Component Types

Stateless Functional Component

JavaScript Functions

```
function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}
```

Stateful Class Component

Class extending Component class

Render method returning HTML

```
class Welcome extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}</h1>;
  }
}
```

Components Summary

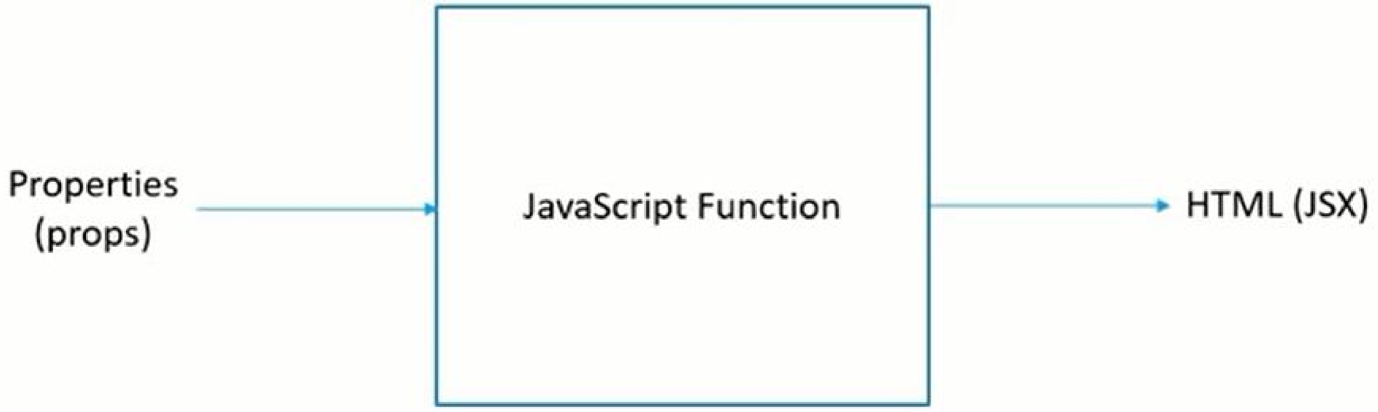
Components describe a part of the user interface

They are re-usable and can be nested inside other components

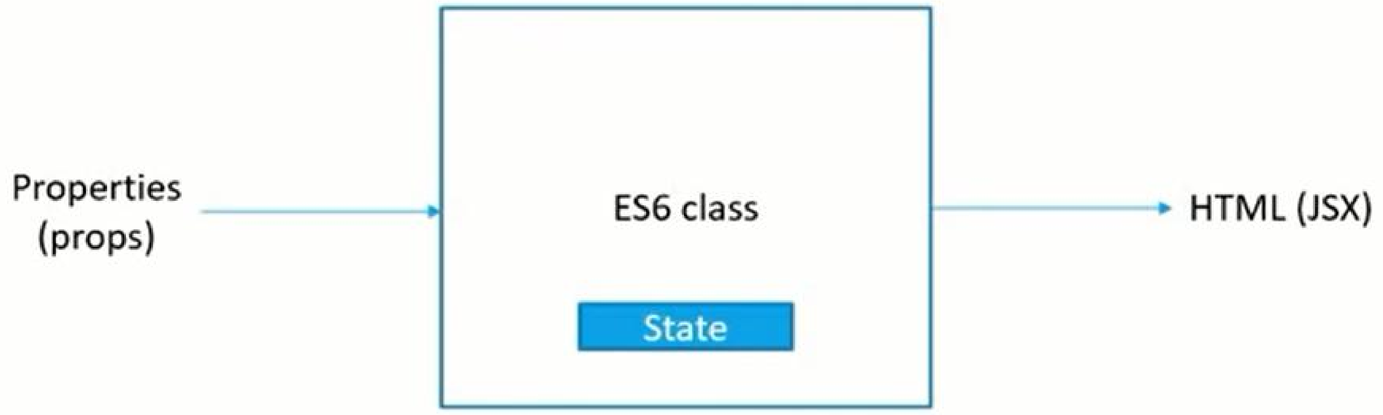
Two Types –

- Stateless Functional Components
- Stateful Class Components

Functional Components



Class Components



Functional vs Class components

Functional

Simple functions

Use Func components as much as possible

Absence of ‘this’ keyword

Solution without using state

Mainly responsible for the UI

Stateless/ Dumb/ Presentational

Class

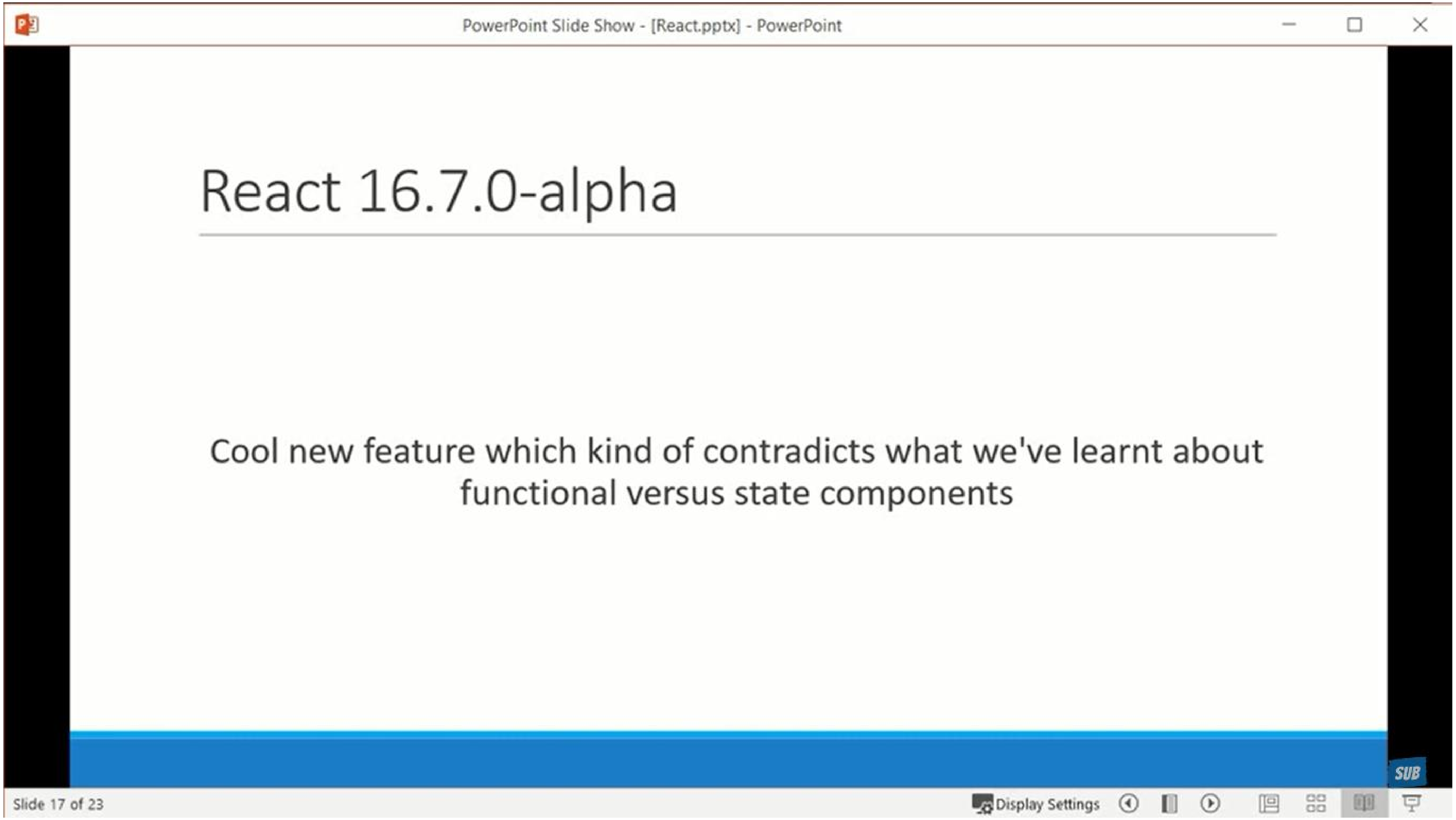
More feature rich

Maintain their own private data - state

Complex UI logic

Provide lifecycle hooks

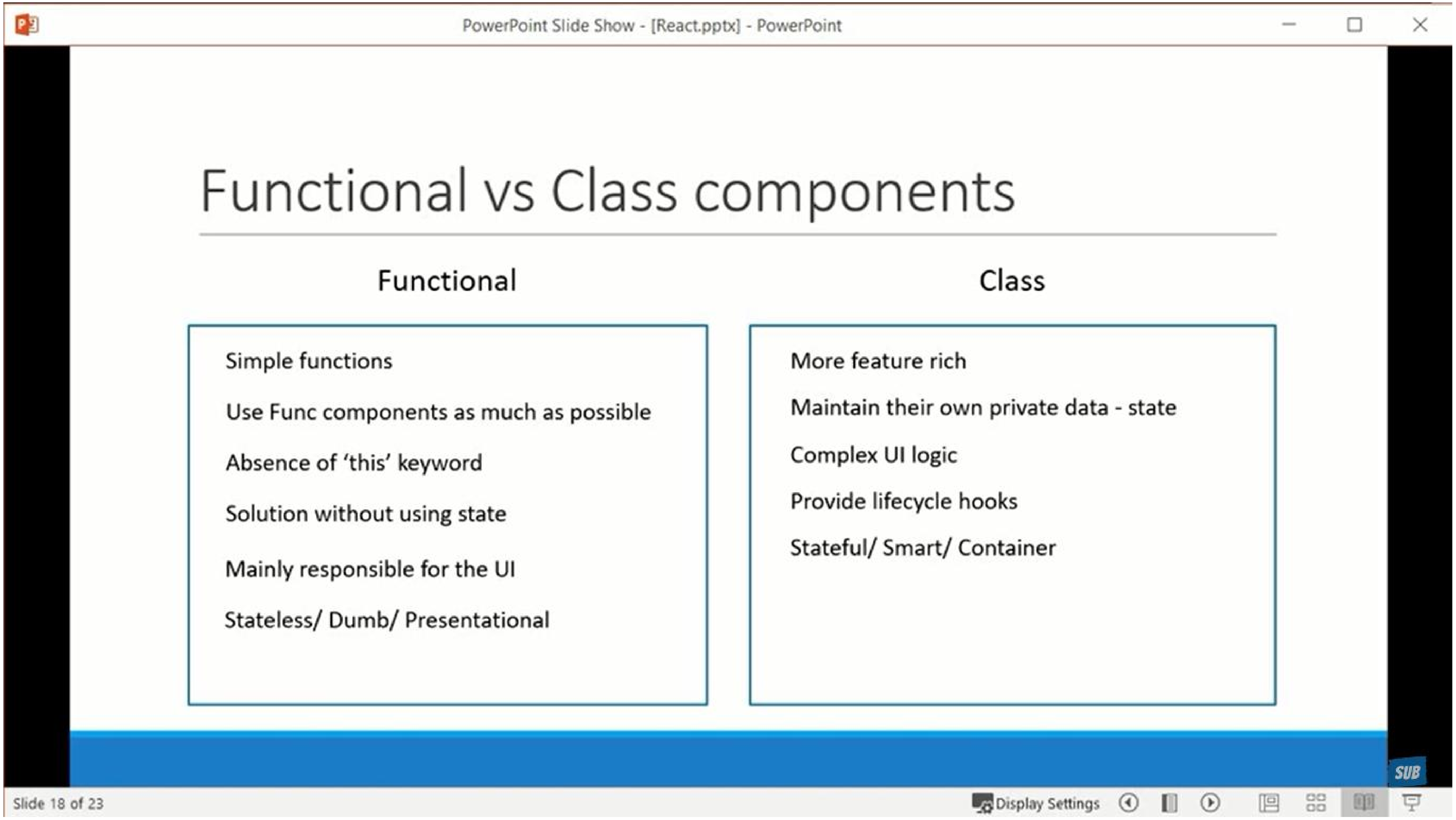
Stateful/ Smart/ Container



PowerPoint Slide Show - [React.pptx] - PowerPoint

React 16.7.0-alpha

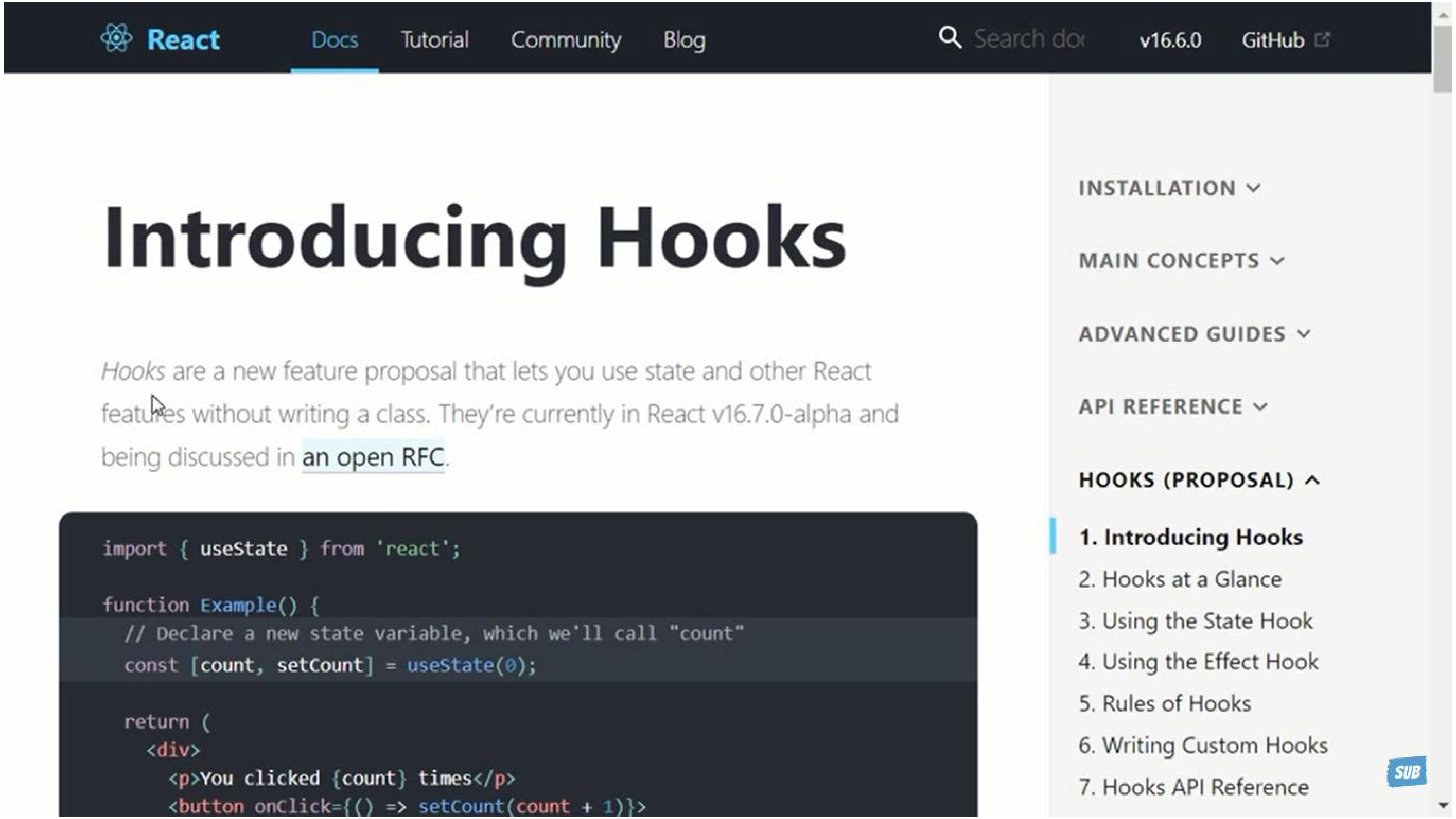
Cool new feature which kind of contradicts what we've learnt about functional versus state components


PowerPoint Slide Show - [React.pptx] - PowerPoint

Functional vs Class components

Functional	Class
Simple functions	More feature rich
Use Func components as much as possible	Maintain their own private data - state
Absence of 'this' keyword	Complex UI logic
Solution without using state	Provide lifecycle hooks
Mainly responsible for the UI	Stateful/ Smart/ Container
Stateless/ Dumb/ Presentational	

Slide 18 of 23



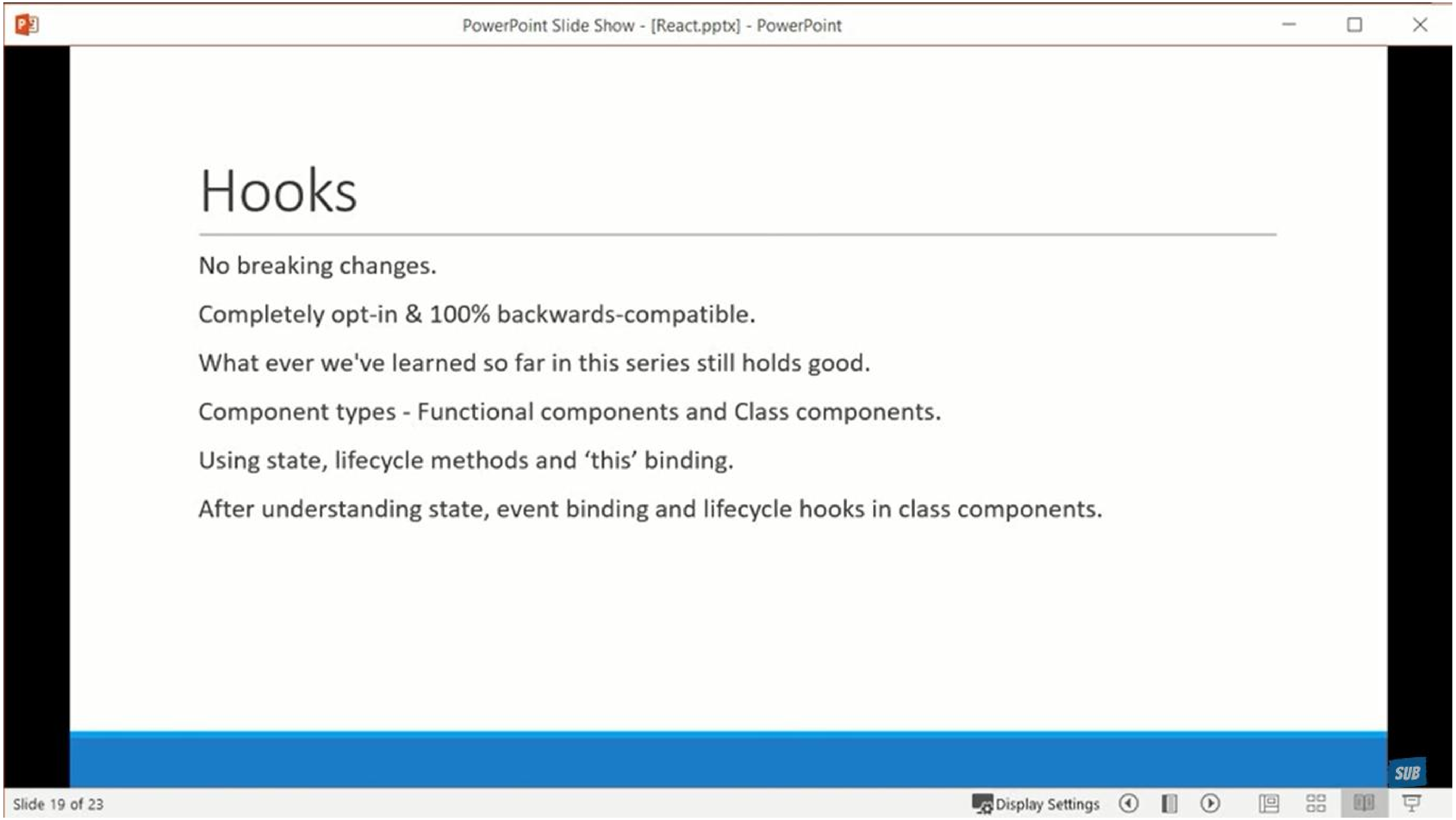
Introducing Hooks

Hooks are a new feature proposal that lets you use state and other React features without writing a class. They're currently in React v16.7.0-alpha and being discussed in an open RFC.

```
import { useState } from 'react';

function Example() {
  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
```



PowerPoint Slide Show - [React.pptx] - PowerPoint

Hooks

No breaking changes.

Completely opt-in & 100% backwards-compatible.

What ever we've learned so far in this series still holds good.

Component types - Functional components and Class components.

Using state, lifecycle methods and 'this' binding.

After understanding state, event binding and lifecycle hooks in class components.

JSX

JavaScript XML (JSX) – Extension to the JavaScript language syntax.

Write XML-like code for elements and components.

JSX tags have a tag name, attributes, and children.

JSX is not a necessity to write React applications.

JSX makes your react code simpler and elegant.

JSX ultimately transpiles to pure JavaScript which is understood by the browsers.

JSX differences

Class -> className

for -> htmlFor

camelCase property naming convention

- onclick -> onClick
- tabindex -> tabIndex