

Cheatsheets / Learn ReactJS: Part II

Advanced React

React CSS styles

React supports inline CSS styles for components. Styles are supplied as a `style` prop to components. Inline styles can be written as attributes in the opening tag of a JSX element, while multi-line styles can be stored in a variable and that variable can be invoked into JSX.

```
// Inline CSS
<h1 style={{ color: 'red' }}>I am red!</h1>

// CSS styles stored in a variable and called in
const color = {
  color: 'blue',
  background: 'sky'
}
<h1 style={color}>Hello</h1>
```

Style Names And Values

In React, style names are written in camelCase, unlike in CSS where it's hyphenated. In most cases, style values are written in strings. When entering number values you don't have to enter "px" because React automatically adds "px" to make them pixel values.

```
//hyphenated style name
font-size:

//camelCase style name
fontSize:

//style values written in string form
color: 'blue';
```

Type Checking

In React, `propTypes` can be used to perform type-checking on props. This gives developers the ability to set rules on what data/variable type each component prop should be and to display warnings when a component receives invalid type props.

Static Property

In React, prop types are set as a static property(`.propTypes`) on a component class or functional component. `.propTypes` is an object with property names matching the expected props and values matching the expected value for that prop type. The code snippet above demonstrates how `.propTypes` can be applied.

```
class Birth extends React.Component {  
  render() { return <h1>{this.props.age}</h1> }  
}  
  
MyComponent.propTypes = {  
  age: React.PropTypes.number  
}
```

.isRequired

If a developer wants to indicate that a prop is required by the component, then the property `.isRequired` can be chained to prop types. Doing this will display a warning in the console if the prop is not passed. The code snippet above demonstrates the use of `.isRequired`.

```
MyComponent.propTypes = {  
  year: React.PropTypes.number.isRequired  
};
```