React + Next.js

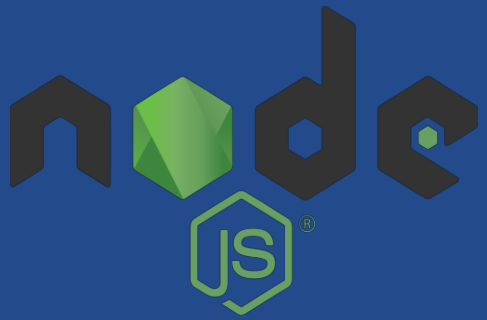
Tech Workshop #1

Feb 9, 2026

Prerequisites

or VS Code

GitHub account



Node.js

IntelliJ WebStorm

KAPPA THETA PI

Agenda

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

KAPPA THETA PI

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?

KAPPA THETA PI

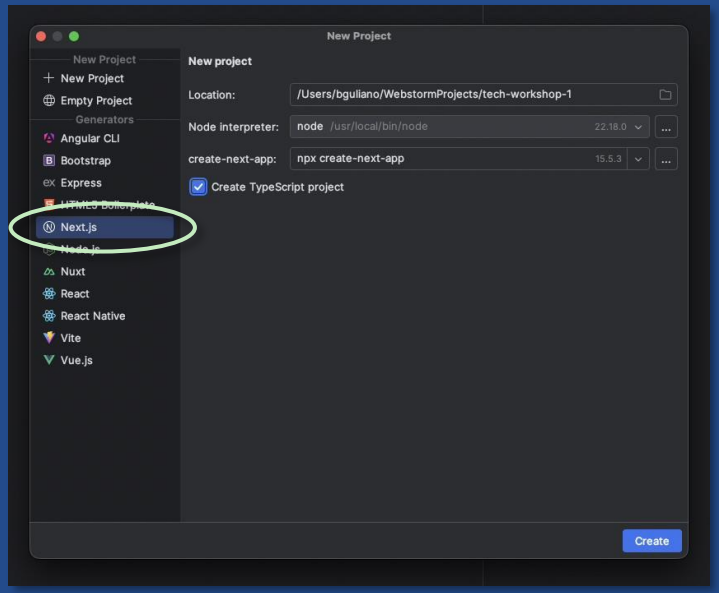
Next.js

- 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)

KAPPA THETA PI

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

KAPPA THETA PI

Setup (VS Code)

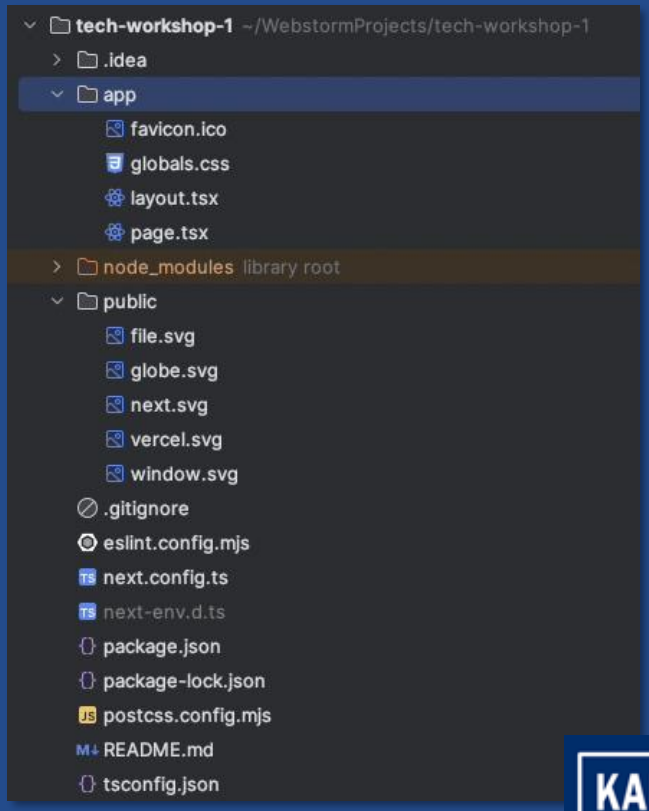
```
bguliano@macbook ~ % mkdir tech-workshop-1
bguliano@macbook ~ % cd tech-workshop-1
bguliano@macbook tech-workshop-1 % npx create-next-app
? What is your project named? > .
? Would you like to use the recommended Next.js defaults? > Yes
```

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

KAPPA THETA PI

Next.js structure

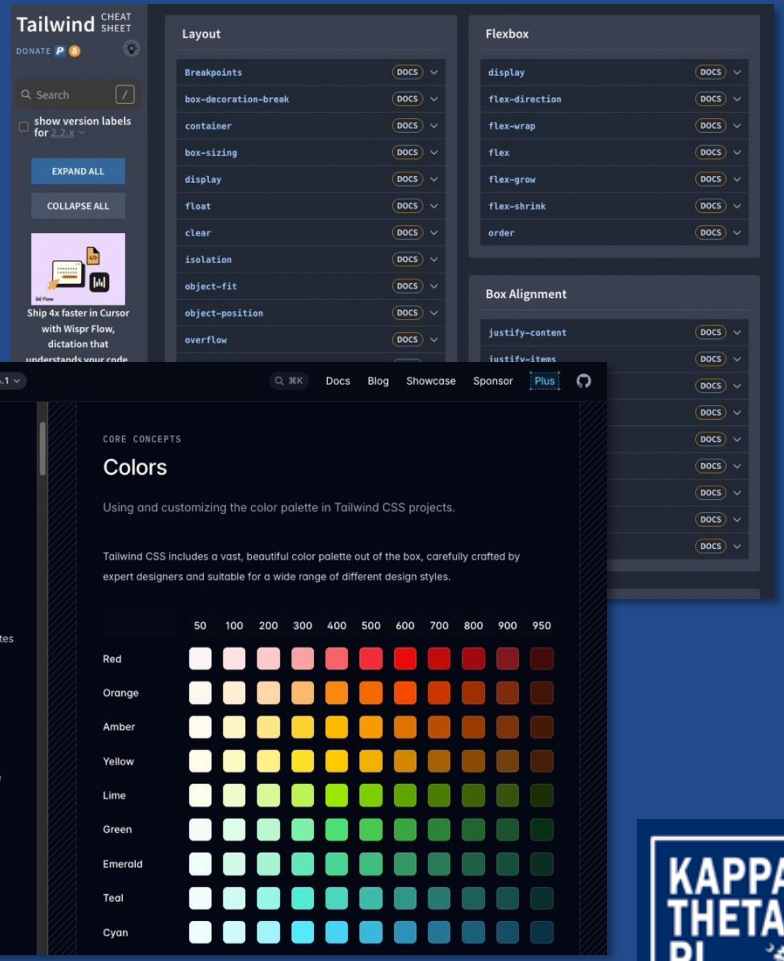
- `/app` → all routes live here
- `layout.tsx` → shared layout
- `page.tsx` → page content
- `globals.css` → global styles
- `/public` → static assets



KAPPA THETA PI

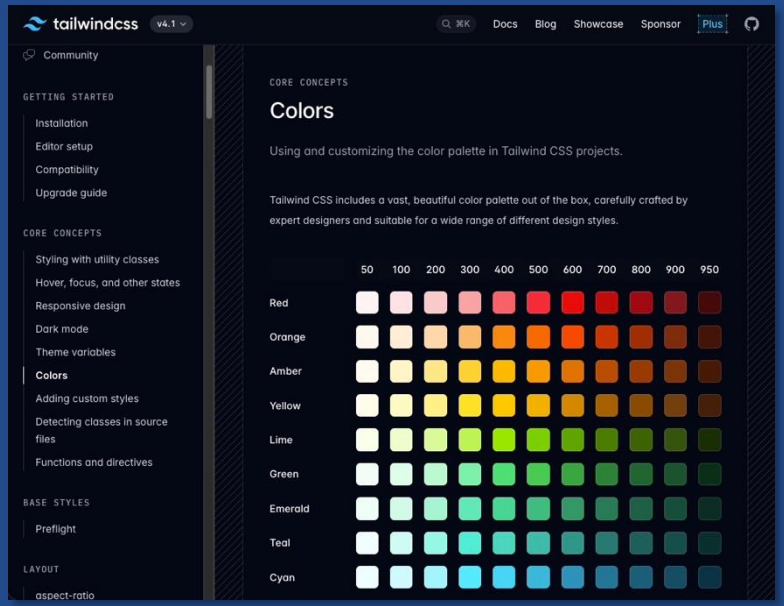
Tailwind CSS

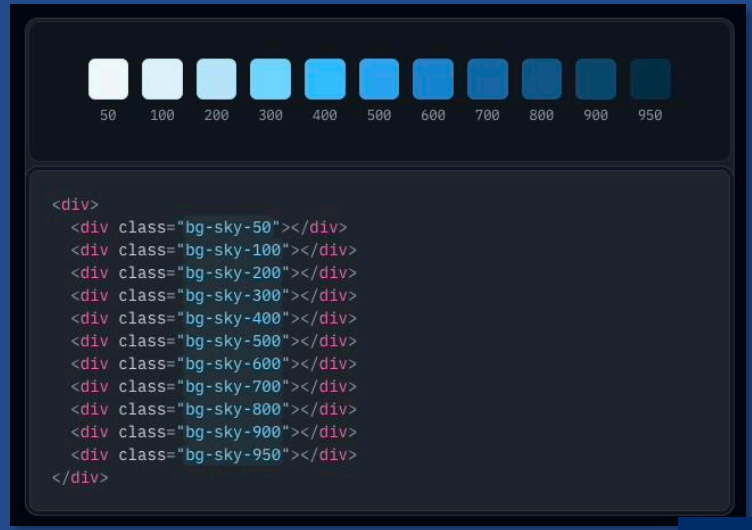
- Utility-first CSS framework
  - ~~Many CSS files~~
  - ~~Switching between files~~
- Styling using classes
- Responsive modifiers
  - `sm:`, `md:`, `lg:`
- Other modifiers
  - `hover:`, `focus:`, `required:`



KAPPA THETA PI

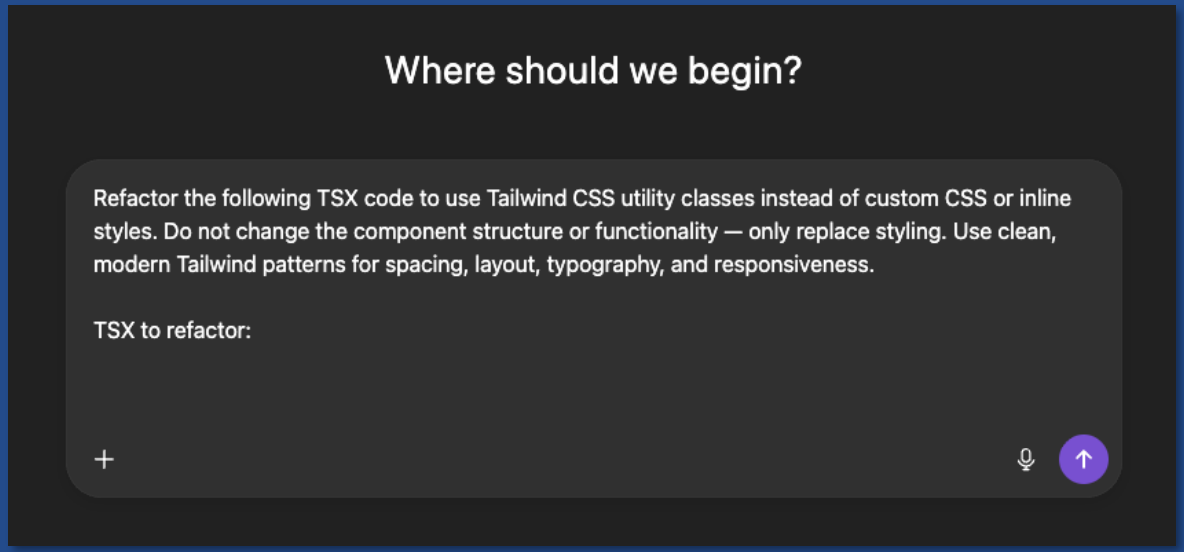
Tailwind CSS





KAPPA THETA PI

Tailwind CSS – Prompting



KAPPA THETA PI

React Components

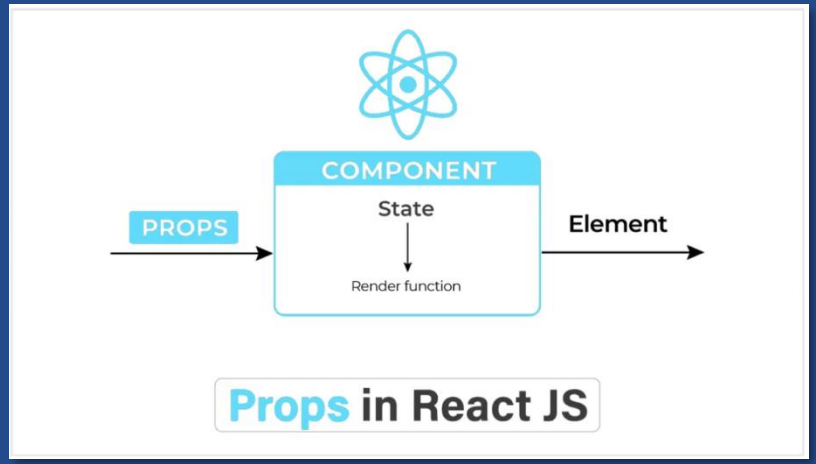
- Components are **reusable building blocks** of UI
- Accept props to customize behavior
- Encourage modular, maintainable code

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

KAPPA THETA PI

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

KAPPA THETA PI

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>
}
```

KAPPA THETA PI

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

KAPPA THETA PI

Questions?

KAPPA THETA PI