

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

HTML (HyperText Markup Language) är standardmärkspråket för att skapa och strukturera webbsidor. Det används för att definiera innehåll såsom text, bilder, länkar, tabeller och formulär på en webbsida. HTML är inte ett programmeringsspråk utan ett märkspråk som beskriver sidans struktur med hjälp av element (taggar). Webbläsare tolkar HTML och visar innehållet enligt dess struktur.

1991 skapades HTML av Tim Berners-Lee, uppfinnaren av World Wide Web. Den första versionen var enkel och designad för att dela dokument mellan forskare. Efter några år släpptes HTML 2.0 som en officiell standard av IETF och innehöll grundläggande funktioner. Sedan kom HTML 3.2 och 4.0 som introducerade fler element för tabeller, skript och stilhantering. På 2000-talet kom XHTML och utvecklades som en striktare variant av HTML. Den senaste versionen kom 2015, HTML5 med stöd för multimedia, semantiska taggar och responsiv design. Idag är HTML den dominerande standarden.

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

CSS (Cascading Style Sheets) är ett stilmallsspråk som används för att styra utseendet och layouten på webbsidor. Med CSS kan man separera innehåll (HTML) från presentation, vilket gör det enklare att hantera och ändra designen på en webbplats.

CSS används bland annat för att styra färger, typsnitt och storlek på text. Man kan även hantera layout och placering av element på sidan. Det finns väldigt mycket mer men det blir mycket att skriva ;).

1996 släpptes CSS1 av W3C (World Wide Web Consortium) för att ge webbdesigners möjlighet att styra webbsidors utseende. Ett par år senare kom CSS2 som introducerade fler egenskaper, inklusive bättre stöd för positionering och media-typer. 2011 kom CSS3, det var uppdelat i flera moduler med nya funktioner såsom flexbox, grid, animeringar och avancerade effekter. Idag är CSS en grundläggande del av modern webbutveckling och används tillsammans med HTML och JavaScript.

3. Förklara vad responsiv design innebär.

Responsiv design är en metod för att bygga webbplatser så att de automatiskt anpassar sig till olika skärmstorlekar och enheter, såsom datorer, surfplattor och mobiltelefoner. Målet är att ge en optimal användarupplevelse oavsett vilken enhet som används.

4. Vad är JavaScript och vad används det till?

JavaScript är ett programmeringsspråk som används för att skapa interaktivitet och dynamiskt beteende på webbsidor. Det gör det möjligt att manipulera HTML och CSS i realtid, hantera användarinmatning och kommunicera med servrar via API:er.

JavaScript är en grundpelare i modern webbutveckling och används tillsammans med HTML och CSS för att skapa interaktiva och dynamiska webbplatser.

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

ECMAScript är den standard som JavaScript följer. Det definierar språket JavaScript, inklusive dess syntax, funktioner och beteenden. JavaScript är alltså en implementering av ECMAScript-standarderna.

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

Inom frontend-utveckling har HTML, CSS, och JavaScript tydliga roller i teknikstacken för att skapa och hantera webbsidors struktur, utseende och interaktivitet.

**HTML:** Bygger upp webbsidans grundläggande innehåll. Definierar element som rubriker, paragrafer, bilder, länkar, tabeller och formulär. Skapar en semantisk struktur för bättre SEO och tillgänglighet.

**CSS:** Styr färger, typsnitt, och textstorlek. Man hanterar sidans layout, inklusive positionering och responsiv design med CSS. Man kan även styra effekter och animationer.

**JavaScript:** Med JavaScript lägger man till funktionalitet som respons på användarhändelser t.ex. klick. Man manipulerar även HTML och CSS i realtid, hanterar API-anrop för att hämta och skicka data utan att ladda om sidan. JavaScript används med ramverk och bibliotek som React, Vue och Angular för att bygga avancerade webbapplikationer.

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

DOM (Document Object Model) är ett programmeringsgränssnitt som representerar strukturen av en webbsida som ett träd av objekt. När en webbläsare laddar en HTML-sida skapas en DOM-struktur där varje HTML-element blir en nod i trädet.

DOM gör det möjligt att dynamiskt ändra en webbsida med JavaScript. Man kan hämta element genom att använda metoder för att hitta element t.ex.

```
let rubrik = document.getElementById("rubrik");
```

Man kan även ändra innehåll och attribut, lägga till eller ta bort element, lyssna på händelser (event listeners).

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

Ett JavaScript-ramverk är en fördefinierad uppsättning kod och verktyg som hjälper utvecklare att bygga webbapplikationer snabbare och mer strukturerat. Ramverk tillhandahåller arkitektur och färdiga funktioner som förenklar utvecklingsprocessen, jämfört med att skriva allt i ren (vanilla) JavaScript.

Ramverk erbjuder en tydlig struktur vilket gör koden mer organiserad och lättare att underhålla i ett större projekt. Istället för att manuellt manipulera DOM använder ramverk virtuell eller reaktiv datahantering vilket gör UI-uppdateringar snabbare och mer optimerade. Inbyggda lösningar för routing, state management och API-hantering gör att utvecklare kan fokusera på funktionalitet snarare än att bygga allt från grunden.

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 (bibliotek, men används ofta som ett ramverk) - Fokus på komponentbaserad UI-utveckling. Det är komponentbaserat och använder en virtual DOM, vilket gör det snabbt och effektivt för UI-uppdateringar. React bygger på JSX (Javascript XML), där HTML och JavaScript kombineras i samma kod.

Vue.js - Lättviktigt och flexibelt, bra för både små och stora projekt. Det används för att bygga interaktiva gränssnitt och ensidiga applikationer. Vue har en enkel inlärningskurva och kombinerar det bästa från React och Angular. Det använder reaktiv datamodell och komponentbaserad utveckling.

Angular - Ett komplett ramverk från Google för att bygga stora skalbara webbapplikationer. Det är opinionsdrivet, vilket betyder att det följer en specifik kodstruktur. Angular använder TypeScript istället för vanlig JavaScript och har inbyggda lösningar för routing, state management och dependency injection.

React är mer av ett bibliotek medan Vue och Angular är ramverk och Angular är ett fullständigt ramverk. React använder JS/JSX, Vue JS och Angular TS som språk. De olika ramverken har ganska olika inlärningskurvor.

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

Tillgänglighetsanpassning innebär att en webbplats utformas och utvecklas så att alla användare, inklusive personer med funktionsnedsättningar, kan ta del av innehållet och använda funktionerna utan hinder. Detta inkluderar anpassningar för personer med synnedsättning, hörselnedsättning, motoriska svårigheter och kognitiva funktionsvariationer.

Detta är viktigt eftersom alla, oavsett funktionsförmåga, ska ha lika möjlighet att använda webbplatser och digitala tjänster. Många länder har lagkrav för webbtillgänglighet. Tillgänglighetsanpassning förbättrar webbplatsens struktur och navigering vilket gynnar alla användare. En tillgänglig webbplats når flera användare och kan öka kundbasen och användarengagemanget.

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

Ett webb-API (Application Programming Interface) är ett gränssnitt som gör det möjligt för olika system och applikationer att kommunicera med varandra över internet. Det fungerar som en mellanhand som tar emot förfrågningar, behandlar dem och skickar tillbaka data, ofta i JSON- eller XML-format.

Man hämtar och skickar data mellan klient och server. Integrera tredjepartstjänster t.ex. använda Google Maps API för att visa kartor. Bygga dynamiska webbapplikationer t.ex. En chattapplikation där nya meddelanden hämtas utan att sidan laddas om.

## 12. Förklara REST och redogör för dess huvudprinciper.

REST (Representational State Transfer) är en arkitekturstil för att bygga webb-API:er som kommunicerar över HTTP. RESTful API:er använder standardmetoder som GET, POST, PUT och DELETE för att hantera resurser på en server. REST bygger på stateless communication, vilket betyder att varje förfrågan från klienten till servern innehåller all information som behövs - servern lagrar ingen information om tidigare förfrågningar.

### 1. Klient-server-arkitektur

- Klienten (ex. en webbläsare eller mobilapp) och servern (API:t) är separata enheter och kommunicerar via HTTP.
- Klienten skicka förfrågningar och servern svarar med data.

### 2. Stateless (tillståndslöst)

- Varje förfrågan från klienten är oberoende av tidigare förfrågningar.
- Ingen information lagras om användaren session på servern mellan förfrågningar.
- Fördel: Skalbarhet och enklare hantering av belastning.

### 3. Cachebarhet

- Respons från servern kan cachas (lagras tillfälligt) för att minska belastningen och förbättra prestandan.
- Exempel: En webbläsare kan lagra API.svar för att undvika onödiga anrop.

### 4. Enhetligt gränssnitt (Uniform Interface)

- Alla API-resurser nås via tydliga URL:er och följer samma struktur.

### 5. Lagerindelning (Layered System)

- API:t kan bestå av flera lager (exempelvis lastbalanserare, autentisering, databaser) utan att klienten behöver känna till dessa.

### 6. Kod på begäran

- Servern kan returnera exekverbar kod till klienten (tex. JavaScript), men detta används sällan i moderna REST API:er.