Component State

Agenda

- React Repetition
- Komponent (hjälp funktioner)
- Destrukturering
- Page re-rendering
- State (intro)

React Repetition (Setup)

Skapa React Project

```
# npm 6.x (outdated, but still used by some):
npm create vite@latest part1 --template react

# npm 7+, extra double-dash is needed:
npm create vite@latest part1 -- --template react
```

React Repetition (Setup)

Installera resurser

```
cd part1
npm install
```

Starta applikation

```
npm run dev
```

React Repetition

```
import ReactDOM from 'react-dom/client'

import App from './App'

ReactDOM.createRoot(document.getElementById('root')).render(<App />)
```

```
const App = () => {
  return (
    <div>
      <p>Hello world</p>
    </div>
  )
}

export default App
```

React Repetition (Intro)

Live kodning: Gör hello world, testa använda ett annat root element

Övning: Gör en app som säger hej <ditt namn>

React Repetition (Komponent)

Live kodning: Gör flera komponenter

(inkludera css)

Övning: Gör en hemsida med

- Head
- Content
  - Lista med Produkter (komponent)
- Footer

```
const App = () => {
  return (
    <div>
      <h1>Greetings</h1>
      <Hello />
      <Hello />
      <Hello />
    </div>
  )
}
```

React Repetition (Props)

Live Kodning: En Person komponent

- Namn
- Telefon
- Ålder
- Adress

Övning: Gör en SpiceProduct komponent

- Namn
- Beskrivning
- Pris / gram
- Paketvikt

```
const Hello = (props) => {
  return (
    <div>
      <p>Hello {props.name}</p>
    </div>
  )
}
```

```
const App = () => {
  return (
    <div>
      <h1>Greetings</h1>
      <Hello name='George' />
      <Hello name='Daisy' />
    </div>
  )
}
```

React Repetition (Props)

Live Kodning: Skicka med objekt

Övning: Modifiera SpiceProduct

m.h.a objekt

```
const App = () => {
  const course = 'Half Stack application development'
  const part1 = {
    name: 'Fundamentals of React',
    exercises: 10
  }
  const part2 = {
    name: 'Using props to pass data',
    exercises: 7
  }
  const part3 = {
    name: 'State of a component',
    exercises: 14
  }

  return (
    <div>
      ...
    </div>
  )
}
```

Komponent (hjälp funktioner)

Vilka props behöver komponenten?

Vad visar komponenten?

```
const Hello = (props) => {
  const bornYear = () => {
    const yearNow = new Date().getFullYear()
    return yearNow - props.age
  }

  return (
    <div>
      <p>
        Hello {props.name}, you are {props.age} years old
      </p>
      <p>So you were probably born in {bornYear()}</p>
    </div>
  )
}
```

Komponent (hjälp funktioner)

Båda dessa funktioner fungerar likadant

```
const bornYear = () => new Date().getFullYear() - age

const bornYear = () => {
  return new Date().getFullYear() - age
}
```

Komponent (hjälp funktioner)

Refaktorering

```
const Hello = (props) => {
  const name = props.name
  const age = props.age

  const bornYear = () => new Date().getFullYear() - age

  return (
    <div>
      <p>Hello {name}, you are {age} years old</p>
      <p>So you were probably born in {bornYear()}</p>
    </div>
  )
}
```

Destrukturering

Vi kan plocka ut båda props

```
const Hello = (props) => {
  const { name, age } = props
  const bornYear = () => new Date().getFullYear() - age

  return (
    <div>
      <p>Hello {name}, you are {age} years old</p>
      <p>So you were probably born in {bornYear()}</p>
    </div>
  )
}
```

Destrukturering

Vi kan plocka ut båda props

```
const Hello = ({ name, age }) => {
  const bornYear = () => new Date().getFullYear() - age

  return (
    <div>
      <p>
        Hello {name}, you are {age} years old
      </p>
      <p>So you were probably born in {bornYear()}</p>
    </div>
  )
}
```

Destrukturering

Vi kan antingen plocka ut i

- variabel
- direkt i komponent

```
const Hello = (props) => {
  const { name, age } = props
```

```
const Hello = ({ name, age }) => {
```

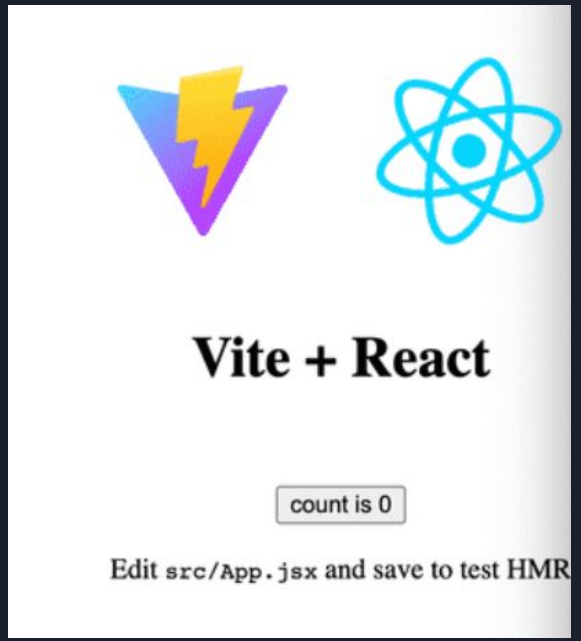
Övning: Modifiera SpiceProduct

- Lägg till en hjälpfunktion som räknar ut pris (pris/gram * vikt/paket)
- Destrukturera props i komponent

Page re-rendering

Hittills så har vi bara bara renderat en applikation en gång, men vad händer om vi skulle trigga en counter (som exempel vite + react).

Hur kommer då renderingen att fungera?



Page re-rendering

Om vi skulle försöka själva?

- Hur skulle vi kunna visa counter = 1, sedan = 2?

```
import ReactDOM from 'react-dom/client'

import App from './App'

let counter = 1

ReactDOM.createRoot(document.getElementById('root')).render(
  <App counter={counter} />
)
```

```
const App = (props) => {
  const {counter} = props
  return (
    <div>{counter}</div>
  )
}

export default App
```

Page re-rendering

Vi skulle kunna skapa en funktion!

```
let counter = 1

const refresh = () => {
  ReactDOM.createRoot(document.getElementById('root')).render(
    <App counter={counter} />
  )
}

refresh()
counter += 1
refresh()
counter += 1
refresh()
```

Page re-rendering

Detta fungerar inte eftersom det går för snabbt att se. Vi kan fixa problemet på följande sätt

```
setInterval(() => {
  refresh()
  counter += 1
}, 1000)
```

Detta funkar men är inte det rekommenderade sättet. Här bootar vi upp appen flera gånger…

State

Demo

```
import { useState } from 'react'

const App = () => {
  const [ counter, setCounter ] = useState(0)

  setTimeout(
    () => setCounter(counter + 1),
    1000
  )

  return (
    <div>{counter}</div>
  )
}

export default App
```

State

Övningar:

- Gör en applikation som gör samma sak (använd slides)
- Gör en applikation som först visar ditt förnamn och efter 3 sekunder visar ditt efternamn
- Gör en applikation som ena sekunden visar ditt förnamn andra sekunden ditt efternamn
  - tips: använd setInterval istället för setTimeout
- Gå tillbaka till hemsidan med krydd produkter
  - Ändra så att produkterna efter 3 sekunder blir gratis (0 kr) (tips: setTimeout och setState i App.jsx)
  - Ändra så att produkterna varannan sekund ändras från gratis till ursprunglig kostnad (tips: använd setInterval istället)