

Johan Hansson SUT24 – Teorihandbok Webbutveckling

1. Vad är HTML och vad används det till? Beskriv också kort historiken för HTML.

HyperText Markup Language är standardspråket för att sätta strukturen för innehåll på webbsidor. Språket ger oss möjlighet att skapa rubriker, texter, bilder och länkar på en webbsida. Språket i sig ger oss bara möjligheten att skapa en struktur på en webbsida men inte utseende eller beteende. För dessa delar använder vi CSS och Java Script. Tim Berners-Lee skapade den första versionen av HTML 1991. Detta i samband med att han uppfann och utvecklade World Wide Web, vilket är ett system för att länka samman dokument och webbsidor via Internet. Inspirationen till HTML kommer från SGML, Standard Generalized Markup Language. Bland annat så kommer idén om att märka upp innehåll med taggar från SGML. Kvarlevor så som `<p>`(paragraf), ``(listföremål) och `<h*>`(rubriksnivåer) är hämtat från SGML.

Sammanfattat är HTML ett flexibelt språk som både ger oss möjligheten att strukturera webbsidor men också ge dem interaktiva och semantiska egenskaper vilket innebär att språket definierar syftet och betydelsen vilket skapar en mer organiserad webbsida.

2. Vad är CSS och vad används det till? Beskriv också kort historiken för CSS.

Om man drar jämförelsen med att en hemsida är som ett hus där HTML är skelettet definierar vi utseendet med CSS. Dvs färger, layout och typsnitt. Innehållet finns ofta i HTML medans stylingen sker med CSS.

CSS är ett sk. Stilbladsspråk som hanterar presentationen av ett webbdokument. Det utvecklades av Håkon Wium Lie 1994 och två år senare kom detta att bli klassat som webstandard av World wide web consortium. Innan CSS utvecklades använde man sk. Inline-styling direkt i HTML-dokumentet. Men tack vare CSS kan vi idag skapa mer avancerade och enhetliga layouter på webbsidor.

3. Förklara vad responsiv design innebär.

Detta är hur en webbsida anpassar sig efter vilken typ av enhet den öppnas från. Texter och bilder på en webbsida annorlunda om den öppnas från en dator kontra en mobil. Responsiv design gör att innehållet blir användarvänligt oavsett vad du använder för enhet. Genom att anpassa designen med flexibla grids och media queries så visas innehållet på det sätt som är mest lämpat för den typen av enhet som användaren använder.

4. Vad är JavaScript och vad används det till?

Om vi fortsätter på jämförelsen till ett hus där HTML och CSS står för innehåll och utseende så kan man kort säga att JS hanterar funktionerna för att webbsidor ska bli interaktiva. Med detta menas att en webbsida kan reagera på användarens val på sidan utan att behöva ladda om sidan hela tiden. Några användningsområden som är kopplat till JS är till exempel att **hantera användarinteraktioner**, om användaren klickar på en knapp så kan JS hantera detta och svara i realtid. **Manipulera DOM**, JS kan justera och ändra innehåll på en sida genom att manipulera Document object model som är en

representation av intern HTML-dokumentet. Dvs du kan ta bort, lägga till eller justera olika element på en sida utan att behöva ladda om sidan varje gång.

Andra användningsområden för JS på en webbsida är att validera användar-input och skapa animeringar för att förbättra användarupplevelsen.

5. Vad är ECMA-script och hur hänger det ihop med JavaScript?

ECMA-script är en standard som definierar ett skriptspråk, dvs det specificerar hur ett skriptspråk ska fungera när det kommer till syntax, typdefinitioner och funktionalitet.

JS är baserat på denna standard men JS omfattar också ytterligare funktioner så som att manipulera webbsidor, kommunicera med servrar och att hantera asynkrona operationer vilket innebär att hantera uppgifter i ett program utan att stanna upp resterande delar medan dessa blir klara, exempel på detta kan vara ett API-anrop. Medans anropet sker så kan det fortfarande uppdatera gränssnittet medan det inväntar anropet.

6. Förklara översiktligt vilket ansvar HTML, CSS och JavaScript har i teknikstacken inom frontend på webben.

HTML: Strukturen för en webbsida. I HTML definierar vi elementen på en sida så som länkar, rubriker, paragrafer/stycken, bilder etc. Vi organiserar innehåller och beskriver strukturen.

CSS: Med CSS styr vi utseende av webbsidan. Med färger, typsnitt, layout etc så styr vi HTML-elementens visuella delar. Vi designar webbsidan genom att använda oss av CSS.

JS: Detta ansvarar för interaktiviteten och dynamiken på en webbsida. Vi kan med hjälp av JS förändra och uppdatera en webbsida i realtid utan att behöva ladda om sidan varje gång.

7. Beskriv vad DOM är och hur vi använder det när vi skapar en hemsida.

Document object model, är ett programmeringsgränssnitt för att manipulera t ex HTML. DOM representerar olika dokument av ett objekt där varje objekt är en del av webbsidan så som olika element, attribut eller texter. DOM gör det möjligt för t ex JS att dynamiskt integrera med en webbsida i realtid så som att ändra texter, bilder, lägga till länkar eller ta bort olika element utan att behöva ladda om sidan hela tiden. Kortfattat kan man säga att DOM är som ett gränssnitt mellan webbsidan och JS som kan ändra dokumentets innehåll och struktur.

8. Vad menas med ett JavaScript-ramverk och vad tillför det till ett projekt jämfört med att bara använda ren JavaScript?

När vi pratar om ett JS-ramverk så är det ett effektivt sätt att bygga webbsidor då det har fördefinierade bibliotek och verktyg som hjälper utvecklaren att skapa webbsidor snabbare. Ett ramverk innehåller strukturer och funktioner för att hantera vanliga problem när det kommer till webbutveckling så som hantering av användargränssnitt, dataflöden och asynkrona operationer. Fördelar med att använda ett ramverk är att du som utvecklare får en struktur att utgå ifrån, jobbar man flera stycken på samma projekt så blir det lättare att samarbeta då alla utgår från samma struktur vilket också leder till

att underhållet av webbsidan blir lättare över tid. Använder du “bara” JS så kan stora projekt vara lite mer problematiska att hålla ordning på vilket i längden. Utan ett ramverk så får utvecklaren en stor flexibilitet men samtidigt måste utvecklaren hantera “allt” från grunden vilket innebär felhantering så som datahantering och routing.

9. Lista tre vanligt förekommande JavaScript-ramverk. För varje ramverk ska du sedan kort beskriva det. Efter din lista med de tre ramverken ska du översiktligt förklara vad som skiljer dessa åt.

De vanligaste ramverken inom JS är Angular, React och Vue.

Angular är ett komplett ramverk som innehåller bl.a. komponenthantering, testning och olika byggverktyg. Detta är utvecklat av Google och har en stor community förutvecklare och kan därför ge bra support vid problem eller frågeställningar.

Angular använder sig av sk. Tvåvägs data bindning vilket innebär att data mellan modell och vy synkroniseras per automatik vilket gynnar dynamiska applikationer. Baksidan av Angular är dock att det är ett avancerat ramverk som inte lämpar sig för nybörjare och på grund av dess storlek så kan det påverka laddningstider och prestanda på vissa webbsidor.

React är som ett flexibelt bibliotek med möjlighet att kombinera med andra verktyg för skräddarsydda lösningar. Med hjälp av virtuell DOM minskas antalet DOM-uppdateringar och på så vis förbättrar hastigheten. Arkitekturen är komponentbaserad vilket underlättar återanvändandet av kod och gör den mer skalbar. Tyvärr är React inte ett komplett ramverk utan behöver integreras med andra verktyg för att bygga kompletta lösningar.

Vue är ett ramverk som är designat för att lätt kunna lära sig och använda vilket passar bra för nybörjare då det har en enkel struktur som ligger nära HTML, CSS och JS.

Till skillnad från exempelvis Angular som kräver Typescript och en mer specifik struktur, så kan man skriva Vue-kod direkt i en vanlig HTML-fil. Vues dokumentation är mycket omfattande och tydlig vilket är en annan aspekt till varför detta är ett ramverk för nybörjare som snabbt kan komma igång och förstå dess funktioner.

Likt React använder Vue sk. komponentbaserad arkitektur vilket innebär att du kan skriva en komponent med all funktion och struktur på ett ställe utan att behöva konfigurera särskilda verktyg eller beroenden av andra delar. Tyvärr har Vue en mindre community jämfört med de andra beskrivna ramverken vilket kan leda till att det blir svårare att hitta resurser. Vue skapades 2015 av en tidigare Google-anställd vid namn Evan You. En riskfaktor att ha med gällande Vue är att det drivs av ett oberoende utvecklingsteam utan någon större aktör i ryggen och på så vis kan framtida utveckling vara osäker.

Beroende på vilken typ av applikation, dess storlek, komplexitet samt utvecklarens egen erfarenhet avgör lite vilket ramverk som hen bör ta sig an.

Angular är ett fullständigt ramverk som passar bra för större och mer komplexa projekt och applikationer. Men vill man jobba med ett ramverk som stödjer hög prestanda och flexibilitet med mycket interaktiva delar så är React ett bra val.

Vue å andra sidan är ett bra val för nybörjare och mindre projekt som kräver snabb utveckling och smidiga lösningar som inte är så tunga.

10. Vad är tillgänglighetsanpassning av webbplatser och varför är det viktigt?

Med tillgänglighetsanpassning menar man att skapa webbsidor som kan användas av alla människor, inklusive t ex en person med funktionsnedsättning. Exempel kan vara blinda, döva eller rörelseförhindrade personer. Tillgänglighetsanpassning skapar möjligheter att kunna nyttja en webbsida gränssnitt som är läsbara och användarvänliga för alla människor.

Exempel på olika typer av tillgänglighetsanpassning är t ex:

Alt-texter som beskriver en bild i ord som en Skärmläsare kan läsa upp för en person med synnedsättning. Dessa Skärmläsare kan även navigera runt resterande del av webbsidan för att läsa upp för användaren. Andra delar av tillgänglighetsanpassning kan vara undertexter på videoklipp, tydlig kontrast på bilder, tangentbordsanpassning för människor som t ex inte kan använda en datamus.

Alla människor bör ha möjligheten att kunna använda en webbsida oavsett personliga förutsättningar så som ett handikapp. Idag finns det lagkrav i många länder som säger att webbsidor ska vara anpassade så att alla ska ha möjligheten att använda dem. Som exempel WCAG, som står för Web content accessibility guidelines och är internationella riktlinjer för webbsidors tillgänglighet.

11. Vad är ett webb-API och vad används det till?

Webb-API(Application programming interface) är en uppsättning med olika funktioner och regler som möjliggör för olika applikationer att kommunicera med varandra online. Ett API innehåller gränssnitt som en applikation kan använda för att t ex skicka förfrågningar och olika begäran för att sen ta emot svar från andra tjänster eller databaser. På så vis behöver inte utvecklaren skriva all kod från grunden själv utan kan nyttja befintliga sidors information för egen utveckling.

Dessa begäran kan t ex vara **GET**, hämta information, **POST**, skicka data, **PUT**, uppdatera information eller **DELETE**, ta bort information. Begäran skickas beroende på vad utvecklaren vill utföra för typ av åtgärd.

Ett API kan användas för att hämta data från externa källor så som nyheter, väderinformation, sociala medier etc. Ett API gör det också möjligt att implementera tredjeparts-tjänster i utvecklarens egen applikation. Exempel på detta kan vara om en webbsida använder en extern betaltjänst.

Fördelarna med ett webb-API kan vara många, t ex, **Återanvändbarhet** då man kan använda samma API till olika webbsidor och mobilappar, **Modularitet** vilket innebär att utvecklaren kan separera olika delar av en applikation med en del för backend som

hanterar logik och databasinteraktion och en annan del för frontend som kommunicerar med backend via det API:et.

12. Förklara REST och redogör för dess huvudprinciper.

REST står för Representational state transfer och är en arkitekтуell still för att bygga webb-API:er. Som jag nämnde i frågan innan så kan man använda HTTP-metoder så som GET, POST, PUT och DELETE för att manipulera resurser på servern.

Huvudprinciperna för REST:

Stateless, en server ska inte hålla något tillstånd mellan förfrågningar och begäran pga att det blir enklare att skalera. Varje gång en klient skickar en begäran till servern behandlas den som helt ny utan att servern kommer ihåg tidigare interaktioner.

HTTP-metoder, För att manipulera resurser så ska HTTP-metoderna GET(hämta data), POST(skapa data), PUT(uppdatera data) och DELETE(ta bort data) användas.

Resurser, Alla objekt i systemet representeras som resurser och kan identifieras med URL:er.

Representation, När en resurs begärs via ett API får klienten en representation av den resursen som t ex en JSON-fil eller XML-fil.