

A dense, dark forest scene with sunlight filtering through the canopy.

React.js

Presentasjon

Oppgaver

Case

TVERRFAGLIGHET

⌚ 08:00 GARTNERS SYN PÅ VERDEN

⌚ 09:00 - 09:45

👤 Business Case

—
Oslo



👤 Dieter Rams: 10 Principles for
Good Design

—
Helsinki



👤 Tech-begreper for ikke-
teknologer

—
København



👤 Hvorfor er en grafdatabase så
utrolig nyttig?

—
Stockholm

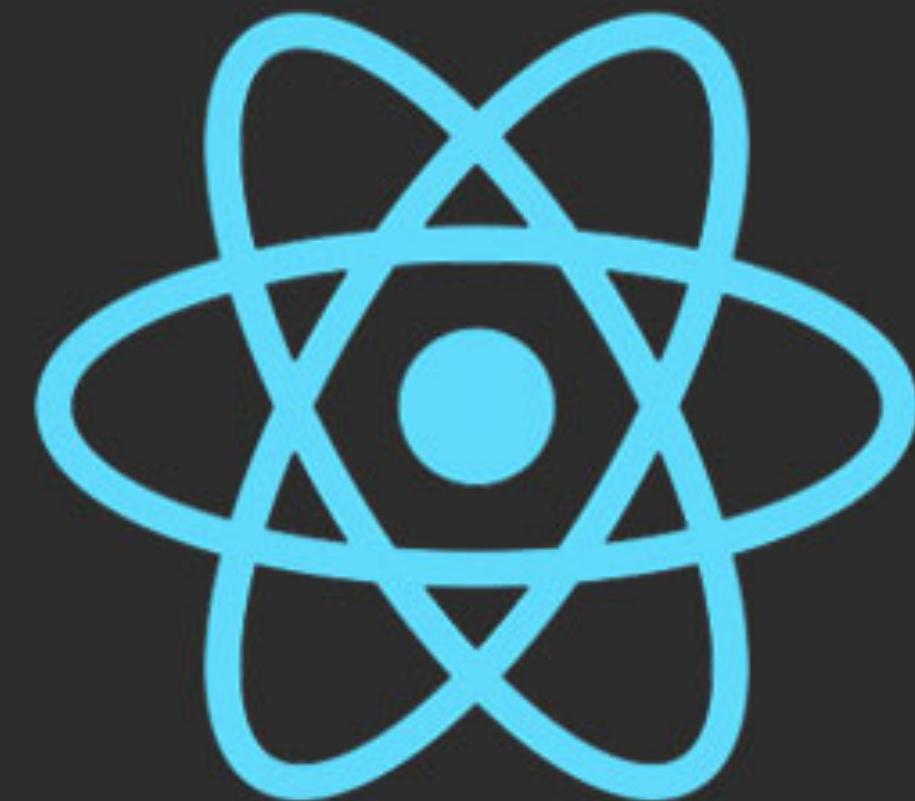


⌚ 10:00 - 10:45

👤 Topp 10 oppgaver til en
prosjektleder

—
Oslo





React

A black and white photograph of a rugged mountain landscape. In the foreground, there are rocky slopes and patches of dry grass. A winding path or stream bed cuts through the terrain. In the middle ground, a large, rounded mountain peak rises, its slopes covered with dense vegetation. The background shows more mountain ridges under a clear sky.

State

Enkelt



Deklarativt



Interaktivitet



Komponenter som byggeklosser

The background of the image is a large, dimly lit industrial facility. It features a complex network of thick, grey pipes running across the ceiling and walls. In the foreground and middle ground, there are numerous large, cylindrical concrete structures, possibly molds or parts of machinery, stacked in rows. The lighting is dramatic, with strong highlights and deep shadows, creating a sense of depth and scale.

Data → DOM



Ytelse?

Virtuel IDOM!

Virtuell DOM

1. Når noe kan ha endret seg, re-render alt til en ny virtuell dom-representasjon
2. Diff den forrige representasjonen med den nye
3. Oppdater kun det som har endret seg

render



diff



batch

Components

```
var Example = React.createClass({  
  render: function() {  
    return <div>  
      Hello World!  
    </div>  
  }  
});  
  
React.renderComponent(<Example/>, document.getElementById('main'));
```

```
var Example = React.createClass({  
  render: function() {  
    return React.DOM.div(null, 'Hello World!');  
  }  
});  
  
React.renderComponent(Example(), document.getElementById('main'));
```

```
/** @jsx React.DOM */\n\nvar HelloWorld = React.createClass({\n  render : function() {\n    return <span>hello</span><span> world!</span>;\n  }\n});
```

```
var HelloWorld = React.createClass({  
  render : function() {  
    return React.DOM.span({  
      children : "hello"  
    })React.DOM.span({  
      children : " world!"  
    });  
  }  
});
```

```
var HelloWorld = React.createClass({
  render : function() {
    return React.DOM.div({
      children : [
        React.DOM.span({
          children : "hello"
        }),
        React.DOM.span({
          children : " world!"
        })
      ]
    });
  }
});
```

Props & State

props = immutable



state = tilstand (helvete)

```
/** @jsx React.DOM */

var HelloWorld = React.createClass({
  render: function() {
    return (
      <p>
        {this.props.name} likes {this.props.likes.join(' and ')}.
      </p>
    );
  }
});

React.renderComponent(
<HelloWorld name="Johnny"
            likes={['dolphins', 'whales']} />
, document.body);
```

```
/* @jsx React.DOM */

var LikeButton = React.createClass({
  getInitialState: function() {
    return {liked: false};
  },
  handleClick: function(event) {
    this.setState({liked: !this.state.liked});
  },
  render: function() {
    var text = this.state.liked ? 'like' : 'unlike';
    return (
      <p onClick={this.handleClick}>
        You {text} this. Click to toggle.
      </p>
    );
  }
});
```

```
var Child = React.createClass({
  render: function () {
    return <div>
      Guybrush Threepwood.
      <button onClick={this.props.userDidClickButton}>
        Mighty pirate.
      </button>;
    }
  );
}

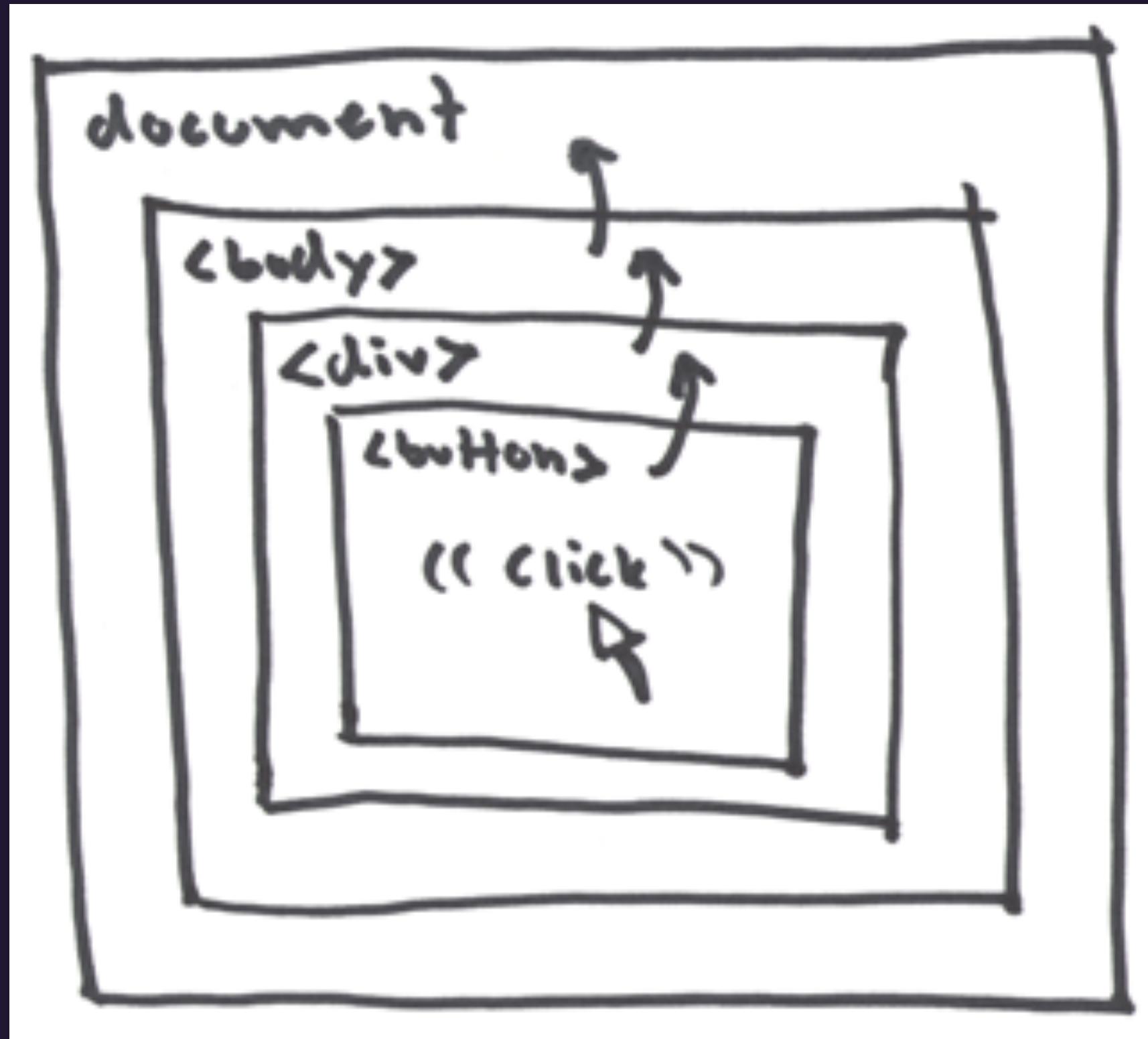
var Parent = React.createClass({
  callback: function() { /* ... */ },
  render: function () {
    return (
      <Child userDidClickButton={this.callback} />
    );
  }
});
```

Dataflyt

data nedover, events oppover

An aerial photograph of a coastal town. In the foreground, there is a large, open, sandy area with some sparse vegetation and a few small structures, including a white van and a small tent-like building. In the middle ground, a long line of white buses is parked along a road. The background features a dense cluster of buildings, mostly one-story houses, nestled among lush green trees and foliage. A prominent white multi-story building with a distinctive tower is visible on the left side of the town.

Eventdelegering

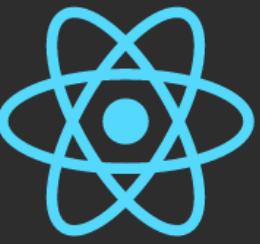


Gjenbruksbarhet

```
var TalkDetailsItem = React.createClass({  
  /* ... */  
  
  render: function() {  
  
    /* ... */  
    var talk = this.props.talk;  
  
    return <article>  
      <SessionHeader talk={talk} />  
      <Paragraphs talk={talk} />  
      <Speakers hosts={talk.hosts} />  
      <FeedbackButton url={talk.feedback_survs_url} />  
    </article>;  
  }  
});
```

Gjenbruksbarhet

```
var Speakers = React.createClass({  
  
  render: function() {  
  
    var speakers = this.props.hosts.map(function(host) {  
      return <Speaker host={host} />;  
    })  
  
    return <div className="speakers-container">  
      {speakers}  
    </div>;  
  }  
});
```



React Components

Component name, keyword or similar

Searchable database of **React** components

How it works

Every module registered on [NPM](#) using the keyword `react-component` will show up in the list. It really is that simple.

How do I add my component to the list?

1. Ensure your `package.json` file contains an array of keywords which includes `react-component`.
2. Publish your component to NPM (learn how at [npmjs.org](#)).
3. Wait for it to show up! Shouldn't take longer than 10-15 minutes.

Missing any features?

[Let us know!](#) We're always looking for ways to improve.

Who made this? Can I contribute?

Developed and currently hosted by [VaffelNinja](#), but it's an open-source, MIT-licensed solution.

Contributions are [very welcome!](#) Please make sure you read the contribution guidelines.

Latest components

- [react-infinity](#)
- [react-croton](#)
- [react-anything-sortable](#)
- [react-form-data](#)
- [react-bootpag](#)
- [react-typeahead](#)
- [react-form-builder](#)
- [react-clipboard](#)
- [react-microspreadsheet](#)
- [react-formly](#)

Recently updated

- [react-visibility-sensor](#)
- [react-treeview](#)
- [react-anchor](#)
- [merry-go-round](#)
- [react-markdown-textarea](#)
- [react-elements](#)
- [plexus-form](#)
- [react-googlemaps](#)
- [react-widgets](#)
- [react-datepicker](#)

Popular

Lifecycle events:

```
function componentDidMount() {...}  
function componentDidUpdate() {...}  
function componentWillUnmount() {...}  
...
```

Lifecycle events

```
var Program = React.createClass({  
  componentDidMount: function() {  
    if (scroll.exists()) {  
      document.body.scrollTop = scroll.get();  
    }  
  },  
  ...  
})
```

Ytelse++

```
shouldComponentUpdate: function(nextProps, nextState) {  
  return nextProps.id !== this.props.id;  
}
```

Fleksibilitet

```
renderComponent( <Example/> , document.getElementById('main'));
```

Lett å bruke React på kun en del av appen

Server-side rendering

```
var app = express();

function index_html(req, res) {
  React.renderComponentToString(Timer(), function(markup) {
    res.send(markup);
  });
}

app.get('/', index_html);
```



Flux

Oppsummert

- utfordrer en del etablerte konvensjoner

Oppsummert

- utfordrer en del etablerte konvensjoner
- har helt drøy ytelse

Oppsummert

- utfordrer en del etablerte konvensjoner
- har helt drøy ytelse
- men hovedformålet er altså å forenkle

Oppsummert

- utfordrer en del etablerte konvensjoner
- har helt drøy ytelse
- men hovedformålet er altså å forenkle
- mindre omfang enn andre rammeverk - håndterer kun DOM

Oppsummert

- utfordrer en del etablerte konvensjoner
- har helt drøy ytelse
- men hovedformålet er altså å forenkle
- mindre omfang enn andre rammeverk - håndterer kun DOM
- aller best til interaktive data
- leker godt med f.eks Backbone

A dense, dark forest scene with sunlight filtering through the canopy.

Lets get cracking