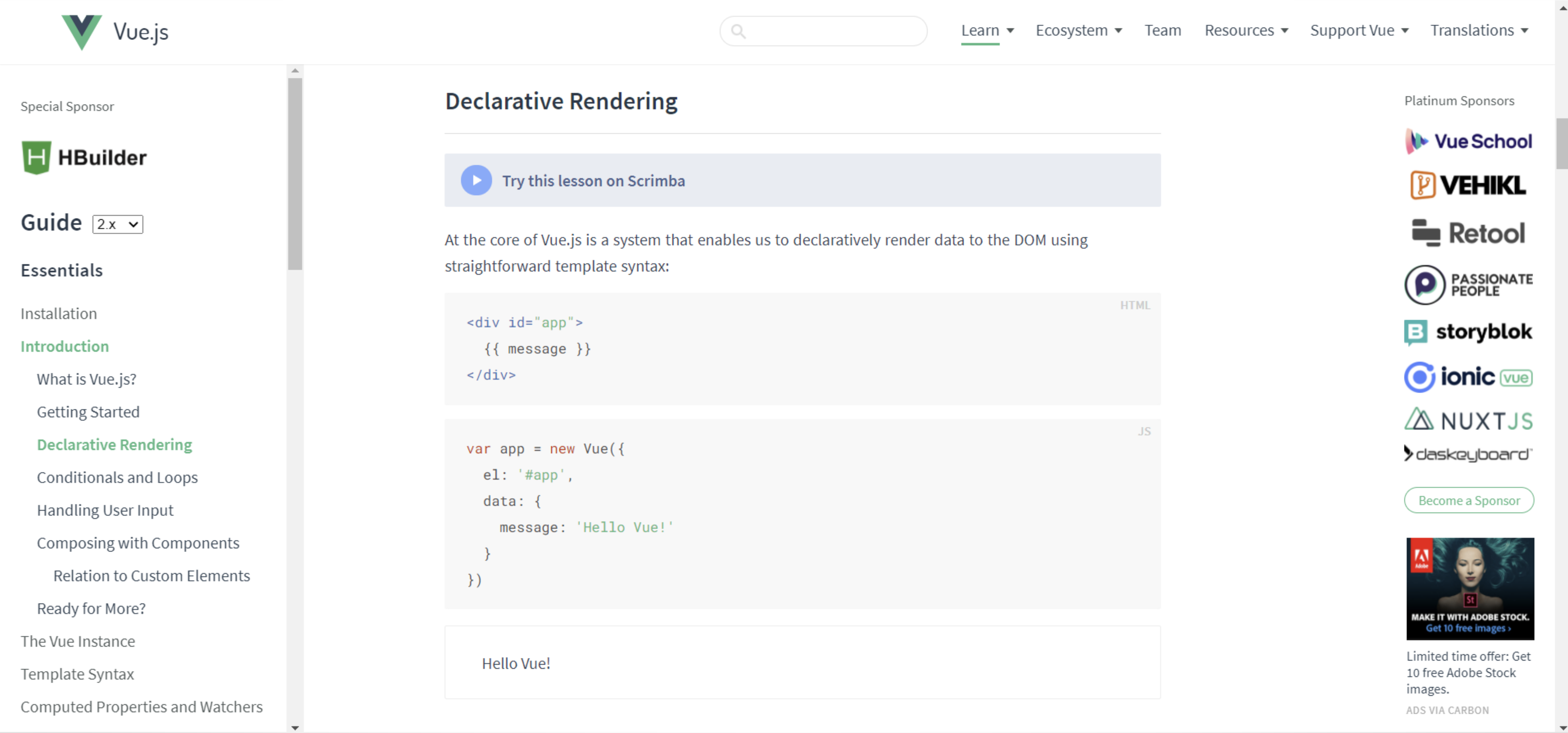
The modern web developer's platform

DEVELOP ACROSS AL

Learn one way to build appli
reuse your code and abilities to build apps for any
deployment target. For web, mobile web, native mobile

```
import { Component } from '@angular/core';
@Component({
  template: `
    <hl [style.font]="font">{{title}}</hl>
    <div>{{person.address.}}></div>
    <span (c
})
export class
```

2. MAIN FRAMEWORK OF SPA : Vue.js



Special Sponsor

HBuilder

Guide 2.x

Essentials

Installation

Introduction

- What is Vue.js?
- Getting Started
- Declarative Rendering
- Conditionals and Loops
- Handling User Input
- Composing with Components
  - Relation to Custom Elements
- Ready for More?

The Vue Instance

Template Syntax

Computed Properties and Watchers

Declarative Rendering

Try this lesson on Scrimba

At the core of Vue.js is a system that enables us to declaratively render data to the DOM using straightforward template syntax:

HTML

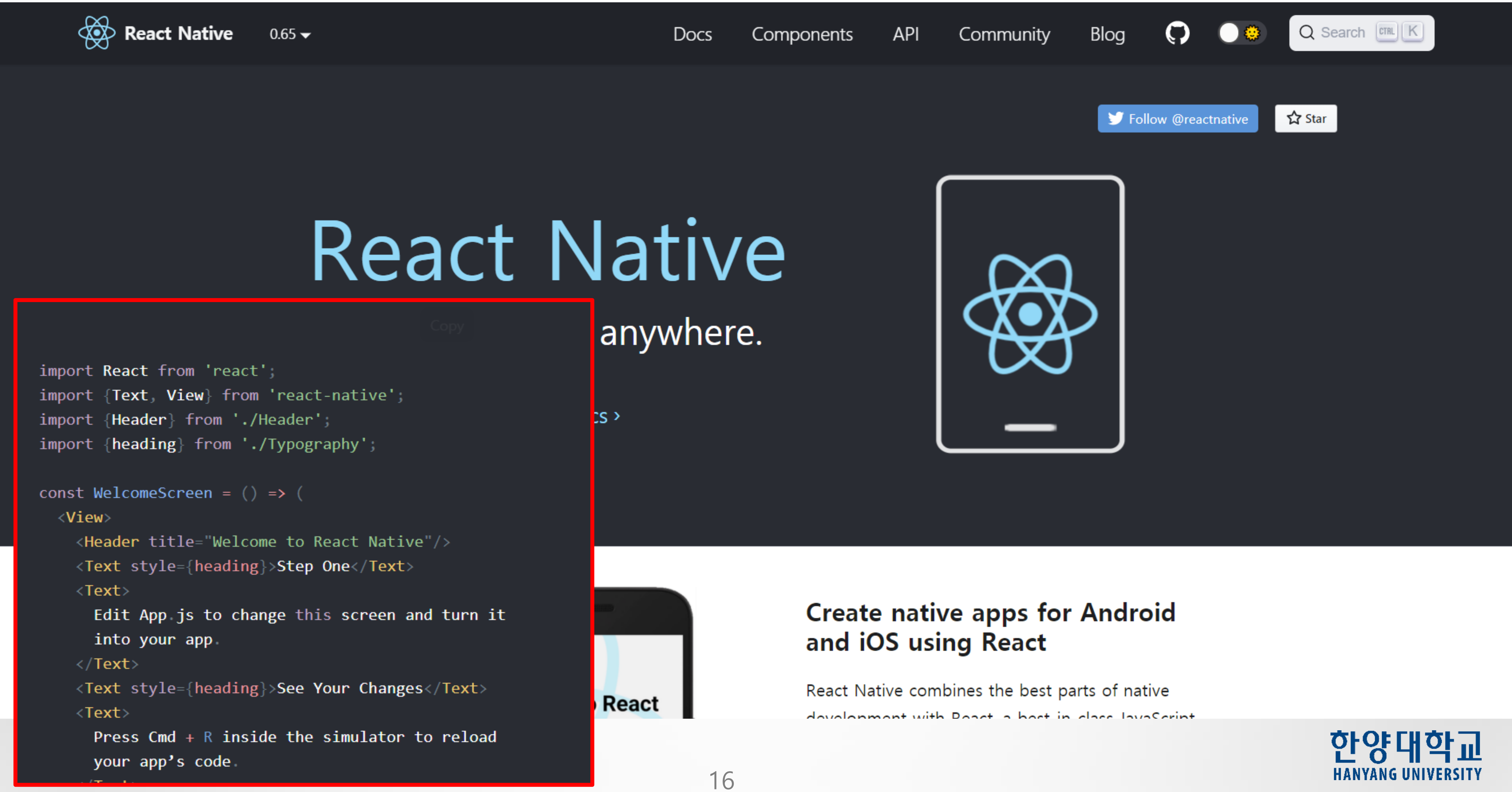
```html
<div id="app">
  {{ message }}
</div>
```

JS

```js
var app = new Vue({
  el: '#app',
  data: {
    message: 'Hello Vue!'
  }
})
```

Hello Vue!

2. MAIN FRAMEWORK OF SPA : React Native

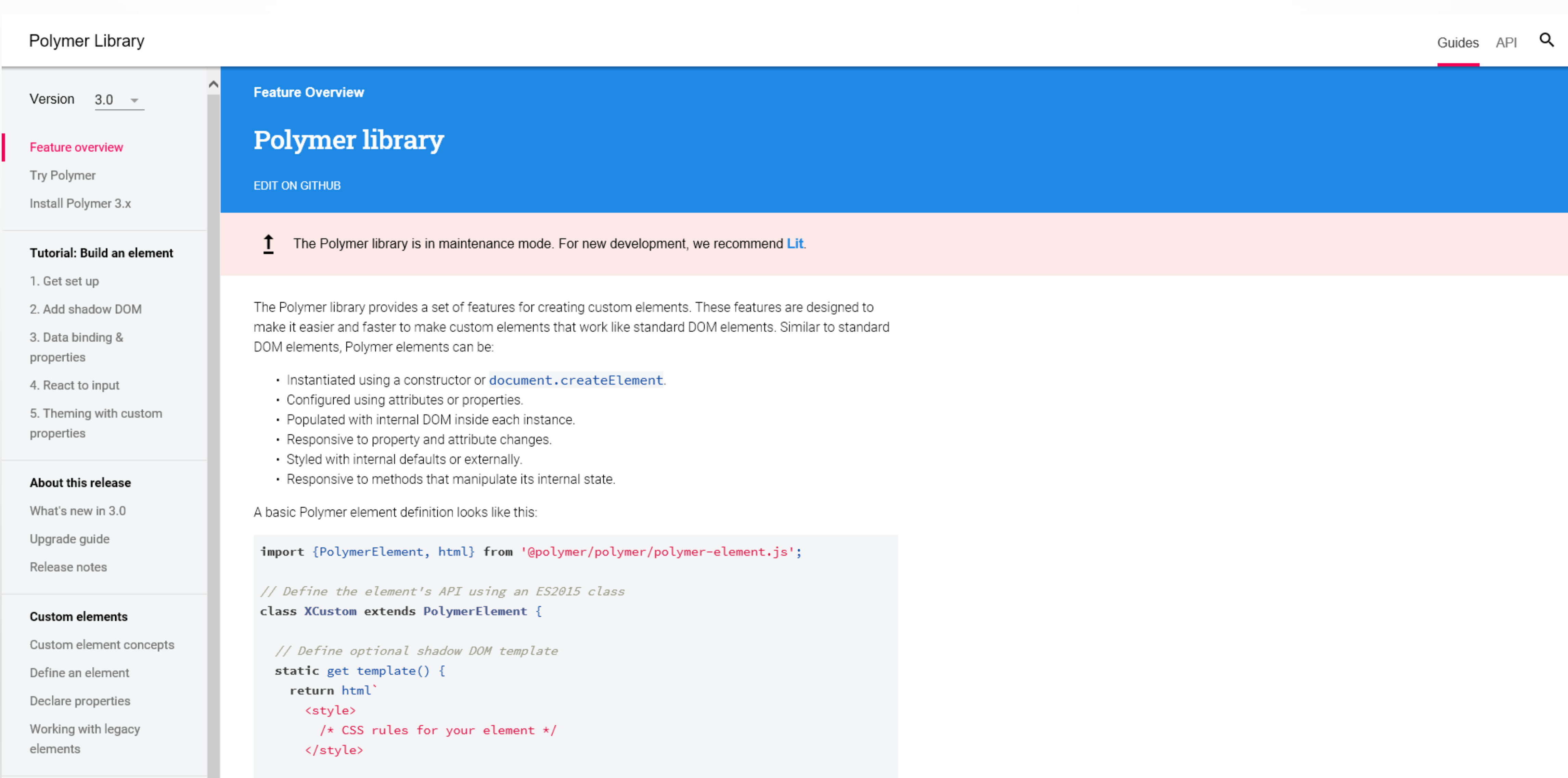


```
import React from 'react';
import {Text, View} from 'react-native';
import {Header} from './Header';
import {heading} from './Typography';

const WelcomeScreen = () => (
  <View>
    <Header title="Welcome to React Native"/>
    <Text style={heading}>Step One</Text>
    <Text>
      Edit App.js to change this screen and turn it
      into your app.
    </Text>
    <Text style={heading}>See Your Changes</Text>
    <Text>
      Press Cmd + R inside the simulator to reload
      your app’s code.
```

Create native apps for Android and iOS using React

React Native combines the best parts of native development with React, a best-in-class JavaScript

3. WEB COMPONENT FRAMEWORK : Polymer



Polymer library

The Polymer library is in maintenance mode. For new development, we recommend Lit.

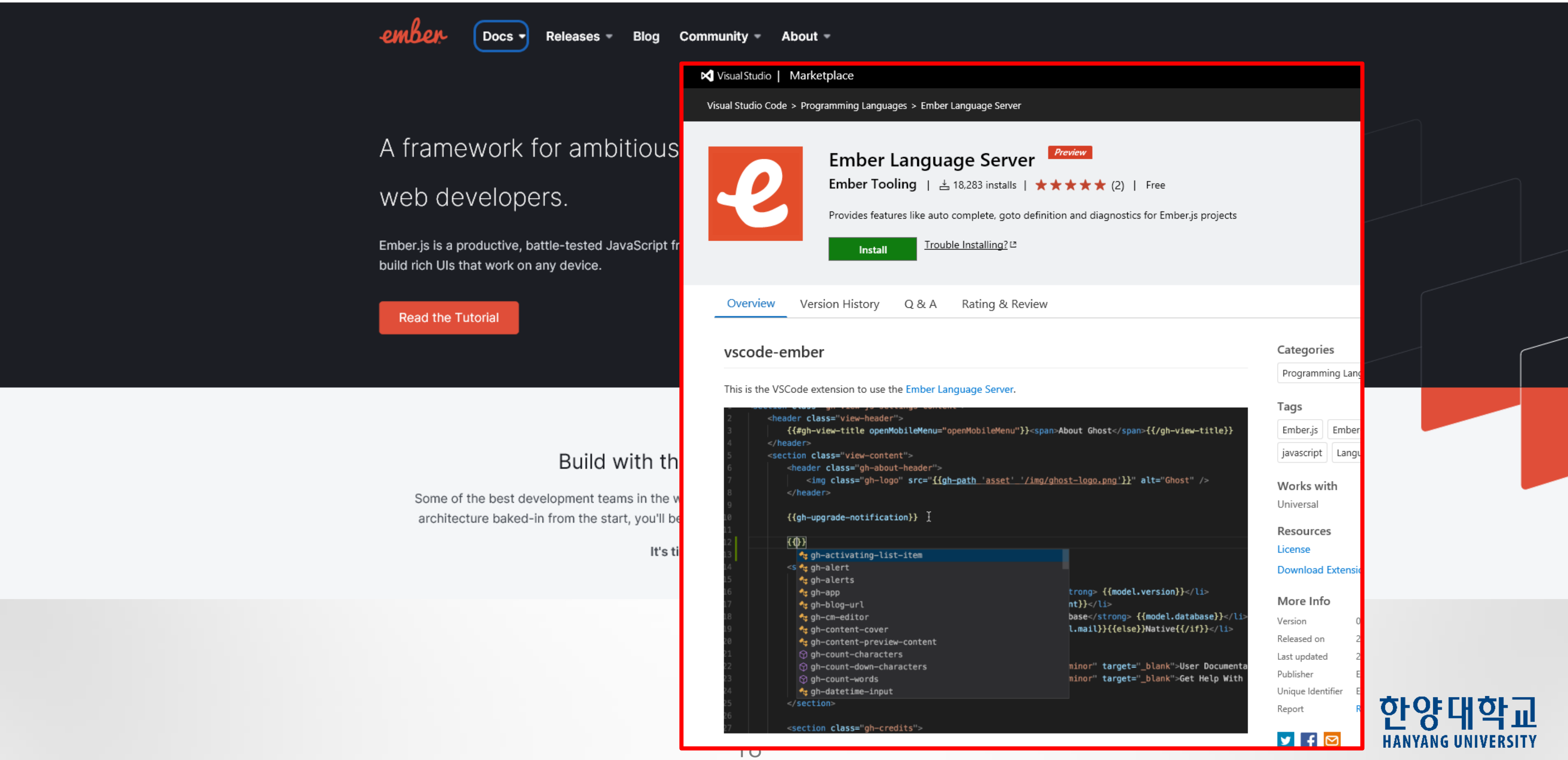
The Polymer library provides a set of features for creating custom elements. These features are designed to make it easier and faster to make custom elements that work like standard DOM elements. Similar to standard DOM elements, Polymer elements can be:

- Instantiated using a constructor or `document.createElement`.
- Configured using attributes or properties.
- Populated with internal DOM inside each instance.
- Responsive to property and attribute changes.
- Styled with internal defaults or externally.
- Responsive to methods that manipulate its internal state.

A basic Polymer element definition looks like this:

```
import {PolymerElement, html} from '@polymer/polymer/polymer-element.js';

// Define the element's API using an ES2015 class
class XCustom extends PolymerElement {

  // Define optional shadow DOM template
  static get template() {
    return html`
      <style>
        /* CSS rules for your element */
      </style>
```

4. FULL STACK FRAMEWORK : Ember.js



A framework for ambitious

web developers.

Ember.js is a productive, battle-tested JavaScript fr
build rich UIs that work on any device.

Build with th

Some of the best development teams in the w
architecture baked-in from the start, you'll be

It's ti

Visual Studio Code > Programming Languages > Ember Language Server

Ember Language Server Preview

Ember Tooling | 18,283 installs | ★★★★★ (2) | Free

Provides features like auto complete, goto definition and diagnostics for Ember.js projects

vscode-ember

This is the VSCode extension to use the Ember Language Server.

```
    <header class="view-header">
        {{#gh-view-title openMobileMenu="openMobileMenu"}}<span>About Ghost</span>{{/gh-view-title}}
    </header>
    <section class="view-content">
        <header class="gh-about-header">
            <img class="gh-logo" src="{{gh-path 'asset' '/img/ghost-logo.png'}}" alt="Ghost" />
        </header>

        {{gh-upgrade-notification}}

        {{{}}
          gh-activating-list-item
        <s  gh-alert
            gh-alerts
            gh-app                                              trong> {{model.version}}</li>
            gh-blog-url                                         nt}}</li>
            gh-cm-editor                                        base</strong> {{model.database}}</li>
            gh-content-cover                                    l.mail}}{{else}}Native{{/if}}</li>
            gh-content-preview-content
            gh-count-characters
            gh-count-down-characters                            minor" target="_blank">User Documenta
            gh-count-words                                      minor" target="_blank">Get Help With
            gh-datetime-input
        </section>

        <section class="gh-credits">
```

Categories

Programming Lang

Tags

Ember.js Ember
javascript Langu

Works with

Universal

Resources

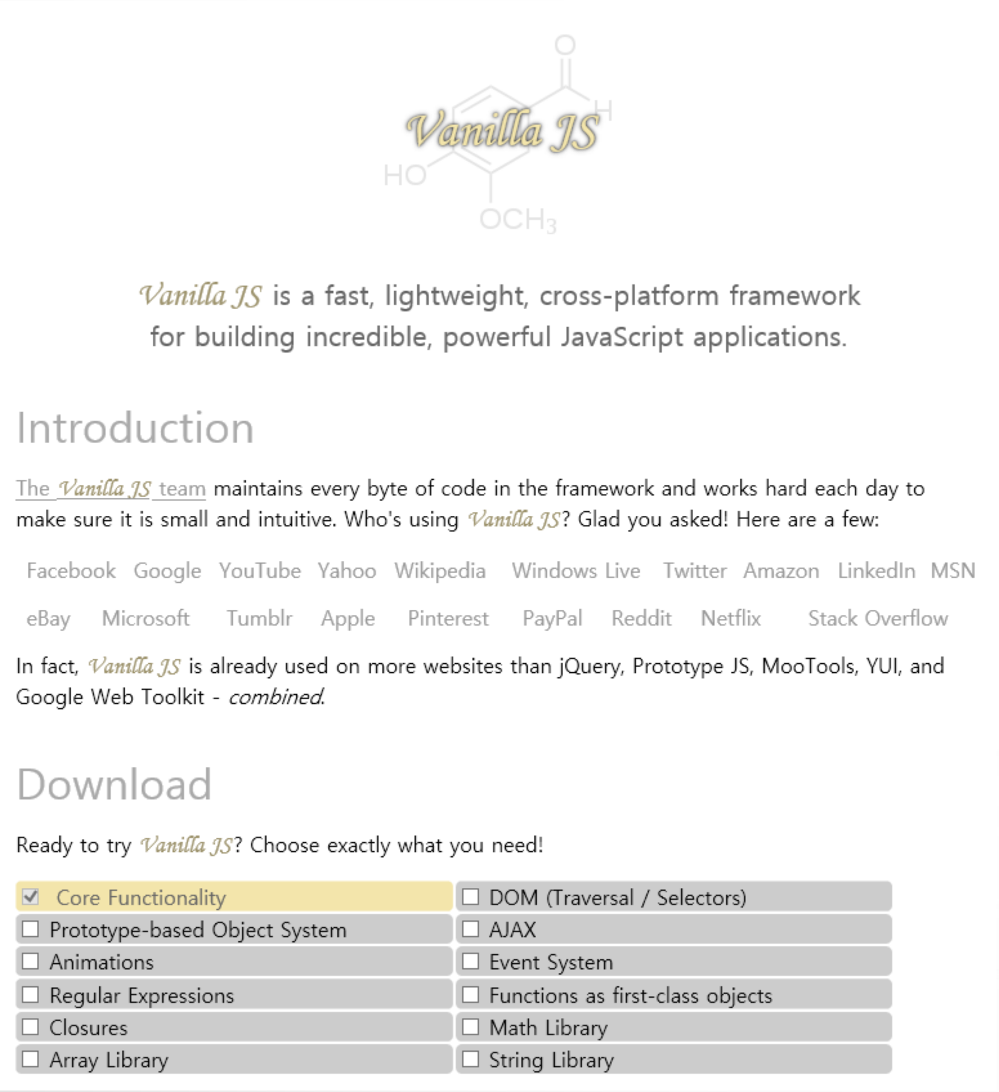
License

Download Extensio

More Info

Version
Released on
Last updated
Publisher
Unique Identifier
Report

4. FULL STACK FRAMEWORK : Vanilla.js



Vanilla JS is a fast, lightweight, cross-platform framework for building incredible, powerful JavaScript applications.

Introduction

The Vanilla JS team maintains every byte of code in the framework and works hard each day to make sure it is small and intuitive. Who's using Vanilla JS? Glad you asked! Here are a few:

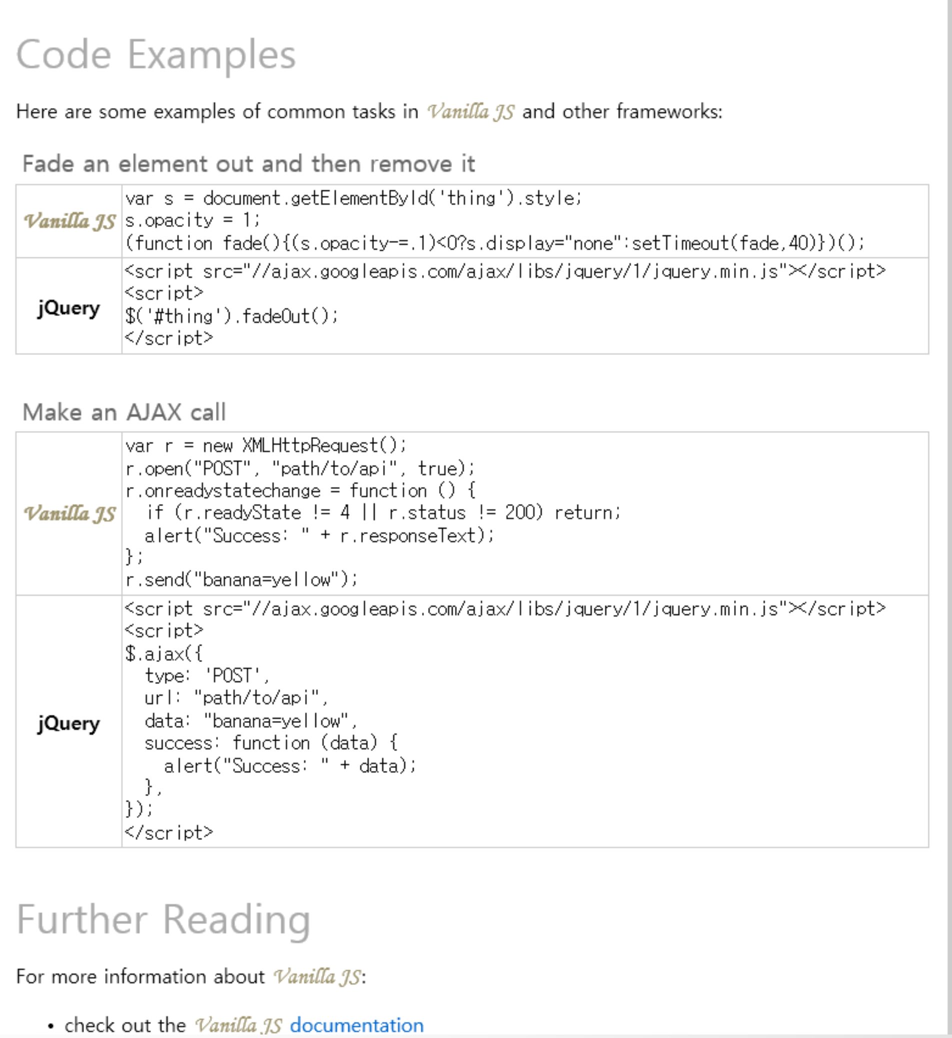
Facebook Google YouTube Yahoo Wikipedia Windows Live Twitter Amazon LinkedIn MSN eBay Microsoft Tumblr Apple Pinterest PayPal Reddit Netflix Stack Overflow

In fact, Vanilla JS is already used on more websites than jQuery, Prototype JS, MooTools, YUI, and Google Web Toolkit - *combined.*

Download

Ready to try Vanilla JS? Choose exactly what you need!

☑ Core Functionality
☐ Prototype-based Object System
☐ Animations
☐ Regular Expressions
☐ Closures
☐ Array Library
☐ DOM (Traversal / Selectors)
☐ AJAX
☐ Event System
☐ Functions as first-class objects
☐ Math Library
☐ String Library



Code Examples

Here are some examples of common tasks in Vanilla JS and other frameworks:

Fade an element out and then remove it

Vanilla JS

```
var s = document.getElementById('thing').style;
s.opacity = 1;
(function fade(){(s.opacity-=.1)<0?s.display="none":setTimeout(fade,40)})();
```

jQuery

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"></script>
<script>
$('#thing').fadeOut();
</script>
```

Make an AJAX call

Vanilla JS

```
var r = new XMLHttpRequest();
r.open("POST", "path/to/api", true);
r.onreadystatechange = function () {
  if (r.readyState != 4 || r.status != 200) return;
  alert("Success: " + r.responseText);
};
r.send("banana=yellow");
```

jQuery

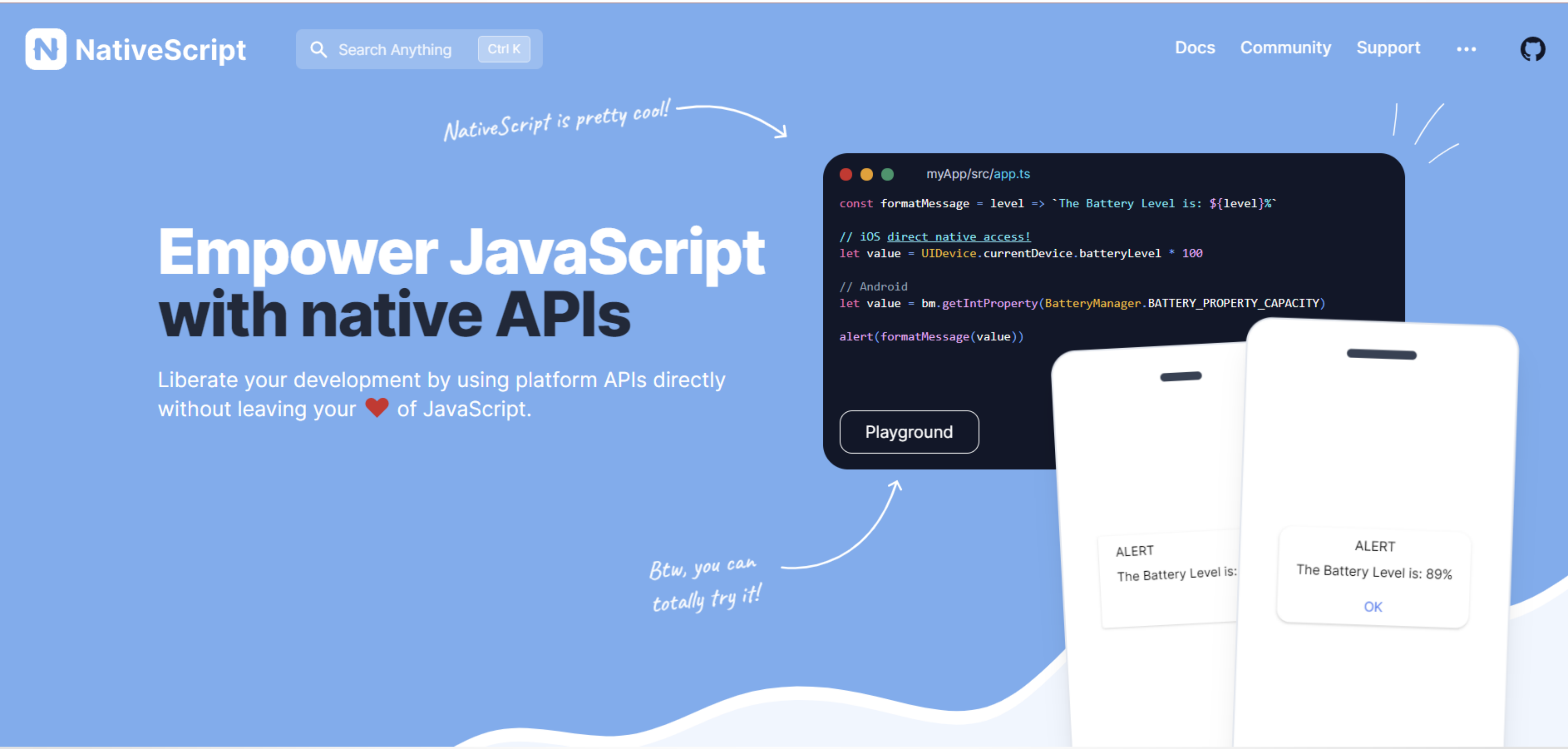
```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"></script>
<script>
$.ajax({
  type: 'POST',
  url: "path/to/api",
  data: "banana=yellow",
  success: function (data) {
    alert("Success: " + data);
  },
});
</script>
```

Further Reading

For more information about Vanilla JS:

- check out the Vanilla JS documentation

5. MOBILE NATIVE FRAMEWORK : NativeScript



NativeScript is pretty cool!

Empower JavaScript with native APIs

Liberate your development by using platform APIs directly without leaving your ❤ of JavaScript.

myApp/src/app.ts

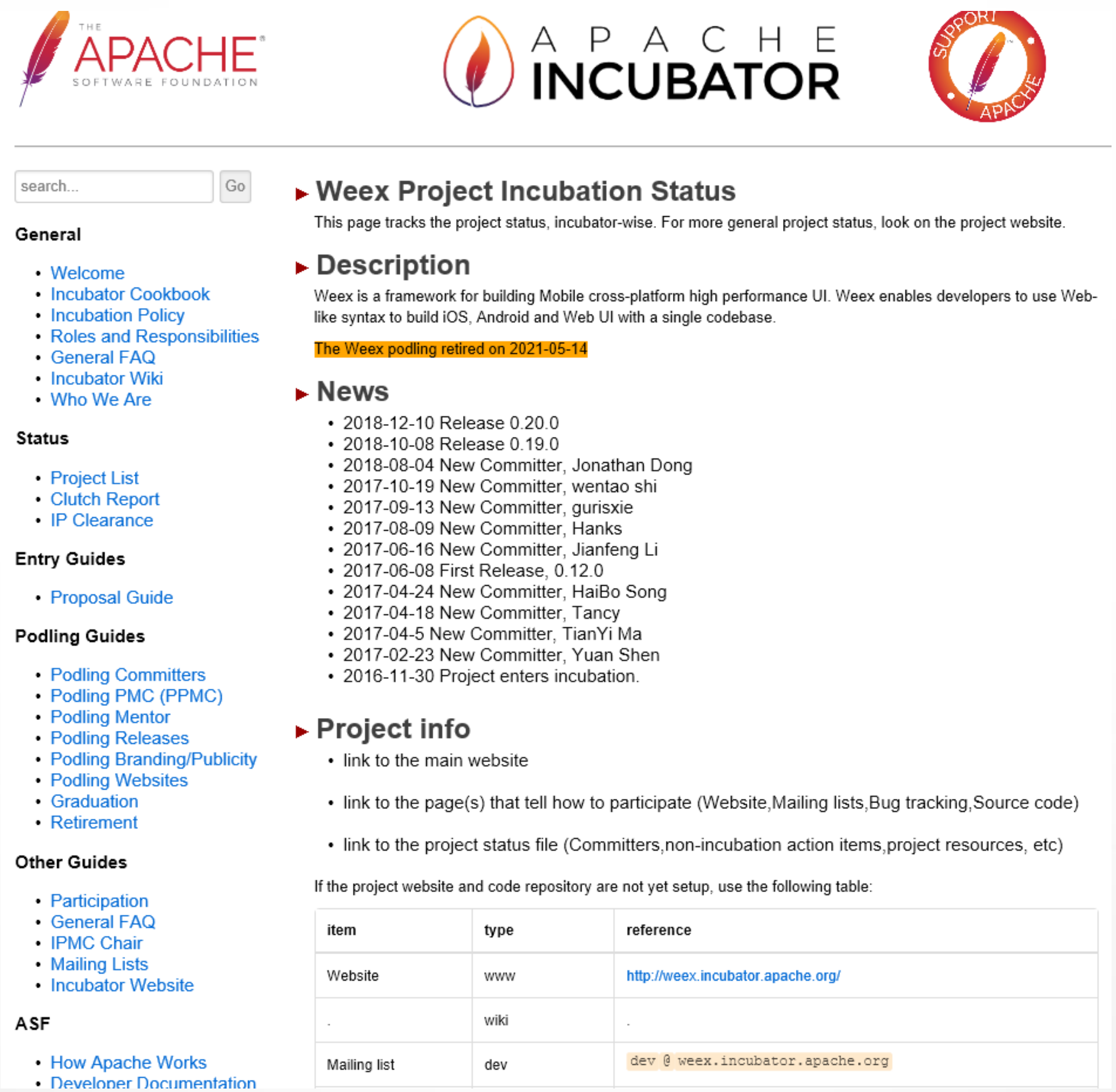
```
const formatMessage = level => `The Battery Level is: ${level}%`

// iOS direct native access!
let value = UIDevice.currentDevice.batteryLevel * 100

// Android
let value = bm.getIntProperty(BatteryManager.BATTERY_PROPERTY_CAPACITY)

alert(formatMessage(value))
```

Btw, you can totally try it!

ALERT
The Battery Level is:

ALERT
The Battery Level is: 89%

5. MOBILE NATIVE FRAMEWORK : Weex



General

- Welcome
- Incubator Cookbook
- Incubation Policy
- Roles and Responsibilities
- General FAQ
- Incubator Wiki
- Who We Are

Status

- Project List
- Clutch Report
- IP Clearance

Entry Guides

- Proposal Guide

Podling Guides

- Podling Committers
- Podling PMC (PPMC)
- Podling Mentor
- Podling Releases
- Podling Branding/Publicity
- Podling Websites
- Graduation
- Retirement

Other Guides

- Participation
- General FAQ
- IPMC Chair
- Mailing Lists
- Incubator Website

ASF

- How Apache Works
- Developer Documentation

▸ Weex Project Incubation Status

This page tracks the project status, incubator-wise. For more general project status, look on the project website.

▸ Description

Weex is a framework for building Mobile cross-platform high performance UI. Weex enables developers to use Web-like syntax to build iOS, Android and Web UI with a single codebase.

The Weex podling retired on 2021-05-14

▸ News

- 2018-12-10 Release 0.20.0
- 2018-10-08 Release 0.19.0
- 2018-08-04 New Committer, Jonathan Dong
- 2017-10-19 New Committer, wentao shi
- 2017-09-13 New Committer, gurisxie
- 2017-08-09 New Committer, Hanks
- 2017-06-16 New Committer, Jianfeng Li
- 2017-06-08 First Release, 0.12.0
- 2017-04-24 New Committer, HaiBo Song
- 2017-04-18 New Committer, Tancy
- 2017-04-5 New Committer, TianYi Ma
- 2017-02-23 New Committer, Yuan Shen
- 2016-11-30 Project enters incubation.

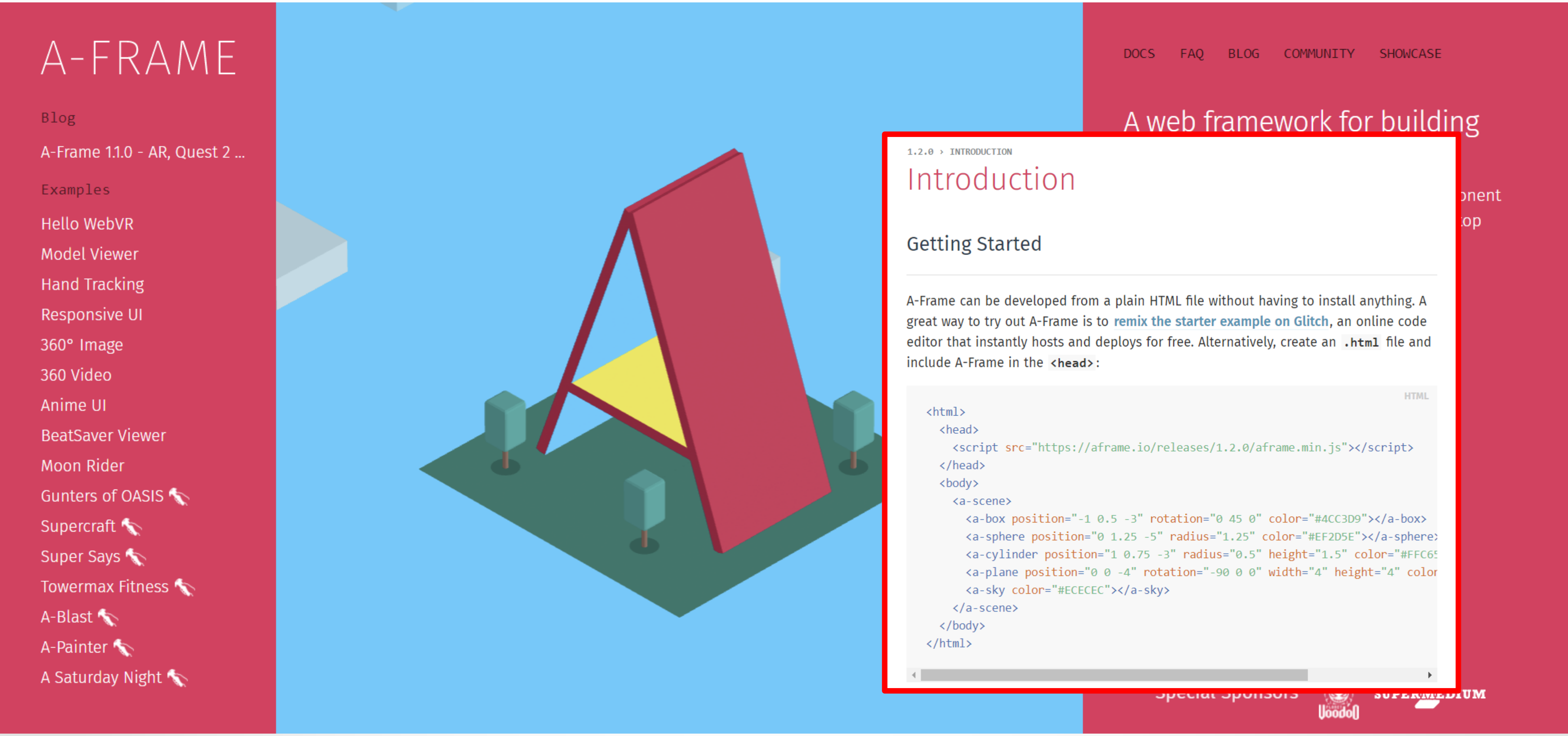
▸ Project info

- link to the main website
- link to the page(s) that tell how to participate (Website,Mailing lists,Bug tracking,Source code)
- link to the project status file (Committers,non-incubation action items,project resources, etc)

If the project website and code repository are not yet setup, use the following table:

item	type	reference
Website	www	http://weex.incubator.apache.org/
.	wiki	.
Mailing list	dev	`dev @ weex.incubator.apache.org`

6. WebVR FRAMEWORK : A-FRAME



1.2.0 › INTRODUCTION

Introduction

Getting Started

A-Frame can be developed from a plain HTML file without having to install anything. A great way to try out A-Frame is to remix the starter example on Glitch, an online code editor that instantly hosts and deploys for free. Alternatively, create an `.html` file and include A-Frame in the `<head>`:

HTML

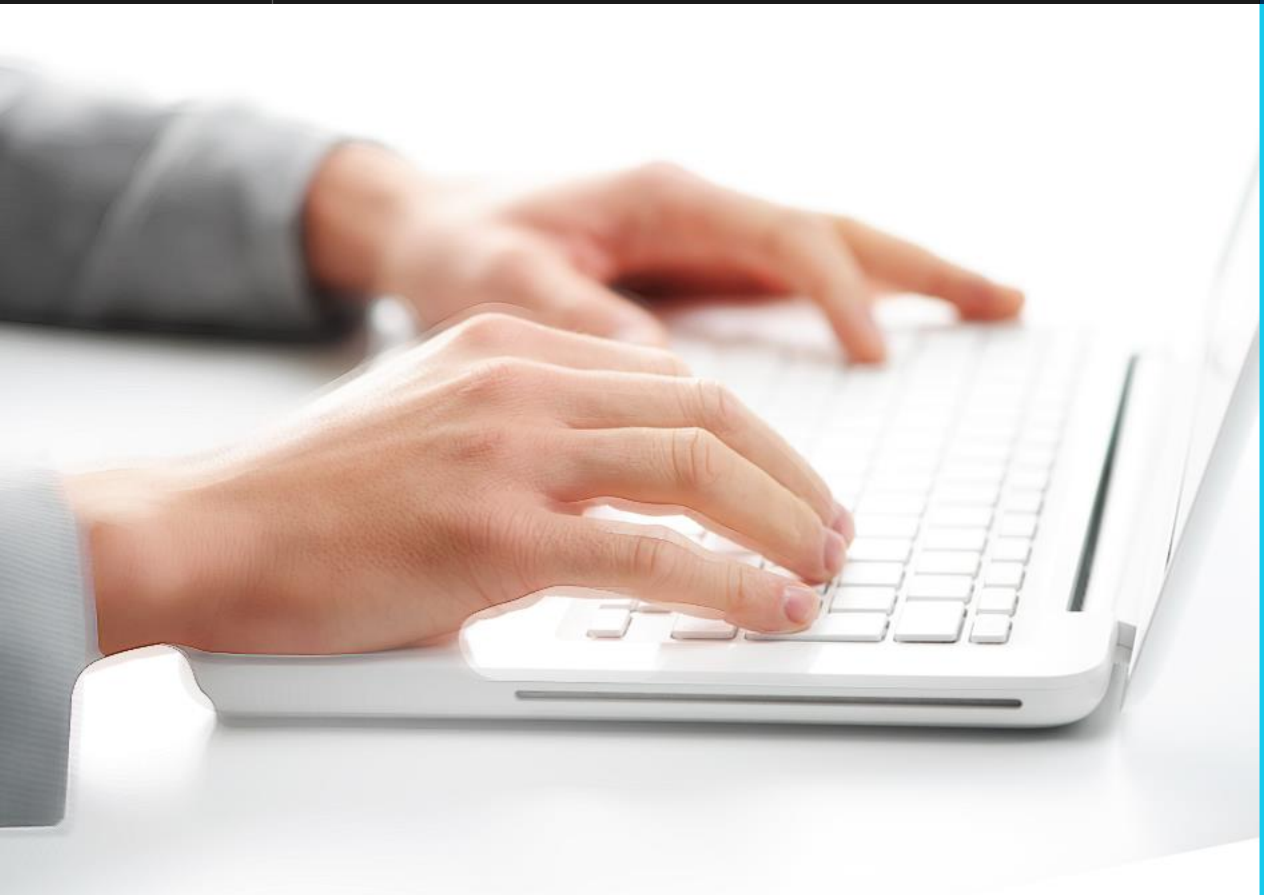
```html
<html>
  <head>
    <script src="https://aframe.io/releases/1.2.0/aframe.min.js"></script>
  </head>
  <body>
    <a-scene>
      <a-box position="-1 0.5 -3" rotation="0 45 0" color="#4CC3D9"></a-box>
      <a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"></a-sphere
      <a-cylinder position="1 0.75 -3" radius="0.5" height="1.5" color="#FFC65
      <a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4" color
      <a-sky color="#ECECEC"></a-sky>
    </a-scene>
  </body>
</html>
```

III

Progressive Web App

PWA

HANYANG UNIVERSITY



1. Front-end implementation technology trend
2. What is the PWA?
3. 6 main techniques of PWA

1. FRONT-END IMPLEMENTATION TECHNOLOGY TREND

Routing and Rendering

- Routing is the process through which the user is navigated to different pages on a website. Rendering is the process of putting those pages on the UI. Every time you request a route to a particular page, you are also rendering that page, but not every render is an outcome of a route. Their close association helps in speculating their effect on the efficiency and speed of an application

1. FRONT-END IMPLEMENTATION TECHNOLOGY TREND

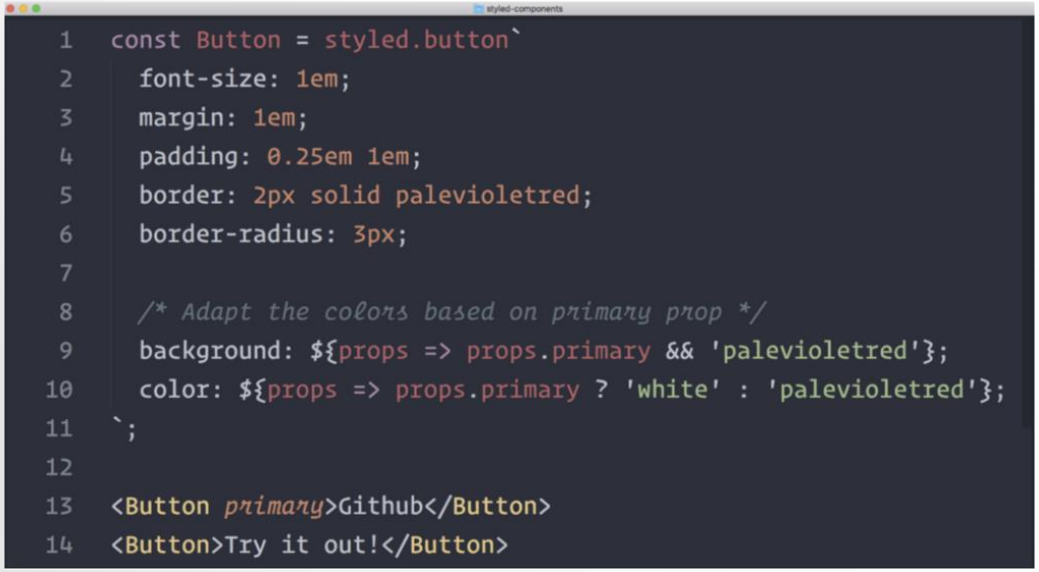
Client-Side Routing

- Client-side routing is handled solely by JavaScript on the page. Whenever a user clicks on a link, the URL bar changes and a different view is rendered on the page. This view could be anything—JSX or HTML. Single-page applications give a smooth sense of navigation as they don't refresh the whole page when a route is performed. Even when a request is made to the server to fetch data, it only seems as if static HTML pages are rendered on the frontend. Thus, single-page applications are direct beneficiaries of client-side routing, and this is one major reason for their growing popularity and delivery of great user experience.

1. FRONT-END IMPLEMENTATION TECHNOLOGY TREND

CSS in JS

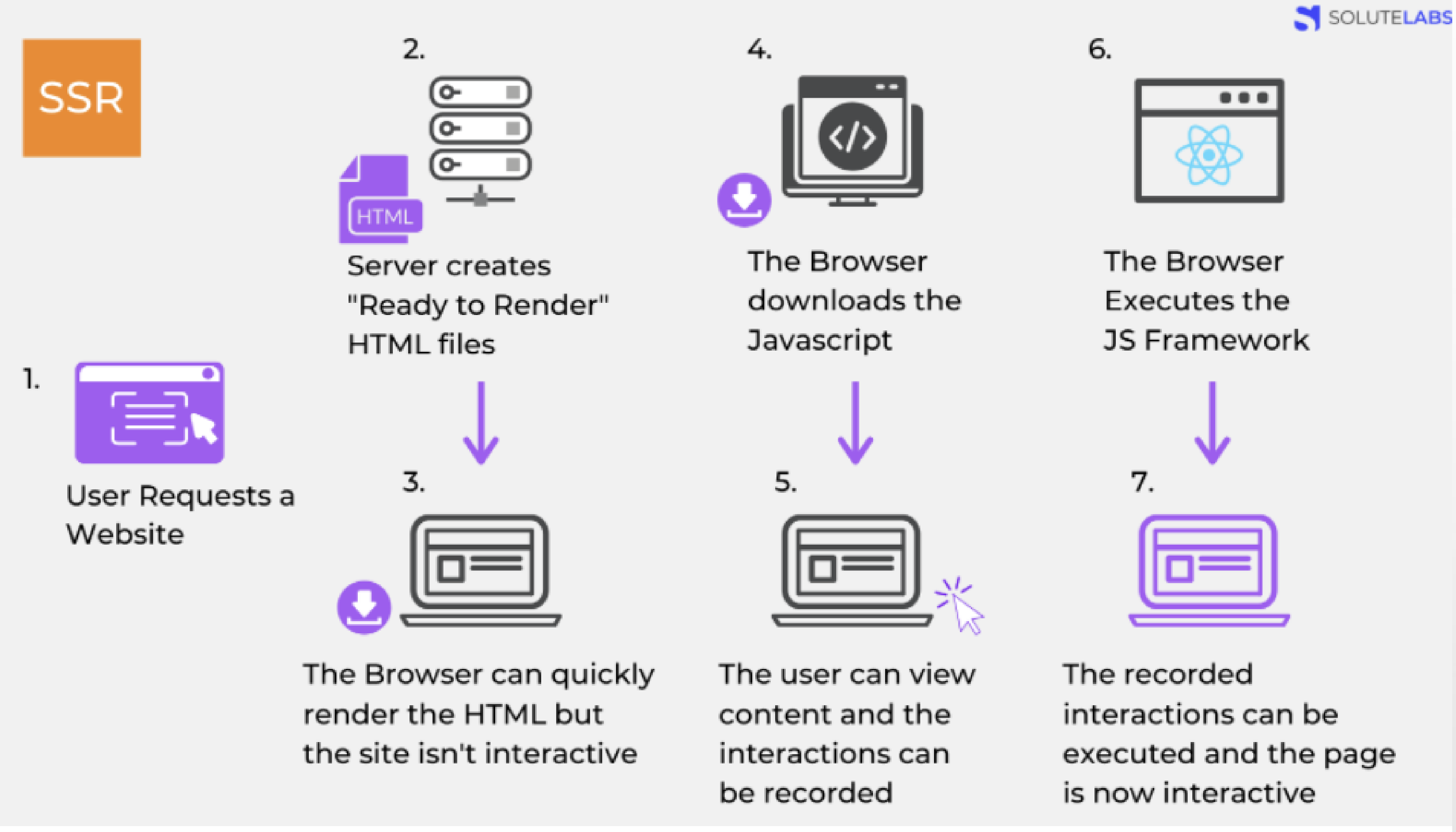
- CSS-in-JS is a styling technique where JavaScript is used to style components. When this JavaScript is parsed, CSS is generated (usually as a <style> element) and attached into the DOM.
  It allows to abstract CSS to the component level itself, using JavaScript to describe styles in a declarative and maintainable way.



```
const Button = styled.button`
  font-size: 1em;
  margin: 1em;
  padding: 0.25em 1em;
  border: 2px solid palevioletred;
  border-radius: 3px;

  /* Adapt the colors based on primary prop */
  background: ${props => props.primary && 'palevioletred'};
  color: ${props => props.primary ? 'white' : 'palevioletred'};
`;

<Button primary>Github</Button>
<Button>Try it out!</Button>
```

1. FRONT-END IMPLEMENTATION TECHNOLOGY TREND

Server Side Rendering



1. User Requests a Website

2. Server creates "Ready to Render" HTML files

3. The Browser can quickly render the HTML but the site isn't interactive

4. The Browser downloads the Javascript

5. The user can view content and the interactions can be recorded

6. The Browser Executes the JS Framework

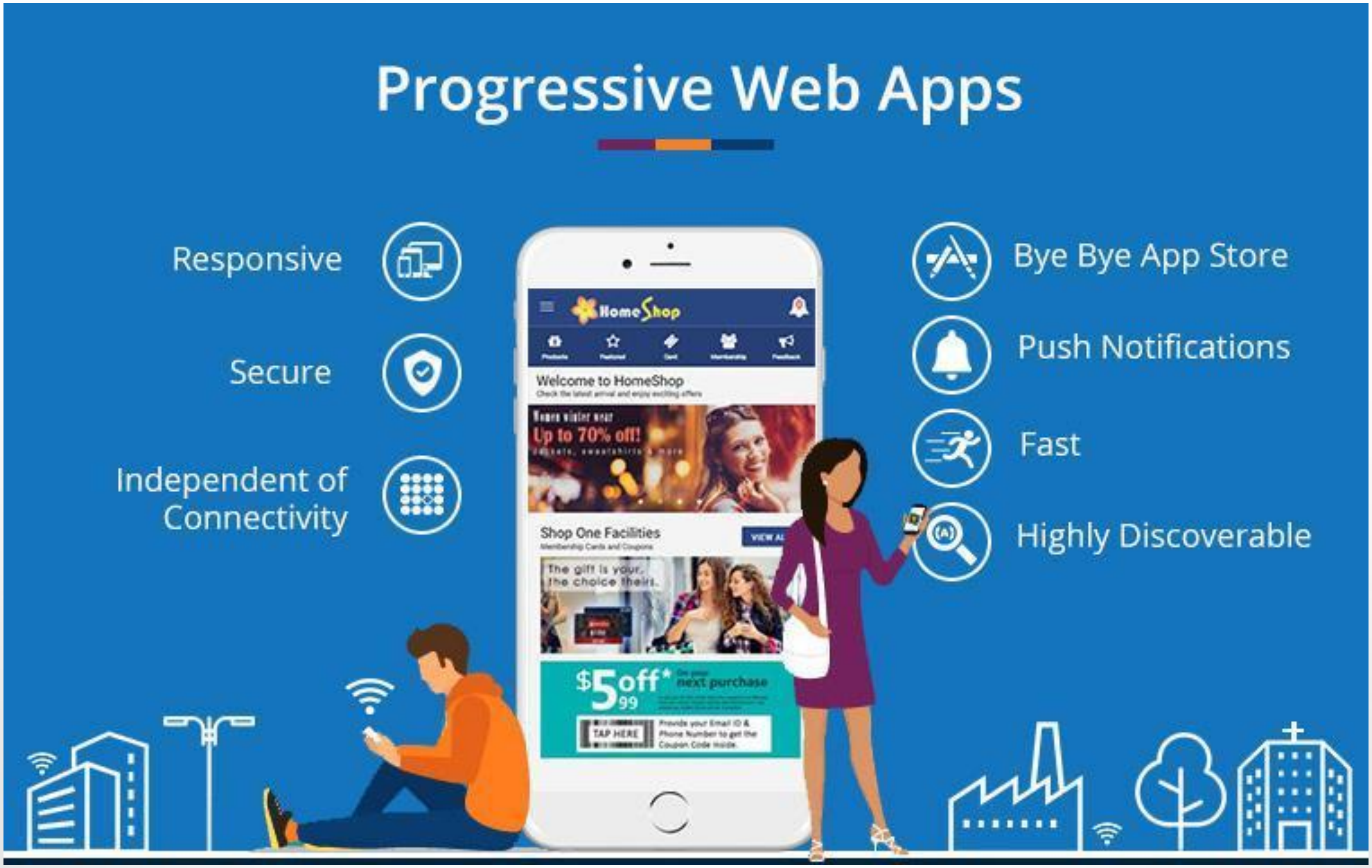
7. The recorded interactions can be executed and the page is now interactive

1. FRONT-END IMPLEMENTATION TECHNOLOGY TREND

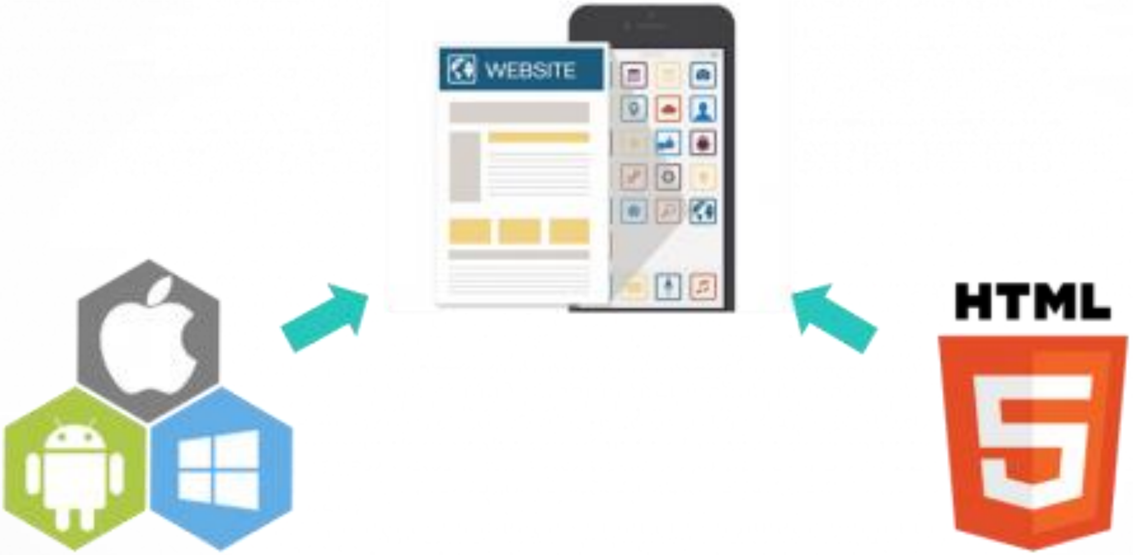
Virtual DOM

- Virtual dom (vdom) is a programming concept that stores ui's ideal or "virtual" representation in memory and synchronizes it with "real" dom by libraries such as reactdom. This process is called readjustment.
- This approach enables the declarative api of react. Tell the react about the desired ui state so that dom matches that state. This method abstracts the attribute manipulation, event processing, and manual dom updates that should be used to build apps.

2. WHAT IS THE PWA?



2. WHAT IS THE PWA?



Native Apps

- → Designed in a native language for a single operating system
- → Fully customisable
- → Takes advantage of the device hardware
- → High performance

Web Apps

- → Built to be used in a common browser
- → Delivered in real time
- → Built using standard HTML, CSS and JavaScript
- → Limited access to a device's features
- → Network performance can affect users experience

2. WHAT IS THE PWA?

For a PWA to work there a two elements which are needed, The Service Worker and The Web App Manifest.

- The Service Worker is vital to allow a PWA to function offline. What it is, is a JavaScript background task that replaces the traditional web cache by using a cache API. The cache API wakes up when needed, listening to different network requests saving the static resources from the visited sites. The Service Worker saves these resources by redirecting the web page from going to the server but instead go to The Service Worker.

2. WHAT IS THE PWA?

For a PWA to work there a two elements which are needed, The Service Worker and The Web App Manifest.

- The second element which a PWA needs is the Web App Manifest, this is a JSON script, making it possible for the Web App to have a home screen icon, theme, splash screen and the ability for the web app to run in full-screen mode and as a standalone application.

3. 6 MAIN TECHNIQUES OF PWA

- PWA's heart that runs 24 hours. "Service Worker".
- PWA's passport, "Webapp Manifest".
- "HTTPS" with enhanced security.
- "Push notification" that approaches users first.
- "Additional function on the home screen" that allows you to connect with just one touch.
- "Web API" that doesn't make you jealous of the native app.
- The reason Starbucks switched to PWA.

Close

감사합니다

질의응답