

Styling in React

There are various ways of styling React components from using style sheets to using external styling libraries.

Inline CSS: In inline styling basically we create objects of style. And render it inside the components in style attribute using the React technique to use JavaScript variable inside the JSX (Using '{ }')

Player.js

```
import React, { Component } from 'react'
function Player(props){
  const {name, runs, match, age} = this.props; //object destructuring
  const ulStyle = {border: '2px solid blue', width:'50%',
listStyleType:'none',background-color:'#eeeeee',boxShadow:'5px 5px 5px gray'}
  const liStyle = {color: 'blue', fontSize:'1.5rem'}
  return(
    <ul style={ulStyle}>
      <li style={liStyle}> Name : {name}</li>
      <li style={liStyle}> Runs: {runs}</li>
      <li style={liStyle}> Match: {match}</li>
      <li style={liStyle}> Age : {age}</li>
    </ul>
  )
}
export default Player
```

index.js

```
import ReactDOM from 'react-dom/client';
import Player from './Player';
function InlineStyleExample(){
  return (
    <div>
      <Player name='Virat Kohli' runs='20000' match='200' age='35' />
      <Player name='Hardik Pandya' runs='30000' match='300' age='28' />
      <Player name='Rohit Sharma' runs='35000' match='350' age='38' />
    </div>
  )
}
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<InlineStyleExample />);
```

Normal CSS: In the external CSS styling technique, we basically create an external CSS file for each component and apply styling using class. And use those class names inside the component. It is a convention (not rule) that names of the external CSS file should be same as the name of the component but extension of the file must be '.css'.

Player.css

```
.PlayerList{
  border: 2px solid green;      width: 40%;
  list-style-type: none;      background-color: aliceblue;
}
.player-details{
  color: blue;      font-size: 1.5rem;
}
```

Player.js

```
import React, { Component } from 'react'
import './Player.css'
class Player extends Component{
  render(){
    const {name, runs, match, age} = this.props
    return(
      <ul className='PlayerList'>
        <li className='player-details'> Name: {name}</li>
        <li className='player-details'> Runs: {runs}</li>
        <li className='player-details'> Match: {match}</li>
        <li className='player-details'> Age : {age}</li>
      </ul>
    )
  }
}
export default Player
```

index.js

```
import ReactDOM from 'react-dom/client';
import Player from './Player'
function InlineStyleExample(){
  return (
    <div>
      <Player name='Virat Kohli' runs='20000' match='200' age='35' />
      <Player name='Hardik Pandya' runs='30000' match='300' age='28' />
      <Player name='Rohit Sharma' runs='35000' match='350' age='38' />
    </div>
  )
}
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<InlineStyleExample />);
```

Styled Components: The styled-components allow us to style the CSS inside the variable created in JavaScript. style components is a third party package using which we can create a component as a JavaScript variable that is already styled with CSS code and used that styled component in our main component. styled-components allow us to create custom reusable components which can be easy to maintain. This requires third party package styled components.

```
npm install --save styled-components
```

```
Player.js
import styled from 'styled-components'
//styled component Li
const Li = styled.li`
  color : blue;
  font-size : 23px
`

//Styled component Ul
const Ul = styled.ul`
  border : 2px solid green;
  width: 40%;
  list-style-type:none;
`

function Player(props){
  const {name, run, match,age} = props; //object destructrin
  return(
    <Ul>
      <Li>Name: {name}</Li>
      <Li>Run: {run}</Li>
      <Li>Match: {match}</Li>
      <Li>age: {age}</Li>
    </Ul>
  )
}
export default Player
```

index.js

```
import ReactDOM from 'react-dom/client';
import Player from './Player'
function StyledComponents(){
  return (
    <div>
      <Player name='Virat Kohli' runs='20000' match='200' age='35' />
      <Player name='Hardik Pandya' runs='30000' match='300' age='28' />
      <Player name='Rohit Sharma' runs='35000' match='350' age='38' />
    </div>
  )
}
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<StyledComponents />);
```