3 Day Workshop - Frontend Development

-Abhijeet Saxena
UI Engineer at

Introduction to Programming with React.js

Day 1

Objectives

- Introduction
- Coding Environment
- CRA template & File Structure
- React Components
- Props & States
- Build our own very basic Counter App

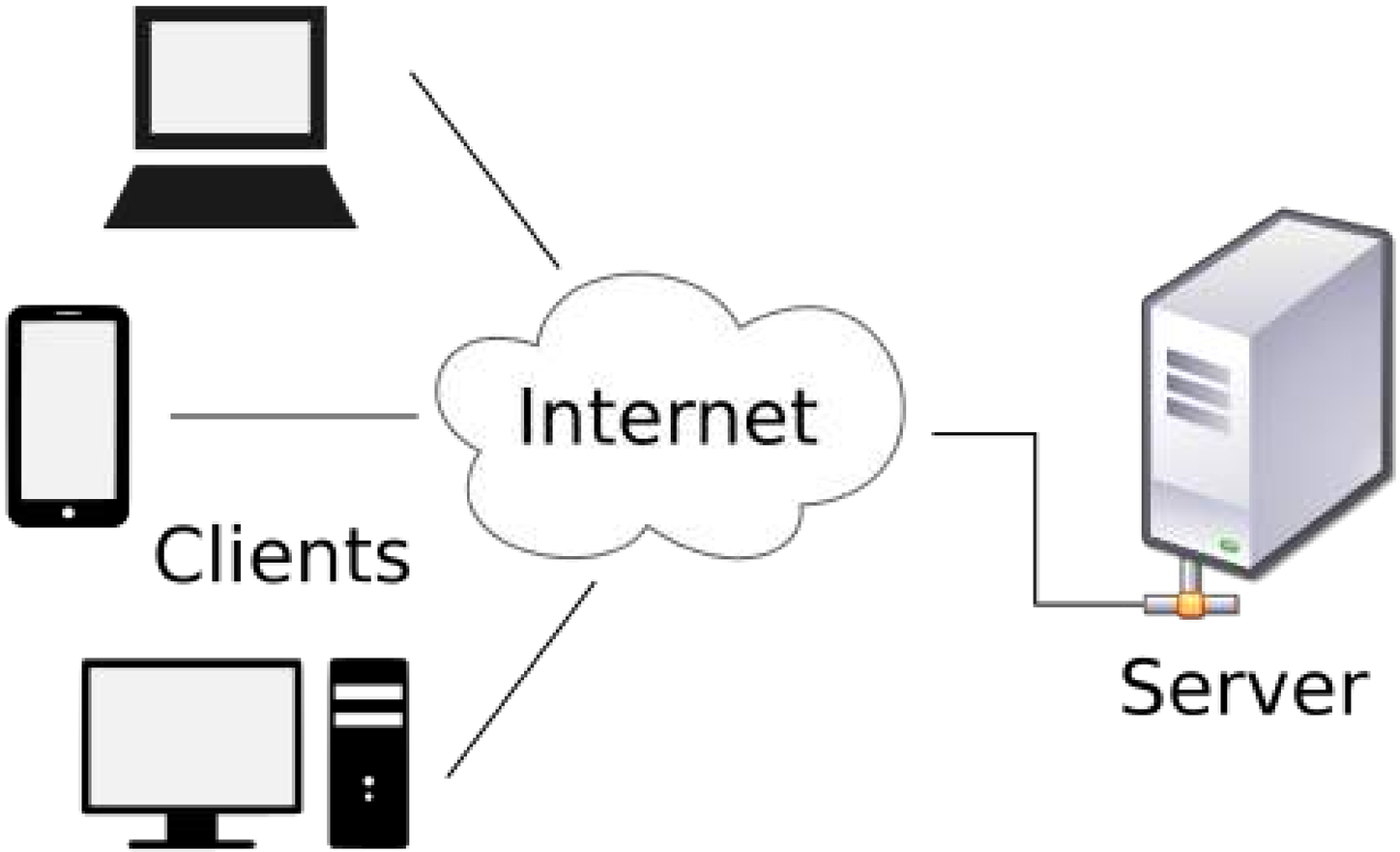
What Is React ?

"A JavaScript library for building user interfaces"

- React makes it painless to create interactive UIs
- Component-Based
- Developed by f in 2013
- React has a community of millions of developers.
- One of the most popular way to build an SPA

https://reactjs.org/

Client Server Architecture



What is SPA (Single Page Application) ?

- A web application or website that interacts with the user by dynamically rewriting the current web page with new data from the web server, instead of the default method of a web browser loading entire new pages
- Resulting in faster transitions that make the website feel quicker
- In a SPA, a page refresh never occurs; instead, all necessary HTML, JavaScript, and CSS code is either retrieved by the browser with a single page load
- Or the appropriate resources are dynamically loaded and added to the page as necessary, usually in response to user actions

https://en.wikipedia.org/wiki/Single-page_application

< />

CODING ENVIRONMENT

There are multiple code sandboxes available online.
For this workshop we'll be using a popular option https://codesandbox.io/

Briefly explain the core features-

- Files & Dependency Pane
- Output Panel
- Integrated Code Editor

File structure of react project

public

- **index.html**

This folder contains the main index.html file sent to the browser
Also may contain static assets like images/media/favicon/manifest files etc.

src

- **App.js**
- **index.js**
- **styles.css**

Source Code folder we would be focussing on-
Contains the React components, styles and other JSX files

package.json

Used by NPM to install required dependencies for CRA and other libraries like Axios, Bootstrap, Lodash etc

build

Contains the production build of your app

What is a React component ?

Components let you split the UI into independent, reusable pieces, and think about each piece in isolation.

Conceptually, components are like JavaScript functions. They accept arbitrary inputs (called “props”) and return React elements describing what should appear on the screen.

Example-

```
const element = <div />;                          // HTML Elements

const element = <Welcome name="Sara" />;          // Custom JSX Components
```

CLASS
STATEFUL

```
class Welcome extends React.Component {
    render() {
        return <h1>Hello, {this.props.name}</h1>;
    }
}
```

FUNCTION
STATELESS

```
function Welcome(props) {
    return <h1>Hello, {props.name}</h1>;
}
```

Composing Components

Components can refer to other components in their output. This lets us use the same component abstraction for any level of detail.
For example, we can create an App component that renders Welcome many times

```
function App() {
    return (
        <div>
            <Welcome name="Sara" />
            <Welcome name="Cahal" />
             <Welcome name="Edite" />
        </div>
    );
}
```

Props

Used by a component **to get data from external environment** i.e another **component** (pure, functional or class) or a **general class** or **javascript** code

In a nutshell

- External
- Controlled by parent who renders it
- Immutable
- Passed between components

State

Used **to manage the internal environment of a component** means the **data changes** inside the **component**

In a nutshell

- Internal
- Controlled by component itself
- Mutable
- Cant be used by other components

See this Stackoverflow Answer

All React components must act like pure functions with respect to their props.

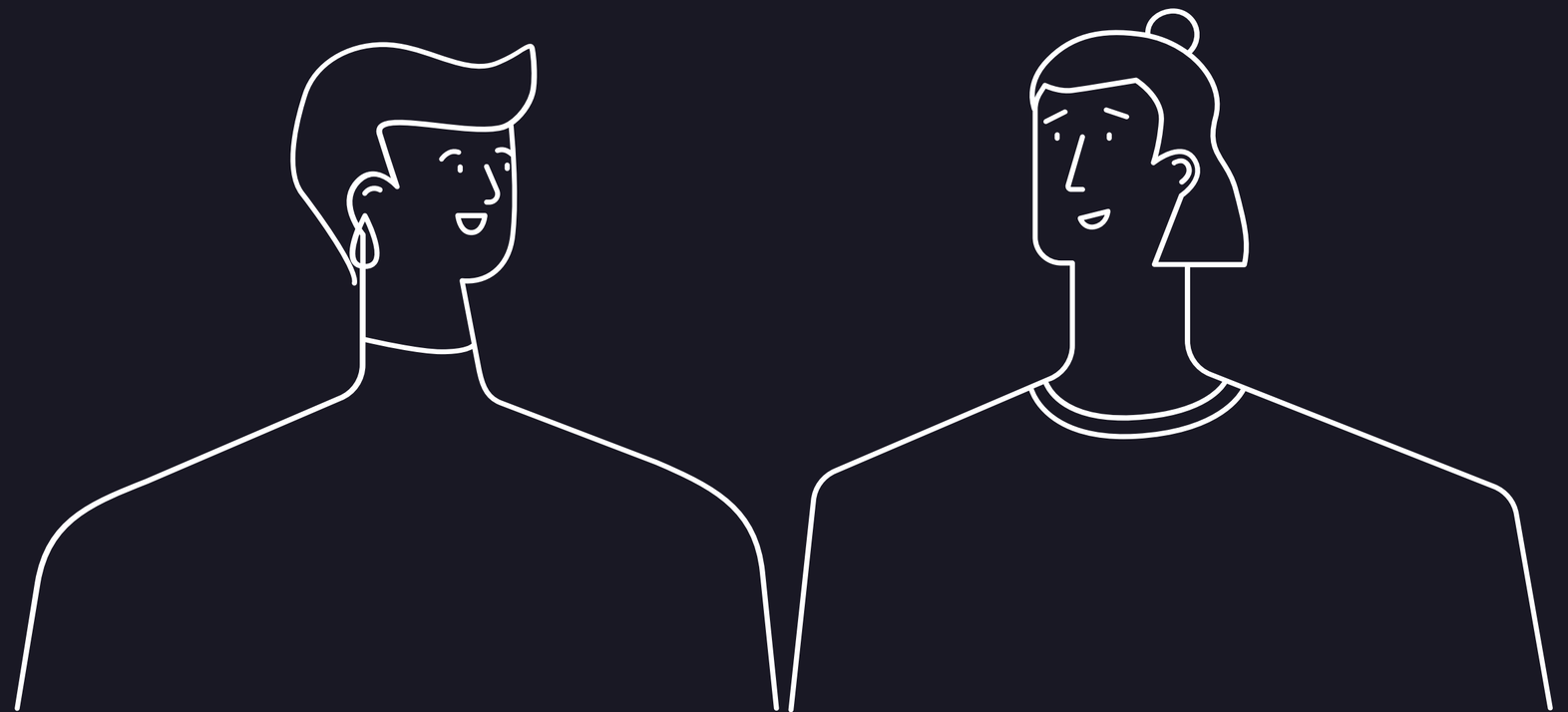
-React.js

Few more useful concepts-

- Pure Components
- useState()
- Prop Drilling
- Re-rendering of Component
- Destructuring props

EXERCISE TIME

Simple Counter App that displays a counter (initialised with 0) and has buttons to increment/decrement it by 1

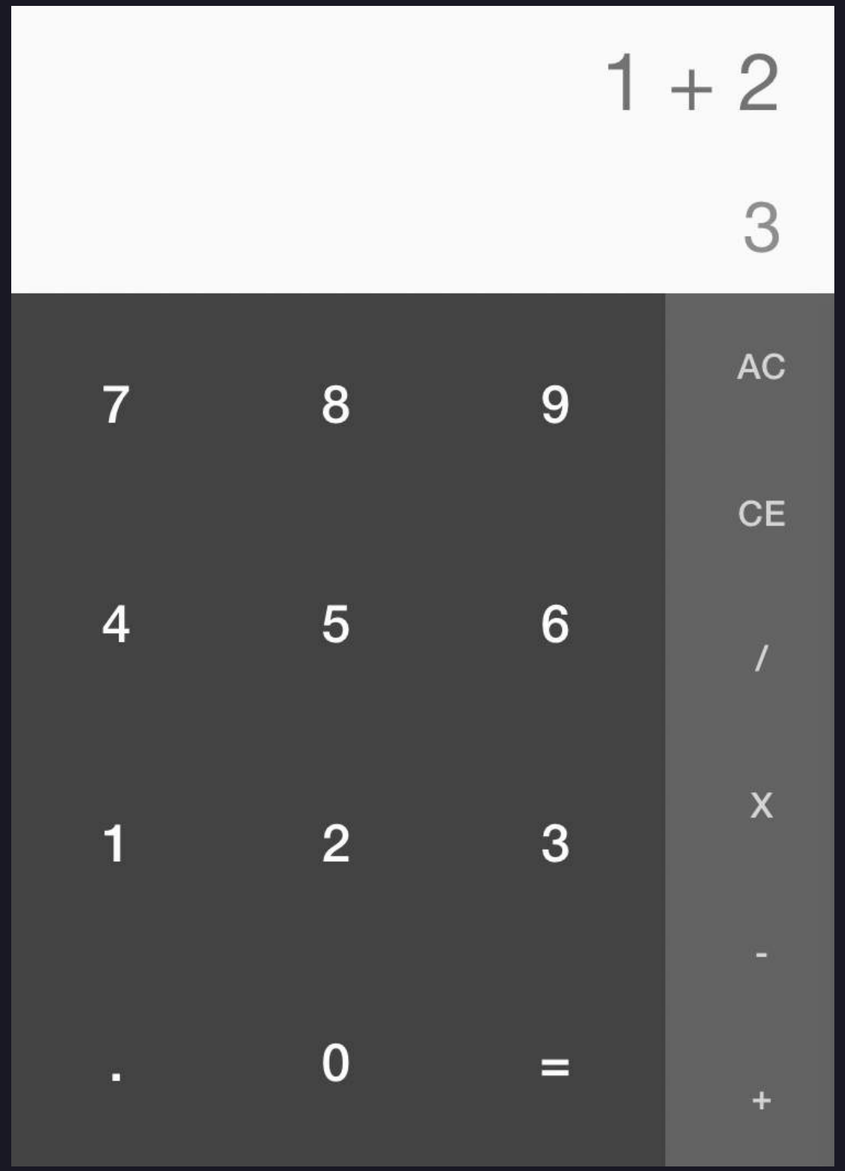


Build your first UI page

Day 2

Objectives

- Develop skills on composite components.
- Learn about conditional rendering
- What are component trees?
- Using HTML/CSS basics to build a calculator UI in React

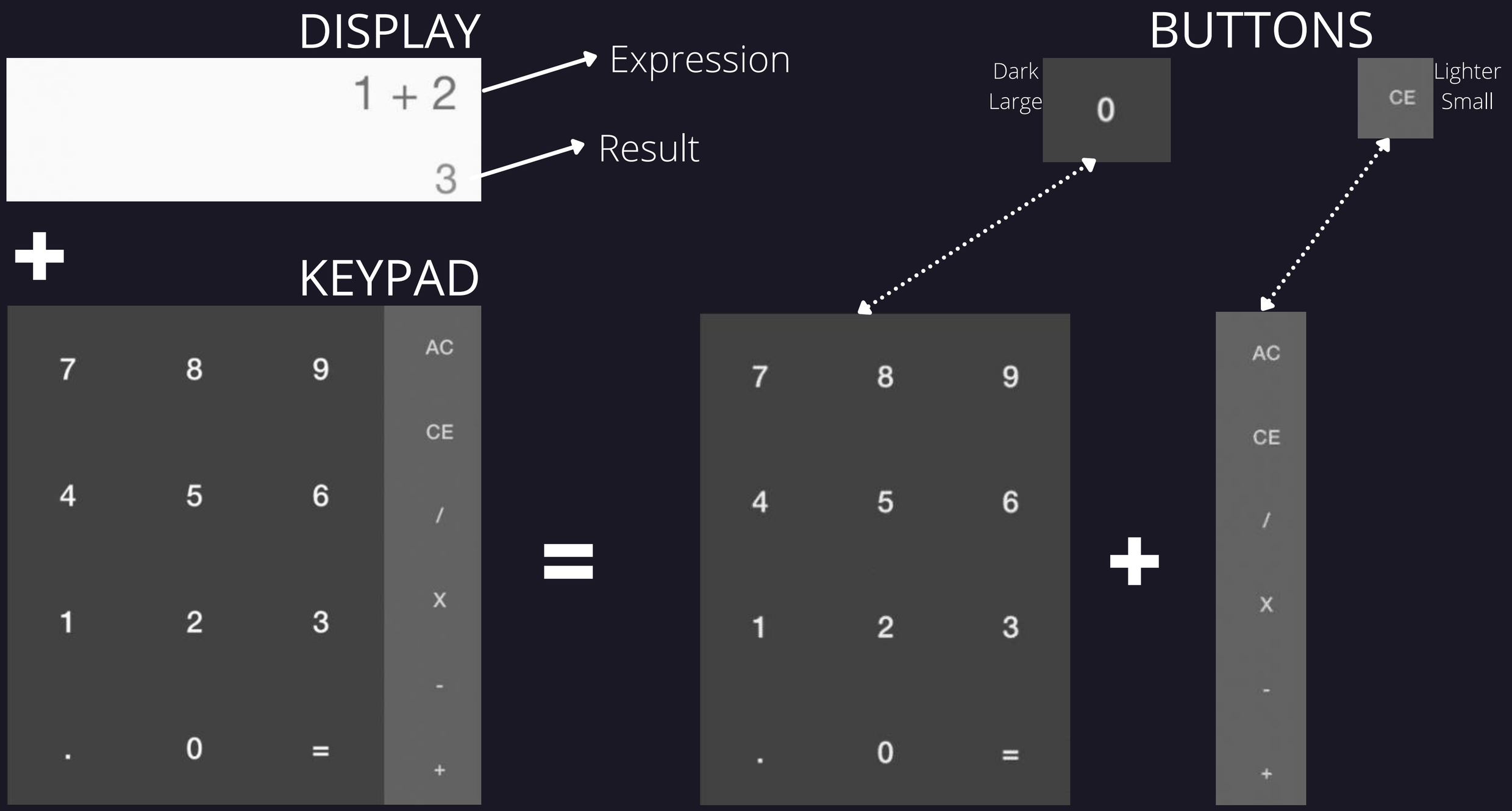


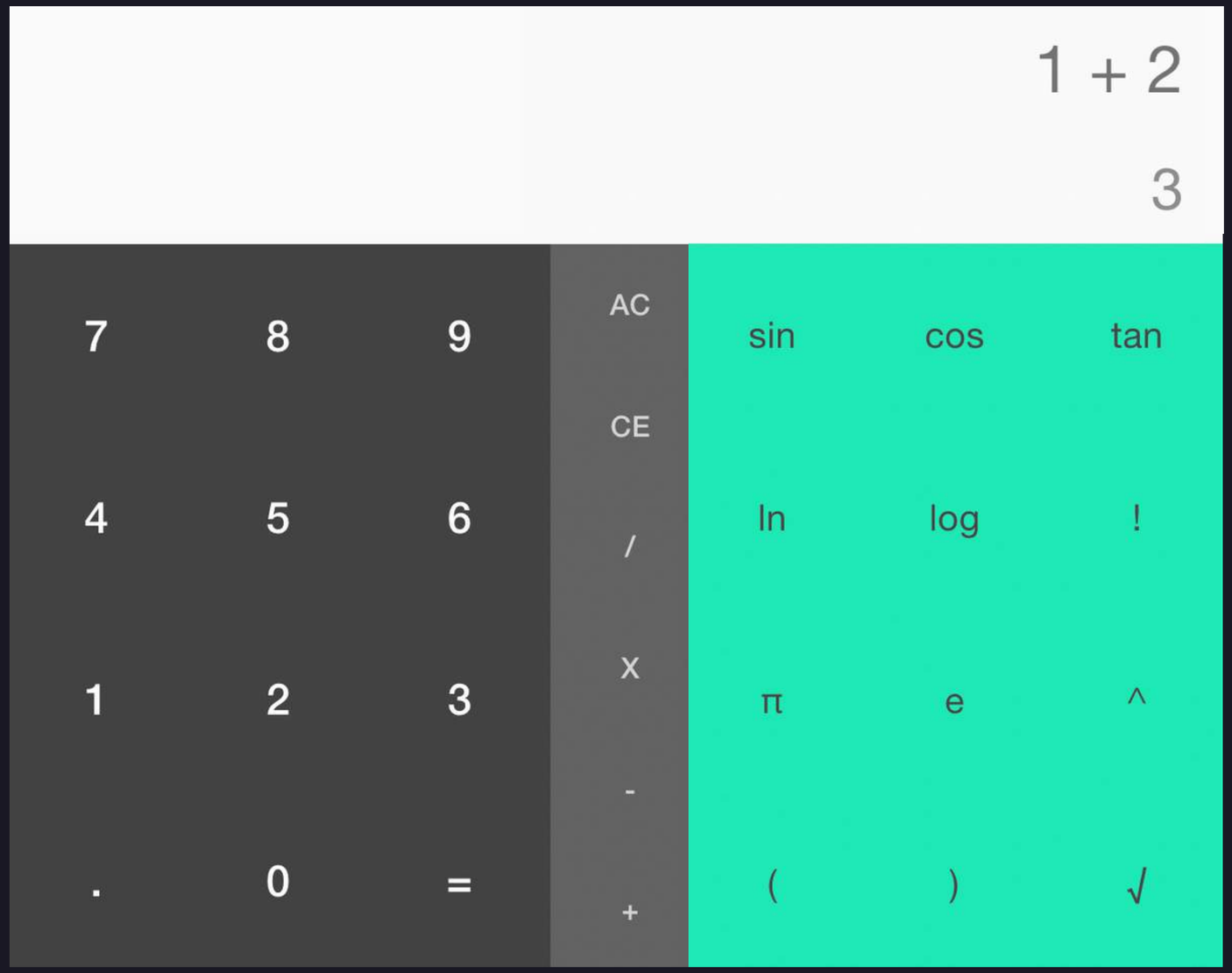
Basic Calculator UI

Component Breakdown

We have the following components -

- App component
- Display Component (Child of App)
- Keypad Component (Child of App)
- Button Component (Child of Keypad)


DISPLAY
1 + 2
3
Expression
Result
+
KEYPAD
7 8 9 AC
CE
4 5 6 /
1 2 3 X
-
. 0 = +
=
BUTTONS
Dark
Large
0
CE
Lighter
Small
7 8 9
4 5 6
1 2 3
. 0 =
+
AC
CE
/
X
-
+



Try Yourself

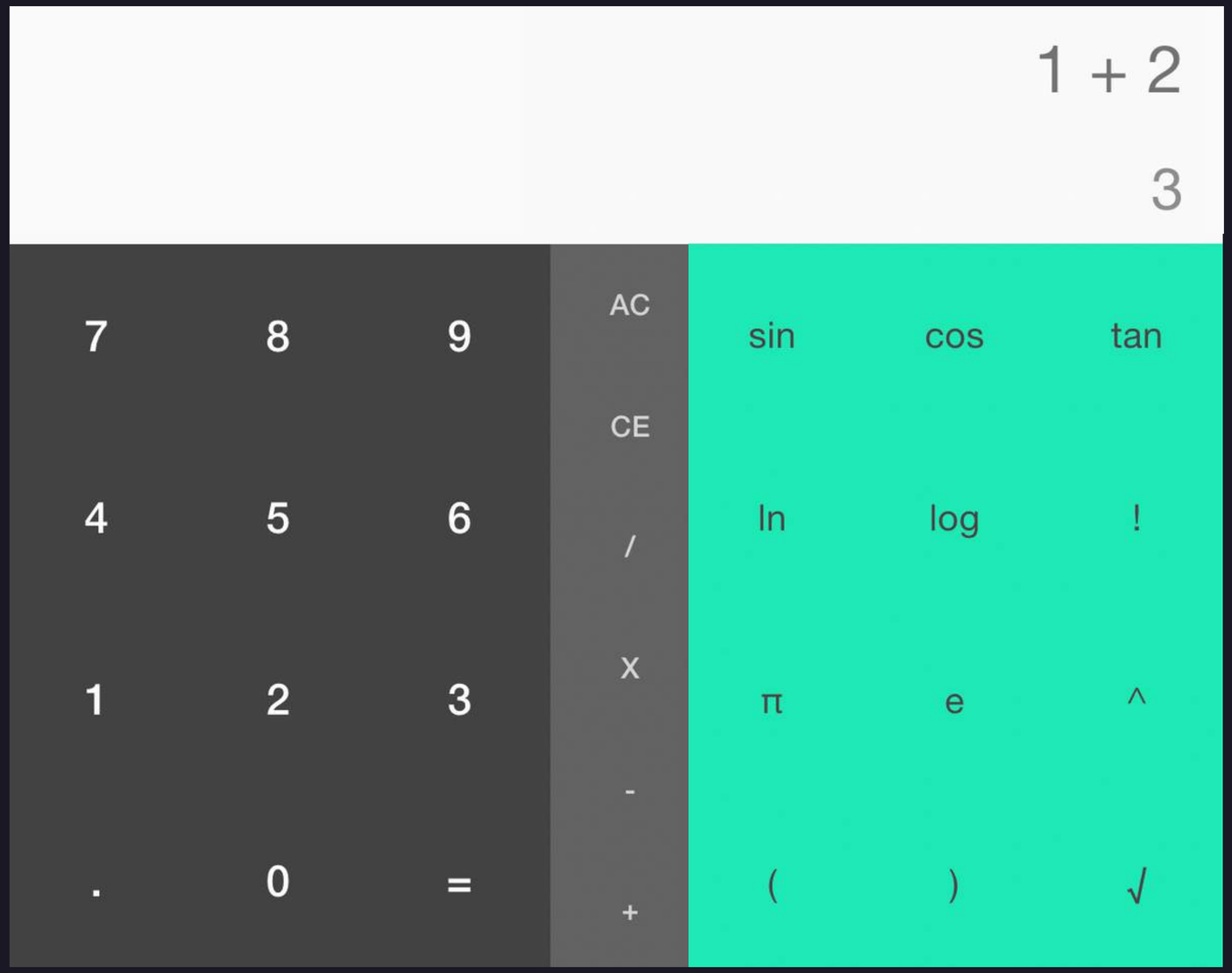
Scientific Calculator UI

Develop a full-fledged calculator

Day 3

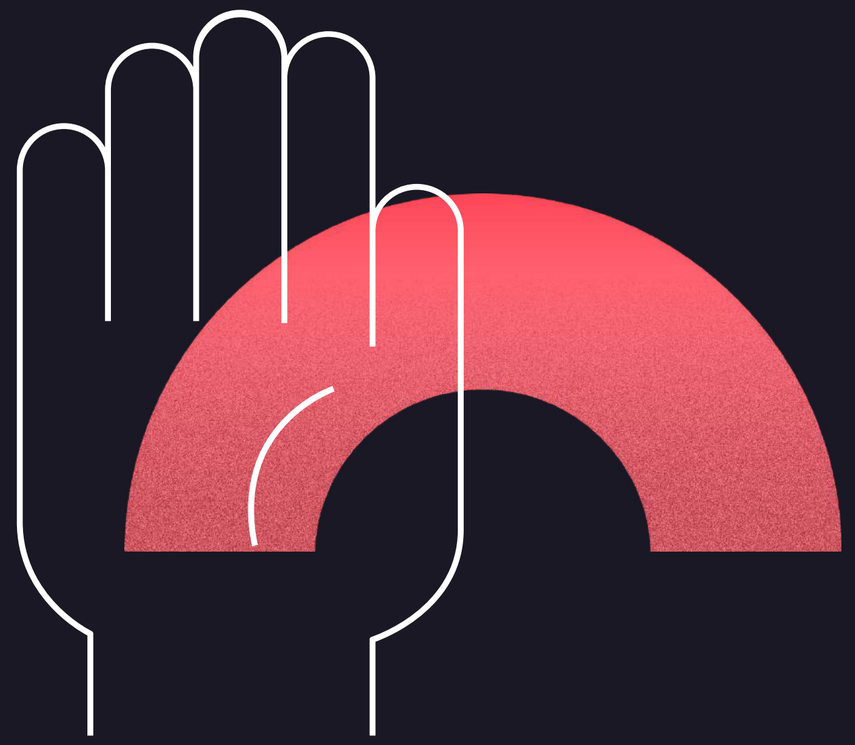
Objectives

- HTML DOM and Event Listener
- Adding dependencies to your project
- Using MathJs for evaluating answer
- Error Handling
- Keyboard Inputs



Scientific Calculator UI

STILL HAVING QUESTIONS ?



In case you still have questions/doubts which you encounter after the class while practising.
Please fill in the details on following Google Form-
https://bit.ly/3D2OomR

All your questions/doubts will be cleared in tomorrow's doubt clearing session.