

# REACT - CONTEXT

Context umožňuje poslat dolů data bez jejich předávání po jednotlivých komponentách. Měli bychom jej ale používat střídavě, protože se potom mohou hůře znovupoužívat komponenty.

## VYTVOŘENÍ CONTEXTU

Context můžeme vytvořit pomocí funkce `React.createContext`, které předáme defaultní hodnotu, kterou by měl context uchovávat.

```
const ThemeContext =
  React.createContext('light');

class App extends React.Component {
  render() {
    // Pokud chceme některým komponentám,
    // které context používají, nastavit
    // jinou než defaultní hodnotu, tak
    // je obalíme do Providera contextu
    return (
      <ThemeContext.Provider
        value="dark"
      >
        <Toolbar />
      </ThemeContext.Provider>
    );
  }
}
```

## POUŽITÍ CONTEXTU

Pokud chceme v některé komponentě použít context, tak této komponentě musíme nastavit statickou vlastnost `contextType` na context, který chceme použít.

K hodnotě contextu se potom dostaneme pomocí `this.context`.

```
class ThemedButton extends React.Component {
  static contextType = ThemeContext;
  render() {
    return <Button theme={this.context}/>;
  }
}
```

## MĚNĚNÍ CONTEXTU

Pokud chceme v contextu něco měnit, tak si na to můžeme v contextu vytvořit funkci.

```
class App extends React.Component {
  constructor(props) {
    super(props);

    // funkce pro měnění statu (vlastně
    // contextu)
    this.toggleTheme = () => {
      this.setState(state => ({
        theme:
          state.theme === themes.dark
            ? themes.light
            : themes.dark
      }));
    };

    // state se nastavuje jako value
    // contextu
    this.state = {
      theme: themes.light,
      toggleTheme: this.toggleTheme
    };
  }

  render() {
    return (
      <ThemeContext.Provider
        value={this.state}
      >
        <Content />
      </ThemeContext.Provider>
    );
  }
}
```

## CONTEXT.CONSUMER

Komponenta `Context.Consumer` bere funkci s hodnotou contextu. Lze ji použít namísto `contextType`.

```
<MyContext.Consumer>
  {value => /* ... */}
</MyContext.Consumer>
```