

Utgå från 0. Skapa komponenter, och se till att ni precis har gjort klart denna när ni hoppar in här.

Interpolation

1. Nu skall vi kika lite på interpolation. Skapa en variabel i er komponent. Ge variabeln ett värde som är ett heltal eller en text, någonting som är lite enklare än en lista med objekt.
2. I er return, skapa en behållare för värdet av er variabel, kanske en section, article eller liknande. Inuti denna behållare, interpolera ut värdet av variabeln genom { ... }.
3. Kontrollera att ert värde syns på skärmen.

Händelser

1. Vi skall nu skapa en ny knapp i vår html och koppla en händelse till knappen (helst en onClick). Syftet med denna knapp är att ändra värdet på vår variabel som vi skapade i avsnittet Interpolation här ovanför. Gör detta genom att skapa en knapp och lägg till attributet onClick. Skapa en funktion i er komponent och koppla denna till onClick-händelsen. Men hur?
2. Genom interpolation! Ni kan ange att funktionen som skall köras när ni gör er onClick skall vara den funktion som står mellan { ... }. Detta betyder att ni kan skriva in funktionsnamnet mellan { och }. Ganska smidigt!
3. I er funktion, ändra nu värdet på er variabel till någonting annat. Visst borde detta göra att ni ser detta nya värde i DOM:en när ni klickar på knappen? Testa och fundera över resultatet.
4. Vi behöver nu ha möjlighet att fånga det som en användare skriver in i textrutorna. Detta kan vi göra med hjälp av traditionell javascript genom att hitta id:t på vårt element och sedan plocka ut value från textrutan.

Skapa en händelse för knappen (från övning 0) (helst onClick) och koppla denna händelse till en funktion som du har i din komponent.

5. Eftersom vi har två textrutor behöver vi hämta värden från de bägge textrutorna i vår funktion. Detta gör vi enklast genom document.getElementById("..."). Gör detta för bägge textrutorna och lagra respektive textrutas värde i en variabel.
6. Ok, så långt, toppen! Nu behöver vi ändra vad som syns i DOM:en. Detta gjorde vi tidigare med hjälp av .innerHTML på ett element. Vi kan göra samma sak även nu. Testa!

Listor

1. Nu skall vi arbeta med listor och se hur vi kan skapa html för en lista med värden. Vi börjar med att skapa en variabel i vår komponent. Denna variabel skall vara en lista med tal. T.ex.

```
let numbers = [1, 1, 2, 3, 5, 8, 13]
```

2. Detta är bra, men det är svårt att skriva ut en lista med tal i vår html. Vi behöver omvandla dessa tal så att varje tal är en html-struktur istället. T.ex. vi vill omvandla talet 1 till 1

Detta är precis syftet med funktionen map. Om ni inte känner er bekväma med map finns det mer information att läsa här: [JavaScript Array map\(\) Method \(w3schools.com\)](https://www.w3schools.com/js/js_array_map.asp)

3. Det vi kan göra är att i vår map-funktion returnera det html-element som vi vill att varje tal skall motsvara. Det betyder att innehållet i vår map-funktion kommer att bli någonting liknande:

```
return <li>{i}</li>
```

Lite beroende på vad ni har valt att er variabel skall heta.

4. Nu behöver vi ta reda på resultatet av vår map. Så lagra resultatet av er map i en variabel, kanske kan den heta lis för att symbolisera att den skall innehålla just en lista med -element.

```
let lis = numbers.map...
```

5. Nu har vi en variabel i vår komponent som vi kan använda genom att skriva ut den i vår html. Då vår lista lis innehåller -taggar känns det som att vi behöver en . Skapa en ul någonstans i er return och inuti denna kan ni återigen använda interpolation för att skriva ut variabeln lis.

```
<ul>{lis}</ul>
```

6. Kontrollera att allting fungerar.

Extra

Från vad ni har i er komponent nu, skapa funktionalitet för att förändra era variabler, t.ex. er lista genom att ta bort, lägga till eller bara ändra någonting i listan. Kommer det ske en förändring på skärmen när ni gör detta? Varför?