

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

HTML (HyperText Markup Language) är ett språk som används för att skapa och strukturera innehåll på webbsidor. Det definierar strukturen på en webbsida genom att använda taggar (element) som anger olika delar av innehållet, såsom rubriker, stycken, bilder, länkar och formulär.

Historik:

- **1991:** HTML skapades av Tim Berners-Lee. Den första versionen av HTML hade bara några enkla taggar för att strukturera text och länkar.
- **1995:** HTML 2.0 lanserades som en standardversion av språket.
- **1997:** HTML 3.2 släpptes och introducerade tabeller, skript (JS) och CSS-stöd.
- **1999:** HTML 4.0 med stöd för multimedia och internationella tecken.
- **2000-talet:** XHTML – en striktare version av HTML baserad på XML. Hårdare syntax.
- **2014:** HTML 5 som är den moderna versionen med stöd för multimedia (film/bild/ljud) och interaktiv/responsiv design.

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

CSS (Cascading Style Sheets) är ett språk som används för att definiera och styra layout och utseende på webbsidor. Med CSS kan man ändra färger, typsnitt, marginaler, positionering och andra visuella aspekter av en webbsida utan att påverka själva innehållet som är strukturerat i HTML.

Historik:

- **1996:** CSS 1 lanserades som en första version av stilarksspråket. Den möjliggjorde grundläggande styling för webbsidor, som t.ex. färger och typsnitt.
- **1998:** CSS 2 släpptes, och introducerade fler funktioner som positionering (t.ex. absolute, relative), media queries (för att anpassa layout för olika enheter) och förbättrad stöd för layout med hjälp av tabeller och fontkontroll.
- **2000-talet:** Arbetet med CSS 3 började, och språket utvecklades i moduler. CSS 3 gav en rad nya funktioner, som exempelvis runda hörn, skuggor, övergångar och animeringar.
- **2010-talet till nu:** CSS 3 och dess nya funktioner, som Flexbox och Grid Layout, blev en standard för responsiv design, och gav utvecklare möjligheten att skapa mer komplexa och användarvänliga webbupplevelser.

Förklara vad responsiv design innebär.

Responsiv design är en metod för webbutveckling där en webbsidas layout och innehåll automatiskt anpassar sig till olika skärmstorlekar och enheter, såsom datorer, surfplattor och mobiltelefoner. Målet är att skapa en användarvänlig och optimal upplevelse oavsett vilken enhet som används.

Grundprinciper för responsiv design:

- **Media Queries** – CSS-teknik som gör det möjligt att ändra layout och styling beroende på skärmens bredd, höjd och upplösning.
- **Flexibla layouter** – Användning av procentbaserade bredder istället för fasta pixelmått, så att elementen anpassar sig till olika skärmstorlekar.
- **Flexbox & CSS Grid** – Moderna layouttekniker som gör det enklare att skapa dynamiska och anpassningsbara webbdesigner.
- **Anpassade bilder** – Bilder och andra media anpassas efter skärmstorlek för att minska laddningstider och förbättra prestanda.
- **Mobilvänlig navigering** – Exempelvis hamburgermenyer och touchvänliga knappar för enklare användning på mindre skärmar.

Vad är JavaScript och vad används det till?

JavaScript är ett programmeringsspråk som används främst för att skapa dynamiskt och interaktivt innehåll på webbsidor. Det är ett av de tre huvudsakliga språken inom webbutveckling, tillsammans med HTML (som definierar sidans struktur) och CSS (som hanterar sidans design och layout).

I korthet – vad gör det?

- **Interaktivitet** – Hanterar klick, menyer, formulär och animationer.
- **Manipulera HTML & CSS** – Ändra innehåll och stil dynamiskt.
- **Hämta data** – Använda AJAX och API:er för att hämta och skicka data.
- **Spel & Grafiska effekter** – Skapa 2D/3D-grafik med t.ex. Canvas och WebGL.
- **Validera formulär** – Kontrollera användarinput innan den skickas till servern.
- **Bygga webbsidor & appar** – Används i ramverk som React, Angular och Vue.

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

ECMAScript är en teknisk standard som beskriver regler och funktioner för skriptspråk. Den skapades för att göra det möjligt att skapa kompatibla skript som fungerar på olika plattformar. JavaScript är ett av de mest kända språken som följer ECMAScript-standarderna och används främst för att bygga interaktiva webbplatser.

ECMAScript beskriver grundläggande saker som syntax (hur koden skrivs), datatyper, funktioner och hur språket ska bete sig. JavaScript implementerar dessa regler, men har också egna funktioner för att hantera saker som webbsidor, användarinteraktioner och nätverkskommunikation. Andra språk, som JScript och ActionScript, följer också ECMAScript, men JavaScript är det mest använda språket baserat på denna standard.

Så, i korthet:

- ECMAScript är standarden som definierar hur ett skriptspråk ska fungera.
- JavaScript är ett skriptspråk som följer denna standard och används för att skapa dynamiska webbsidor.

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

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

- **HTML (HyperText Markup Language) – Struktur**
HTML är skelettet på en webbsida. Det används för att skapa sidans grundläggande struktur med element som rubriker, paragrafer, bilder och knappar.
- **CSS (Cascading Style Sheets) – Design och layout**
CSS ger webbsidan stil och utseende. Det används för att ändra färger, typsnitt, placering och anpassa layouten för olika skärmstorlekar.
- **JavaScript – Interaktivitet och funktionalitet**
JavaScript gör webbsidan dynamisk genom att hantera händelser, uppdatera innehåll utan omladdning och kommunicera med servrar.

Tillsammans skapar dessa tre teknologier en modern, responsiv och interaktiv webbupplevelse.

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

DOM (Document Object Model) är en modell som representerar en webbsidas struktur som ett träd av element. Det gör det möjligt att ändra innehåll, layout och funktionalitet med JavaScript.

När vi skapar en hemsida används DOM för att:

- **Hämta och ändra element** (t.ex. uppdatera text eller färg).
- **Lägga till eller ta bort element** (som nya knappar eller bilder).
- **Reagera på händelser** (som klick eller tangenttryckningar).
- **Styra sidans utseende** genom att ändra CSS-egenskaper.

DOM gör det alltså möjligt att skapa interaktiva och dynamiska webbsidor utan att ladda om sidan.

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 uppsättning fördefinierade verktyg och strukturer som gör det lättare att bygga webbapplikationer. Det innehåller ofta funktioner för hantering av användargränssnitt, data och interaktioner utan att behöva skriva all kod från grunden.

Fördelar med att använda ett ramverk jämfört med ren JavaScript:

- **Snabbare utveckling:** Ramverket erbjuder färdiga lösningar på vanliga problem.
- **Bättre struktur:** Det hjälper till att organisera koden på ett effektivt sätt.
- **Skalbarhet:** Ramverk gör det lättare att bygga större och mer komplexa applikationer.
- **Underhåll:** Ramverk ger bättre stöd för långsiktig underhåll och samarbete i team.

Exempel på JavaScript-ramverk är React, Angular och Vue.

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.

Tre vanliga JavaScript-ramverk:

1. **React:** Ett bibliotek utvecklat av Facebook för att bygga användargränssnitt. React fokuserar på komponentbaserad utveckling och används ofta för att skapa dynamiska webbapplikationer med hög prestanda.
2. **Angular:** Ett ramverk utvecklat av Google som är fullständigt och omfattande. Angular använder TypeScript och erbjuder ett komplett verktyg för att bygga stora, skalbara webbapplikationer med funktioner som tvåvägsbindning av data och beroendeinjektion.
3. **Vue.js:** Ett progressivt ramverk för att bygga användargränssnitt. Vue är lätt att komma igång med och erbjuder flexibilitet för att integreras i både små och stora projekt. Det är känt för sin enkelhet och sin reaktiva datamodell.

Skillnader:

- **React** är ett bibliotek snarare än ett fullständigt ramverk och fokuserar främst på UI-komponenter. Det är mer flexibelt men kräver ofta att andra verktyg används för att bygga en komplett applikation.
- **Angular** är ett omfattande ramverk som erbjuder en hel lösning för att bygga applikationer, inklusive routing, formulärhantering och mer. Det kräver mer inläring och är bättre för stora applikationer.
- **Vue.js** erbjuder en balans mellan enkelhet och funktionalitet. Det är lättare att komma igång med än Angular och mer strukturerat än React, vilket gör det populärt för både små och medelstora projekt.

Vad är tillgänglighetsanpassning av webbplatser och varför är det viktigt?

Tillgänglighetsanpassning av webbplatser innebär att göra dem användbara för alla, inklusive personer med olika funktionsnedsättningar (t.ex. syn-, hörsel- eller motoriska nedsättningar). Detta kan inkludera att använda tydlig text, alternativtext för bilder, rätt kontrast och stöd för skärmläsare.

Det är viktigt för att säkerställa att alla kan få tillgång till information och tjänster på webben, oavsett funktionsförmåga. Dessutom är tillgänglighetsanpassning ofta ett lagkrav och kan förbättra användarupplevelsen för alla.

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 program och system att kommunicera med varandra över webben. Det fungerar som en mellanhand

där en applikation kan skicka förfrågningar (t.ex. begära data) och ta emot svar (t.ex. information i JSON-format) från en annan tjänst eller server.

Webb-API:er används för att hämta eller skicka data mellan olika system, som exempelvis att få väderdata från en vädertjänst eller uppdatera användarprofiler på sociala medier. De underlättar integration och automatisering mellan olika applikationer och plattformar.

Förklara REST och redogör för dess huvudprinciper.

REST (Representational State Transfer) är en arkitekturstil för att bygga webb-API:er där klienter och servrar kommunicerar över HTTP. REST används för att skapa lättanvända, skalbara och flexibla API:er.

Huvudprinciper för REST:

1. **Stateless:** Varje förfrågan från en klient till servern måste innehålla all nödvändig information (ingen session eller lagrad status på servern).
2. **Client-Server:** Klienten och servern är separata och interagerar via tydliga gränssnitt.
3. **Cacheable:** Svar från servern ska kunna cachas för att förbättra prestanda.
4. **Uniform Interface:** Alla resurser (data) ska ha en tydlig och standardiserad struktur, ofta via URL:er och HTTP-metoder som GET, POST, PUT och DELETE.
5. **Layered System:** Arkitekturen kan bestå av flera lager (t.ex. mellanlager för caching eller säkerhet), där varje lager hanterar sin specifika funktion.

REST är populärt för sin enkelhet och flexibilitet i att bygga webbtjänster.