



# Assignment 10 - Jo dikhta hai, vo bikta hai



Owner



Pankaj Kumar

## 1: Explore all the ways of writing css.

There are different ways of using CSS in our application.

1. Writing normal Native CSS : By creating `.css` files and adding them to our `html` files
2. By using CSS preprocessors like Sass or SCSS.
3. By writing inline CSS : writing CSS inside the components itself.
4. Using Libraries like Chakra UI, Tailwind CSS etc.

## 2: How do we configure Tailwind?

For configuring the Tailwind CSS with Parcel.

1. We need to install Tailwind library.

```
npm install -D tailwindcss postcss
```

1. Then we need to setup the config file for Tailwind.

```
npx tailwindcss init
```

This will create a `tailwind.config.js` file.

### 3: In tailwind.config.js, what does all the keys mean (content, theme, extend, plugins)?

The `tailwind.config.js` file contains the configurations for the our application. Following are the configurations we require to setup tailwind.

1. `content` : This configuration of the file formats, on which the styles are applied.

e.g.

```
content: [  
  "*/src/**/*.{html,js,ts,jsx,tsx}",  
],
```

This configuration means that in files extensions of `html , js, ts ,tsx, jsx` files will use the stylings of Tailwind.

2. `theme` : Theme is where we design our own custom configurations for our project like colors and font-families for our application.

```
module.exports = {  
  theme: {  
    screens: {  
      sm: '480px',  
      md: '768px',  
      lg: '976px',  
      xl: '1440px',  
    },  
    colors: {  
      'blue': '#1fb6ff',  
      'purple': '#7e5bef',  
      'pink': '#ff49db',  
    },  
  },  
}
```

3. `extend` : Here we can extend more properties for Tailwind like adding values that does not exist for tailwind or overriding the existing the values for tailwind.

```
extend: {  
  spacing: {  
    '128': '32rem',  
  },  
}
```

```

    '144': '36rem',
  },
  borderRadius: {
    '4xl': '2rem',
  }
}

```

4. **plugins** : We can use to inject new style in our project using JavaScript instead of CSS.

const plugin = require('tailwindcss/plugin')

```

module.exports = {
  plugins: [
    plugin(function({ addUtilities, addComponents, e, config }) {
      // Add your custom styles here
    }),
  ]
}

```

## 4: Why do we have **.postcssrc** file?

It is just a transpiler that turns a special PostCSS plugin syntax into a Vanilla CSS. You can think of it as the Babel tool for CSS. Which will contain the PostCSS settings.