



Introduksjon til React.js

**Erik Wendel
fra BEKK**
SpareBank1 siden august 2012
Mobilbank
Fagleder Webteknologier

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

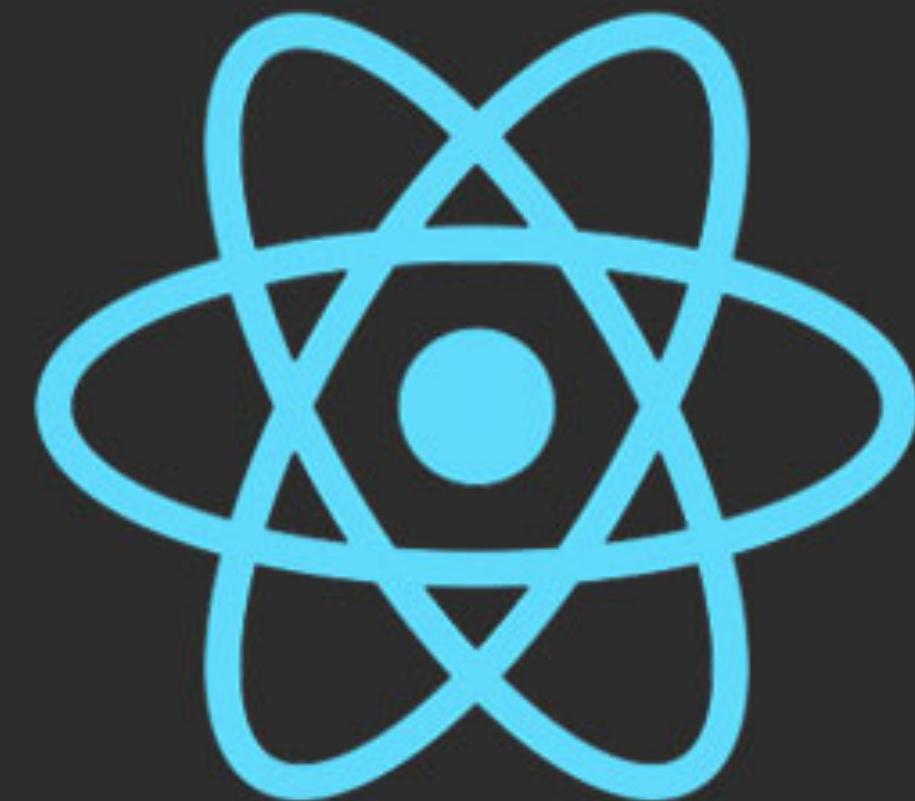


⌚ 10:00 - 10:45

👤 Topp 10 oppgaver til en
prosjektleder

—
Oslo





React

A dark, hazy landscape featuring rolling hills and mountains in the background. The foreground is dominated by a dark, textured surface, possibly volcanic rock or ash, with patches of dry, yellowish-brown grass in the lower left corner.

State

**Enkelt
Deklarativt
Interaktivitet
Komponenter som byggeklosser**

A black and white photograph of a construction site. In the foreground, there are several large, round concrete pipes stacked on wooden pallets. Above them, a complex network of scaffolding and steel beams spans across the frame. The scene is industrial and shows various stages of construction.

Data → DOM



Ytelse?

virtueDOM!

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

Components

"The way React does componentization is the successful invention. Not the virtual-dom. The virtual-dom is just hacky immediate mode graphics."

-- Sebastian Markbåge

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

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

```
var HelloWorld = React.createClass( {  
  render : function() {  
    return <span>hello</span><span> world!</span>;  
  }  
} );
```

```
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 = immutabel
state = tilstand (helvete)

```
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);
```

```
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>
    </div>;
  }
});
```

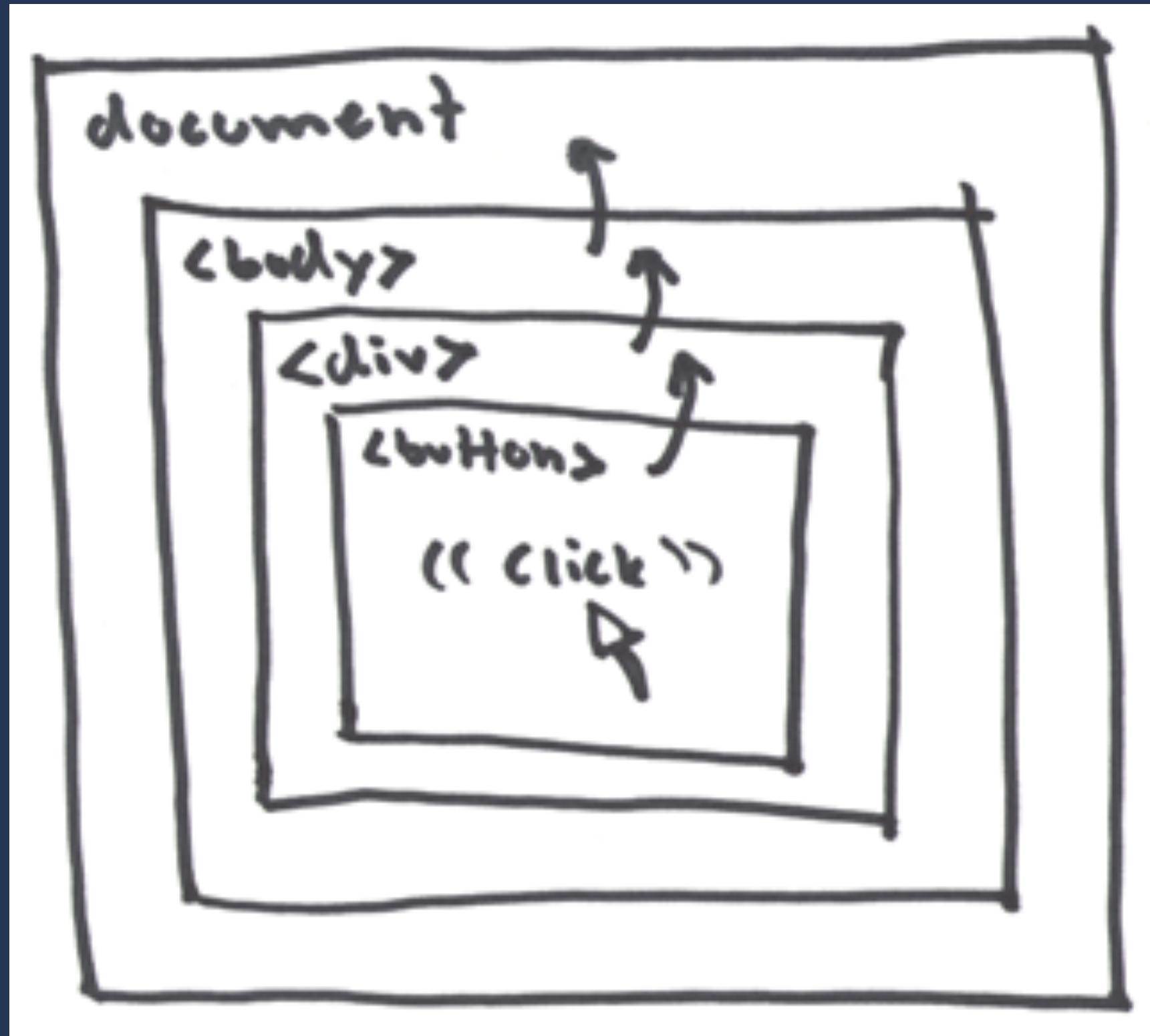
```
var Parent = React.createClass({
  callback: function() { /* ... */ },
  render: function () {
    return (
      <Child userDidClickButton={this.callback} />
    );
  }
});
```

Dataflyt

data nedover, events oppover

An aerial photograph of a coastal town. In the foreground, there's a large, open grassy field with some scattered trees and a few small buildings or vehicles. In the middle ground, the town extends towards the sea, featuring numerous buildings of various sizes, some with red roofs. Hills are visible in the background, and the sky is overcast.

Eventdelegering

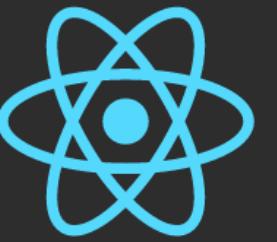


Gjenbruksbarhet

```
var TransferView = React.createClass({  
  /* ... */  
  
  render: function() {  
  
    /* ... */  
    return <div>  
      ...  
      <AccountSelector accounts={this.props.accounts}  
        selectedId={this.props.fromAccount}  
        onChange={...} />  
      <AccountSelector accounts={this.props.accounts}  
        onChange={...} />  
      <AmountInputs onChange={...} />  
      <MessageField onChange={...} />  
      <MainButton onClick={...} />  
      <CancelButton onClick={...} />  
    </div>;  
  }  
});
```

Gjenbrukbarhet

```
<AmountInputs />
<MessageField />
<AccountSelector />
<MainButton />
<CancelButton />
<ErrorMessageField />
```



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 (!this.props.accounts.hasResponse()) {  
      this.props.accounts.fetch();  
    }  
  },  
  ...  
})
```

Ytelse++

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

Fleksibilitet

```
renderComponent( <Example/> , document.body );
```

Lett å bruke React på kun en del av appen

Server-side rendering

```
React.renderComponentToString(Mobilbank(), function(markup) {  
  // ...  
});
```

A wide-angle photograph of a waterfall cascading down a rocky cliff into a pool of water. The waterfall is the central focus, with water falling in multiple streams over different levels of rocks. The surrounding environment is a dense jungle with various shades of green foliage, trees, and vines. The lighting suggests it's daytime, with sunlight filtering through the canopy.

Flux

React vs Backbone



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*
- begrenset omfang (DOM)

Oppsummert

- utfordrer en del etablerte konvensjoner
- har helt drøy ytelse
- men hovedformålet er altså å *forenkle*
- begrenset omfang (DOM)
- skinner med interaktive data

A dense forest scene with sunlight filtering through the leaves.

Takk for meg

@ewndl