

Teorihandbok

Webbutveckling Frontend

Frågor

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

HTML är ett "mark up language" som webbläsare använder för att tolka och ordna text, bilder och annan information på webbsidor. Den grundläggande funktionen av HTML-element definieras i webbläsaren. HTML togs från början fram i början av 1990-talet av forskare på CERN, som ett sätt att dela text (tekniskt sett "hypertext" - text som visas på en skärm) på internet. Först 1995 togs en standard fram (HTML 2.0), men inte förrän år 2000 blev det en internationell standard för webbsidor. Den senaste versionen av HTML (5.0) färdigställdes 2014.

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

CSS (Cascading style sheets) är ett "presentationsspråk" som används för att designa utseende på främst webbsidor. CSS separerar utseendet från innehållet.

Style sheets av olika slag har funnits längre än CSS, men i takt med att internet växte på 1990-talet uppstod diskussioner om behovet att kunna anpassa utseendet på webbsidor efter behov. Ett förslag/utkast till CSS presenterades 1994 på en internetkonferens och en specifikation publicerades för första gången i slutet av 1996.

Utvecklingen av CSS och dess implementation pågår fortfarande under översikt av en ideell arbetsgrupp - "CSS Working Group". Efter standard för CSS sattes med version 2.1 beslöts det att dela upp utvecklandet av CSS i olika moduler med egna versionsnummer.

CSS kan användas i HTML på tre olika sätt:

Inline -> där css-koden skrivs som en egenskap i ett HTML-element.

Internal -> där koden skrivs i ett eget stycke som är inbäddat i ett HTML-dokument

External -> där all CSS ligger i en egen fil, som sedan länkas till från de HTML-dokument som ska använda den.

3. Förklara vad responsiv design innebär.

Responsiv design innebär att layouten av en websida anpassas efter det medium ("viewing environment") som används för att titta på den. Syftet med responsiv design är att öka användarvänligheten och möjligheten att behålla funktionalitet över flera olika medium.

I responsiv design används "relativa enheter" (% , vh, vw mfl.), obestämda placeringar eller rent av helt alternativa vyer (via t.ex. @media) för att en websida ska kunna anpassa sitt innehåll efter skärmen som den visas i.

4. Vad är JavaScript och vad används det till?

JavaScript är ett programmeringsspråk (eller "scriptspråk") som främst används för att skapa dynamiska element på webbplatser. Ett vanligt exempel är att uppdatera utseendet på en webbplats efter användarens preferenser.

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

ECMA-script är en standard för scriptspråk som används för att säkerställa att de fungerar likadant i olika webbläsare. Standarden upprätthålls av en internationell ideell grupp (ecma international) som tillsammans med olika andra organisationer arbetar för att ta fram och utveckla standarder i olika teknikområden.

ECMA-script 262 specificerar syntax för grundläggande funktioner i JavaScript (t.ex. array och function).

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

HTML ansvarar för strukturen - hur en hemsida skall vara uppbyggd och organiserad.

CSS ansvarar för layout och design - hur hemsidan ska se ut och presenteras för användaren.

JavaScript ansvarar för interaktivitet - så att hemsidan kan utbyta information med dess användare.

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

DOM står för "Document Object Model" och är ett programmeringsinterface för webbsidor. Det representerar själva "sidan" i formen av noder och objekt, så att andra program kan interagera med den för att lägga till/ändra innehåll och utseende.

Alla egenskaperna, metoderna och "eventsen" som kan användas för att redigera en webbsida är alltså objekt i DOM:en. Det är dessa som vi sedan använder för att skapa hemsidan. document-objektet som vi använder för att komma åt / lägga till specifika element är t.ex. ett eget objekt i DOM:en.

Utan DOM:en skulle vår kod inte ha möjlighet att tolka vad t.ex. en webbsida eller ett HTML-dokument är för något. DOM:en är fristående från programmeringsspråk, så även fast den används mest av javascript-utvecklare går det att bygga implementeringar av den i andra programmeringsspråk.

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

Gustav Eriksson Söderlund

SUT24

VT25

Ett Javascript-ramverk är en samling av kodbibliotek med färdigskriven kod för återkommande rutinuppgifter. Ett ramverk är en struktur med en speciell kontext som utvecklare kan använda för att skapa webbsidor. Till skillnad från ett rent kodbibliotek så har ett ramverk en struktur som hjälper till att skapa och organisera en applikation.

Den största fördelen är att all kod inte behöver skrivas från början, utan att man kan bygga på redan existerande funktioner för de vanligare uppgifterna koden behöver utföra.

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 är tekniskt sett inte ett ramverk utan ett bibliotek för att bygga användarinterface. Men det kombineras med ett eller fler React-specifika bibliotek (för t.ex. state management) för att uppnå samma funktionalitet som ett ramverk.

React är känt för hög prestanda tack vare användandet av en virtuell DOM - den minimerar tiden som krävs för att uppdatera ett UI eftersom bara de komponenter som faktiskt uppdaterats renderas om.

Vue.js

Vue används för att bygga användarinterface och single-page-applikationer. Det är designat för att vara anpassningsbart, så man kan lägga till interaktivitet i en applikation utan att behöva skriva om all kod.

Next.js

Next.js är ett open source front-end ramverk, byggt på React, som används för att bygga interface och statiska hemsidor. Det kan även användas för att skapa server-side-applikationer. Det använder javascript eller typescript ihop med React-komponenter för att skapa UI. Det kan till skillnad från andra ramverk automatiskt generera sidor baserat på ett projekts filsystem.

Jämförelse

Vue är snabbare än React och Next när det gäller just mindre applikationer, men i mer komplexa system med mycket interaktivitet är det istället React som har högre prestanda. Next.js "för-renderar" innehåll på servern, vilket kan kombineras med t.ex. Reacts virtuella dom-hantering för positiva effekt på laddtider. React kan med sitt component-system enkelt skapa delar som går att återanvända, medan Vue kan vara enklare att börja med och få översikt över i och med systemet med "single page application" (en sida för allt innehåll, som laddas om med nytt beroende på hur användaren interagerar med den.

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

Tillgänglighetsanpassning handlar om att göra en webbplats användbar för så många människor som möjligt, oavsett eventuella begränsningar i till exempel syn eller hörsel.

I praktiken bör en tillgänglighetsanpassad webbsida också vara enklare att interagera med för användaren och ha en tydligare design.

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

Ett API (Application Programming Interface) är ett slags "mjukvaruinterface", som erbjuder en service till andra mjukvaror. De abstraherar mer komplex kod och gör enklare syntax tillgängligt för användaren av det.

JavaScript har tillgång till en mängd olika API:er, som inte är en del av själva språket utan byggda av det. De brukar delas in i två olika kategorier:

Browser-API:er som är inbyggda i webbläsare och gör det möjligt att för JavaScript att använda sig av data från webbläsaren och enheten den befinner sig på. Några av de vanligast använda är t.ex. DOM-api:et som hanterar HTML/CSS i javascript och Fetch API:et som hämtar data från en server.

Tredjeparts-API:er som inte är inbyggda i webbläsaren, utan dess kod och information måste hämtas från något annat ställe på internet. Några vanliga tredjeparts-api:er är Google Maps och Youtubes API:er som gör det möjligt att bädda in deras respektive innehåll i en annan webbsida.

12. Förklara REST och redogör för dess huvudprinciper.

Rest (REpresentational State Transfer) är en typ av arkitektur som används vid byggandet av web-baserade API:er. Det är alltså inte en standard eller ett protokoll, utan en "stil" som man utgår ifrån vid byggandet. Målet med REST-arkitektur är att göra sin applikation enklare och mer skalbar.

Det finns sex huvudprinciper som arbetar tillsammans för att begränsa arkitekturen i ett REST API.

- Uniformt interface

Innebär att serversvar borde, utöver att innehålla data, också innehålla information om vilka andra resurser och åtgärder som finns tillgängliga. Att skicka en förfrågan om information, ska alltså också göra så att klienten sedan kan fråga efter andra saker, utan att ha vetat att de fanns från början. Data i JSON-format innehåller t.ex., förutom datan, också info om struktur, namn, relaterade data och mängd entiteter. När klienten vet detta, kan hen också använda det för att fråga efter ytterligare information.

- Klient-Server design pattern

Innebär att klientens och serverns ansvarsområden är åtskilda (se ovan). Detta ökar överförbarheten mellan plattformar och gör det enklare att skala upp eftersom serverkomponenterna kan hållas relativt simplistiska.

- Statelessness

Innebär att sessionsdata skall hanteras på klientsidan (och att varje "request" som görs behandlas som helt ny). Det innebär också att varje request ska innehålla all information som behövs för att fullfölja den.

- Cacheable

Innebär att ett svar på en request ska tydliggöra (antingen explicit eller implicit) huruvida det är cacheable eller inte. Om det är cacheable, kan klienten återanvända datan vid liknande requests (ofta inom en viss period).

- Layered System

I ett "skiktat" system är komponenter låsta i vissa "skikt" eller "lager". Det innebär att komponenterna inte kan se eller interagera med komponenter i andra "skikt". Alltså separeras deras användning från varandra, vilket bör leda till att det blir enklare att utveckla och underhålla en applikation.

- Code on demand

Code on demand beskrivs som "valfritt", det innebär att kod kan hämtas och användas (exekveras!) på klientdatorn. Det gör att klientapplikationerna kan hållas enklare - serverdatorn skickar med data i form av kod, som klienten bara behöver köra.