Design

Reactivity

Declare state

```
export default function Name() {
  const name = "John"

  return <h1>Hello {name}</h1>;
}
```

Update state

```
export default function Name() {
  let name = "John"
  name = "Jane"

  return <h1>Hello {name}</h1>;
}
```

Computed state

```
export default function DoubleCount() {
  const count = 10
  const doubleCount = count * 2;

  return <div>{doubleCount}</div>;
}
```

Templating

Minimal template

```
export default function HelloWorld() {
  return <h1>Hello world</h1>;
}
```

Styling

```
import "./style.css";

export default function CssStyle() {
  return (
    <>
      <h1 className="title">I am red</h1>
      <button style={{ fontSize: "10rem" }}>I am a button</button>
    </>
  );
}
```

Loop

```
export default function Colors() {
  const colors = ["red", "green", "blue"];
  return (
    <ul>
      <for each={colors}>{(color) => <li>{color}</li>}</for>
    </ul>
  );
}
```

Event click

```
export default function Counter() {
  let count = 0;

  function incrementCount() {
    count ++;
  }

  return (
    <>
      <p>Counter: {count}</p>
      <button onClick={incrementCount}>+1</button>
    </>
  );
}
```

Dom ref

```
export default function InputFocused() {
  let inputElement;

  didMount(() => {
    inputElement.focus();
  })

  return <input type="text" element={inputElement} />;
}
```

Conditional

```
const TRAFFIC_LIGHTS = ["red", "orange", "green"];

export default function TrafficLight() {
  let lightIndex = 0

  const light = TRAFFIC_LIGHTS[lightIndex];

  function nextLight() {
    lightIndex = (lightIndex + 1) % TRAFFIC_LIGHTS.length;
  }

  return (
    <>
      <button onClick={nextLight}>Next light</button>
      <p>Light is: {light}</p>
      <p>
        You must
        <if cond={light === "red"}>
          <span>STOP</span>
        </if>
        <else-if cond={light === "orange"}>
          <span>SLOW DOWN</span>
        </else-if>
        <else-if cond={light === "red"}>
          <span>GO</span>
        </else-if>
      </p>
    </>
  );
}
```

Lifecycle

On mount

```
export default function PageTitle() {
  let pageTitle = "";

  didMount(() => {
    pageTitle = document.title;
  });

  return <p>Page title: {pageTitle}</p>;
}
```

On unmount

```
export default function Time() {
  let time = new Date().toLocaleTimeString();

  const timer = setInterval(() => {
    time = new Date().toLocaleTimeString()
  }, 1000);

  willUnmount(() => clearInterval(timer));
  // or
  didUnmount(() => clearInterval(timer));

  return <p>Current time: {time}</p>;
}
```

Component composition

Props

```
export default function UserProfile({
  name = "",
  age = null,
  favouriteColors = [],
  isAvailable = false,
}) {
  return (
    <>
      <p>My name is {name}!</p>
      <p>My age is {age}!</p>
      <p>My favourite colors are {favouriteColors.join(", ")}!</p>
      <p>I am {isAvailable ? "available" : "not available"}</p>
    </>
  );
}

// ~
import UserProfile from "./UserProfile.jsx";
```

```
export default function App() {
  return (
    <UserProfile
      name="John"
      age={20}
      favouriteColors={["green", "blue", "red"]}
      isAvailable
    />
  );
}
```

Emit to parent

```
export default function AnswerButton({ onYes, onNo }) {
  return (
    <>
      <button onClick={onYes}>YES</button>

      <button onClick={onNo}>NO</button>
    </>
  );
}

// ~
import AnswerButton from "./AnswerButton.jsx";

export default function App() {
  let isHappy = true;

  function onAnswerNo() {
    isHappy = false
  }

  function onAnswerYes() {
    isHappy = true
  }

  return (
    <>
      <p>Are you happy?</p>
      <AnswerButton onYes={onAnswerYes} onNo={onAnswerNo} />
      <p style={{ fontSize: 50 }}>{isHappy ? "😀" : "😥"}</p>
    </>
  );
}
```

Slot

```
export default function FunnyButton({ children }) {
  return (
    <button
      style={{
        background: "rgba(0, 0, 0, 0.4)",
        color: "#fff",
        padding: "10px 20px",
        fontSize: "30px",
        border: "2px solid #fff",
        margin: "8px",
        transform: "scale(0.9)",
        boxShadow: "4px 4px rgba(0, 0, 0, 0.4)",
        transition: "transform 0.2s cubic-bezier(0.34, 1.65, 0.88, 0.925) 0s",
        outline: "0",
      }}
    >
      {children}
    </button>
  );
}

// ~
import FunnyButton from "./FunnyButton.jsx";

export default function App() {
  return <FunnyButton>Click me!</FunnyButton>;
}
```

Slot fallback

```
export default function FunnyButton({ children }) {
  return (
    <button
      style={{
        background: "rgba(0, 0, 0, 0.4)",
        color: "#fff",
        padding: "10px 20px",
        fontSize: "30px",
        border: "2px solid #fff",
        margin: "8px",
        transform: "scale(0.9)",
        boxShadow: "4px 4px rgba(0, 0, 0, 0.4)",
        transition: "transform 0.2s cubic-bezier(0.34, 1.65, 0.88, 0.925) 0s",
        outline: "0",
      }}
    >
      {children || <span>No content found</span>}
    </button>
  );
}

// ~
import FunnyButton from "./FunnyButton.jsx";
```

```
export default function App() {
  return (
    <>
      <FunnyButton />
      <FunnyButton>I got content!</FunnyButton>
    </>
  );
}
```

Context

```
export default function UserProfile(_props, { username, email, updateUsername }) {
  return (
    <div>
      <h2>My Profile</h2>
      <p>Username: {username}</p>
      <p>Email: {email}</p>
      <button onClick={() => updateUsername("Jane")}>
        Update username to Jane
      </button>
    </div>
  );
}
// ~
import UserProfile from "./UserProfile";

export default function App() {
  let user = {
    id: 1,
    username: "unicorn42",
    email: "unicorn42@example.com",
  };

  function updateUsername(newUsername) {
    user = { ...user, username: newUsername };
  }

  return (
    <>
      <h1>Welcome back, {user.username}</h1>
      <env {...user} updateUsername={updateUsername}>
        <UserProfile />
      </env>
    </>
  );
}
```