

Draft.js workshop

Hvorfor rich text?

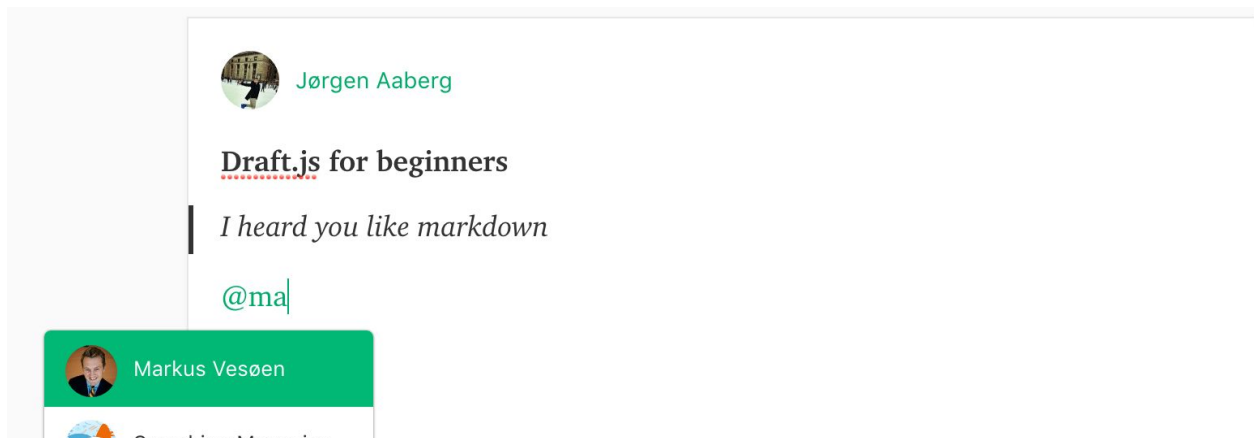
Hvorfor rich text?



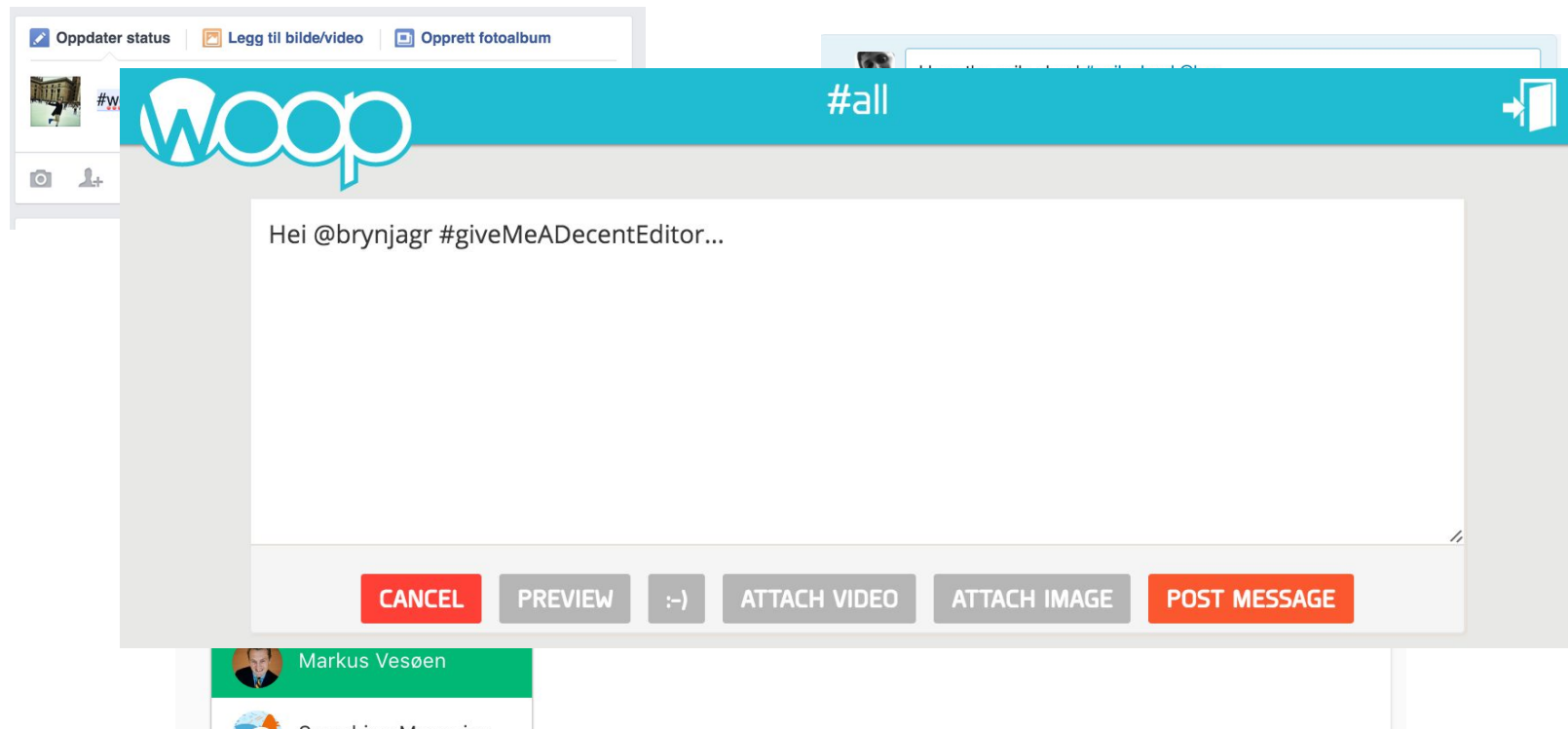
Hvorfor rich text?



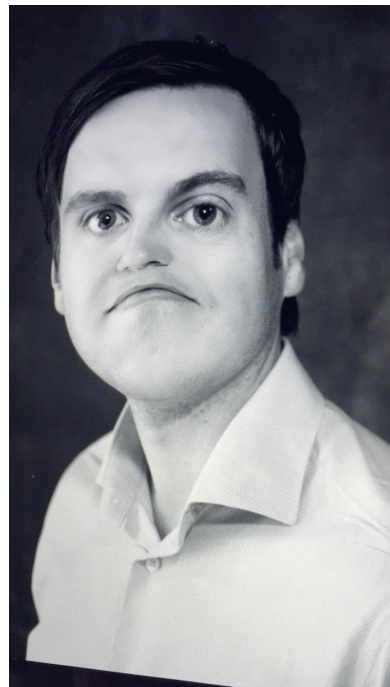
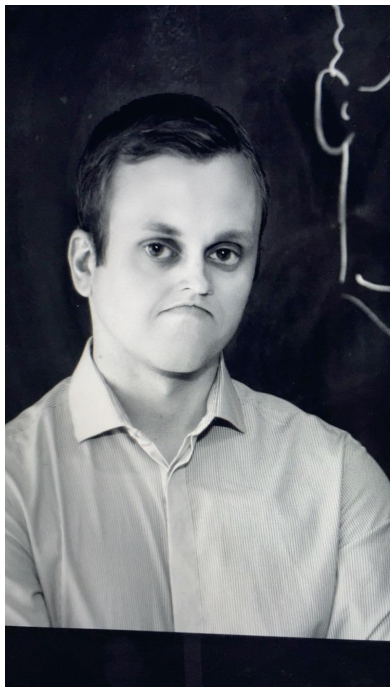
Hvorfor rich text?

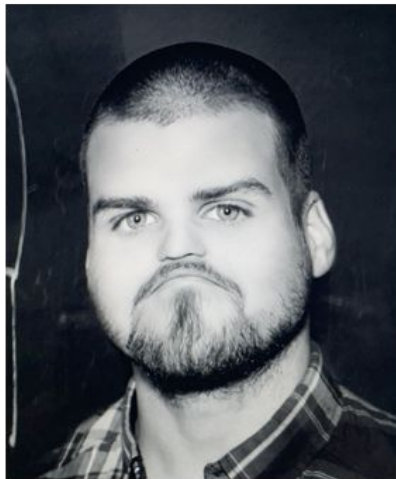
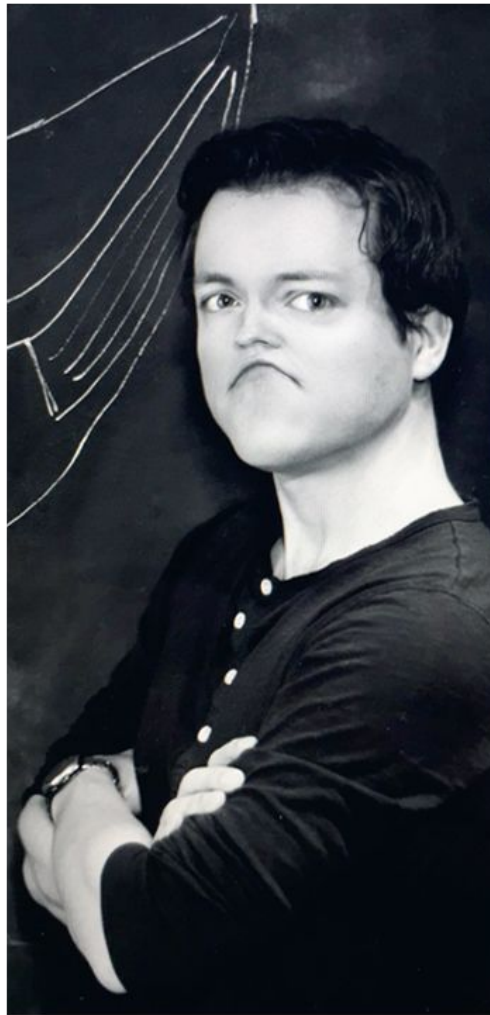


Hvorfor rich text?











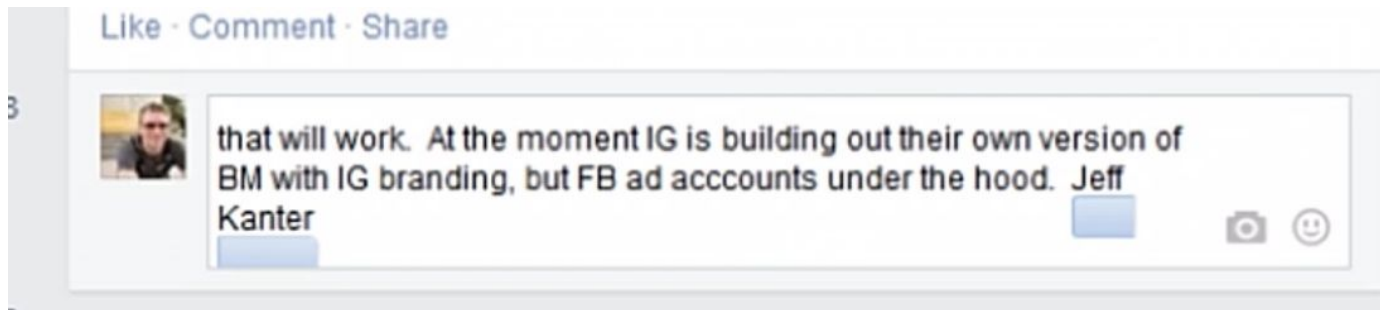
Hvordan har man gjort dette før?

Hvordan har man gjort dette før?

<textarea> + <div> + DOM-hacks

Hvordan har man gjort dette før?

<textarea> + <div> + DOM-hacks



Hvilke alternativer har man?

Hvilke alternativer har man?

1. `<textarea>` + `<div>` + DOM-hacks

Hvilke alternativer har man?

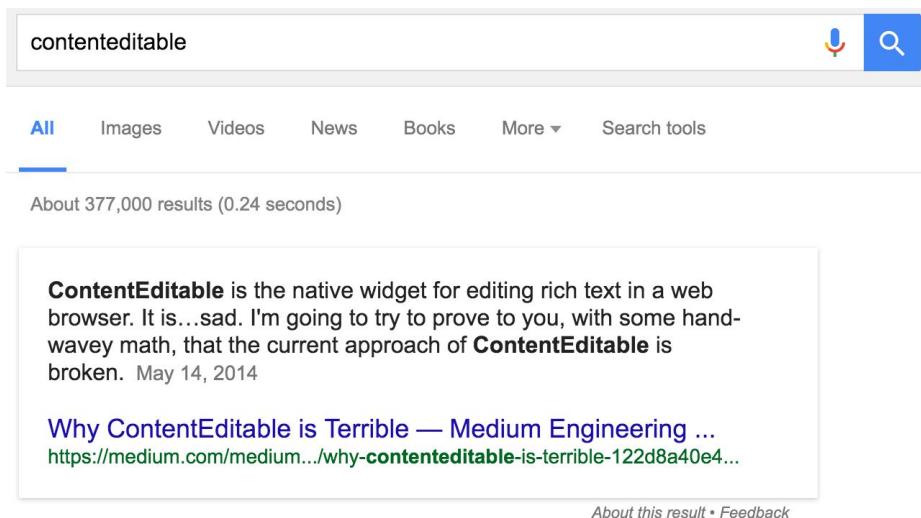
1. `<textarea>` + `<div>` + DOM-hacks
2. Manuelt tegne DOMen

Hvilke alternativer har man?

1. `<textarea>` + `<div>` + DOM-hacks
2. Manuelt tegne DOMen
3. ContentEditable

Hvilke alternativer har man?

1. `<textarea>` + `<div>` + DOM-hacks
2. Manuelt tegne DOMen
3. ContentEditable



Hvilke alternativer har man?

1. <tex
2. Man
3. Con



broken. May 14, 2014

[Why ContentEditable is Terrible — Medium Engineering ...](https://medium.com/medium.../why-contenteditable-is-terrible-122d8a40e4...)
<https://medium.com/medium.../why-contenteditable-is-terrible-122d8a40e4...>

[About this result](#) • [Feedback](#)

ContentEditable – The good parts

ContentEditable – The good parts

1. Virker i alle browsere

ContentEditable – The good parts

1. Virker i alle browsere
2. Native håndtering av cursor og selection

ContentEditable – The good parts

1. Virker i alle browsere
2. Native cursor og selection
3. Native input events (keyEvents o.l.)

ContentEditable – The good parts

1. Virker i alle browsere
2. Native cursor og selection
3. Native input events (keyEvents o.l.)
4. Muligheter for rich text

ContentEditable – The good parts

1. Virker i alle browsere
2. Native cursor og selection
3. Native input events (keyEvents o.l.)
4. Muligheter for rich text
5. Autogrowing

<LetsCode />

<LetsCode />

```
import {Editor} from 'draft-js';
```

ContentEditable med en onChange-property som kjører hver gang det skjer noe. F.eks. ved flytting av cursor, markerer noe, undo/redo, taster en bokstav.

<LetsCode />

```
import {Editor} from 'draft-js';
```

En *controlled* ContentEditable med en onChange-property som kjører hver gang det skjer noe. F. eks. ved flytting av cursor, markerer noe, undo/redo, taster en bokstav.

```
import {EditorState} from 'draft-js';
```

Inneholder all informasjon om teksten som er skrevet, f.eks. om den er stylet, undo/redo-stacken, hva som er selektert nå osv.

Controlled input

1. En *verdi* som representerer tilstanden til en input
2. En *onChange*-funksjon som mottar oppdateringer til inputen

```
class MyInput extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = {value: ''};  
    this.onChange = (evt) => this.setState({value: evt.target.value});  
  }  
  render() {  
    return <input value={this.state.value} onChange={this.onChange} />;  
  }  
}
```

Controlling rich text

1. En tekststreng holder ikke til å representere en kompleks rich editor
2. Det finnes ingen *onChange*-event for et ContentEditable-element

```
import {Editor, EditorState} from 'draft-js';

class MyEditor extends React.Component {
  constructor(props) {
    super(props);
    this.state = {editorState: EditorState.createEmpty()};
    this.onChange = (editorState) => this.setState({editorState});
  }
  render() {
    return <Editor editorState={this.state.editorState} onChange={this.onChange} />;
  }
}
```

<Oppgave0 />

Lag en plain text-editor i Draft.js med en tom EditorState

Tips: Sjekk ut <https://facebook.github.io/draft-js/docs/overview.html>

Persistent Data

Four score and seven years ago our fathers brought forth, upon this continent, a new nation, conceived in liberty, and dedicated to the proposition that "all men are created equal."



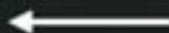
Record

Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived, and so dedicated, can long endure. We are met on a great battle field of that war. We come to dedicate a portion of it, as a final resting place for those who died here, that the nation might live. This we may, in all propriety do. **Lorem ipsum dolor sit amet adipiscing.**



Record

But, in a larger sense, we can not dedicate – we can not consecrate – we can not hallow, this ground – The brave men, living and dead, who struggled here, have hallowed it, far above our poor power to add or detract. The world will little note, nor long remember what we say here; while it can never forget what they did here.



Record

It is rather for us, the living, we here be dedicated to the great task remaining before us – that, from these honored dead we take increased devotion to that cause for which they here, gave the last full measure of devotion – that we here highly resolve these dead shall not have died in vain; that the nation, shall have a new birth of freedom, and that government of the people, by the people, for the people, shall not perish from the earth.



Record

A rich text editor

This is a title

This is the second *paragraph*

This is the third paragraph

Log state to console

Log markdown

```
▼ Object {allowUndo: true, currentConte
  allowUndo: true
  ▼ currentContent: Object
    ▼ blockMap: Object
      ► 4kdme: Object
      ► a4buf: Object
      ► b9429: Object
      ► bmpr5: Object
      ► __proto__: Object
      ► selectionAfter: Object
      ► selectionBefore: Object
      ► __proto__: Object
    ► decorator: CompositeDraftDecorator
    ► directionMap: Object
      forceSelection: false
      inCompositionMode: false
      inlineStyleOverride: null
      lastChangeType: "split-block"
      nativelyRenderedContent: null
    ► redoStack: Array[0]
    ▼ selection: Object
      anchorKey: "b9429"
      anchorOffset: 5
      focusKey: "bmpr5"
      focusOffset: 4
      hasFocus: false
      isBackward: false
      ► __proto__: Object
    ► treeMap: Object
    ► undoStack: Array[52]
    ► __proto__: Object
```



Rich text





Rich text



```
import {RichUtils} from 'draft-js'
```

RichUtils holder informasjon om vanlige WYSIWYG-kommandoer, f.eks. **fet**, *kursiv*, understreket



Rich text



```
import {RichUtils} from 'draft-js'
```

RichUtils holder informasjon om vanlige WYSIWYG-kommandoer, f.eks. **fet**, *kursiv*, understreket

```
<Editor handleKeyCommand={(command) => {<handle the command>}} />
```

Command er en tastatur-kommando, feks cmd+b



Rich text



```
import {RichUtils} from 'draft-js'
```

RichUtils holder informasjon om vanlige WYSIWYG-kommandoer, f.eks. **fet**, *kursiv*, understreket

```
<Editor handleKeyCommand={(command) => {<handle the command>}} />
```

Command er en tastatur-kommando, feks cmd+b

```
RichUtils.handeKeyCommand(currentEditorState, <command>)
```

Returnerer en oppdatert EditorState

<Oppgave1 />

Lag en rich text-editor som bruker RichUtils til å støtte vanlige formateringsoperasjoner

Bonus: bruk RichUtils.toggleInlineStyle og lag en knapp som gjør teksten **fet**

Tips: <https://facebook.github.io/draft-js/docs/quickstart-rich-styling.html>

<Oppgave1 />

Lag en rich text-editor som som bruker RichUtils til å støtte vanlige formatteringsoperasjoner

Bonus: bruk RichUtils.toggleInlineStyle og lag en knapp som gjør teksten **fet**



Tips: <https://facebook.github.io/draft-js/docs/quickstart-rich-styling.html>

Decorators

Decorators



Decorators



Scanner teksten basert på en strategi (regex)

Erstatter den med en valgfri komponent

Decorators



Scanner teksten basert på en strategi (regex)
Erstatter den med en valgfri komponent

```
import {CompositeDecorator} from 'draft-js';  
EditorState.createEmpty(new CompositeDecorator(...));
```

<Oppgave2 />

Lag en editor som setter farge på hashtags ved bruk av decorators

Hello there you #sexything|

Tips: <https://facebook.github.io/draft-js/docs/advanced-topics-decorators.html>

<Oppgave3 />

Lag en editor som støtter mentions, altså at du kan skrive @ og få opp navneforslag.



A screenshot of a web application interface. At the top, there is a text input field containing the text "Hello there @m". Below the input field, on the left, is a button labeled "Log State". To the right of the button, a dropdown menu is open, displaying a list of names: "Martin", "Marte", "Martinus", and "Markus". The dropdown menu has a light gray background and a thin border.

Tips: Lag en hardkodet liste med brukere. Bruk `filter`, `startsWith` og `props.decoratedText` (sendes automatisk med til Decorator-komponenten)

```
Record {  
  text: "Live from the Grand Hyatt ...",  
  metadata: [..., ..., 1, 1, 1, 1, ..., 1, 1],  
}
```

```
function addMention(initialState, ...) {  
  const stateAfterRemoval = removeText(initialState, ...);  
  const stateAfterInsertion = insertText(stateAfterRemoval, ...);  
  const stateWithMention = applyMention(stateAfterInsertion, ...);  
  return stateWithMention;  
}
```

```
Record {  
  text: "Live from the @grand hyatt",  
}  
  
Record {  
  text: "Live from the ",  
}  
  
Record {  
  text: "Live from the Grand Hyatt ...",  
}
```



DraftJS Plugins

High quality plugins with great UX



GitHub

825

<Oppgave4 />

Ta i bruk draft-js-plugins og implementer mentions

Tips: <https://github.com/draft-js-plugins/draft-js-plugins>