

Del 1

Begrepp och koncept

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

React är ett populärt JavaScript-ramverk för att bygga användargränssnitt och interaktiva webbapplikationer. Det använder en komponentbaserad arkitektur och har en unik funktion kallad Virtual DOM för snabbare rendering. Fördelarna med React inkluderar komponentåteranvändning, effektiv uppdatering av gränssnittet med Virtual DOM, enkel integration och ett stort community och ekosystem.

Nackdelar kan vara inlärningskurvan och att det kan vara överkvalificerat för enklare webbsidor. Sammanfattningsvis är React ett kraftfullt verktyg för att bygga interaktiva webbapplikationer, men kan vara överdrivet för enklare projekt.

2. Vad innebär Rendering och Virtual DOM?

Rendering:

Rendering handlar om att göra programkod synlig för användaren. I webbutveckling innebär det att skapa och visa användargränssnittet i webbläsaren baserat på kod och data. Det inkluderar tolkning av HTML, CSS och JavaScript för att visa innehållet och funktionaliteten på en webbsida.

Virtual DOM:

Virtual DOM (virtuell dokumentobjektmodell) är en teknik som används av React och vissa andra JavaScript-ramverk. Det är en virtuell kopia av den faktiska DOM-strukturen i webbläsaren. Virtual DOM används för att effektivt hantera och uppdatera användargränssnittet.

Processen med Virtual DOM kan sammanfattas i följande steg:

1. När tillståndet i en React-komponent ändras, skapas en ny virtuell representation av DOM:en för komponenten.
2. Den virtuella DOM:en jämförs med den tidigare virtuella DOM:en för att identifiera skillnader eller uppdateringar som behöver göras i gränssnittet.

3. Endast de delar av den virtuella DOM:en som har ändrats uppdateras i den faktiska DOM:en genom en "diff"-operation.

4. Uppdateringarna i den faktiska DOM:en återspeglas i webbläsaren och användaren ser ändringarna på skärmen.

Virtual DOM används för att optimera prestandan vid uppdateringar av användargränssnittet. Istället för att direkt ändra den faktiska DOM:en för varje ändring, görs ändringarna i den virtuella DOM:en, vilket är snabbare. Sedan jämförs de virtuella DOM:arna för att bara göra de nödvändiga ändringarna i den faktiska DOM:en. Detta minskar antalet ändringar och gör renderingen mer effektiv.

Genom att använda Virtual DOM kan React optimera prestandan och göra användargränssnittsuppdateringar snabbare och mer effektiva, vilket ger en bättre användarupplevelse.

3. Vad är JSX? Vad används det till?

JSX står för "JavaScript XML" och är en syntaktisk utökning av JavaScript som används främst i ramverket React för att skapa användargränssnitt. JSX låter utvecklare skriva HTML-liknande kodsyntax direkt inuti JavaScript-kod.

JSX gör det möjligt att skapa och manipulera DOM-element på ett deklarativt sätt genom att kombinera HTML-strukturer med JavaScript-logik. Det låter utvecklare skapa återanvändbara komponenter och rendera dem i webbläsaren.

När en React-komponent används med JSX skapar utvecklaren en trädstruktur av komponenter som beskriver hur användargränssnittet ska se ut. Detta möjliggör en effektiv och flexibel hantering av gränssnittskomponenter och deras uppdateringar baserat på applikationslogik eller användarinteraktioner.

En förenklad JSX-exempelkod kan se ut så här:

```
```jsx const element = <h1>Hello, World!</h1>; ```
```

I det här fallet skapas en variabel `element` som innehåller ett HTML-element `

# ` med texten "Hello, World!". Detta element kan sedan renderas i en React-applikation för att visas i webbläsaren.

Genom att använda JSX kan utvecklare kombinera JavaScript-kod och HTML-strukturer på ett smidigt sätt, vilket gör det lättare att utveckla och underhålla användargränssnittet i React-baserade applikationer.

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

Inom programmering används termen "undantag" eller "exception" för att beskriva oväntade fel eller avvikelser som kan uppstå när ett program körs. Ett undantag uppstår när något går fel och avbryter det vanliga flödet i programmet.

Det finns olika skäl till varför undantag kan inträffa, till exempel felaktig inmatning av data, oförmåga att hitta en fil, otillgängliga resurser eller försök att utföra ogiltiga operationer. När ett undantag inträffar, bryts den normala körningen av programmet och övergår till en speciell hanteringsmekanism för undantag.

Undantagshantering är en teknik som används för att hantera undantagssituationer på ett strukturerat sätt istället för att programmet kraschar eller ger vagt formulerade felmeddelanden till användaren. Genom att implementera undantagshantering kan utvecklare fånga och hantera undantag genom att definiera specifika åtgärder som ska vidtas när ett visst undantag uppstår. Detta kan innebära att logga fel, visa tydliga felmeddelanden för användaren, återställa till en tidigare tillstånd eller vidta andra lämpliga åtgärder för att hantera felet.

Undantagshantering bidrar till att göra program mer robusta och tillförlitliga genom att hantera oväntade fel och undvika programkrascher eller odefinierat beteende. Genom att identifiera och hantera undantag på ett lämpligt sätt kan utvecklare förbättra användarupplevelsen och underlätta felsökning och felkorrigering i programmet.

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

Autentisering inom webbapplikationer handlar om att kontrollera och bekräfta identiteten hos en användare eller enhet för att ge åtkomst till specifika funktioner eller data. Det görs vanligtvis genom att användaren anger inloggningsuppgifter, som användarnamn och lösenord, som jämförs med sparade uppgifter för att bekräfta att användaren är auktoriserad. Det finns också andra autentiseringsmetoder, som säkerhetsfrågor eller biometrisk data, för att verifiera identiteten. Autentisering är

viktigt för att skydda användardata och säkerställa att endast rättmätiga användare får åtkomst till känslig information och funktioner i en applikation.

## **Del 2 - sammanfattning av projekt**

I denna del har jag redogjort hur punkterna ovan hänger ihop med min portfölj. När jag renderar en av mina komponenter, vart hamnar den först? I DOM eller i Virtual DOM? När jag skriver en return inuti en komponent, vad kallas det som finns inuti returnen?

När datorn renderar en React-komponent för första gången, hamnar den först i Virtual DOM. React skapar en virtuell representation av komponentens DOM-struktur i minnet, vilket kallas Virtual DOM. Denna virtuella representation används för att jämföra och uppdatera den verkliga DOM:en effektivt.

När jag skriver en ``return``-sats inuti en React-komponent kallas det som finns inuti ``return``-satsen för JSX (JavaScript XML). Det är en syntaktisk utökning av JavaScript som tillåter mig att skriva HTML-liknande struktur och komponenter direkt i JavaScript-koden. JSX används för att definiera och beskriva hur komponenten ska renderas i användargränssnittet. När komponenten renderas och JSX-koden utvärderas, skapar React element som representerar de olika delarna av användargränssnittet. Dessa element används sedan för att skapa den virtuella DOM:en och uppdatera den verkliga DOM:en vid behov.