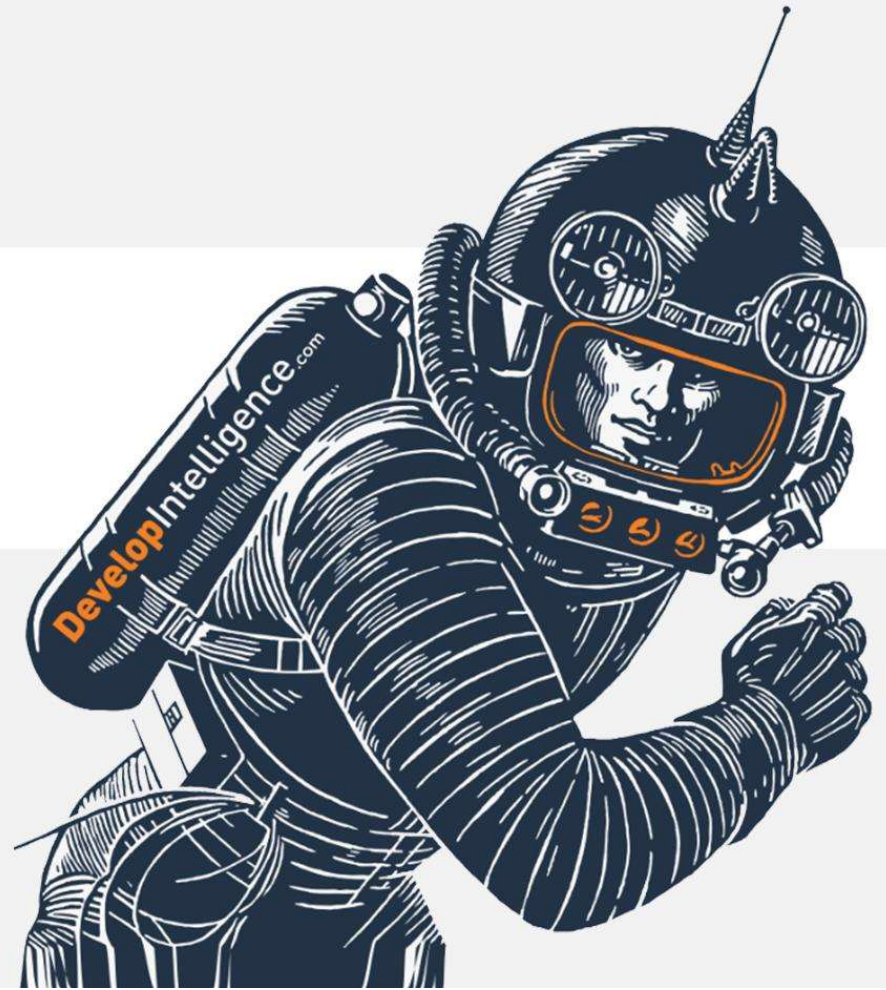


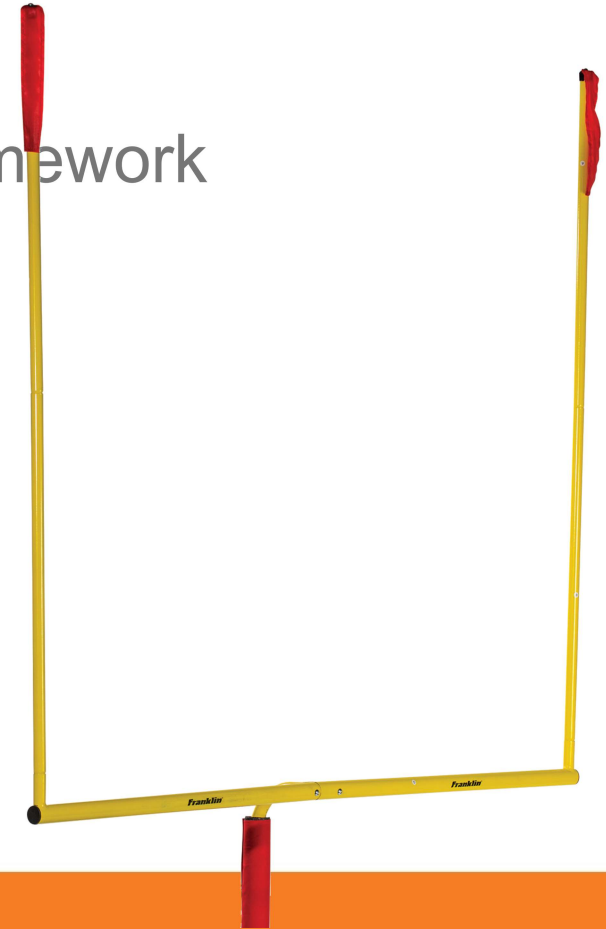
Overview





Goals

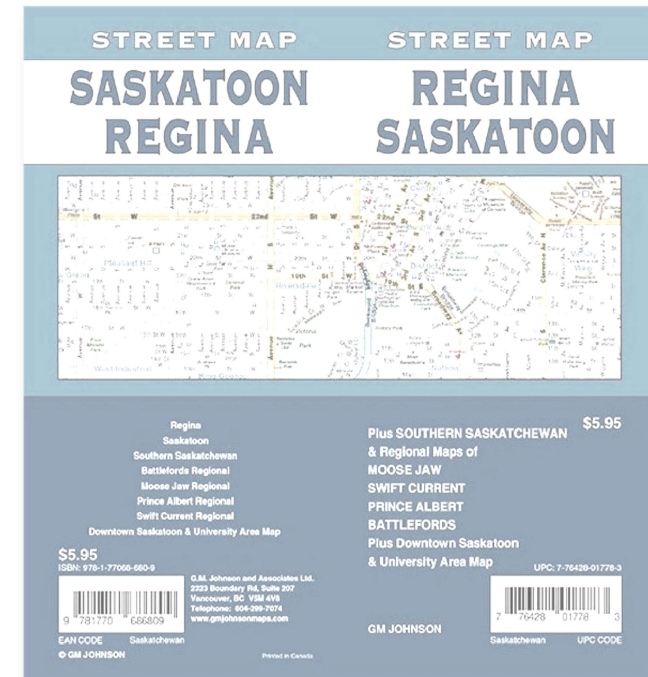
1. Compare React v. Angular
2. Explain the advantage of using a React framework
3. Describe Vite

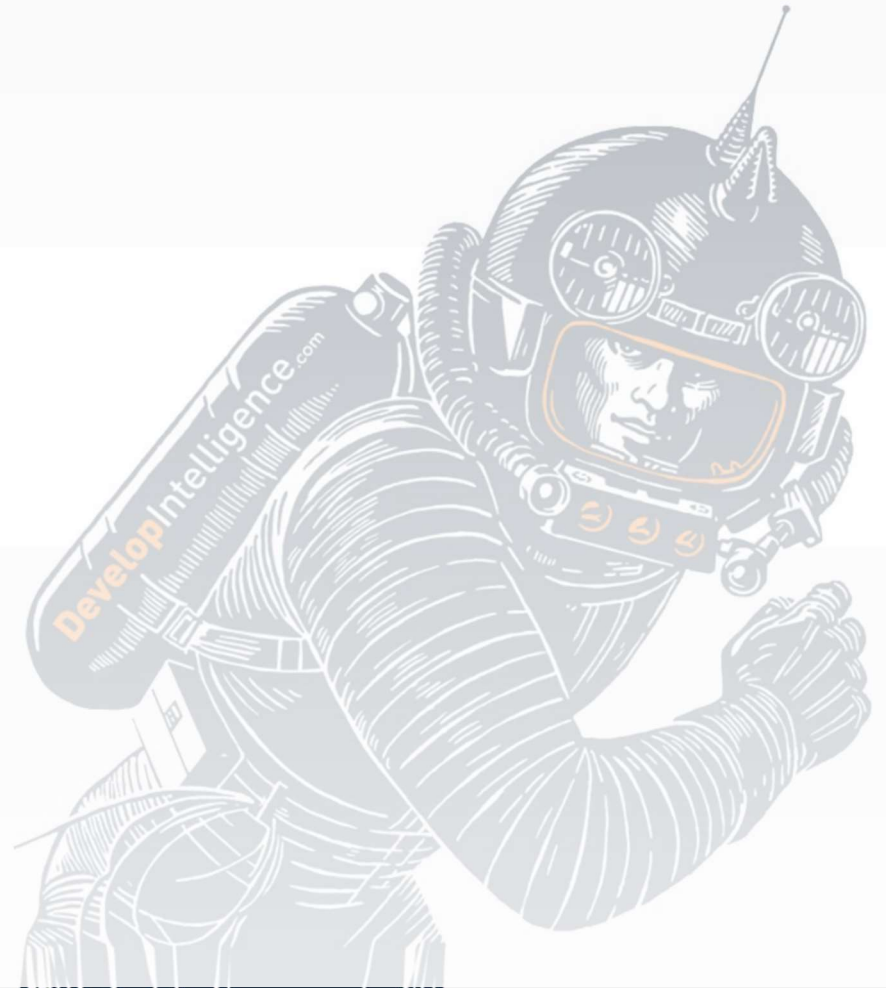




Roadmap

1. What's React?
2. Creating a React App
3. Project Anatomy





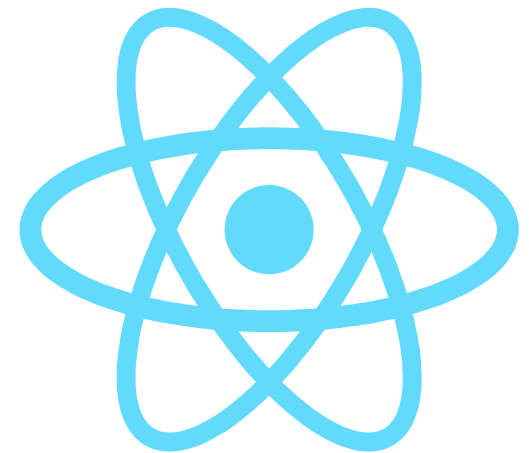
What's React?





What's **React**?

- **Library** for creating user interfaces
- Created at Facebook in 2011
- Open source
- Hugely popular: **40% of new web apps**
 - (Angular is ~18% and falling)





Compared to Angular

Angular

Framework

Opinionated

Batteries included

Templates

2-way binding

Component-based

React

Library

Laissez-faire

A la carte

JSX

Unidirectional data flow

Same!





React is a Library

- Library for interactive UIs
- View-only
 - **Not** MVC
 - **Not** bundling
 - **Not** transpiling



React is an **Architecture**

- React software outputs a dynamic tree
 - Of DOM nodes
 - Of iOS controls
 - Etc...
- A ***renderer*** mediates between React and its in-memory tree
- ***Reconciliation*** makes the host tree match the in-memory tree



React is an **Ecosystem**

- React is 'just a library'
- Most projects rely on a vast ecosystem for other things



Things You Might Need

- **Dependency management** - npm, yarn
- **Bundler** - Webpack, Rollup
- **Compiler** -- jsx, typescript, modern javascript
- **SSR** - Next.js
- **Development tools** - Hot reload, etc.
- **State management** - Zustand, Redux
- **Styling** - CSS / CSS-in-js
- **Routing** - React Router
- **HTTP** - React Query

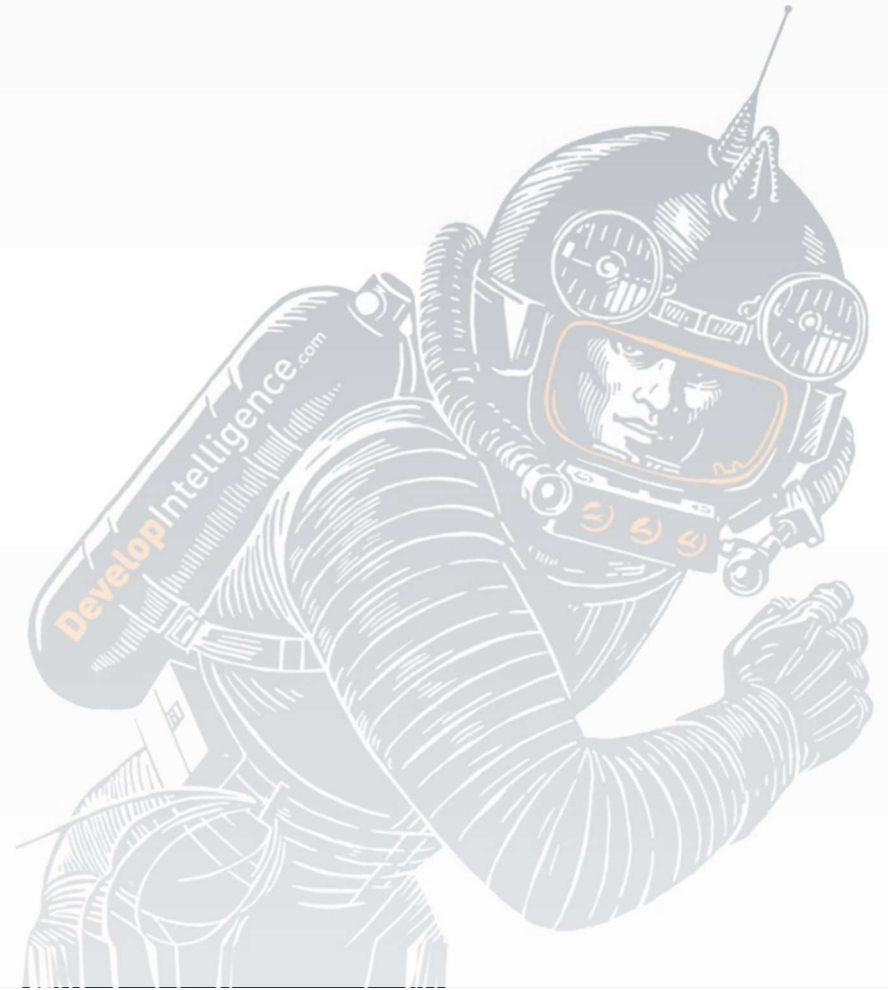


Frameworks

- You can configure everything a-la-carte
 - Configuration is hard
 - Making choices is harder
 - Upgrades are the worst
- Or you can use a more opinionated framework
 - Next.js
 - Remix
 - Astro



NEXT.js



Creating a React App





Option 0: Framework-Free

- All you really need--
 - **React** - For the core framework
 - **React-dom** - For web stuff
 - **Babel** - For JSX, and ES6+
- **Pros** No build process, Good for learning
- **Cons** No build process, no web server, not industrial strength



Demo: Framework-Free

```
1 <html>
2   <head>
3     <script src="https://unpkg.com/react@18/umd/react.development.js" crossorigin="anonymous"></script>
4     <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js" crossorigin="anonymous"></script>
5     <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
6   </head>
7   <body>
8     <div id="root">
9       <script type="text/babel">
10        const App = () => <h1>Hello World!</h1>;
11        ReactDOM.createRoot(document.getElementById('root'))
12          .render(<App />);
13      </script>
14    </div>
15  </body>
16 </html>
```




Option 0.5: create-react-app

- Go-to project scaffolding for a long time
- There are better options now
- **Pros:** Familiar to lots of React developers
- **Cons:** DX, Not really evolving
- Magic words:

```
1 | npx create-react-app some-spa
```



Option 1: Vite



- Modern front-end tooling
- Major parts:
 - Dev server with Hot Module Replacement
 - Compiles with esbuild
 - Builds with Rollup
 - Provides scaffolding for various project types
- Magic words:

```
1 npm create vite@latest
```



Vite: Pros and Cons

Pro

- Unopinionated (Relatively)
 - Good for learning plain React
 - Good for building pure old-school SPAs
- Great DX: Fast HMR, small bundles

Con

- Unopinionated (Relatively)
 - Still lots of configuration todo before production
- On your own for code splitting, routing, SSR



Dan on Vite



danabra.mov 

@dan_abramov

“why not recommend vite” we do recommend it, for existing projects adding a bit of react to their pages. if you’re creating a new application today with more than a single route, a template that produces an empty HTML + monolith chunk of JS is years behind

Harsh!



Consensus: Use a Framework



Andrew Clark

@acd lite · [Follow](#)



If you use React, you should be using a React framework. If your existing app doesn't use a framework, you should incrementally migrate to one. If you're creating a new React project, you should use a framework from the beginning.

12:52 PM · Jan 23, 2023



1.9K



Reply



Share

[Read 113 replies](#)



Lab: Create a React App

1. Use Vite to scaffold a new project

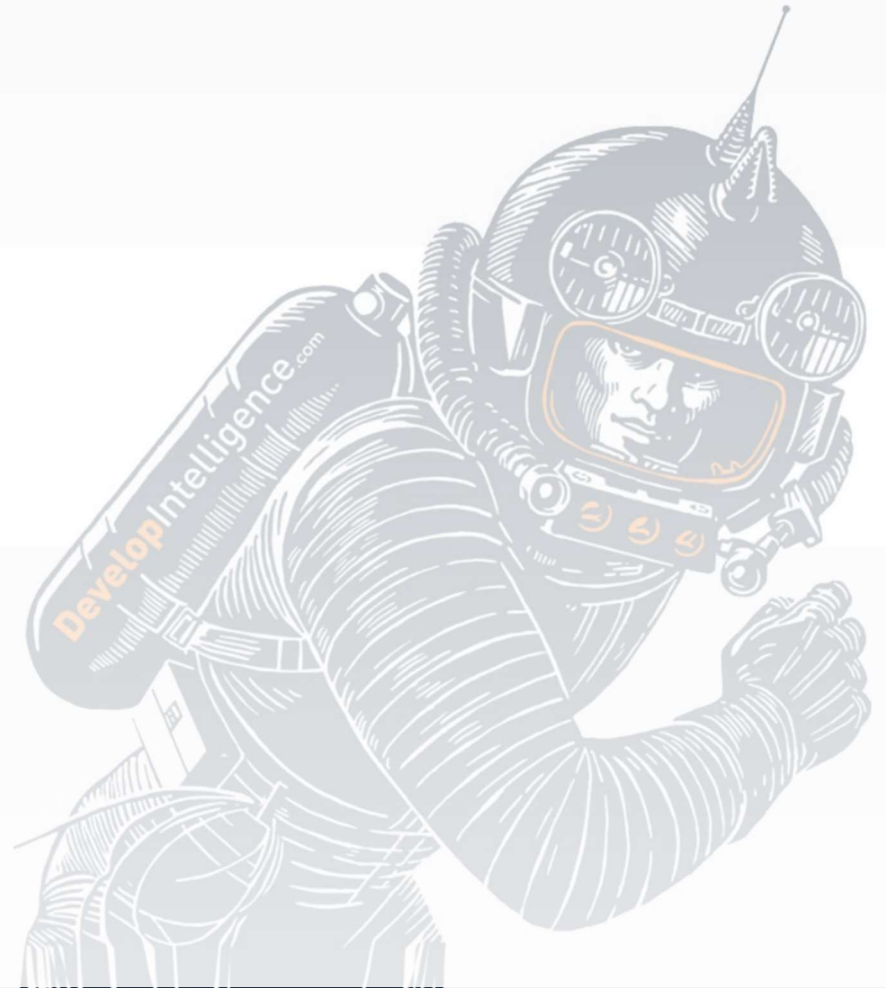
```
1 npm create vite@latest some-throwaway-project
2 cd ./some-throwaway-project
3 npm install
4 code .
```

2. Examine the project

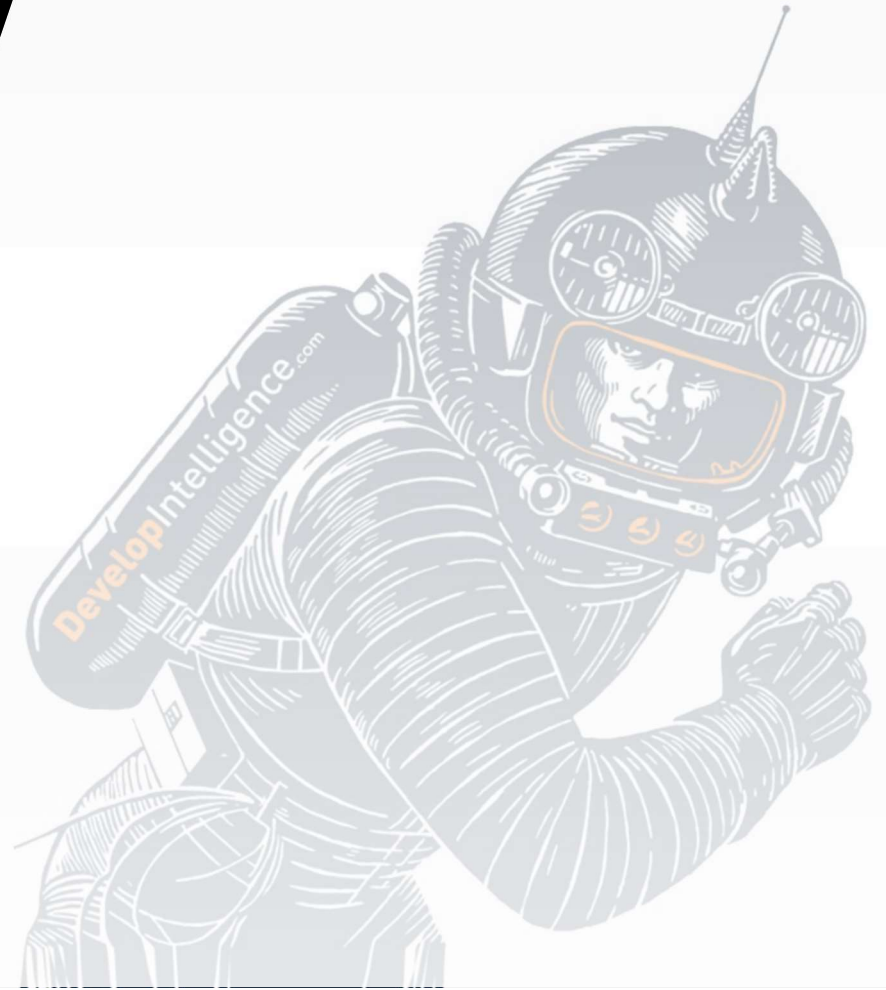
3. Fire up the dev server

```
1 npm run dev
```





Project Anatomy





Folder Structure

- React is flexible, so projects are different
- Usually a project consists of
 - **Root** contains tooling configuration. e.g. `package.json`
 - **/public** contains everything *not* managed by the build system
 - **/src** holds all executable code, styling, and assets

```
▼ FIESTR
  > node_modules
  ▼ public
    vite.svg
  ▼ src
    > assets
    # App.css
    TS App.tsx
    # index.css
    TS main.tsx
    TS vite-env.d.ts
    .eslintrc.cjs
    .gitignore
    <> index.html
    {} package-lock.json
    {} package.json
    ⓘ README.md
    TS tsconfig.json
    {} tsconfig.node.json
    TS vite.config.ts
```



Familiar Faces

- If you know Angular, lots of the files should look familiar
 - `tsconfig.json` -
 - `package.json` / `package-lock.json` -



File: index.tsx

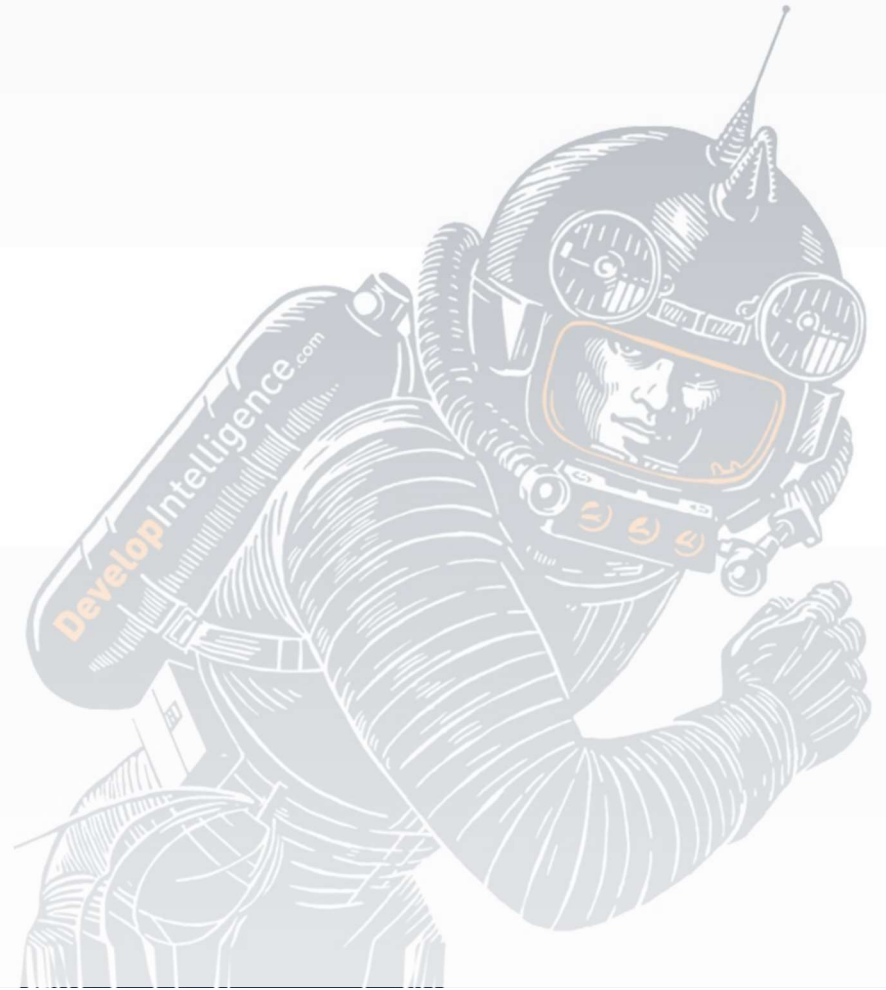
- Entrypoint
- Loads the root component (often called **App**)

```
1 import React from 'react';
2 import { Root } from './App';
3
4 ReactDOM.render(
5   <React.StrictMode>
6     <Root />
7   </React.StrictMode>,
8   document.getElementById('root'),
9 );
```



About React.StrictMode

- Turns on dev-mode
 - Double-rendering, double-effects
 - Checks for deprecations





Review

1. Compare React v. Angular
2. Explain the advantage of using a React framework
3. Describe Vite

