ChildComponent.js

import React from 'react';

class ChildComponent extends React.Component {

constructor(props){

super(props);

}

render() {

  let { inputVal } = this.props;

return (

<container>

<input type="text" value={inputVal} readonly />

<h2 style={{color:inputVal}}>Child Component Input Value Color is {this.props.inputVal}</h2>

</container>

);

}

}

export default ChildComponent

Hello.js

import React from 'react';

export default ({ name }) => <h1>Hello {name}!</h1>;

index.js

import React, { Component } from 'react';

import { render } from 'react-dom';

import ParentComponent from './ParentComponent';

import './style.css';

class App extends Component {

constructor() {

super();

}

render() {

return (

<div>

<ParentComponent />

<p>

Start editing to see some magic happen :)

</p>

</div>

);

}

}

render(<App />, document.getElementById('root'));

ParentComponent.js

import React from 'react';

import ChildComponent from './ChildComponent';

class ParentComponent extends React.Component {

constructor(props){

super(props);

this.state = {

color: "blue"

};

    this.onChangeTextColor = this.onChangeTextColor.bind(this);

}

  onChangeTextColor(e) {

var newColor = e.target.value;

this.setState({color: newColor});

}

render() {

return (

<container-wrapper>

<ChildComponent inputVal={this.state.color} />

        Enter your color name <input type="text" value={this.state.color} onChange={(e) => this.onChangeTextColor(e)}/>

<h5>We have set input value is {this.state.color} inside Parent Component</h5>

</container-wrapper>

);

}

}

export default ParentComponent

{

"name": "react-tbul4t",

"version": "0.0.0",

"private": true,

"dependencies": {

"react": "16.1.1",

"react-dom": "16.1.1"

},

"scripts": {

"start": "react-scripts start",

"build": "react-scripts build",

"test": "react-scripts test --env=jsdom",

"eject": "react-scripts eject"

},

"devDependencies": {

"react-scripts": "latest"

}

}

h1, p {

font-family: Lato;

}

<div id="root"></div>