React + Next.js

Braden Guliano

Director of Technical Development

Kappa Theta Pi | Tech Workshop #1

Prerequisites

GitHub account

Node JS

IntelliJ WebStorm

KTP Tech Stack

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

KTP Tech Stack

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

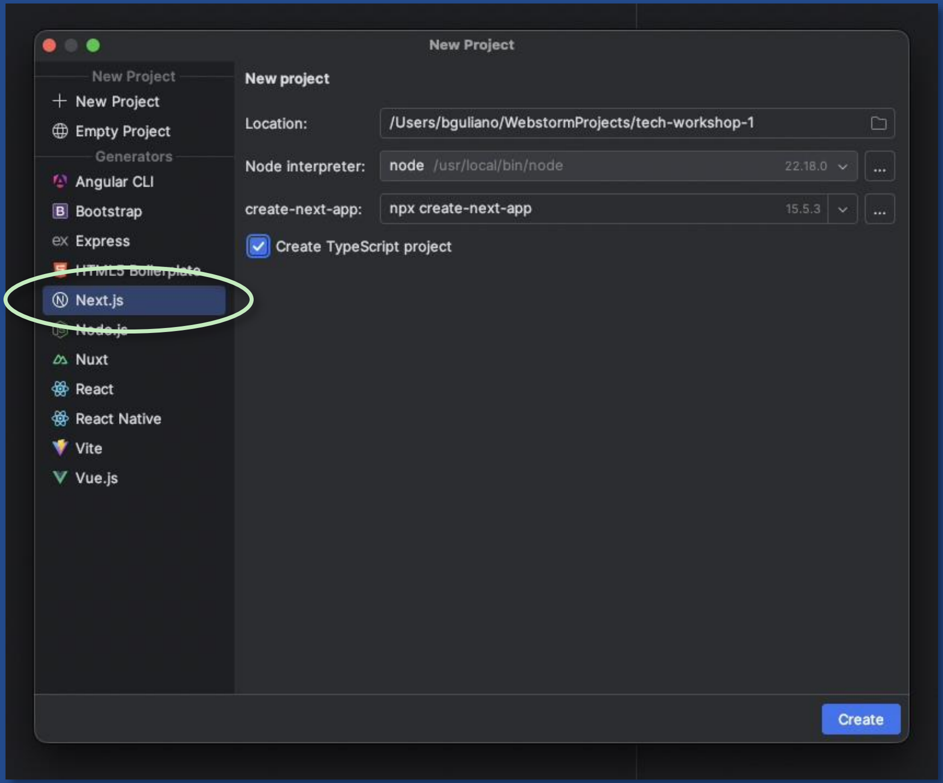
What happened to React?

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)



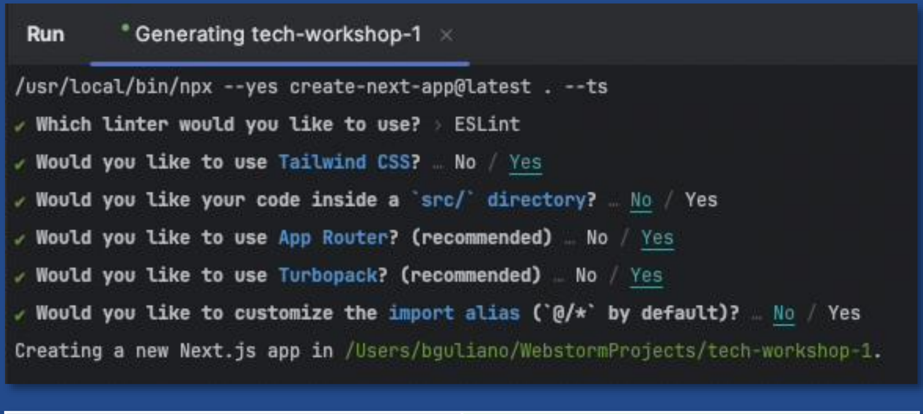
New project

Location: /Users/bguliano/WebstormProjects/tech-workshop-1

Node interpreter: node /usr/local/bin/node 22.18.0

create-next-app: npx create-next-app 15.5.3

☑ Create TypeScript project



```
/usr/local/bin/npx --yes create-next-app@latest . --ts
✓ Which linter would you like to use? › ESLint
✓ 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 use Turbopack? (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?	Yes
Tailwind CSS?	Yes
src/ directory?	No
App Router?	Yes
Turbopack?	Yes
Customize import?	No

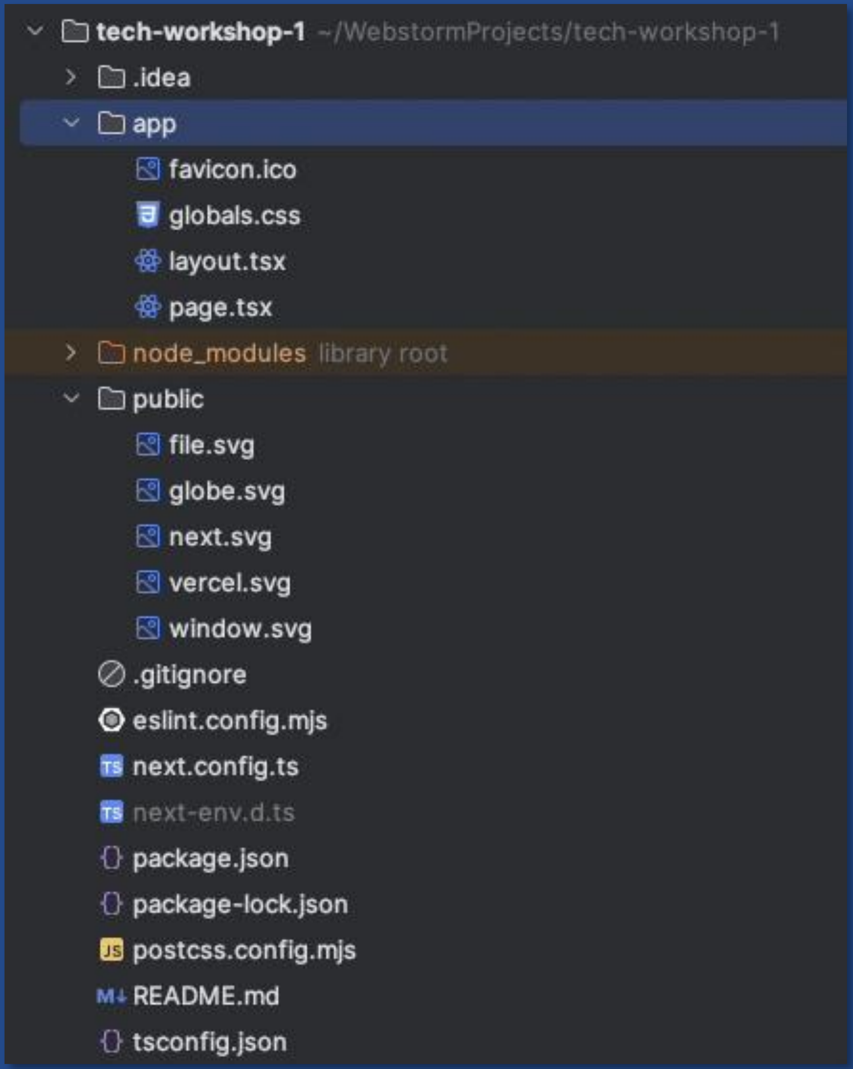
Setup (VS Code)

```
bguliano@macbook tech-workshop-1 % npx create-next-app
? What is your project named? > .
? Would you like to use TypeScript? > Yes
```

Linter?	Yes
Tailwind CSS?	Yes
src/ directory?	No
App Router?	Yes
Turbopack?	Yes
Customize import?	No

Next.js structure

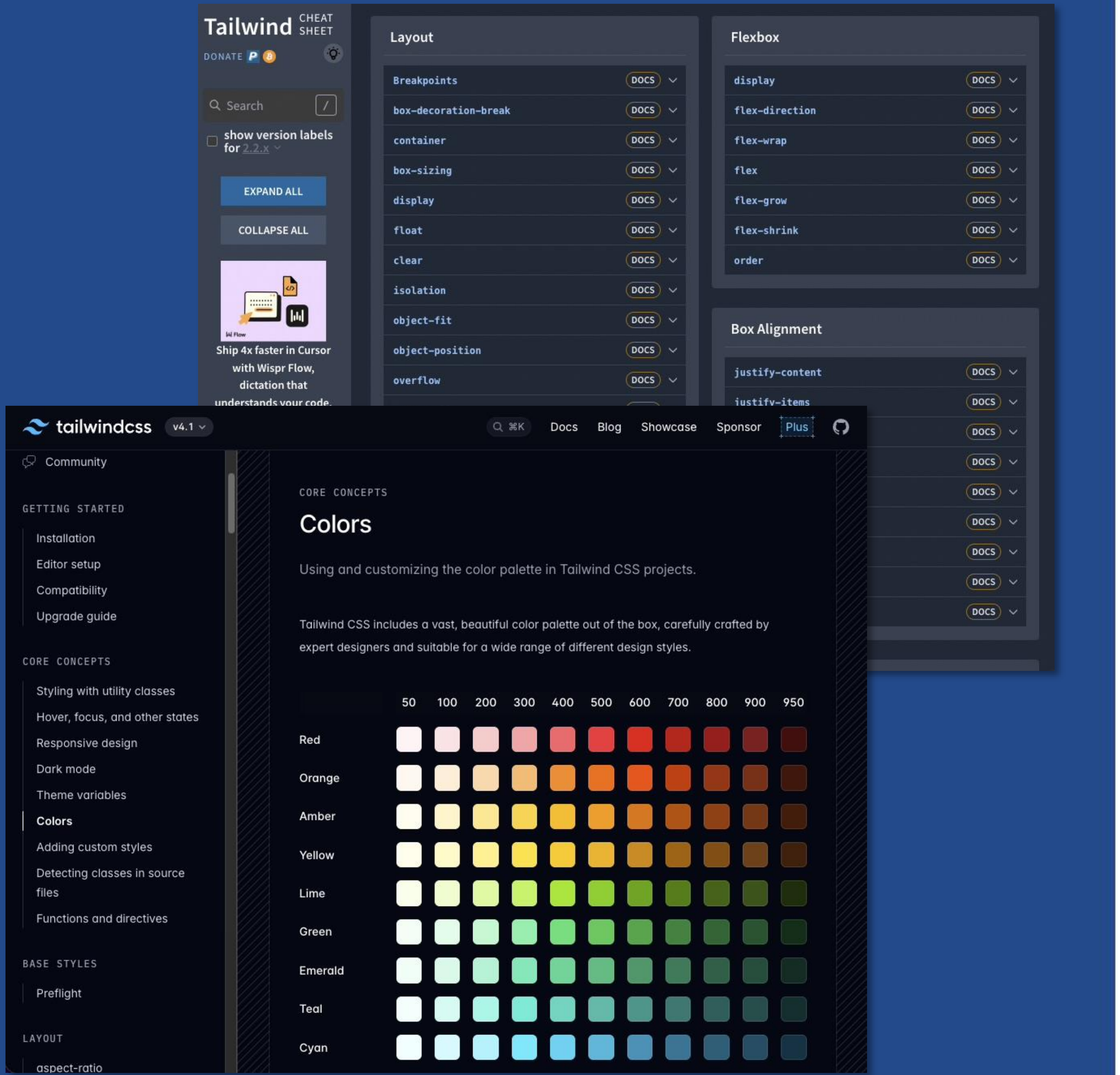
- `/app` → all routes live here
- `layout.tsx` → shared layout
- `page.tsx` → page content
- `globals.css` → global styles
- `/public` → static assets



```
tech-workshop-1 ~/WebstormProjects/tech-workshop-1
  .idea
  app
    favicon.ico
    globals.css
    layout.tsx
    page.tsx
  node_modules library root
  public
    file.svg
    globe.svg
    next.svg
    vercel.svg
    window.svg
  .gitignore
  eslint.config.mjs
  next.config.ts
  next-env.d.ts
  package.json
  package-lock.json
  postcss.config.mjs
  README.md
  tsconfig.json
```

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:`



Tailwind CHEAT SHEET

DONATE

Search /

□ show version labels for 2.2.x

EXPAND ALL

COLLAPSE ALL

Layout

- Breakpoints DOCS
- box-decoration-break DOCS
- container DOCS
- box-sizing DOCS
- display DOCS
- float DOCS
- clear DOCS
- isolation DOCS
- object-fit DOCS
- object-position DOCS
- overflow DOCS

Flexbox

- display DOCS
- flex-direction DOCS
- flex-wrap DOCS
- flex DOCS
- flex-grow DOCS
- flex-shrink DOCS
- order DOCS

Box Alignment

- justify-content DOCS
- justify-items DOCS

tailwindcss v4.1 ⌘K Docs Blog Showcase Sponsor Plus

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

LAYOUT

- aspect-ratio

CORE CONCEPTS

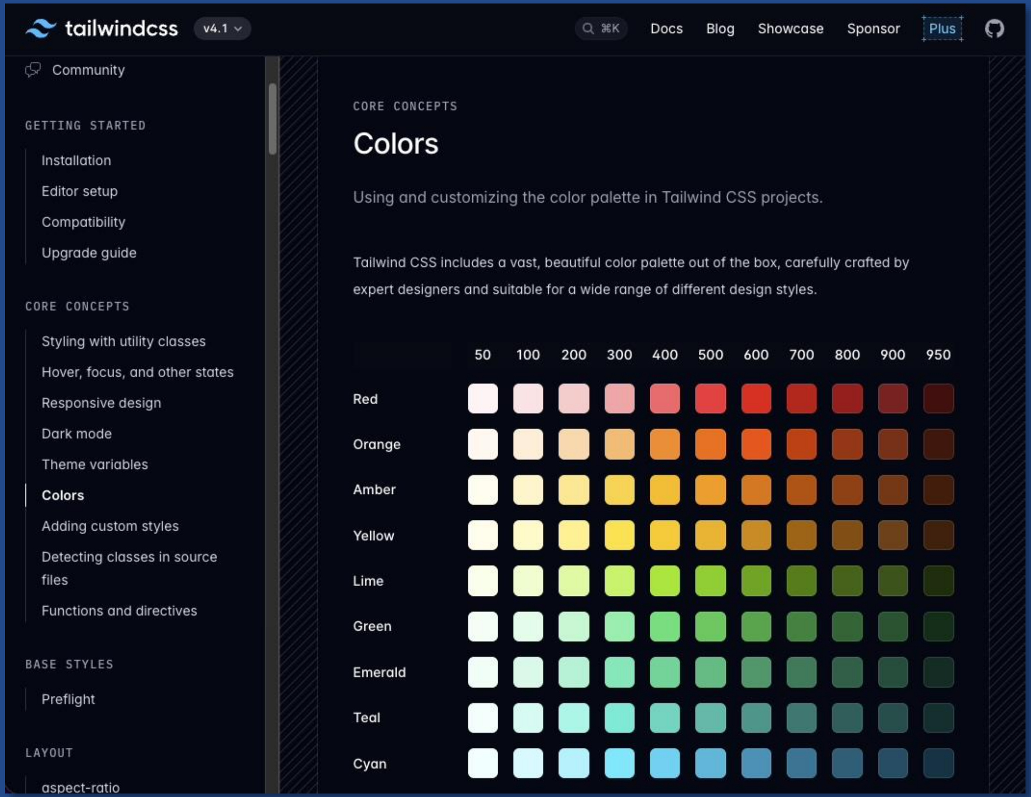
Colors

Using and customizing the color palette in Tailwind CSS projects.

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.

	50	100	200	300	400	500	600	700	800	900	950
Red	■	■	■	■	■	■	■	■	■	■	■
Orange	■	■	■	■	■	■	■	■	■	■	■
Amber	■	■	■	■	■	■	■	■	■	■	■
Yellow	■	■	■	■	■	■	■	■	■	■	■
Lime	■	■	■	■	■	■	■	■	■	■	■
Green	■	■	■	■	■	■	■	■	■	■	■
Emerald	■	■	■	■	■	■	■	■	■	■	■
Teal	■	■	■	■	■	■	■	■	■	■	■
Cyan	■	■	■	■	■	■	■	■	■	■	■

Tailwind CSS



tailwindcss v4.1

⌘K Docs Blog Showcase Sponsor Plus

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

LAYOUT

- aspect-ratio

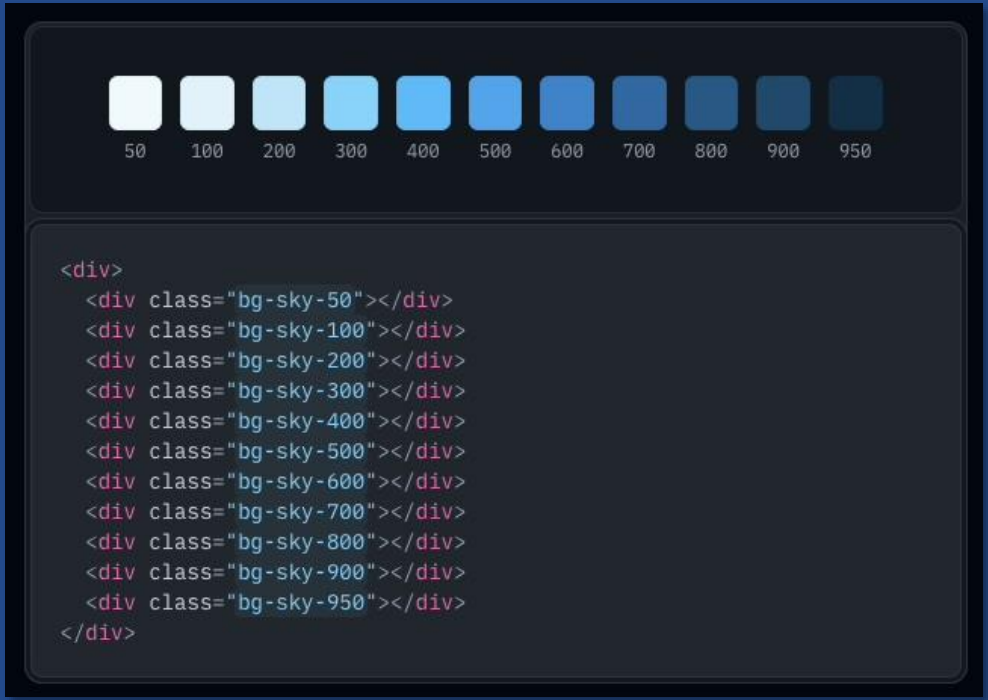
CORE CONCEPTS

Colors

Using and customizing the color palette in Tailwind CSS projects.

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.

	50	100	200	300	400	500	600	700	800	900	950
Red	■	■	■	■	■	■	■	■	■	■	■
Orange	■	■	■	■	■	■	■	■	■	■	■
Amber	■	■	■	■	■	■	■	■	■	■	■
Yellow	■	■	■	■	■	■	■	■	■	■	■
Lime	■	■	■	■	■	■	■	■	■	■	■
Green	■	■	■	■	■	■	■	■	■	■	■
Emerald	■	■	■	■	■	■	■	■	■	■	■
Teal	■	■	■	■	■	■	■	■	■	■	■
Cyan	■	■	■	■	■	■	■	■	■	■	■



50	100	200	300	400	500	600	700	800	900	950
■	■	■	■	■	■	■	■	■	■	■

```html
<div>
  <div class="bg-sky-50"></div>
  <div class="bg-sky-100"></div>
  <div class="bg-sky-200"></div>
  <div class="bg-sky-300"></div>
  <div class="bg-sky-400"></div>
  <div class="bg-sky-500"></div>
  <div class="bg-sky-600"></div>
  <div class="bg-sky-700"></div>
  <div class="bg-sky-800"></div>
  <div class="bg-sky-900"></div>
  <div class="bg-sky-950"></div>
</div>
```

React Components

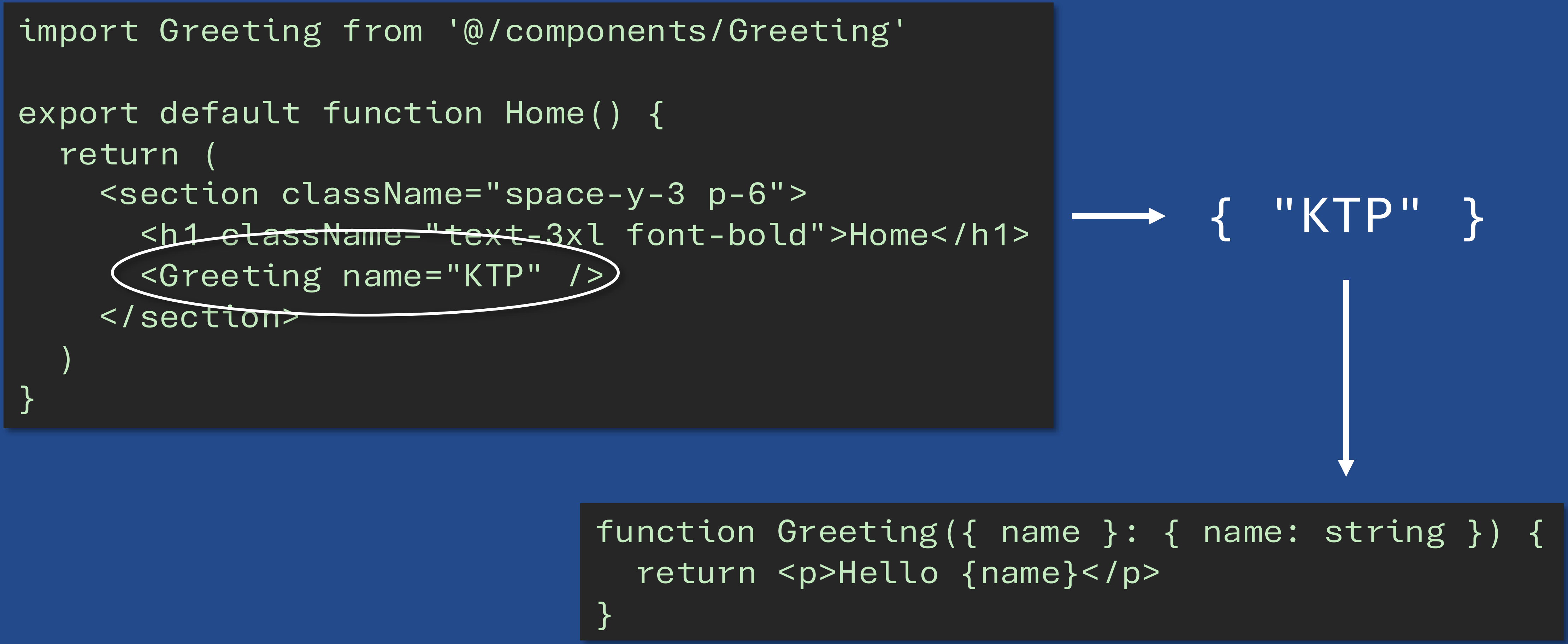
- Components are **reusable building blocks** of UI
- Accept props to customize behavior
- Encourage modular, maintainable code

```
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

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

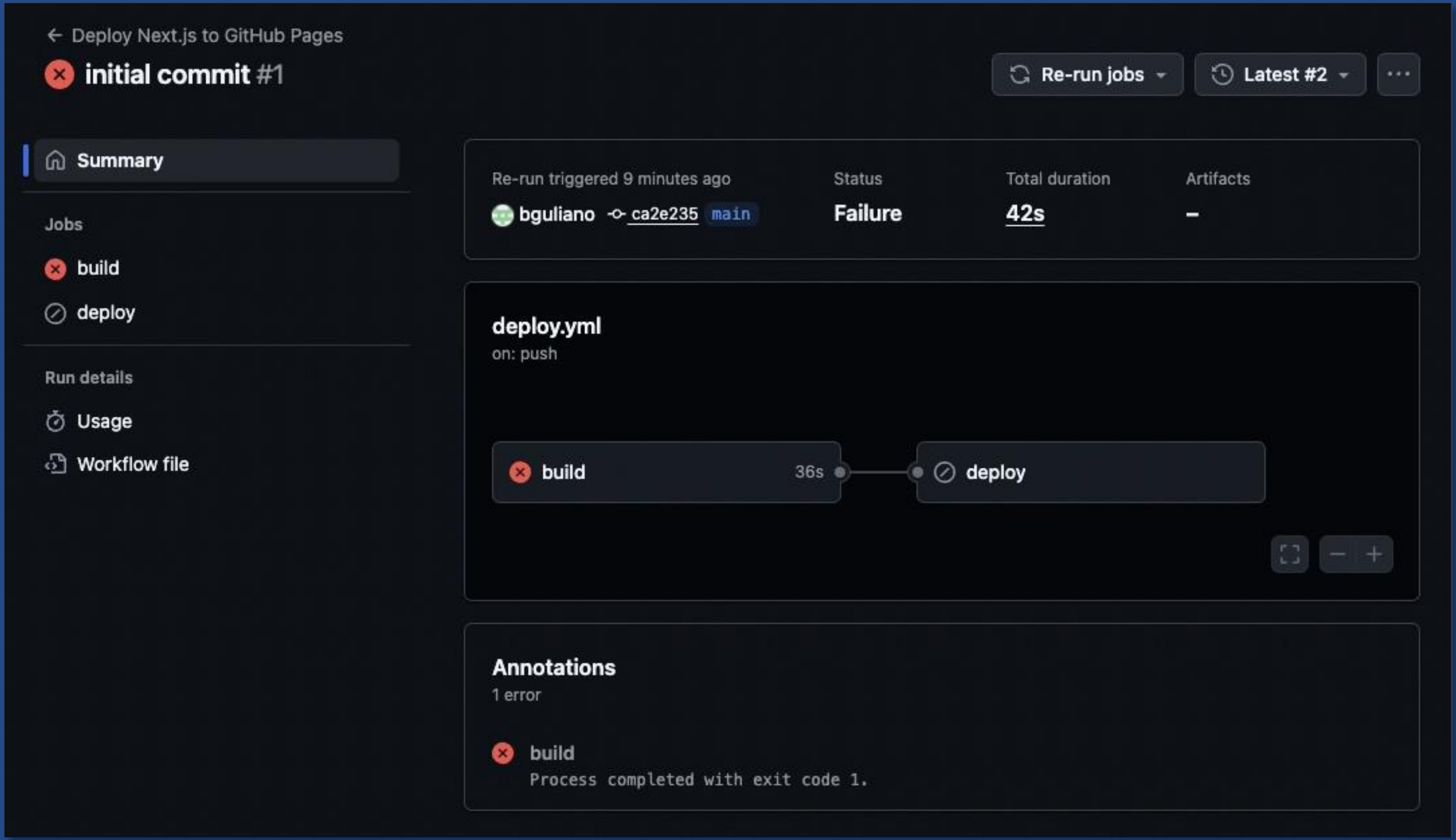
- Next.js uses **file-based routing** (`/app` folder)
- Each file = one route (`page.tsx` → URL)
- Navigate with `<Link>`
- Navigation is typically a shared component used in `layout.tsx`

- Dynamic routes: use `[slug]` for flexible paths

Deploying to GitHub

1. Edit `next.config.js`
2. Create GitHub Actions workflow
3. Change Actions > General to Read/Write
4. Set Pages > Build and deployment > Source to GitHub Actions

Why did it fail?



☒ initial commit #1

Re-run triggered 9 minutes ago
bguliano ca2e235 main

Status: Failure

Total duration: 42s

Artifacts: –

deploy.yml
on: push

Annotations
1 error

☒ build

```
Process completed with exit code 1.
```

Why did it fail?

```
import React from "react";

type Params = {
    projectName: string;
}

const ProjectNamePage = ({ params }: { params: Params }) => {
    return (
        <p>This page features project &#34;{params.projectName}&#34;</p>
    );
};
export default ProjectNamePage;
```

Client-side

VS

Server-side

Questions?

ΚΘΠ