Tasks React Hooks

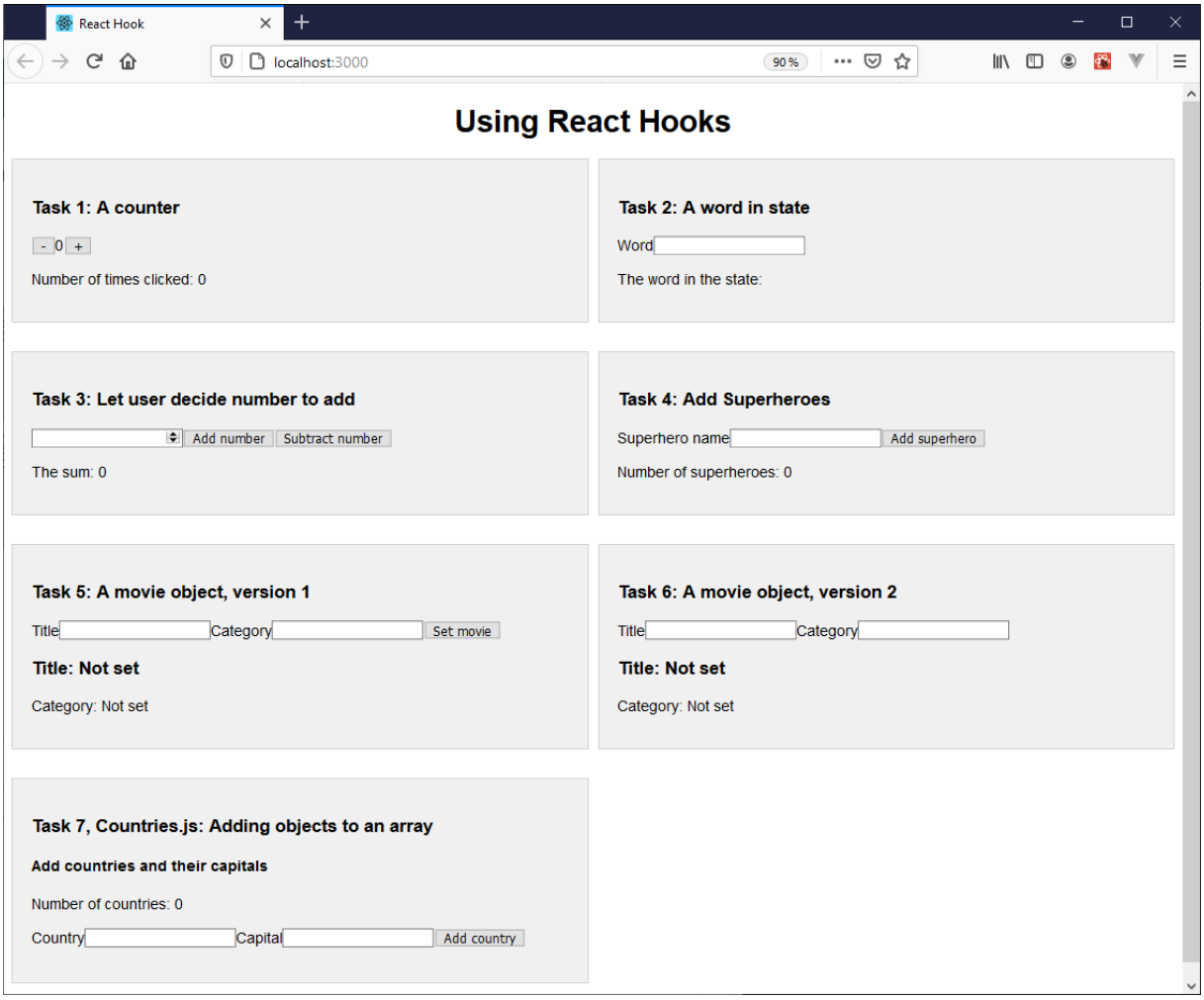
Mål: lær å bruke useState for å håndtere endringer som krever at komponenten skal endres

Intro til React Hooks: https://reactjs.org/docs/hooks-intro.html

Hvordan jobbe med taskene

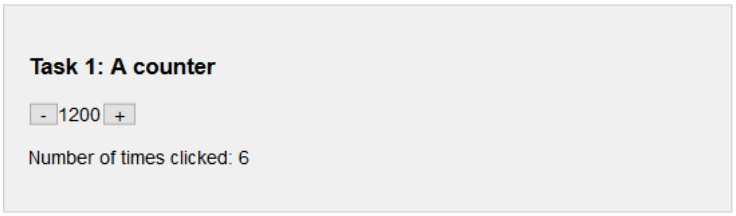
Skap en ny React boilerplate (npx create-react-app valgfritt-prosjekt-navn). Du skal kode komponenter som gjør bruk av useState på flere nivåer av kompleksitet. Hver task er 1 komponent, men noen av dem kan splittes opp i 2 komponenter (ref. List og Item). Stilsett gjerne med enkel CSS for å øve på importering av CSS-fil.

Merknad: noen ganger kan useRef være bedre å bruke enn useState i forms – dette gjelder hvor det ikke er nødvendig med sanntids endringer i grensesnittet.



Task 1, Counter.js: Incrementing and decrementing a number in state

You need a state for a counter. When the user clicks minus, the counter decrements with 200, when the user clicks plus the counter increments with 200.



Task 2: Let user set a word in the state through a textbox

In this task you should let the user input a word which will be in the state of the component. You don't need a button, only the onChange event on the textbox. While the user is typing in the textbox you should see the text appear in the <p> below it immediately.

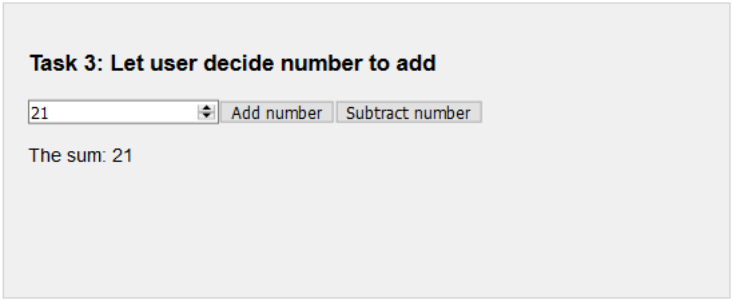
Notice that when you use the onChange you can get the textboxes value:

<input onChange={ nameOfMethod } type="text">

const nameOfMethod = (e) => { alert(e.target.value) }

Task 3: Incrementing and decrementing a number by a user defined number

Let the user be able to input a number into a textbox and add the number to a counter. In this task you should use the onChange event on the textbox to change the state of a value which later is to be used to add or subtract from the counter. I.e. you should use two useState sentences.

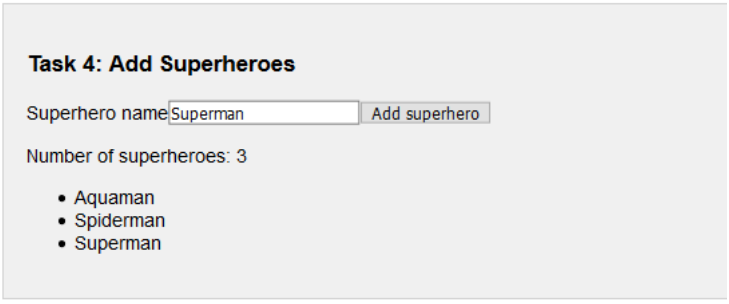


Task 4, Superheroes.js: Let user add words in a string array in a state

Let the state have an array of texts. The user can input texts in the textbox and add these will be added to the array in the state when the button is pressed.

Notice that to add something in an array you should use the spread operator as the regular JavaScript functions such as push will not trigger an update:

setAnArray([…nameOfArray, newThingInArray])



Task 5, Movie.js: Working with a single object in a state

In this task the state should contain 1 object created with object initializer: { title: “”, category: “” }.

Coding tip:

When you have an input element in your code which has an onChange event attached, you can get information from it in the handleChange method. You can get its value, type and if you add id or name to it you can identify which input element you have changed.

```
<input onChange={ handleChange } type="text"/>

const handleChange = ( e ) => {
    alert( e.target.value );
}
```

Task 6: Working with a single object in a state, without button

In this task the state should contain 1 object created with object initializer: { title: “”, category: “” }.

Coding tip:

If you add the *name* attribute to a textbox, you can get it afterwards and its value in the handleChange method.

Input:

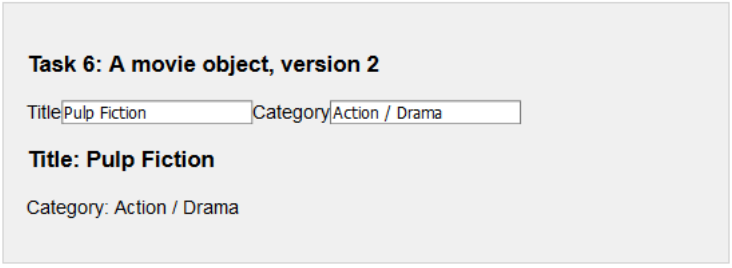
```
<input name="nameThatMatchesAProperty" onChange={ handleChange } type="text"/>
```

Method handling the onChange:

```
const handleChange = ( e ) => {
    const { name, value } = e.target;
    console.log( name + " " + value);
}
```

Changing a property of an object (this will be in the onChange method):

```
setObjectState({ ...objectFromBefore, [name]: value });
```



Task 7: Adding objects in an array in a state

In this task you shall have an array of objects in your state. Each object has a name and a capital as properties.