# Styled Components

W12d3 – extra session from Helena

Styled-components allows you to use CSS in your JavaScript. It uses SASS so you can nest components inside.

Store this in a variable e.g. StyledButton

Call the library, then a . then whatever html you want e.g. styled.button

Then open a string with backticks and add CSS

Call the styled component instead of the JSX e.g. <StyledButton/>

Using functions to dynamically render a component that can be individually styled:

So app.js and button.js:

App.js

import Button from "./component/Button";

export default function App() {

return (

<>

<p>Hello world</p>

<Button background="red" color="green">

Hello from button 1

</Button>

<Button background="yellow">Hello from button 2</Button>

<Button background="blue">Hello from button 3</Button>

<Button>Boring Button</Button>

</>

);

}

Button.js

import styled from "styled-components";

export default function Button({ children, background, color }) {

return (

<StyledButton background={background} color={color}>

{children}

</StyledButton>

);

}

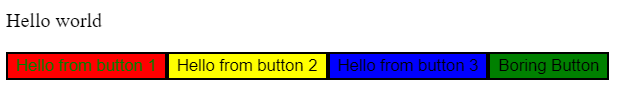
const StyledButton = styled.button`

*/\*CSS goes here\*/*

background-color: ${({ background }) => background || 'green'};

color: ${({ color }) => color};

`;

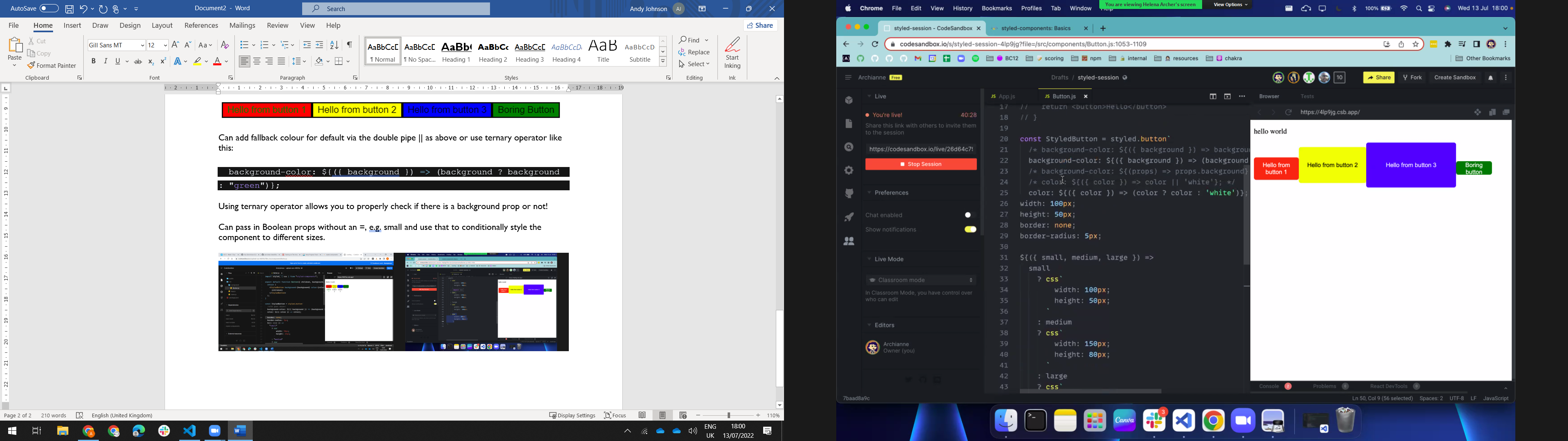


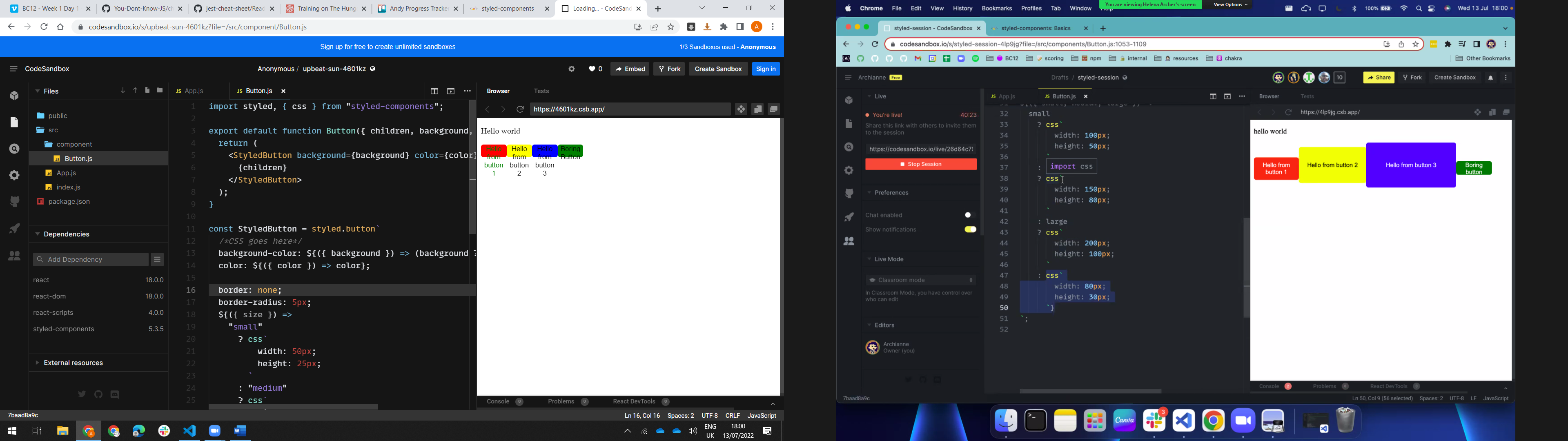
Can add fallback colour for default via the double pipe || as above or use ternary operator like this:

background-color: ${({ background }) => (background ? background : "green")};

Using ternary operator allows you to properly check if there is a background prop or not!

Can pass in Boolean props without an =, e.g. small and use that to conditionally style the component to different sizes:





My code:

App.js

import Button from "./component/Button";

export default function App() {

return (

<>

<p>Hello world</p>

<Button background="red" color="green" size="small">

Hello from button 1

</Button>

<Button size="medium" background="yellow">

Hello from button 2

</Button>

<Button size="large" background="blue">

Hello from button 3

</Button>

<Button>Boring Button</Button>

</>

);

}

Button.js

import styled, { css } from "styled-components";

export default function Button({ children, background, color, size }) {

return (

<StyledButton background={background} color={color} size={size}>

{children}

</StyledButton>

);

}

const StyledButton = styled.button`

*/\*CSS goes here\*/*

background-color: ${({ background }) => (background ? background : "green")};

color: ${({ color }) => color};

border: none;

border-radius: 5px;

${({ size }) =>

size === "small"

? css`

width: 50px;

height: 25px;

`

: size === "medium"

? css`

width: 100px;

height: 50px;

`

: css`

width: 150px;

height: 100px;

`}

`;