

JAVASCRIPT COURSE

PART 12 – 23.11.2017.



IN THIS CLASS

- Introduction to React.
- Homework 😊

BEFORE WE BEGIN

Presentations and homework:

<https://github.com/JSBelgrade/course-2017>

PREVIOUS HOMEWORK



NEXT HOMEWORK

CREATE A TODO API

API should have following routes:

- **GET** `/todos` // return all items
- **GET** `/todos/{id}` // return a single item
- **POST** `/todos` // add a new item
- **PUT** `/todos/{id}` // update an item
- **DELETE** `/todos/{id}` // delete an item

EXAMPLE RESPONSES

GET /todos

```
[  
  {  
    id: 1,  
    text: 'Prepare homework',  
    completed: true  
  },  
  {  
    id: 2,  
    text: 'Prepare presentation',  
    completed: false  
  }  
]
```

GET /todos/1

```
{  
  id: 1,  
  text: 'Prepare homework',  
  completed: true  
}
```

POST /todos

Body:

```
{  
  text: 'Buy milk'  
}
```

Response:

201 Created

```
{  
  id: 3,  
  text: 'Buy milk',  
  completed: true  
}
```

PUT /todos/2

Body:

```
{  
  completed: true  
}
```

Response: status 200

```
{  
  id: 2,  
  text: 'Prepare presentation',  
  completed: true  
}
```

DELETE /todos/2

Response: status 200
{ }

INTRO TO REACT

React.js is a JavaScript library
for building user interfaces

<https://reactjs.org>

Not a framework?

Sample code

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

```
ReactDOM.render(  
  <HelloMessage name="Slobodan" />,  
  document.getElementById("root")  
);
```

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

```
ReactDOM.render(  
  <HelloMessage name="Slobodan" />,  
  mountNode  
);
```

JSX SYNTAX

JSX is a preprocessor step that adds XML **syntax** to JavaScript. You can definitely use React without **JSX** but **JSX** makes React a lot more elegant.

```
<div>  
  Hello <i>friends</i>!  
</div>
```

```
React.createElement(  
  "div",  
  null,  
  "Hello ",  
  React.createElement(  
    "i",  
    null,  
    "friends"  
  ),  
  "!"  
);
```

Just like XML, **JSX** tags have a tag name, attributes, and children. If an attribute value is enclosed in quotes, the value is a string.

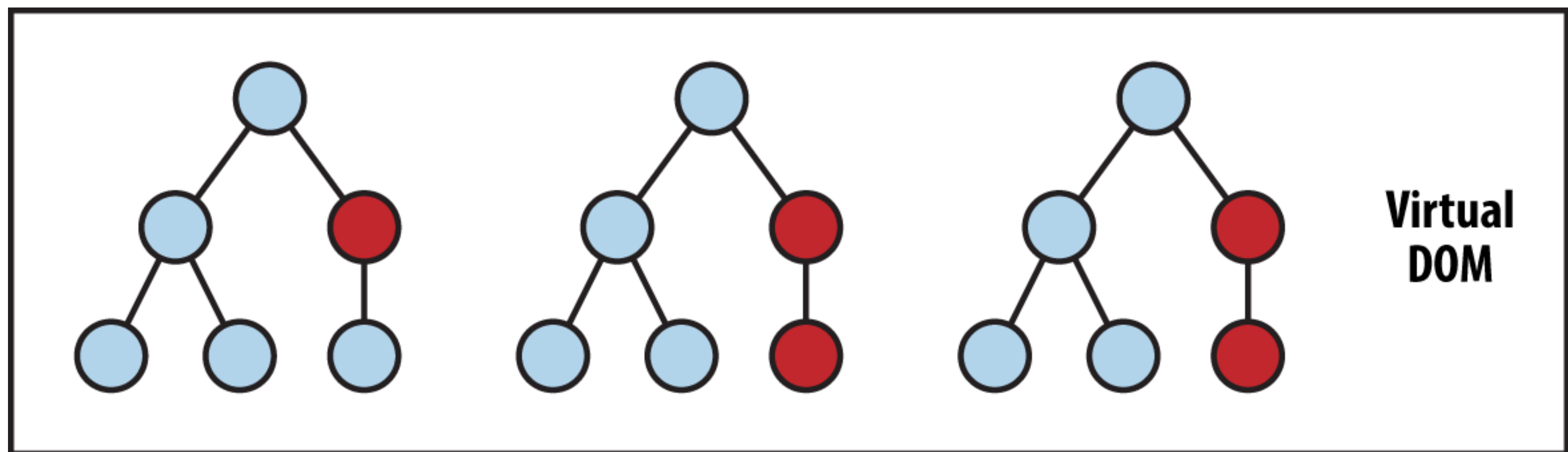
VIRTUAL DOM

DOM manipulation is the heart of the modern, interactive web. Unfortunately, it is also a lot slower than most JavaScript operations.

This slowness is made worse by the fact that most JavaScript frameworks update the DOM much more than they have to.

A virtual DOM object is a representation of a DOM object, like a lightweight copy.

A virtual DOM object has the same properties as a real DOM object, but it lacks the real thing's power to directly change what's on the screen.



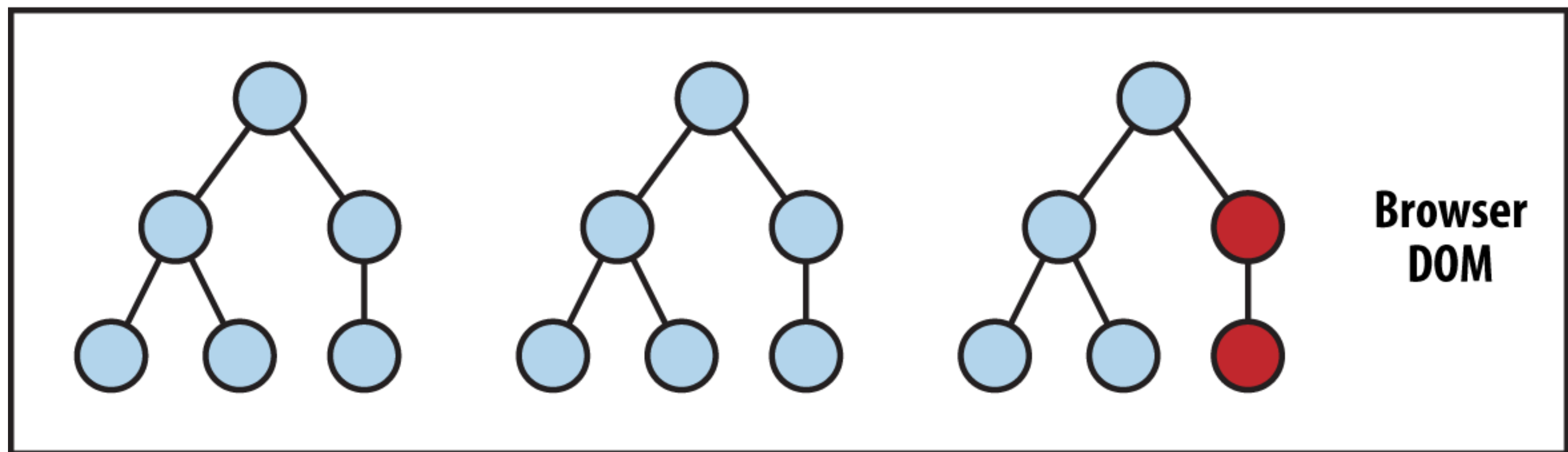
State Change



Compute Diff



Re-render



STATEFUL VS STATELESS

LIFECYCLE EVENTS

Initialization

setup props and state

Mounting

componentWillMount

render

componentDidMount

Updation

props

componentWillReceiveProps

shouldComponentUpdate

true

false

componentWillUpdate

render

componentDidUpdate

states

shouldComponentUpdate

true

false

componentWillUpdate

render

componentDidUpdate

Unmounting

componentWillUnmount

COMMON REACT PROJECT

- ES6 Syntax + imports
- React components
- Redux for state
- Webpack
- ...

CREATE REACT APP

github.com/facebookincubator/create-react-app

```
> npm install create-react-app -g
```

```
> create-react-app todo  
> cd todo  
> npm start
```




Welcome to React

To get started, edit `src/App.js` and save to reload.

LIVE CODING!



HOMEWORK



CREATE A TODO API

API should have following routes:

- **GET** `/todos` // return all items
- **GET** `/todos/{id}` // return a single item
- **POST** `/todos` // add a new item
- **PUT** `/todos/{id}` // update an item
- **DELETE** `/todos/{id}` // delete an item

EXAMPLE RESPONSES

GET /todos

```
[
  {
    id: 1,
    text: 'Prepare homework',
    completed: true
  },
  {
    id: 2,
    text: 'Prepare presentation',
    completed: false
  }
]
```

GET /todos/1

```
{  
  id: 1,  
  text: 'Prepare homework',  
  completed: true  
}
```

POST /todos

Body:

```
{  
  text: 'Buy milk'  
}
```

Response:

201 Created

```
{  
  id: 3,  
  text: 'Buy milk',  
  completed: true  
}
```

PUT /todos/2

Body:

```
{  
  id: 2,  
  completed: true  
}
```

Response: status 200

```
{  
  id: 2,  
  text: 'Prepare presentation',  
  completed: true  
}
```

DELETE /todos/2

Response: status 200
{ }

ADDITIONAL HOMEWORK

Connect TODO API with MongoDB:

<https://mlab.com>



TIPS & TRICKS (& TUTORIALS):

https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/mongoose

<https://university.mongodb.com>

READ (AND LEARN) MORE

NodeSchool

<https://nodeschool.io>

Free Code Camp
Learn to code for free.

<https://www.freecodecamp.org>

Eloquent JavaScript

Marijn Haverbeke

<https://eloquentjavascript.net>

You Don't know JavaScript

Kyle Simpson

<https://github.com/getify/You-Dont-Know-JS>

JavaScript: The Definitive Guide

David Flanagan

<http://shop.oreilly.com/product/9780596805531.do>

JavaScript: The Good Parts

Douglas Crockford

<http://shop.oreilly.com/product/9780596517748.do>

THE END

OF PART TWO