

setState

1. För att kunna använda state och inte skriva en klasskomponent så behöver vi introducera hooks. Hooks tillåter en funktionell komponent att använda sig av state. Detta fungerar genom den så kallade State Hook. Skapa ett nytt projekt genom att använda

```
npm create vite <myAppName>
```

I denna nya applikation vill jag att du skapar en ny funktionell komponent. Om du inte kommer ihåg hur detta skall göras, titta på följande url: <https://reactjs.org/docs/components-and-props.html>

Komponenten behöver inte göra någonting ännu, bara fungera. Så till så att den renderar en tom div eller en span eller någonting annat.

2. När din komponent fungerar, d.v.s. applikationen kompilerar och du kan se applikationen i en webbläsare, är det dags att introducera State Hook. Denna hook tillåter oss att använda state i en funktionell komponent, vilket tidigare inte gick. Skapa ett state som heter counter genom att använda `useState()` i din komponent.

Här kan du titta på följande url: <https://reactjs.org/docs/hooks-overview.html>

Kom ihåg att använda startvärde 0 för ditt state

3. När ditt state finns på plats så ser du att det ser ut som en lista, en array, med olika saker i sig. Anledningen till att vi behöver definiera en array med funktioner och värden är därför att den hook, `useState`, returnerar en sådan array. Det första objektet i listan är vårt state-värde. Det andra objektet i listan är den funktion som tillåter oss att ändra den första objektet. Motsvarigheten till detta i klasskomponenter är `setState`.

Skapa en knapp i din html genom att ändra i din return så att den tomma div:en från övning 1 nu blir en knapp. Skapa också upp en klickhändelse för knappen och koppla den till en funktion som skall köras när knapptryckningen sker.

4. I funktionen som körs när användaren klickar på knappen från övning 3, kör funktionen `setCounter` och skicka med det nuvarande värdet på state + 1.
5. Lägg till ytterligare ett html-element som skriver ut värdet av vårt state-objekt. Prova att klicka loss på knappen för att se hur det fungerar.