

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

HTML står för HyperText Markup Language och används för att bygga upp strukturen på en webbsida. Det är som skelettet för en hemsida där man lägger till rubriker, text, bilder och länkar. HTML skapades av Tim Berners-Lee 1991 och har utvecklats mycket sedan dess. Den senaste versionen, HTML5, är väldigt kraftfull och används idag för att stödja multimedia och moderna webbapplikationer.

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

CSS betyder Cascading Style Sheets och används för att styla en webbsida, till exempel för att ändra färger, typsnitt och layout. Det gör att en hemsida ser snygg och professionell ut. CSS kom 1996 för att separera design från innehåll, och idag använder vi CSS3 som har många nya funktioner som animationer, flexbox och grid-system för avancerad layout.

3. Förklara vad responsiv design innebär.

Responsiv design handlar om att en hemsida ska fungera och se bra ut på alla typer av enheter, som mobiler, surfplattor och datorer. Man använder CSS-tekniker som media queries för att anpassa layouten beroende på skärmstorlek. Detta gör att användarupplevelsen blir optimal oavsett vilken enhet eller skärmstorlek som används.

4. Vad är JavaScript och vad används det till?

JavaScript är ett programmeringsspråk som gör webbsidor interaktiva. Det används för att skapa saker som knappar som reagerar när man klickar, formulär som kontrollerar om allt är rätt ifyllt, eller för att hämta data från internet. Det körs direkt i webbläsaren och är en av de viktigaste teknikerna för att bygga moderna webbapplikationer.

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

ECMAScript är en standard som JavaScript följer. Det är som en regelbok som ser till att JavaScript fungerar på samma sätt i alla webbläsare. JavaScript är alltså en version av ECMAScript. Standardiseringen gör det möjligt att utveckla nya funktioner som fungerar konsekvent över olika miljöer och plattformar.

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

HTML: Bygger upp strukturen på sidan, som rubriker, text och bilder.

CSS: Gör sidan snygg genom att styra färger, typsnitt och layout.

JavaScript: Läger till interaktivitet, som att saker händer när man klickar eller scollar.

Tillsammans skapar de en komplett frontend-upplevelse för användaren.

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

DOM står för Document Object Model och är en slags karta över allt innehåll på en webbsida. När vi använder JavaScript kan vi ändra DOM för att uppdatera sidan, till exempel lägga till nya element eller ändra text utan att ladda om sidan. Det gör att vi kan skapa dynamiska och interaktiva webbsidor som reagerar på användarens handlingar.

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 som en verktygslåda som gör det enklare att bygga webbapplikationer. Det innehåller färdiga funktioner och strukturer som sparar tid och gör koden mer organiserad. Med ett ramverk slipper man skriva allt från grunden och kan fokusera på att bygga funktionalitet istället för att lösa grundläggande problem.

9. Lista tre vanligt förekommande JavaScript-ramverk.

React: Ett bibliotek från Meta (Facebook) som används för att bygga användargränssnitt med återanvändbara komponenter. Det är populärt för att skapa snabba och skalbara applikationer.

Angular: Ett ramverk från Google som används för att bygga stora och komplexa webbapplikationer. Det erbjuder många inbyggda funktioner som routing och formulärhantering.

Vue.js: Ett lättanvänt ramverk som passar både små och stora projekt. Det är känt för sin enkelhet och flexibilitet, vilket gör det lätt att komma igång med.

Skillnader:

React är ett bibliotek och fokuserar på gränssnitt, medan Angular är ett komplett ramverk med fler funktioner.

Vue.js är enklare att lära sig och mer flexibelt än både React och Angular, vilket gör det populärt bland nybörjare och mindre team.

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

Tillgänglighetsanpassning innebär att göra webbplatser användbara för alla, även personer med funktionsnedsättningar. Det kan handla om att använda tydliga färger, lägga till textbeskrivningar för bilder och göra sidan kompatibel med skärmläsare. Det är viktigt för att alla ska kunna använda webben och för att följa lagar som WCAG (Web Content Accessibility Guidelines).

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

Ett webb-API är ett sätt för olika program att prata med varandra över internet. Det används för att hämta eller skicka data, till exempel för att visa väderdata eller skicka meddelanden. API:er gör det möjligt att integrera externa tjänster i en applikation, vilket sparar tid och resurser.

12. Förklara REST och redogör för dess huvudprinciper.

REST står för Representational State Transfer och är ett sätt att designa webb-API:er. Huvudprinciperna är:

Stateless: Varje förfrågan är oberoende och innehåller all information som behövs.

Resurser: Data representeras som resurser med unika URL:er.

HTTP-metoder: Man använder metoder som GET (hämta data), POST (skicka data), PUT (uppdatera data) och DELETE (ta bort data).

Representationer: Resurser kan skickas i olika format, som JSON eller XML. REST gör det enkelt att bygga skalbara och lättförståeliga API:er.

1. **VG-fråga**

Lista fem (5) vanligt förekommande missar inom tillgänglighet på webbplatser.

1. Brist på textalternativ för bilder.

1.1. Vad det innebär: Bilder saknar beskrivande text i alt-attributet, vilket gör att skärmläsare inte kan förmedla bildens innehåll till användare med synnedsättning.

1.2. Vad det beror på: Utvecklare eller innehållsskapare glömmer ofta att lägga till alt-texter eller använder generiska texter som bild1.

1.3. Tekniska utmaningar: Det kan vara svårt att automatiskt generera meningsfulla alt-texter. Utvecklare måste förstå bildens kontext för att skriva en relevant beskrivning. Tidskrävande att lägga till alt-texter på en stor webbplats med många bilder.

- 1.4. Att tänka på: Använda verktyg som påminner en om att lägga till alt-texter.
2. Otillräcklig kontrast mellan text och bakgrund.
 - 2.1. Vad det innebär: När texten har för låg kontrast mot bakgrunden leder det till att det blir svårt att läsa för personer med synnedsättning eller färgblindhet.
 - 2.2. Vad det beror på: Okunskap om kontrastkrav enligt WCAG eller designprioriteringar som sätter estetik före läsbarhet.
 - 2.3. Tekniska utmaningar: Att balansera designens visuella estetik med tillgänglighetskrav. Dynamiska teman eller användargenererat innehåll kan skapa oförutsägbara kontrastproblem. Svårt att använda verktyg för att mäta kontrast i designprocessen.
 - 2.4. Att tänka på: Använda kontrastkontrollverktyg som WCAG contrast checker och testa olika färgkombinationer, implementera CSS-variabler för att enkelt justera färger vid behov.
3. Otillräckligt med navigeringsmöjligheter för tangentbordsanvändare
 - 3.1. Vad det innebär: Webbplatsen kan inte navigeras med tangentbordet, vilket påverkar personer som inte kan använda en datormus. Saknas fokusmarkeringar eller logisk tabb-ordning.
 - 3.2. Vad det beror på: Utvecklare testar inte webbplatsen med enbart tangentbord.
 - 3.3. Tekniska utmaningar: Anpassning av tabb-ordning kan vara problematiskt/komplext på sidor med dynamiskt innehåll. Fokusmarkeringar kan ibland tas bort av designskäl, vilket gör det svårt att se vart fokus är.
 - 3.4. Att tänka på: Testa webbplatsen med tangentbordet och använd semantisk HTML-element. Lägg till tydliga fokusmarkeringar med CSS och undvik att maipulera tabb-ordning i onödan.
4. Bristande stöd för skärmläsare
 - 4.1. Vad det innebär: Webbplatsen är inte optimerad för skärmläsare, vilket gör att personer med synnedsättning inte kan navigera eller förstå innehållet.
 - 4.2. Vad det beror på: Utvecklare saknar kunskap om hur skärmläsare fungerar eller använder felaktiga tekniker som bryter tillgängligheten.
 - 4.3. Tekniska utmaningar: Att använda ARIA-attribut korrekt kräver en djup förståelse för deras funktion. Felaktig användning gör mer skada än nytta. Testning med skärmläsare kräver tid och tillgång till rätt verktyg.

- 4.4. Att tänka på: Följ riktlinjer för ARIA och använd semantisk HTML som standard. Testa webbplatsen med olika skärmläsare och säkerställ att dynamiska uppdateringar meddelas korrekt.

5. Formulär utan tydliga etiketter och felmeddelanden

- 5.1. Vad det innebär. formulärfält saknar tydliga etiketter eller ger inte användaren feedback vid fel, vilket gör det svårt för personer med kognitiva eller synrelaterade funktionsnedsättningar att fylla i formulär
- 5.2. Vad det beror på: Utvecklare använder visuella ledtrådar (placeholder-text) istället för ritkiga etiketter, eller glömmer implementera felhantering.
- 5.3. Tekniska utmaningar: Placeholder-text försvinner när användaren börjar skriva, vilket gör det svårt att veta vad fältet representerar. Dynamiska felmeddelanden kan vara svåra att implementera på ett sätt som fungerar för både skärmläsare och visuella användare. Att säkerställa att alla fält har unika och tydliga etiketter kan vara tidskrävande i stora formulär.
- 5.4. Att tänka på: Använd alltid <label>-element kopplade till formulärfält med for-attribut. Implementera tydliga och tillgängliga felmeddelande som fungerar för alla.