Styled Components

This library allows us to write css code and create a new React Component with those styles applied

# Install

* Npm install styled-components
* Special VS Code extension for styiling the css
  + Vscode-styled-components

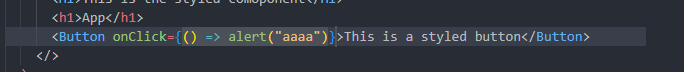
# Use

* Import styled from ‘styled-components’
* We are using the **styled** import to create the new component
* We are using the **``** to add the css (ES6 **tagged template literal**)
* This will return a new comoponent
  + We save the component
  + We can use the component

A screen shot of a computer program

Description automatically generated

* The Styled components can receive all the props and event handlers as a normal prop



* To customize the **COMPONENT ITSELF** (the main div) that is not an single component
  + Const StyledApp = styled.div`
  + Font
  + Background
  + `
* A screen shot of a computer program

  Description automatically generated

# Global Styles with Styled Components

* When using Styled Components, we don’t use the index.css
* We create a **brand new styled component that we will use as the GLOBAL component**
* For this, we use the **createGlobalStyle``** function from ‘styled-components’
  + We add the whole CSS file from index.css
  + Save it as a const
  + Export it
  + The GlobalStyles **cannot accept any children**
    - We have to define it as a sibling of the main component
    - A screen shot of a computer program

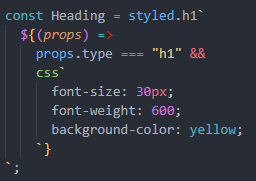
      Description automatically generated

# Styled Components with props and css function

* We can define JS functions inside the `` since it’s a template literal at base
* We can write some ‘’external’’ css
  + Cont test = `text-align: center `;
  + The issue is we will not have syntax highlighting
    - Here comes the **css function**
      * Test = css`text-align: center` 🡺 now we have syntax highlight

## Passing props to a styled comoponent

* We can receive the prop by using the **JS mode ${}**
  + ${props => props.type === h1}



* Use the **JS mode** **${}** to check the prop
* If it’s ‘h1’ we return the css code using the **css function**  and `` for syntax highlighting

# “AS” prop

* Using the method above, we style the headings in different ways, but in the HTML tree, it still shows as H1
* A screenshot of a computer code

  Description automatically generated
* Styled components offers the ‘**AS´property** and we can define what HTML we want to be shown
* A screenshot of a computer code

  Description automatically generated A screenshot of a computer program

  Description automatically generated