

Intro to React

Braden Guliano

Director of Technical Development

Kappa Theta Pi | Spring 2026 Rush

Prerequisites



GitHub account



Node.js



Git



IntelliJ WebStorm

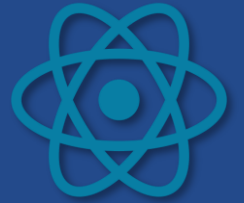


or VS Code



github.com/ktp-usc

What is React?



- A JavaScript library for building user interfaces
- Created and maintained by Facebook/Meta
- Component-based: build reusable UI pieces that combine into apps
- Works with modern tools for full stack web development

Why React?

- Traditional websites reload entire pages
- React apps feel faster and smoother
 - **SPA**: Single-Page Application
 - **MPA**: Multi-Page Application
- Used by companies worldwide for better user experience



Key differences

Normal pages

```
<a href="/about.html">
```

React

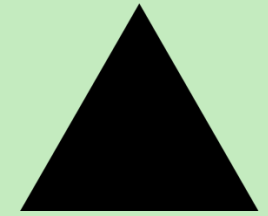
```
<a onClick={goToAbout}>About</a>
```

Becomes...

```
<a>About</a>
```

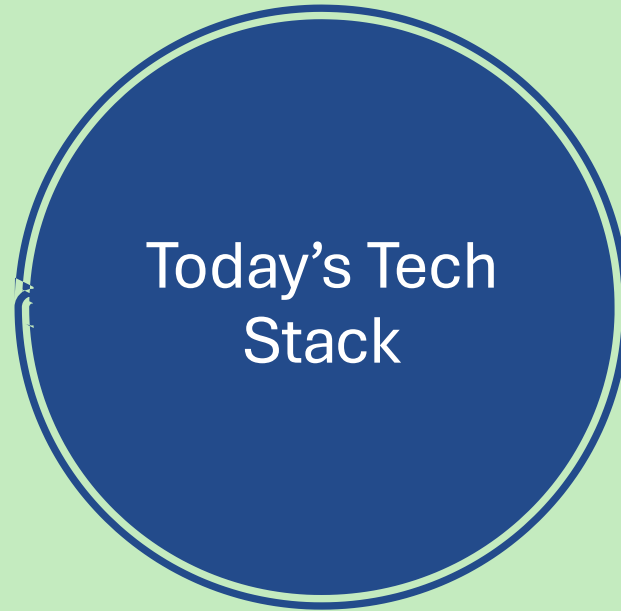


KTP Tech Stack



Tool	Purpose
Next.js	Core framework
TypeScript	Safer coding practices (catches errors before they occur with typing)
Tailwind CSS	Styling
Vercel	Deployment and branch previewing
GitHub	Remote version control
Figma	Design, wireframing, prototyping
Zustand	Easy and minimal global state management
React Query	Server state management and better fetching





Today's goals



Set up React project with
WebStorm + Vite



Learn the basics of
components, props,
states, and basic hooks



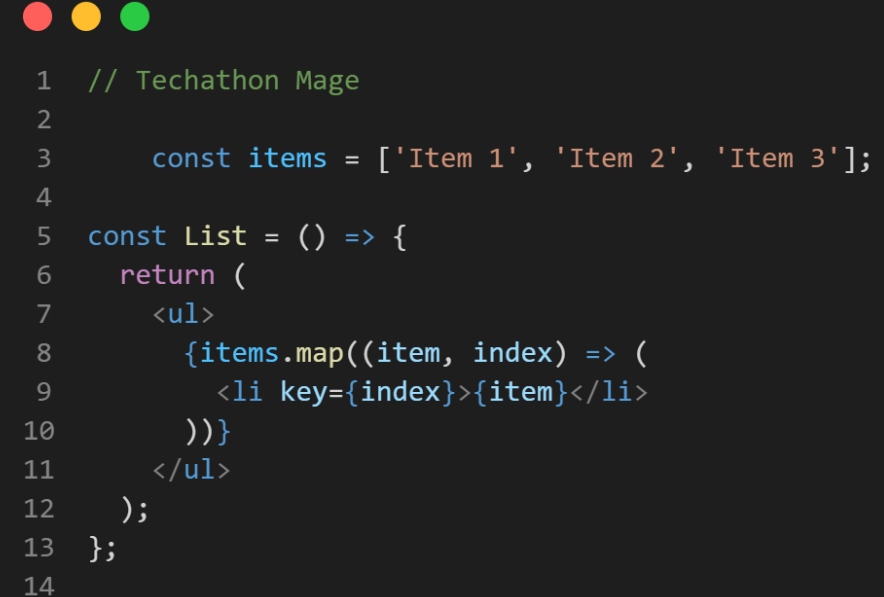
Build a small, interactive
site



Add routing and simple
styling

JSX – JavaScript XML

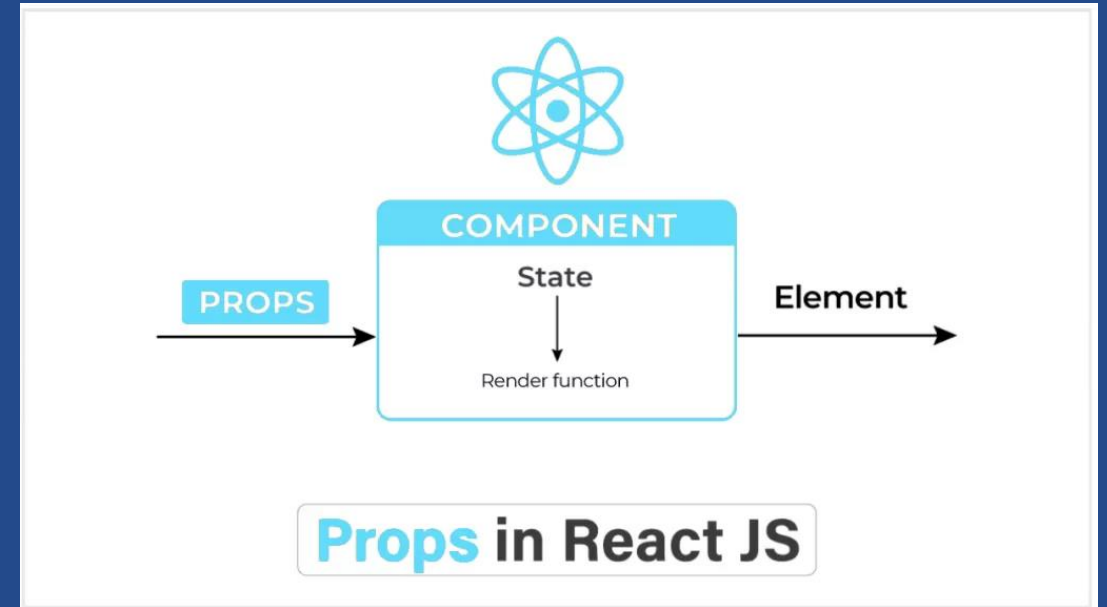
- Looks like HTML, but also includes JS expressions
- Must return a single root element
- Class -> className, inline events, curly braces vs quotes



```
1 // Techathon Mage
2
3     const items = ['Item 1', 'Item 2', 'Item 3'];
4
5 const List = () => {
6     return (
7         <ul>
8             {items.map((item, index) => (
9                 <li key={index}>{item}</li>
10             ))}
11         </ul>
12     );
13 };
14
```

Components + State + Props

- It's what makes React so powerful
- A component is a reusable piece of UI
- State is what information is stored inside the component
- Props are inputs passed into a component
- Input *props*, output slightly different UI *component*



You can think of **components** like **functions** in programming languages

Live setup

- Create project with Vite
- View pre-generated site
- Discover template contents



<Let's Build! />

Installing Tailwind

```
npm install tailwindcss @tailwindcss/vite
```

Installing react-router

```
npm i react-router
```

Git + GitHub Integration

```
git init
git add .
git commit -m "Initial commit"
git branch -M main
git remote add origin
    https://github.com/<username>/<repo_name>.git
git push -u origin main
```


Where do I go from here?

- Expand the project into your own portfolio
 - Work on the About page!
- Feel free to reference KTP GitHub
- Share link on application, optional

Questions?

