

TEORIHANDBOKEN

WEBBUTVECKLING

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

HTML står för Hyper Text Marked Language och det är med HTML-koden som definierar och strukturerar basinformationen till en hemsida (ex text, bilder, länkar och multimedia). HTML markerar för en web-browser vad som är titel, paragraf, fet text, kursiv, länkar, bilder etc.

HTML utvecklades -89/-90 av Tim Berners-Lee, som en del av "World Wide Web-projektet" vid CERN (European Organization for Nuclear Research). HTML utvecklades för att beskriva strukturen på forskningsrapporter för att göra delning av dessa enklare och effektivare tidigare versionerna av internet.

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

CSS står för Cascading Style Sheet. CSS är ett stilspråk som används tillsammans med HTML (eller annat märkspråk) och som definierar utseende, stil och layout på webbsidor. CSS styr element som färg, typsnitt, marginaler och responsiv design och brukar även kallas just stilmall.

CCS introducerades första gången 1996 av W3C (World Wide Web Consortium) för att separera innehåll (HTML) från design men har sedan utvecklats till en avancerad teknik för att bygga moderna hemsidor och det fortsätter att spela en avgörande roll i framtiden för webbutveckling.

3. Förklara vad responsiv design innebär.

Responsiv design innebär att en webbplats skalas per automatik efter skärmupplösning, dvs oberoende om du tittar på den via en mobil, dator etc så ska innehållet se bra ut.

4. Vad är JavaScript och vad används det till?

Javascript är ett skriptspråk som används för att skapa dynamiska och interaktiva webbsidor. Det körs på klientsidan (direkt i webbläsaren) och kan manipulera innehåll och utseende på webbsidor, svara på olika användarinteraktioner (klick, input mm), hämta och kommunicera med servrar, skapa animationer, lagra data mm mm.

* dvs ett datorspråk som genom små program (script) kan ge instruktioner till och styra annan programvara (t ex till webbläsaren, servrar mm) eller automatisera uppgifter.

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

ECMA international tar fram standarder för I/O industrin. ECMAScript är det skriptspråk som är standardiserat av ECMA i standarden ECMA-262. Den första utgåvan av ECMA-262 gavs ut 1997 och var då sprungen ur Javascriptspecifikation (Javascript presenterades första gången 1996).

Både Javascript och Microsofts JScript siktar idag på att vara kompatibla med ECMA-262. Men eftersom både Javascript och JScript innehåller utökningar som inte är specificerade i ECMA-262 så kan man säga att ECMAScript utgör en delmängd av Javascript och JScript. ECMAScript kan tolkas av både JS och JScript men en ECMAScript-tolk kan nödvändigtvis inte tolka all kod i JS och JScript.

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

En kan tänka att HTML är substantiv, CSS adjektiv och Javascript är verb. En grön(CSS) dino (HTML) dansar (JavaScript). Mao, HTML ansvarar för VAD - det är med HTML-koden som vi definierar basinformationen till en hemsida (fr a text, titlar, stycken, länkar, bilder etc). CCS hantera HUR det ser ut. JavaScript handlar om vad som HÄNDER när man ex klickar på en knapp, skrolla med musen etc.

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

DOM står för Document Object Model och är en Javascript-representation av en webbsida och består av en massa Javascript-objekt som vi kan interagera med via JS.

Javascriptobjekten i DOMen är organiserade i en trädstruktur som motsvarar hierkin i HTML-innehållet. Dokumentobjektet är vår entrédörr till DOMen, den högsta noden, och innehåller alltså sJS-objekt-representationer av allt innehåll på sidan samt en massa metoder och egenskaper som vi kan använda oss av..

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

JS-ramverk förser oss med mallar och färdiga kodbibliotek som gör det enklare och snabbare att utveckla applikationer och webb-sidor.

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.

React: React skapades 2011 av Facebookutvecklaren Jordan Walke. Det används idag av bolag som Apple, Paypal, Netflix och Facebook. Det är baserat på återanvändningsbara och oberoende komponenter.

Angular: Javascript-baserat ramverk, skrivet i Typescript och "cross-platform". Utvecklat av Google.

Vue.js: Skapades av tidigare Google-anställd, Evan You.

Skillnader: Angular är ett komplett och övergripande ramverk som kommer med många färdiga funktioner och verktyg men är med det också det kanske mest komplexa att sätta sig in i. React är på andra sidan av skalan - det beskrivs som ett UI-ramverk det är avskalat som möjligt för att erbjuda maximal flexibilitet men det tar också längre tid att komma igång. Vue lägger sig däremellan.

React och Vue har en virtuell DOM gör att hela DOMen inte måste renderas av vid en förändring, utan enbart berörda delar.

I React flödar dessutom data enbart i en riktigt, uppifrån och ner, vilket gör att så få delar som möjligt av DOMen påverkas vid en förändring. Få saker behöver ritas om vilket ger en snabbare applikation.

Medan Angular som sagt verkar täcka in allt, så är React framtaget för att bygga javascriptapplikationer med fokus på snabba användargränssnitt. Det finns därför funktionalitet

som har tonats ner eller helt utelämnats Ett sådant exempel är datahantering. För större applikationer bör kodbibliotek som specialiserar på detta tas in, ex Redux och MobX. Detta kan ses som nackdel om en vill arbeta enbart i ett ramverk. Men specialiseringen - att se till att göra en sak riktigt bra och nyttja kodbibliotek som gör sin grej riktigt bra ses av många även som en styrka.

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

Att en webbplats är tillgänglighetsanpassad innebär att den är tillgänglig för alla - oavsett eventuella funktionsvariationer. Det innebär att design och innehåll ska vara tydligt och det ska vara möjligt att interagera med sidan på olika sätt - mus, tangentbord, läsare mm. Det innefattar att:

- innehållet ska vara presenterat så att alla kan uppfatta och ta del av det.
- innehållet ska gå att hantera och interagera med för alla
- det ska vara lätt att tolka och förstå sida - ex genom enkel text och smidig navigering
- robust teknik ska användas som fungerar på ett brett spektrum av program.

Praktiska exempel på att tillgänglighetsanpassa är inte skapa en sida som enbart består av div-ar utan att välja html-element som så bra som möjligt beskriver vad de visar så att synskadade med skärmläsare lättare kan navigera. Det kan ex också handla om att allt på en sida ska gå att göra med både tangentbord och mus ex. osv.

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

API står för Application Programming Interface och är en bred teknisk term för ett gränssnitt igenom vilket olika program kan kommunicera med varandra. Web-APIer är webb/http-baserade och agerar som portal eller dörr in i andra servrar, applikationer mm. I fallet webb-APIer så skickas en förfrågan till en URL-endpoint och tillbaka fås en data som för det mesta kommer i ett json eller xml-format.

12. Förklara REST och redogör för dess huvudprinciper.

REST står för Representational State Transfer och är en uppsättning av begränsningar eller principer för hur en klient och server ska kommunicerar och utföra så kallade CRUD-operationer på en specifik resurs. CRUD står för Create, Read, Update, Delete (eller Destroy).

RESTful är att följa REST-principerna som beskrivs mer nedan.

Klient-server-arkitektur

Handlar om att separera krav från klientgränssnittet från krav på servern som lagrar data. Klienten för förfrågningar till servrarna och ansvarar för presentationen. Däremellan ansvarar servern för logik och lagring av data.

Genom att separera de två kan de utvecklas oberoende av varandra. Dessutom blir klientkoden mer flexibel och portabel mellan olika plattformar och på serversidan innebär förenkling och separation ökad skalbarhet.

Stateless

Varje förfrågan från klienten måste innehålla all information som behövs för att förstå och hantera/bearbeta den. Varje förfrågan är fristående och det finns inga beroenden till tidigare förfrågningar.

Resurser och representation

Resurser i REST syftar till något som kan identifieras, hämtas, uppdateras eller tas bort via ett unikt URI (Uniform Resource Identifier) och kan vara vad som helst eg som har ett identifierbart tillstånd. Ex på resurser är en användare, ett dokument, en bild, samling av data mm.

Lagerindelat System/Lagerkomponentsystem

I REST-arkitektur kan systemet vara uppdelat i mellanlager med ansvar för olika saker, ex caching och autentisering. Klienten behöver inte tänka på detta utan kommunicerar med servern som en helhet.

Enhetligt gränssnitt

Gränssnitt för att kommunicera med resurser ska vara enhetliga så att det är enkelt att som utvecklare förstå. Man ska resurserna vara tillgängliga via ett API som följer ett standardiserat mönster. Fyra grundläggande HTTP-metoder som ofta används för detta är GET, POST, PUT, DELETE.

Cache

Resurser som skickas mellan server och klient kan cachas för att förbättra prestanda. Klienten kan spara ner cache-bara resurser för att använda senare och behöver då inte upprepa samma förfrågan om och om igen.