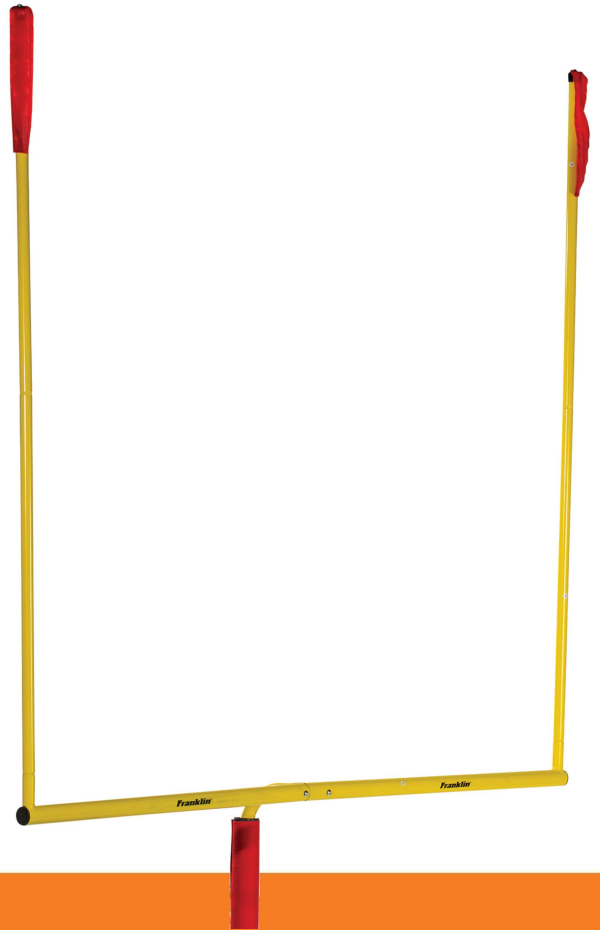
Potpourri

Develop Intelligence

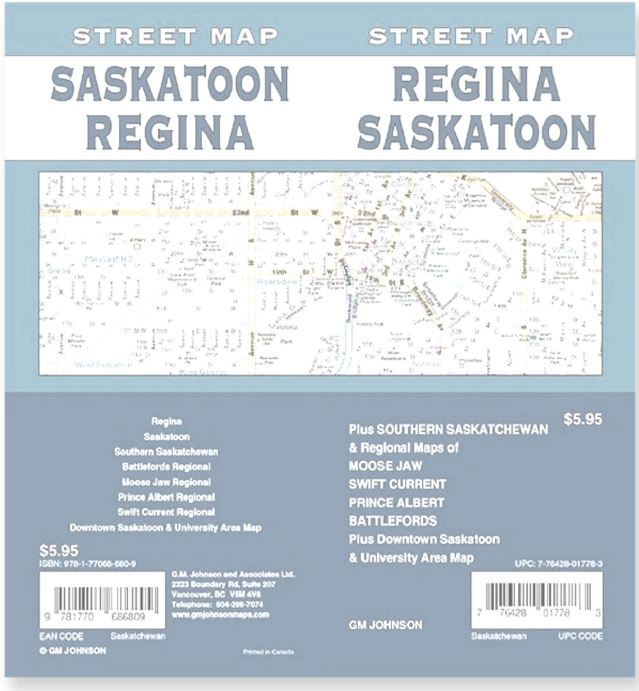
Goals

1. Explain what 'lifting state up' means
2. Describe how to use `createContext`



Roadmap

1. Context
2. Routing



Develop
Intelligence

Context

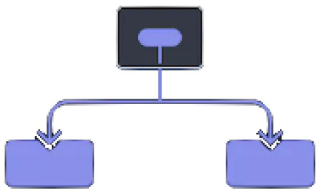
Motivating Example

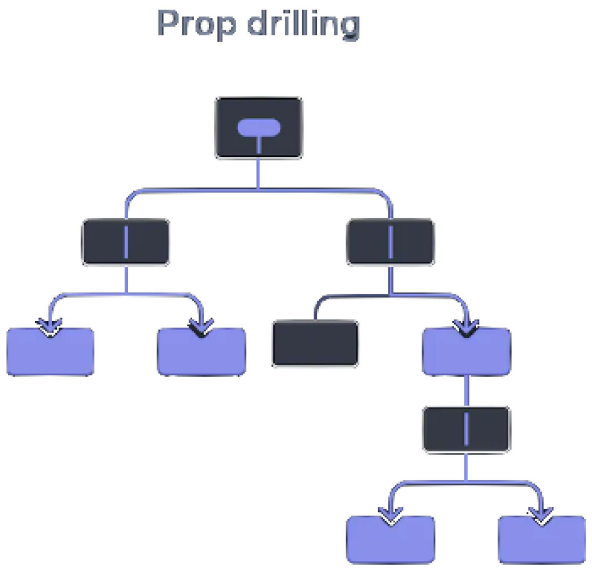
Develop Intelligence

```
export const Page = () =>
    <Section>
      <Heading level={1}>Title</Heading>
      <Section>
        <Heading level={2}>Heading</Heading>
        <Heading level={2}>Heading</Heading>
        <Heading level={2}>Heading</Heading>
        <Section>
          <Heading level={3}>Sub-heading</Heading>
          <Heading level={3}>Sub-heading</Heading>
          <Section>
            <Heading level={4}>Sub-sub-heading</Heading>
            <Heading level={4}>Sub-sub-heading</Heading>
          </Section>
        </Section>
      </Section>
```

Lifting State Up

- Known as 'prop drilling'





Solution: Context

Develop Intelligence

Usually, you will pass information from a parent component to a child component via props. But passing props can become verbose and inconvenient if you have to pass them through many components in the middle, or if many components in your app need the same information. Context lets the parent component make some information available to any component in the tree below it—no matter how deep—without passing it explicitly through props.

Step 1: createContext


```
1 import { createContext } from 'react';
2
3 // Note: This is the default context.
4 export const LevelContext = createContext(1);
```

Step 2: Use the Context

Develop Intelligence

```tsx
export const Heading = ({ children }: PropsWithChildren) => {
  const level = useContext(LevelContext);
  switch (level) {
    case 1:
      return <h1>{children}</h1>;
    case 2:
      return <h2>{children}</h2>;
    // Etc....
```

Step 3: Inject the Context

Develop Intelligence

```tsx
interface SectionProps extends PropsWithChildren {
  level: number;
}

export const Section = ({ children, level }: SectionProps) => {
  return (
    <section className="section prose">
      <LevelContext.Provider value={level}>{children}</LevelContext.Provider>
    </section>
  );
};
```

Context Use Cases

Develop Intelligence

- Theming
- User information
- Routing
- Services (with reducers)

Develop
Intelligence

Routing

Overview

- Common options:
  - Roll your own
  - Comes with your framework
  - React Router
- Often 'routing' includes more than just routing

Example: React Router

Develop Intelligence

- Common solution when you don't have a framework

```
const router = createBrowserRouter([
  {
    path: '',
    element: <App />,
    errorElement: <ErrorPage />,
    children: Object.values(routes),
  },
]);

ReactDOM.createRoot(document.getElementById('root')!).render(
  <React.StrictMode>
    <RouterProvider router={router} />
  </React.StrictMode>,
);
```

App.js

Develop Intelligence

```
import { Navbar } from './components/nav-bar/Navbar.tsx';
import { Outlet } from 'react-router-dom';
import { ThemeProvider } from '@components/theme/theme-provider.tsx';

export function App() {
  return (
    <ThemeProvider
      defaultTheme="light"
      storageKey="vite-ui-theme"
    >
      <Navbar />
      <Outlet />
    </ThemeProvider>
  );
}
```

Routes

Develop Intelligence

```tsx
export interface RouteInfo {
  path: string;
  element: ReactNode;
  title: string;
}

export const demoRoutes: { [key: string]: RouteInfo } = {
  Planets: {
    path: '/demo/planets',
    element: <PlanetsDemo />,
    title: 'Planets',
  },
  Headings: {
    path: '/demo/headings',
    element: <HeadingsDemo />,
    title: 'Headings',
```

Links

Develop Intelligence

```
interface Props extends PropsWithChildren {
  to: string;
}

export const NavbarMenuLink = ({ to, children, ...props }: Props) => (
  <Link
    className="block select-none space-y-1 rounded-md p-3 leading-none no-und
    to={to}
    {...props}
  >
    <NavigationMenuLink>{children}</NavigationMenuLink>
  </Link>
);
```

Develop
Intelligence

Review

1. Explain what 'lifting state up' means
2. Describe how to use `createContext`