

# Teorihandbok

## Del 1 - endast begrepp och koncept

### 1. Ramverket React (en översikt, vad som är unikt för ramverket, fördelar, ev nackdelar, osv)

React är ett populärt JavaScript-baserat ramverk för att bygga användargränssnitt (UI) på webbsidor och webbapplikationer. React fokuserar på att skapa interaktiva och skalbara UI-komponenter med hjälp av en komponentbaserad arkitektur.

En av fördelarna med React är att det är komponentbaserat. En komponent representerar en del av användargränssnittet och kan innehålla logik och återanvändbara delar av kod. Komponenter kan vara enkla och återanvändbara, eller de kan sättas samman för att bilda mer komplexa användargränssnitt. Genom att dela upp gränssnittet i komponenter blir koden mer organiserad och lättare att hantera.

React använder också ett koncept som kallas "Virtual DOM" (Virtuell DOM) för att effektivt hantera uppdateringar av webbsidor, istället för att direkt manipulera DOM (Document Object Model). React använder sig även av JSX som är en syntaktisk utvidgning av JavaScript. JSX gör det möjligt att skriva HTML-liknande kod direkt i JavaScript-filer. Detta gör det enklare att skapa och manipulera UI-komponenter i React.

### 2. Vad innebär Rendering och Virtual DOM?

Rendering är processen att generera en visuell representation av en webbsida eller ett användargränssnitt baserat på underliggande data och komponenter. När en webbsida laddar tolkar webbläsaren HTML-, CSS- och JavaScript-koden och visar innehållet på skärmen.

I vanliga webbutveckling innebär rendering ofta att manipulera den faktiska DOM (Document Object Model) direkt. Det innebär att du gör ändringar i HTML-elementen och deras egenskaper för att uppdatera användargränssnittet. Den här metoden är ofta långsam och inte så effektiv om man har lite mer avancerade komponenter som ska renderas.

För att optimera renderingsprestanda introducerade JavaScript-ramverk begreppet Virtual DOM.

Virtual DOM är en abstraktion av den faktiska DOM som finns i minnet. Det är en lätt kopia av den verkliga DOM, som representerar strukturen och egenskaperna hos UI-komponenterna.

När du använder ett ramverk som React, istället för att göra direkta ändringar i den verkliga DOM, arbetar man med Virtual DOM. Alla ändringar som görs i den virtuella DOM jämförs med den tidigare versionen av den virtuella DOM, och ramverket fastställer den minsta uppsättning verkliga DOM-uppdateringar som krävs för att återspegla dessa ändringar, detta gör att renderingen går mycket snabbare och blir mer smooth.

### 3. Vad är JSX? Vad används det till?

JSX står för JavaScript XML. Det är ett syntaxtillägg för JavaScript som låter dig skriva HTML-liknande kod i JavaScript. JSX används främst i samband med React. JSX är ett sätt att beskriva strukturen och utseendet på UI-komponenter på ett enkelt sätt. Den kombinerar kraften i JavaScript med flexibiliteten i HTML-liknande syntax, vilket gör det möjligt för utvecklare att definiera UI-hierarkin och komponentlogiken i en enda fil.

#### 4. Vad är ett undantag inom programmering? Eng. Exception

I programmering är ett undantag en ovanlig eller oönskad händelse eller felaktig situation som uppstår under körningen av ett program. När ett undantag inträffar bryter det normala flödet av programmet och kan leda till att programmet avbryts eller att specifika åtgärder vidtas för att hantera undantaget.

Undantag används för att hantera och signalera felaktiga förhållanden eller ohanterade situationer som kan uppstå vid körning av program. De kan orsakas av felaktiga indata, programlogikfel, problem med resursåtkomst (t.ex. filer eller nätverk) eller andra oväntade omständigheter.

#### 5. Vad innebär autentisering inom webbapplikationer? Vad används det till?

Autentisering inom webbapplikationer handlar om att verifiera identiteten för en användare eller enhet innan de får åtkomst till skyddade resurser eller funktioner. Det är en viktig del av att säkerställa att endast behöriga användare kan använda och interagera med en webbapplikation.

Autentisering används för att skydda applikationens resurser och data från obehörig åtkomst och för att säkerställa att bara behöriga användare kan interagera med vissa funktioner. Det ger också en grund för att implementera behörighetskontroller, som definierar vilka användare som har rättigheter att utföra vissa åtgärder eller få åtkomst till specifika resurser inom applikationen.

Vanliga autentiserings som används i webbapplikationer är användarnamn och lösenord, tvåfaktorsautentisering, social autentisering med facebook, google eller twitter.

#### Del 2 - sammanfattning av projekt

I min portfölj som jag byggt med hjälp av React har jag använt mig utav React biblioteket React Hook Form som ger en enkel och smidig metod för att hantera formulärstaten och validering i mina React-komponenter med hjälp av Reacts hooks-koncept.

Jag har också använt mig av React biblioteket react-anchor-link-smooth-scroll som används för en smidig scrollning till ankareffekter. Det gör det möjligt att skapa länkande navigeringslänkar som kan användas för att navigera till olika delar av en sida med hjälp av en mjuk scrollningseffekt.

I exemplet nedan visas hur ankareffekten sätts till "#home", vilket antyder att klicket på länken kommer att scrolla till elementet med id "home" på samma sida.

```
<AnchorLink
  className={` ${
    selectedPage === 'home' ? selectedStyles : "bg-dark-grey"
  } w-3 h-3 rounded-full`}
  href="#home"
  onClick={() => setSelectedPage('home')}
/>
```

Detta navigeras via min komponent som heter DotGroup som utgör 4 runda navigeringscirklar på höger sida av webbsidan. Den tar emot tar emot selectedPage och setSelectedPage som egenskaper och använder selectedStyles för att definiera CSS-klasser för den valda stilen.

```
const DotGroup = ({ selectedPage, setSelectedPage }) => {  
  const selectedStyles = `relative bg-yellow before:absolute before:w-6  
before:h-6  
before:rounded-full before:border-2 before:border-yellow before:left-[-  
50%]  
before:top-[-50%]`;  
}
```

Jag har även använt mig utav React biblioteket Framer Motion som används till animationshantering i React. I exemplet nedan används motion.div-komponenten från Framer Motion för att animera en div som innehåller olika egenskaper och props för att skapa animationen av att header texten glider in i bild från vänster sida.

```
<motion.div  
  initial="hidden"  
  whileInView="visible"  
  viewport={{ once: true, amount: 0.5 }}  
  transition={{ duration: 0.5 }}  
  variants={{  
    hidden: { opacity: 0, x: -50 },  
    visible: { opacity: 1, x: 0 },  
  }}  
>
```

Det som finns inuti return i en komponent kallas JSX, i JSX har jag skapat element som liknar HTML-kod, men det är egentligen JavaScript-uttryck som kommer att renderas till den verkliga DOMen när komponenten används. I JSX har jag använt HTML-taggar, attribut och händelser.

```
<div className="md:flex justify-center md:justify-between text-center ">  
  <p className="font-playfair font-semibold text-2xl text-yellow">  
    JOSEFIN BERTSSON  
  </p>  
</div>
```

I det här exemplet kan man se JSX-element i form av html syntax för <div> och <p> .

I min portfölj har jag med hjälp av react delat upp mina kodfiler i 3 olika undergrupper, det är components, hooks och scenes. I components ligger mina komponenter som jag återanvänder i portföljen på olika sidor som till exempel min Line. Med hjälp av react kan jag skriva en component för att sedan återanvända samma component som jag gör på contact, my skills och projects.

```
<LineGradient width="w-2/3" />
```

Jag har även använt mig utav en anpassad hook som heter "useMediaQuery", den används för att övervaka och returnera om rätt givet mediefrågeuttryck matchar. useMediaQuery-hooken används alltså för att kontrollera om fönstrets bredd är mindre än eller lika med 768 pixlar och visar sedan en text som indikerar om det är en mobil enhet eller inte.

```
// Lyssna på ändringar i fönstrets storlek
const listener = () => setMatches(media.matches);
window.addEventListener("resize", listener);
```

När jag renderar en komponent i React placeras den först i Virtual DOM, där den jämförs med den befintliga virtuella DOM-en och uppdateras vid behov. Sedan appliceras ändringarna på den faktiska DOM-en för att visa de uppdaterade komponenterna på webbsidan.