

# React Component

Intro to building a component with React

# About React

- React is an open source JavaScript library
- Used for building UI components.
- Maintained by Facebook
- The main idea: You create component with a state, when state is changed the component will rerender.
- Together with Application Architecture Design like Flux or Redux becomes a powerful tool for creating complex web applications

# Why React

- Fast DOM manipulation
- Small footprint
- Fast loading time
- Large community support
- Usually combined with better application architecture than MVC

# Is it production ready?



Instagram



# React License

From September 2017 React is licensed under MIT license



facebook/react is licensed under the

## MIT License

A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

### Permissions

- ✓ Commercial use
- ✓ Modification
- ✓ Distribution
- ✓ Private use

### Limitations

- ✗ Liability
- ✗ Warranty

### Conditions

- ⓘ License and copyright notice

This is not legal advice. [Learn more about repository licenses.](#)

# Hello World with React

- To create our React component we need to load the React script
- A React component extends `React.Component`
- The class render method determine the UI representation of the component
- The render method needs to return `React.createElement`

# ReactDOM

- To render the component to the DOM we will need to load the ReactDOM script
- The ReactDOM.render method will render our component to the DOM

# React-JSX

- JSX stands for JavaScript XML
- We can use it to add HTML like syntax to JavaScript
- React-JSX can be used to replace `React.createElement`
- We can easily inject JS code in our JSX using `{ <JS code> }`
- Browsers don't understand JSX
- We need to transpile the JSX to proper JS code
- We will use Babel to transpile our code



# Hello World JSX

- We will now use React-JSX to improve our Hello World Component
- Load the babel core from CDN
- when loading the hello world script use `type="text/babel"`
- this means that those scripts at runtime will be transpiled by Babel

# What else can a React Component Do?

- Properties
- Events
- State
- Ref

# React Component - Properties

- You can pass data from parent to child using properties
- properties will arrive at component: **this.props**
- With JSX properties are passed similar to attributes in HTML
- if attribute has string syntax like HTML then the attribute will be passed as string
- if the attribute has curly braces than it will be passed as JS expression
- you can use **static propTypes** in the component to verify the properties are sent
- To demonstrate let's extend the hello world component to receive property of the message to display
- Can we use it to pass data from child to parent?

# React Component - Events

- component can have also forms, buttons, and events that will change the state
- events are added like DOM events only **camelCased**
- events get curly braces js inject not like HTML where we pass string
- The original DOM event will be passed to the function
- It's common practice that events will be class methods
- Careful by default the context of the events will be similar to DOM events (this will be undefined)
- Let's add a button that will pop an alert with the message we got from the parent

# React Component - state

- Each component in React has a private state
- You init the state in the constructor
- You have to use **this.setState** if you want to cause the component to rerender
- The **setState** can get a dictionary which will be merged with the current state
- The **setState** can also get a function which will be called with the current state and return the dictionary of the new state
- Let's add a toggle visibility button in our hello world component

# React Component - Ref

- ref is used to create a reference to DOM element or React Component
- ref accepts a function that will be called with DOM element or React class instance
- This function will be called after component is mounted
- It's common usage to use ref to grab input values
- Let's extend the hello world component and add a form with text input and a list that will display the things we type

# Student Ex

## Todo app with no server

- App will contain 3 components
  - app component
  - todo form component
  - todo list component
- app component will hold the list of tasks
- app component will pass to the todo form a callback that will add a new task to the list of tasks
- todo list will get the list of tasks from the parent app component and will display a list of tasks