1. **Explore all the ways of writing css.**

* Inline CSS
* SCSS
* Native CSS (external files)
* Styled Components
* Libraries – MUI, Chakra UI, Base UI, Tailwind

1. **How do we configure taliwind?**

* Install Tailwind CSS
  + Install tailwindcss and its peer dependencies via npm, and then run the init command to generate tailwind.config.js.
  + npm install -D tailwindcss postcss
  + npx tailwindcss init
* Configure PostCSS
  + Create a .postcssrc file in your project root, and enable the tailwindcss plugin.
  + {

"plugins": {

"tailwindcss": {}

}

}

* Configure your template paths
  + Add the paths to all of your template files in your tailwind.config.js file.
  + module.exports = {

content: [

"./src/\*\*/\*.{html,js,ts,jsx,tsx}",

],

theme: {

extend: {},

},

plugins: [],

}

* Add the Tailwind directives to your CSS
  + Create a ./src/index.css file and add the @tailwind directives for each of Tailwind’s layers.
  + @tailwind base;

@tailwind components;

@tailwind utilities;

1. **In tailwind.config.js, what does all the keys mean (context, theme, plugins, extend)?**

* Content
* The content section is where you configure the paths to all of your HTML templates, JS components, and any other files that contain Tailwind class names.
* Theme
* The theme section is where you define your color palette, fonts, type scale, border sizes, breakpoints — anything related to the visual design of your site.
* Plugins
* The plugins section allows you to register plugins with Tailwind that can be used to generate extra utilities, components, base styles, or custom variants.
* Extend –
* If you’d like to preserve the default values for a theme option but also add new values, add your extensions under the extend key in the theme section of your configuration file.

1. **Why do we have .postcssrc file?**

* To tell bundler(parcel) that we are using tailwind classes so compile them to normal css while bundling.