Building Rich Front-End Applications with React and ES6
Module 1 Building Rich Front-End Applications with React and ES6

Package/Method	Description	Code Example
let and const	let allows you to restrict the scope of variables within the block where they are declared. const allows you to declare constants whose values cannot be changed.	1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10 11. 11 1. { 2. let a = 10 3. console.log(a) 4. a = 15 5. console.log(a) 6. } 7. console.log(a) 8. const num = 5 9. console.log(num) 10. num = 8 11. console.log(num) Copied!
Arrow function	Arrow functions allow you to write shorter function syntax.	1. 1 2. 2 3. 3 4. 4 1. hello = () => 2. { 3. return "Hello World!"; 4. } Copied!

Promises	The Promise object represents the eventual completion (or failure) of an asynchronous operation and its resulting value.	1. 1
		2. 2
		3. 3
		4. 4
		5. 5
		6. 6
		7. 7
		8. 8
		9. 9
		10. 10
		11. 11
		12. 12
		
		1. let promiseArgument = (resolve, reject) =>
		2. setTimeout (() => {
		3. let currTime = new Date().getTime();
		4. if(currTime % 2 === 0){
		5. resolve(“Success”)
		6. }else{
		7. Reject(“Failed!!!”)}
		8. }
		9. ,2000)
		10. }
		11. Let myPromise = new Promise(promiseArgument);
		12.
		
		Copied!

class	Class is a template or blueprint for creating object.	1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8

```
9. 9
10. 10
```

```
1. function car(name,year)
2. {
3.     this.name = name
4.     this.year = year
5.     return this;
6. }
7. Let car = car("Ford", 2014)
8. console.log(car)
9. console.log(car.name)
10. console.log(car.year)
```

Copied!

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
```

Inheritence

A class created with a class inheritance, inherits all the methods from another class.

```
1. class Square extends Rectangle
2. {
3.     constructor(height,width)
4.     {
5.         if(height === width)
6.         {
7.             super(height,width)
8.         }
9.         else
10.         {
11.         Super(width,width)
12.         }
13.     }
14. }
15. Let mySquare = new Square(5,5)
```

Copied!

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
```

React components

Components are reusable segments of code that come under the class and functional component types.

```
1. importReact from 'react';
2. import {Text} from 'react-native’;
3. constHelloworld= ()=>
4. {
5.     return
6.     (Hello, World!);
7. }
8. exportdefault Helloworld;
```

Copied!

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
```

React class Component

React class component contains-
Props: set from outside the class
State: internal to the class

```
1. import React from "react";
2. class App extends React.Component {
3. constructor(props) {
```

```
4. super(props);
5. this.state={change: true };
6. }
7. render() {
8. return(
9. <buttononClick={()=>{this.setState({change: !this.state.change});}}>Click Here!</button>
10. {this.state.change?(Hello!!):(Welcome to the React Course)}
    );}}
11. export default App;
```

Copied!

onClick	When an event fires, event handlers decide what should happen next. This could involve pressing a button or altering a text entry.

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 7
7. 8
8. 9
9. 10
```

```
1. function changeColor() {
2. const shoot = () => {
3.     alert("Color Changed!");
4. }
5. return (
6. <button onClick={change}>Change the Color! </button>
7. );
8. }
9. const root = ReactDOM.createRoot(document.getElementById('root'));
10. root.render(<changeColor />);
```

Copied!

Changelog

Date	Version	Changed by	Change Description
20-10-2022	1.1	Sapthashree K S	Cheatsheet updated