

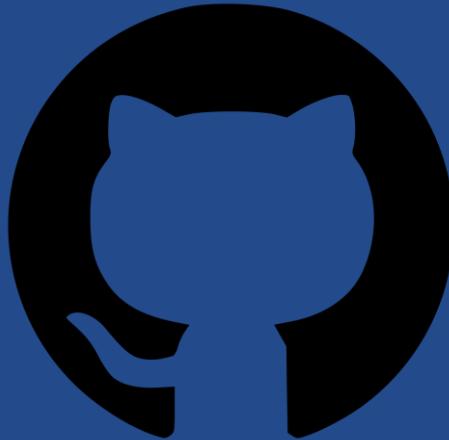


React + Next.js

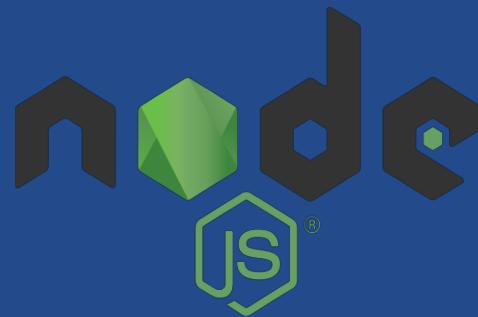
Tech Workshop #1

Feb 9, 2026

Prerequisites



GitHub account



Node.js



or VS Code



IntelliJ WebStorm



Agenda

1. Next.js overview
2. Setup live site
3. Structure + Tailwind
4. Components + Props
5. Navigation



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



What happened to React?

NE~~X~~T.JS

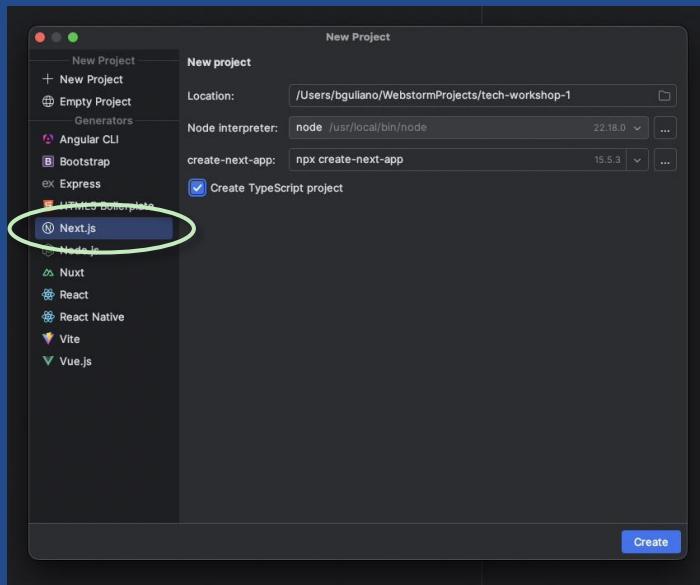


Next.js

- Created in 2016
- Maintained by Vercel + open-source community
- A framework built on React that adds production features out-of-the-box
- Benefits over plain React:
 - File-based routing
 - React **Server Components** by default (fast loads, smaller bundles)
 - Image/font optimization, streaming, prefetching
 - Strong developer experience (TypeScript, ESLint, Tailwind, Turbopack included)



Setup (WebStorm)



```
/usr/local/bin/npx --yes create-next-app@latest . --ts
✓ Which linter would you like to use? > ESLint
✓ Would you like to use React Compiler? ... No / Yes
✓ Would you like to use Tailwind CSS? ... No / Yes
✓ Would you like your code inside a 'src/' directory? ... No / Yes
✓ Would you like to use App Router? (recommended) ... No / Yes
✓ Would you like to customize the import alias ('@/*' by default)? ... No / Yes
Creating a new Next.js app in /Users/bguliano/WebstormProjects/tech-workshop-1.
```

Linter?	ESLint
React Compiler?	No
Tailwind CSS?	Yes
src/ directory?	No
App Router?	Yes
Customize import?	No

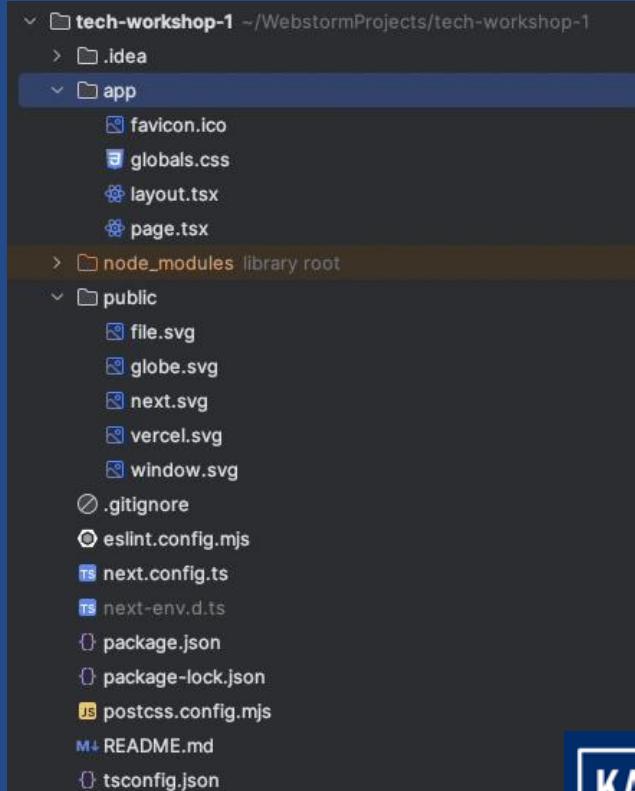
Setup (VS Code)

```
bguliano@macbook ~ % mkdir tech-workshop-1  
bguliano@macbook ~ % cd tech-workshop-1  
bguliano@macbook tech-workshop-1 % npx create-next-app  
? What is your project named? > .  
? Would you like to use the recommended Next.js defaults? > Yes
```

Linter?	ESLint
React Compiler?	No
Tailwind CSS?	Yes
src/ directory?	No
App Router?	Yes
Customize import?	No

Next.js structure

- / app → all routes live here
- layout.tsx → shared layout
- page.tsx → page content
- globals.css → global styles
- / public → static assets



Tailwind CSS

- Utility-first CSS framework
 - Many CSS files
 - Switching between files
- Styling using classes
- Responsive modifiers
 - sm:, md:, lg:
- Other modifiers
 - hover:, focus:, required:

The image shows two screenshots of the Tailwind CSS website. The top screenshot is the 'CHEAT SHEET' page, which lists utility classes like 'breakpoints', 'box-decoration-break', 'container', 'box-sizing', 'display', 'float', 'clear', 'isolation', 'object-fit', 'object-position', and 'overflow'. The bottom screenshot is the main website homepage, showing the 'Community' sidebar and the 'CORE CONCEPTS' section. The 'Colors' section is highlighted, showing a grid of color swatches for Red, Orange, Amber, Yellow, Lime, Green, Emerald, Teal, and Cyan.

Tailwind CSS

The screenshot shows the Tailwind CSS website's "Colors" section. The page has a dark header with the Tailwind logo and navigation links for Docs, Blog, Showcase, Sponsor, and Plus. A sidebar on the left contains links for Community, Getting Started (Installation, Editor setup, Compatibility, Upgrade guide), Core Concepts (Styling with utility classes, Hover, focus, and other states, Responsive design, Dark mode, Theme variables, Colors, Adding custom styles, Detecting classes in source files, Functions and directives), Base Styles (Preflight), and Layout (aspect-ratio). The main content area is titled "CORE CONCEPTS" and "Colors". It includes a sub-section header "Using and customizing the color palette in Tailwind CSS projects." and a paragraph stating "Tailwind CSS includes a vast, beautiful color palette out of the box, carefully crafted by expert designers and suitable for a wide range of different design styles." Below this is a color palette grid for Red, Orange, Amber, Yellow, Lime, Green, Emerald, Teal, and Cyan, each with a corresponding row of 10 color swatches labeled 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, and 950.



Tailwind CSS – Prompting

Where should we begin?

Refactor the following TSX code to use Tailwind CSS utility classes instead of custom CSS or inline styles. Do not change the component structure or functionality — only replace styling. Use clean, modern Tailwind patterns for spacing, layout, typography, and responsiveness.

TSX to refactor:

+

0

↑

React Components

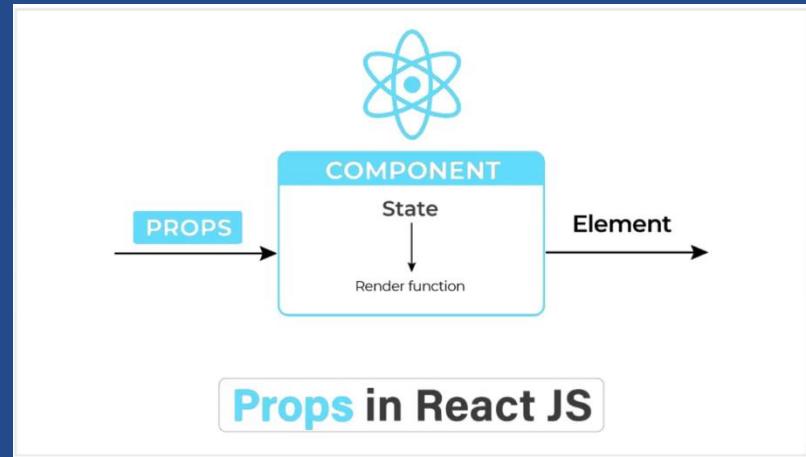
- Components are **reusable building blocks** of UI
- Accept props to customize behavior
- Encourage modular, maintainable code

```
function Greeting({ name }: { name: string })  
{  
  return <p>Hello {name}</p>  
}
```



React Props

- Props allow for passing data into child components from parent components
- It's what makes components reusable
- React backend recognizes changes in these props to know when to re-render the screen
- Using TypeScript enforces correct data types for props
 - Makes it harder to make mistakes



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

React Props

```
import Greeting from '@/components/Greeting'

export default function Home() {
  return (
    <section className="space-y-3 p-6">
      <h1 className="text-3xl font-
bold">Home</h1>
      <Greeting name="KTP" />
    </section>
  )
}
```

→ { "KTP" }



```
function Greeting({ name }: { name: string }) {
  return <p>Hello {name}</p>
}
```

Navigation

- Next.js uses **file-based routing** (/ app folder)
- Each file = one route (page . tsx → URL)
- Navigate with <Link>
- Navigation is typically a shared component used in layout . tsx
- Dynamic routes: use [slug] for flexible paths

Questions?

