

Florent Hajdari - Teorihandbok - Webbutveckling

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

Hypertext Markup Language som det egentligen heter är det grundläggande språket för att strukturera och sätta ihop innehåll på webbsidor. Man skapar text, bilder, länkar och dylikt. Man bygger ihop en struktur med HTML för att sedan fullända med CSS och JavaScript.

HTML skapades av Tim Berners-Lee år 1991 och då hade den första versionen mindre än 20 taggar. Idag har HTML hela 142 varav 115 används mer flitigt.

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

Cascading Style Sheets som det också heter är ett designspråk som används för att ta fram en personlig och unik design av en webbsida. Med anpassningar för färger, typsnitt, layout, avstånd, bilder och andra effekter kan man skapa något som är personligt. Helt enkelt ett redigeringsprogram fast i kodform.

CSS kom till under 1990-talet som ett komplement till HTML. CSS1 släpptes 1996, CSS2 1998 och den mer kraftfulla versionen CSS3 släpptes 2010.

3. Förklara vad responsiv design innebär.

Responsiv design har som mål att skapa en anpassning för användarens enhet och skärm. Alltså anpassas designen för att maximera användarens skärmutrymme och skapa bästa möjliga användarvänlighet. Detta gör man med Media Queries, Flexbox och Grid.

Exempel på anpassning för mobiler:

```
body {  
  font-size: 16px;  
  margin: 0;  
  padding: 0;  
}
```

4. Vad är JavaScript och vad används det till?

Det är ett programmeringsspråk som majoriteten av webbsidor använder. Nästintill alla. Det är ett språk som är integrerat med HTML och även plattformsoberoende. Man utvecklar större applikationer och backend-system med hjälp av olika ramverk, exempelvis Node.JS och React. JavaScript är the main guy när det gäller webbsidor.

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

ECMA är ett skriptspråk som är standard och ligger till grund för JavaScript. Målet med ECMA var att skapa ett enhetligt skriftspråk eftersom att det fanns flera olika. Man kan säga att ECMA är mallen som JavaScript följer, ECMA är själva regelboken. Javascript är med andra ord en implementering av ECMA.

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

Man kan säga att dessa tre samarbetar för att få ut en fullständig produkt. Utan den ena, andra eller tredje blir produkten lidande. HTML sätter skelettet av hemsidan och bestämmer vilka element som ska visas. CSS ansvarar för designen av webbsidan genom anpassning av färger, typsnitt, bilder och mera. Sedan har vi Javascript som gör att användaren kan interagera med sidan och inte bara stirra blint på den. Den tar in användarens klick, hovringar, tangenttryck osv. Allt detta skapar en helhet.

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

DOM står för Document Object Model och är en form av trädliknande struktur för innehållet på en webbsida. Genom DOM kan man läsa, ändra och styra element, stilar och events direkt med hjälp av JavaScript.

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

Ett JS-ramverk är ett fördefinierat kodbibliotek och verktyg som har som mål att förenkla utvecklingen av applikationer. Ramverket erbjuder regler och struktur för hur man ska bygga projektet.

Ramverk tillför struktur först och främst för hur man ska bygga koden. Sedan tillkommer färdiga funktioner och komponenter vilket sparar tid. Genom den tydliga strukturen blir koden enklare att underhålla.

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: JS-bibliotek med öppen källkod som används för att bygga användargränssnitt, speciellt för single page applications. Använder även en virtuell DOM som gör gränssnittet extremt snabbt.

Vue: Smidigt och lätt ramverk som kan vara extremt enkelt men även kraftfullt. Används främst för single page applications. Används för både webb- och mobila appar med hjälp av Electron-ramverket.

Angular: Ramverk skriver i typescript. Även detta funkar mycket bra för single page applications. Kända för sin strikta struktur och fina uppsättning av verktyg.

Uppbyggnad: React fungerar mer som ett bibliotek och är flexibelt. Angular med komplett ramverk med inbyggda funktioner. Vue är mittemellan och men kan anpassas för högre nivåer.

Svårighet: Vue anses vara lättast att lära sig. Angular är klart svårare då det är mer komplext men även krävs förståelse för Typescript. React lätt att komma igång med men det krävs förståelse för JSX.

Användning: React används oftast i dynamiska applikationer. Vue i projekt som kräver flexibilitet och snabb utveckling. Angular i större och komplicerade projekt.

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

Det innebär att man anpassar webbplatser så att det är lättillgängligt för så många som möjligt. Exempelvis folk som är döva, då kan man skapa funktioner i textformat eller dylikt som förklarar läte. Alla ska ha tillgång till webbplatsen och kunna utnyttja funktionerna i allra högsta grad.

Genom att göra detta inkluderar man så många som möjligt vilket skapar mer spridning osv. Sedan handlar det också om att respektera gemene man.

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

Web application programming interface är ett gränssnitt som gör det möjligt för olika program/system att kommunicera med varandra. Det blir som en bro mellan en kund och en server där information skickas per automatik utan att kunden behöver förstå hur det funkar.

API:er används till massvis med saker. Appar kan hämta väderinformation med hjälp av API:er, betalningssystem som PayPal använder API:er. Ska man till exempel bygga en databas med sportstatistik kan man fixa en API som uppdaterar din databas med senaste omgångens statistik per automatik. Detta är en finurlig lösning som gör vardagen smidigare.

12. Förklara REST och redogör för dess huvudprinciper.

Representational State Transfer är en stil på uppbyggnad för att designa API:er som kopplar samman backend och frontend i en applikation. REST används i moderna applikationer för att hämta och uppdatera data på ett effektivt sätt.

Huvudprinciper för REST:

1. **Stateless**: Respektive begäran är oberoende och innehåller all information som behövs. Ingen historik lagras mellan varje begäran.
2. **Resurser**: API hanterar resurser (användare eller produkter) som identifieras med unika URL:er. Skickas sedan i JSON-format eller XML-format.
3. **Enhetliga gränssnitt**: Använder standardmetoder som GET, POST, PUT och DELETE.
4. **Cache**: Information kan cashas för att förbättra prestanda och minska belastning på servern. Fungerar som ett understöd.
5. **Separation**: Kunden och servern har en tydlig barriär. Kunden (frontend) och servern (backend) är separata och är oberoende av varandra.