**Jeg har meget svært ved at forklare mig på skrift, men jeg må give det et forsøg. Har lavet alle opgaverne for ugen + lidt af ekstra opgaven, så jeg føler jeg er godt med, men det har ikke sat sig helt endnu.**

**Link til hele ugen: https://github.com/Pelle-pr/Flow3Week1**

1. **Kunne bruge react komponenter og kommunikation mellem Komponent**

React komponenter er lidt som javascript funktioner, som taget et argument som man normalt kalder ”props” som indeholder attributter som vi kan bruge i komponentet eller sende videre til andre komponenter.

Komponentet returnere et react element

I alle opgaverne i denne uge har vi lavet kommunikation mellem komponenter.

1. **Kunne bruge JSX samt forklare hvilket problem JSX løser for os.**

JSX sørger for at konvertere vores html tags til react elementer.

Så vi ikke behøver at skrive f.eks ”React.createElement”

Eksempel uden JSX:

const myelement = React.createElement('h1', {}, 'I do not use JSX!');

Eksempel med JSX:

const myelement = <h1>I Love JSX!</h1>;

1. **Kunne forklare og demonstrere brug af state i React med Hooks via useState**

Use state bruger vi til at sætte en ”viable” på et komponent, f.eks i count projektet(se Onsdag mappen i count.js)

<https://github.com/Pelle-pr/Flow3Week1/blob/main/Onsdag/day2_ex/src/Count.js> linje 7

Når vores komponent bliver ”kaldt” sætter jeg count til initialCount(f.eks 0)

Derefter hvis man trykker på f.eks Increment knappen, bliver Increment metoden kaldt som

Som ved hjælp af ”setCount” ændre ”staten” fra 0 til 1, ved at sige count +1.

Hvis der trykkes på Increment igen, vil den igen sige count+1 men da CountStaten nu har ændret sig fra 0 til 1, vil det sige 1 + 1.

1. **Kunne bruge useEffect og folklare forskellig anvendelse af dette.**

UseEffect bruges at lave side effecter i React komponentet når .feks Staten ændre sig.

I count projektet(se Onsdag mappen i count.js)

<https://github.com/Pelle-pr/Flow3Week1/blob/main/Onsdag/day2_ex/src/Count.js> linje 9-11

Laver jeg en UseEffect som gør at hver gang staten i count ændre sig så kalder jeg localStorage.setItem(”count”, count) som gemmer count værdien.

Eller f.eks i den ene opgave hvor der skal addes personer til en liste(torsdags mappen, react-crud-rest-exercise-master)

<https://github.com/Pelle-pr/Flow3Week1/blob/main/Torsdag/react-crud-rest-exercise-master/react-crud-rest-exercise-master/src/components/App.js> linje 10-15

Der har jeg lavet en useEffect som henter alle personerne ned igen fra vores API og lægger dem ind i tabellen, hver gang staten i PersonToAddEdit ændre sig.

1. **Kunne bruge list og keys i react, samt bruge controlled komponents til formdata.**

I projektet fra Torsdag (statelift-demo) laver jeg en simple liste med navne på personer.

React skal bruge en key på alle liste elementer, så React entydigt kan finde ud af hvor i listen den virtuelle dom har ændret sig. Derfor når laver <ul> eller <tr> sætter vi en unik key ind. I mit eksempel i statelift-demo har jeg bare brugt personers navn, men denne key er ikke unik da flere kan have samme navn, så i de andre opgaver.

F.eks. Torsdag/ react-crud-rest-exercise-master, der bruger jeg id’et fra hver person som key

<https://github.com/Pelle-pr/Flow3Week1/blob/main/Torsdag/react-crud-rest-exercise-master/react-crud-rest-exercise-master/src/components/AllPersons.js> linje 20

I denne opgave har jeg også brugt controlled komponents til formdata.

1. **Kunne forklare og demonstrere begrebet Lifting State i React**

<https://github.com/Pelle-pr/Flow3Week1/tree/main/Torsdag/statelift-demo>

I Torsdags opgaven (statelift-demo) bruger vi ”single source of truth” dvs i vores App komponent har vi alt vores data, som skal bruges i vores child komponenter. Dataen som benyttes er (names) som er en liste af navne, og (newName) som er det nye navn som skal indsættes i listen.

Min App komponent starter med at rendere siden som består af 2 child komponenter, NewPerson og PersonList.

Personlist komponentet bruger props(”persons”) til at rendere en liste af personer.

NewPerson rendere en form hvor i man kan skrive et navn man vil tilføje, derudover har jeg lavet en prop hvor jeg giver AddPerson funktionen med ned i komponentet.

Det gør at jeg nede i mit child NewPerson komponent kan bruge addPerson metoden til at tilføje en ny person til ”names” staten. Som så opdatere min personlist da ”staten” på names har ændret sig.

Kort forklaret så betyder det at dataen i vores App komponent ”flyder” ned gennem de andre komponenter. Og ikke fra et child komponent til et andet.