

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

HTML (HyperText Markup Language) är ett märkspråk som används för att strukturera och presentera innehåll på webben. Det är liksom grunden för alla webbsidor och används för att skapa rubriker, stycken, bilder, länkar och andra element på en hemsida.

HTML skapades av Tim Berners-Lee i början av 90-talet när han utvecklade World Wide Web. Första versionen var rätt enkel, men med tiden har det uppdaterats med fler funktioner. En av de största uppdateringarna kom med HTML5, som gjorde det möjligt att hantera multimedia som video och ljud direkt i webbläsaren utan extra plugins.

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

CSS (Cascading Style Sheets) är ett stilmall språk som används för att styla HTML-element på en webbsida. Det bestämmer hur saker ska se ut som till exempel färger, typsnitt, layout, avstånd osv. Med CSS kan man göra webbsidor snygga och responsiva, så att de funkar på olika enheter som datorer, surfplattor och mobiler.

CSS utvecklades på 90-talet för att separera design från struktur i HTML. Tidigare var all styling inbakad i HTML-koden, vilket blev rörigt och svårt att underhålla. Den första versionen, CSS1, kom 1996, men det var med CSS2 (1998) och särskilt CSS3 (början av 2000-talet) som det verkligen tog fart. CSS3 gjorde det möjligt att använda saker som skuggor, animationer och flexibla layouter utan att behöva bilder eller JavaScript.

3. Förklara vad responsiv design innebär.

Responsiv design innebär att en webbsida anpassar sig automatiskt efter skärmens storlek och enhetstyp, så att den ser bra ut och fungerar bra oavsett om du kollar på den från en dator, surfplatta eller mobil.

Det handlar främst om att använda flexibla layouter, procentbaserade breddmått och CSS-tekniker som media queries för att justera designen beroende på skärmens bredd. Till exempel kan en meny visas som en horisontell list på en stor skärm men som en hamburgarmeny på mobilen.

4. Vad är JavaScript och vad används det till?

JavaScript (JS) är ett programmeringsspråk som används för att göra webbsidor interaktiva. Medan HTML strukturerar innehållet och CSS styr designen, så används JavaScript för att lägga till dynamik – till exempel knappar som reagerar när man klickar, bildspel, animationer och realtidsuppdateringar utan att ladda om sidan.

Javascript används till:

- **Interaktivitet** – exempelvis popup-fönster, knappar som ändrar färg när man hoverar över dem, eller formulär som valideras innan de skickas.
- **Dynamiskt innehåll** – kan hämta och uppdatera data utan att ladda om sidan (t.ex. nyhetsflöden eller kommentarer i realtid).
- **Spel och animationer** – många webbaserade spel och grafiska effekter använder JavaScript.
- **Webbappar** – stora plattformar som Instagram, Twitter och Google Docs använder JavaScript för att fungera smidigt i webbläsaren.

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

ECMA-Script (ES) är den standard som JavaScript är baserat på. Det definierar regler, funktioner och syntax för hur JavaScript ska fungera. Man kan säga att ECMA-Script är "regelboken" och JavaScript är ett av språken som följer den boken.

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

I frontend-utveckling har HTML, CSS och JavaScript olika ansvarsområden som tillsammans skapar en fungerande och snygg webbsida:

- **HTML (Struktur)** – Ansvarar för sidans innehåll och uppbyggnad, t.ex. rubriker, stycken, bilder och länkar. Det är skelettet i webbsidan.
- **CSS (Design & Layout)** – Hanterar färger, typsnitt, placering och responsivitet. Det är det som gör att sidan ser snygg och användarvänlig ut.
- **JavaScript (Interaktivitet & Logik)** – Gör sidan dynamisk genom att hantera saker som klickhändelser, animationer och datahämtning i realtid.

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

DOM (Document Object Model) är en trädstruktur som representerar hela webbsidan i webbläsaren. När en webbsida laddas omvandlas HTML-koden till en DOM-struktur, där varje element (t.ex. `<div>`, `<h1>`, `<p>`) blir en nod i trädet.

Hur DOM används:

- JavaScript kan manipulera DOM för att ändra innehåll, stil och struktur på sidan i realtid.
- Vi kan använda DOM-metoder som `document.getElementById()` eller `document.querySelector()` för att hämta och ändra element.
- Eventlyssnare som `addEventListener()` gör att vi kan reagera på användarens interaktioner (t.ex. klick eller tangenttryck).

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 samling färdiga funktioner och verktyg som underlättar utvecklingen av webbapplikationer. Ramverk ger en strukturerad kodbas och hjälper utvecklare att skriva kod snabbare och mer effektivt.

Att bara använda ren JavaScript (vanilla JS) innebär att du själv måste skriva all logik för att hantera DOM, skapa komponenter och uppdatera UI. Ett ramverk automatiserar många av dessa saker och erbjuder smartare lösningar.

Fördelar med ett ramverk:

- **Struktur & Skalbarhet** – Håller koden organiserad och gör det lättare att bygga stora projekt.
- **Effektivitet** – Du slipper uppfinna hjulet på nytt, eftersom många funktioner är färdiga att använda.
- **Prestandaoptimering** – Ramverk använder smarta tekniker som **virtual DOM** (React) eller **two-way data binding** (Vue, Angular).
- **Komponentbaserad utveckling** – Gör det lättare att återanvända kod och skapa modulära UI-element.

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.

a. React:

- React är ett bibliotek (inte ett fullständigt ramverk) som används för att bygga användargränssnitt.
- Använder en virtuell DOM för att effektivt uppdatera UI.
- Komponentbaserat, vilket gör det enkelt att återanvända kod.
- Stort ekosystem och används av företag som Facebook och Instagram.

b. Vue.js:

- Ett lättviktigt och flexibelt ramverk för att bygga interaktiva webbapplikationer.
- Har en enkel syntax, vilket gör det lätt att lära sig.
- Stöder two-way data binding, vilket innebär att data i UI och i koden hålls synkroniserad automatiskt.
- Bra för både små och stora projekt, populärt bland frilansare och startups.

c. Angular:

- Ett fullständigt ramverk utvecklat av Google, används ofta i stora företag och enterprise-appar.
- Bygger på TypeScript, vilket ger stark typning och bättre struktur.
- Har en brantare inlärningskurva men erbjuder många inbyggda funktioner.
- Använder en kraftfull dependency injection-modell för bättre kodhantering.

Vad skiljer dem åt?

- React är lättviktigt och fokuserar på UI, men kräver externa bibliotek för avancerade funktioner.
- Vue.js är enklast att lära sig och kombinerar det bästa från React och Angular.
- Angular är stort och komplett, perfekt för stora skalbara projekt men svårare att lära sig.

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

Tillgänglighetsanpassning innebär att skapa webbplatser som är användbara för alla, inklusive personer med olika typer av funktionsnedsättningar (t.ex. synnedsättning, hörselskador eller motoriska svårigheter). Det handlar om att göra webbplatsen tillgänglig och användbar för så många som möjligt genom att följa specifika riktlinjer och använda tekniker som underlättar för dessa användare.

Varför det är viktigt:

- **Inkludering** – Alla ska kunna använda webbplatsen, oavsett funktionsnedsättning.
- **Juridiska krav** – Många länder har lagar som kräver att webbplatser ska vara tillgängliga, som **WCAG (Web Content Accessibility Guidelines)**.
- **Breddad målgrupp** – En tillgänglig webbplats når fler användare och kan förbättra användarupplevelsen för alla, inte bara de med särskilda behov.
- **SEO-fördelar** – Tillgänglighetsanpassning kan också förbättra sökmotoroptimering (SEO), eftersom det gör webbplatsen lättare att indexera och använda.

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

Ett webb-API (Application Programming Interface) är ett sätt för olika program eller system att kommunicera med varandra över webben. API:er tillhandahåller en uppsättning regler och protokoll som gör det möjligt för en applikation att skicka och ta emot data från en annan applikation eller tjänst via HTTP (det vill säga genom webben).

Vad ett webb-API används till:

Webb-API:er används för att möjliggöra integration och kommunikation mellan olika system, t.ex.:

- **Hämta data** – Du kan hämta data från en annan tjänst, som väderinformation från en vädertjänst eller användardata från ett socialt nätverk.
- **Skicka data** – API:er gör det möjligt att skicka data till en server, som när en användare fyller i ett formulär på en webbplats och skickar informationen till en databas.
- **Tredjepartsintegrationer** – API:er gör det möjligt att integrera externa tjänster på din webbplats, som betalningslösningar (PayPal, Stripe) eller inloggning med sociala medier (Google, Facebook).
- **Automatisering** – Genom API:er kan system kommunicera automatiskt med varandra utan mänsklig inblandning, till exempel genom att synkronisera information mellan appar.

12. Förklara REST och redogör för dess huvudprinciper.

REST (Representational State Transfer) är en stil för att bygga webb-API:er som gör det möjligt för applikationer att kommunicera med varandra över HTTP. REST används för att skapa skalbara, effektiva och enkla webb-API:er genom att följa vissa arkitekturella principer. Det är en av de vanligaste arkitekturerna för att bygga moderna webb-API:er.

Huvudprinciper för REST:

1. Klient-server arkitektur

- I en REST-arkitektur är klienten (t.ex. en webbläsare eller mobilapp) separerad från servern.
- Servern hanterar logik och lagring, medan klienten fokuserar på presentation och användarinteraktion.
- Denna separation gör det möjligt för båda att utvecklas och skalas oberoende av varandra.

2. Stateless (tillståndslös)

- Varje begäran från en klient till en server ska vara självständig och innehålla all information som behövs för att bearbeta den.
- Servern behåller inget tillstånd om tidigare begärningar. Detta betyder att servern inte lagrar någon session eller användardata mellan begärningarna.
- Till exempel: Om du gör två API-anrop till en server, kommer varje anrop att vara fristående, utan att servern behöver känna till det föregående.

3. Cacheable (cachning)

- Svar från servern kan vara cachade, vilket innebär att de kan lagras på klienten för att snabba upp framtida begärningar.
- Om ett svar är cachebart kan klienten lagra det i en cache och använda det vid behov, utan att göra en ny begäran till servern.
- Detta förbättrar prestanda och minskar belastningen på servern.

4. Uniform Interface (enhetligt gränssnitt)

- REST definierar ett standardiserat sätt att interagera med servern genom att använda vanliga HTTP-metoder (GET, POST, PUT, DELETE, etc.).
- Detta innebär att alla RESTful API:er följer samma grundläggande principer för att hämta, skicka, uppdatera och ta bort data.
- Detta förenklar förståelsen och användningen av API:er eftersom samma metoder används konsekvent.

5. Resurser identifieras via URIs (Uniform Resource Identifiers)

- I REST representeras data som resurser, och varje resurs har en unik identifierare, vanligen en URL (Uniform Resource Locator).
- Till exempel kan en användare i ett API identifieras via en URL som `/users/{id}`.

- Resurserna manipuleras genom de HTTP-metoder som är associerade med dem:
 - GET för att hämta en resurs
 - POST för att skapa en ny resurs
 - PUT för att uppdatera en resurs
 - DELETE för att ta bort en resurs

6. Representation

- När du begär en resurs från en RESTful server får du inte själva resursen direkt, utan en representation av den (ofta i JSON eller XML-format).
- Denna representation innehåller alla de data du behöver för att visa eller manipulera resursen, men själva resursen ligger på servern och kan förändras.