# Flashcards

**HTML**

1. **Hva er HTML, og hvordan fungerer det i nettleseren?**
2. **Hva står de forskjellige HTML5-doktype-erklæringene for, og hvorfor er de viktige?**
3. **Forklar betydningen av semantisk HTML og gi eksempler på semantiske elementer.**
4. **Hvordan kan du lage en intern lenke i HTML?**
5. **Hva er forskjellen mellom <div> og <section>?**
6. **Hvordan kan du bruke <data>-taggen, og hva er dens formål?**
7. **Hva er forskjellen mellom <strong> og <b>?**
8. **Hvordan kan du implementere tilgjengelighet i HTML ved hjelp av ARIA-attributter?**
9. **Hva er en HTML5-video, og hvordan setter du opp en?**
10. **Forklar hvordan du bruker <canvas>-elementet for å tegne grafikk.**

**HTML Svar**

**Svar:**

1. **HTML (HyperText Markup Language) er standard markup-språket for å lage nettsider. Det fungerer ved å strukturere innholdet med tags som nettleseren tolker og viser.**
2. **Doctype-erklæringen informerer nettleseren om hvilken HTML-versjon som brukes. HTML5s doctype, <!DOCTYPE html>, er enkel og sikrer korrekt rendering.**
3. **Semantisk HTML refererer til bruk av elementer som gir mening i forhold til innholdet, som <article> og <header>.**
4. **Bruk <a>-taggen med href som peker til en ID.**
5. **<div> er en generisk beholder, mens <section> representerer en selvstendig seksjon med relaterte innholdstyper.**
6. **<data> brukes til å representere data i et maskinlesbart format.**
7. **<strong> indikerer betydning, mens <b> kun er for visuell stil.**
8. **ARIA-attributter gir mer kontekst til skjermlesere, som role og aria-label.**
9. **En HTML5-video settes opp med <video>-taggen.**
10. **<canvas> er en bitmap-beholder for dynamisk grafikk, manipulert med JavaScript.**

**CSS**

1. **Hva er CSS, og hvordan skiller det seg fra HTML?**
2. **Forklar forskjellen mellom klasse- og ID-selektorer i CSS.**
3. **|Hordan kan du bruke CSS for å tilpasse stilen til et bestemt nettleservindu?**
4. **Hva gjør !important i CSS, og hvordan påvirker det spesifisitet?**
5. **Forklar hva en CSS-variabel er, og gi et eksempel på hvordan den brukes.**
6. **Hvordan kan du implementere CSS-overganger, og hva er bruken av dem?**
7. **Hva er forskjellen mellom display: none; og visibility: hidden;?**
8. **Gi et eksempel på hvordan du bruker CSS Flexbox for å lage en responsiv navigasjonsmeny.**
9. **Hva er CSS-in-JS, og hva er fordelene og ulempene ved denne tilnærmingen?**
10. **Hvordan kan du bruke pseudoklasser i CSS? Gi eksempler.**

**Svar:**

1. **CSS (Cascading Style Sheets) brukes til å style innholdet i HTML.**
2. **Klasse-selektorer (.) brukes for flere elementer, mens ID-selektorer (#) er unike.**
3. **Bruk media queries for å justere stil basert på skjermstørrelse.**
4. **!important gir høyere prioritet til regler og kan overskrive spesifisitet.**
5. **CSS-variabler lagrer verdier for gjenbruk, f.eks. --primær-farge.**
6. **Overganger gir jevn endring fra en stil til en annen med transition.**
7. **display: none; fjerner elementet, mens visibility: hidden; skjuler det men beholder plassen.**
8. **Flexbox kan brukes for å justere og distribuere plass mellom elementer.**
9. **CSS-in-JS lar deg skrive CSS i JavaScript-filer, med fordeler som modulær struktur.**
10. **Pseudoklasser stiliserer elementer basert på tilstand, som :hover.**

**Avansert CSS**

1. **Forklar hva CSS Grid er, og gi et eksempel på hvordan du setter opp et grid.**
2. **Hvordan kan du lage en animasjon i CSS, og hva er de viktigste egenskapene?**
3. **Hva er forskjellen mellom absolute, relative, fixed, og sticky posisjonering?**
4. **Hvordan kan du bruke media queries for å lage en responsiv design? Gi et eksempel.**
5. **Forklar hva en "clearfix" er, og hvordan det brukes i CSS.**
6. **Hvordan kan du implementere et rutenettlayout med CSS Grid?**
7. **Hva er clip-path, og hvordan kan det brukes til å lage uvanlige former?**
8. **Hvordan bruker du @keyframes for å definere animasjoner i CSS?**
9. **Hva er forskjellen mellom margin og padding, og når bruker du hver?**
10. **Forklar hvordan CSS Modules fungerer og deres fordeler.**

**Svar: AVANSERT CSS**

1. **CSS Grid er et layoutsystem for komplekse rutenett.**
2. **Animasjoner defineres med @keyframes og animation-egenskapen.**
3. **relative posisjonerer i forhold til sitt normale sted, absolute i forhold til nærmeste posisjonerte element, fixed i forhold til vinduet, og sticky veksler mellom relative og fixed.**
4. **Media queries endrer stil basert på enhetens egenskaper.**
5. **Clearfix brukes for å rydde opp flytende elementer.**
6. **Rutenettlayout settes opp med grid-template-columns og grid-template-rows.**
7. **clip-path skjuler deler av et element for å lage former.**
8. **@keyframes definerer animasjonssekvenser.**
9. **margin er rommet utenfor et element, mens padding er rommet inni.**
10. **CSS Modules gir scoped stilregler for å unngå navnekollisjoner.**

**SASS**

1. **Hva er SASS, og hvordan forbedrer det CSS-utvikling?**
2. **Forklar forskjellen mellom SASS og SCSS.**
3. **Hvordan kan du bruke mixins i SASS, og hva er fordelene ved å bruke dem?**
4. **Gi et eksempel på nesting i SASS og hvordan det forbedrer koden.**
5. **Hva er SASS-funksjoner, og hvordan kan de brukes til å manipulere verdier?**
6. **Hvordan kan du organisere SASS-filer i et prosjekt for bedre vedlikehold?**
7. **Forklar hva en SASS-loop er, og gi et eksempel.**
8. **Hvordan kan du bruke SASS til å lage responsive designklasser?**
9. **Hva er @import i SASS, og hvordan hjelper det med modulær CSS?**
10. **Forklar hvordan du kan bruke SASS til å implementere temaer i designet.**

**Svar: SASS**

1. **SASS (Syntactically Awesome Style Sheets) er en CSS-preprosessor som forbedrer organiseringen av stilene.**
2. **SASS bruker en eldre syntaks, mens SCSS er mer lik CSS-syntaksen.**
3. **Mixins lar deg gjenbruke stilregler, noe som reduserer duplisering.**
4. **Nesting organiserer CSS-regler hierarkisk.**
5. **SASS-funksjoner lar deg manipulere verdier, som å utføre beregninger.**
6. **Organiser SASS-filer i moduler for bedre vedlikehold.**
7. **Looper lar deg gjenta stilregler for flere elementer.**
8. **Bruk medieforespørsel i SASS for å lage responsive klasser.**
9. **@import inkluderer andre SASS-filer for modulær organisering.**
10. **Bruk variabler for farger og skrifttyper for å lage ulike temaer.**

**Responsivt design**

1. **Hva er de viktigste prinsippene for responsiv design?**
2. **Hvordan kan du bruke fleksible bilder for å forbedre responsivitet?**
3. **Forklar hvordan "mobile-first" designprinsippet fungerer.**
4. **Hvordan kan du bruke rem og em enheter for responsiv design?**
5. **Hva er forskjellen mellom max-width og min-width i media queries?**
6. **Hvordan kan du bruke CSS Grid for å lage et responsivt rutenett?**
7. **Hva er "viewport" og hvordan påvirker det designet?**
8. **Hvordan kan du implementere responsiv typografi?**
9. **Forklar hvordan du kan bruke flex for responsiv layout.**
10. **Hvordan kan du teste responsiviteten til en nettside?**

**Svar: Responsiv design**

1. **Viktige prinsipper inkluderer fleksible rutenett, fleksible bilder, og medieforespørsel.**
2. **Bruk max-width: 100%; for bilder for å tilpasse dem.**
3. **"Mobile-first" design begynner med design for små skjermer og bygger opp.**
4. **rem og em enheter gjør det enklere å lage responsiv typografi.**
5. **max-width begrenser bredden, mens min-width setter en minimumsgrense.**
6. **CSS Grid lar deg definere rader og kolonner for responsiv layout.**
7. **"Viewport" refererer til den synlige delen av en nettside på en enhet.**
8. **Responsiv typografi kan implementeres med relative enheter og medieforespørsel.**
9. **Flexbox lar deg justere elementer responsivt basert på tilgjengelig plass.**
10. **Du kan teste responsivitet ved å bruke utviklerverktøy i nettleseren.**

**HTML-skjema**

1. **Hvordan kan du lage et HTML-skjema med forskjellige input-felt?**
2. **Forklar bruken av autocomplete-attributtet i skjemaer.**
3. **Hva er fieldset og legend i HTML-skjemaer, og hvordan brukes de?**
4. **Hvordan kan du validere skjemaer med HTML5-attributter?**
5. **Forklar forskjellen mellom GET og POST metoder i skjemaer.**
6. **Hva er en submit-knapp, og hvordan kan den brukes?**
7. **Hvordan kan du implementere tilpassede valideringsmeldinger i skjemaer?**
8. **Hva er input type="range" og hvordan brukes det?**
9. **Forklar hvordan du kan bruke radio-knapper og avmerkingsbokser i skjemaer.**
10. **Hvordan kan du implementere et flertrinns skjema?**

**Svar: HTML-SKJEMA**

1. **Et HTML-skjema lages med <form>-taggen og ulike <input>-felt.**
2. **autocomplete-attributtet hjelper nettleseren med å huske brukerens input.**
3. **<fieldset> grupperer relaterte elementer, mens <legend> gir en tittel.**
4. **HTML5 tilbyr attributter som required og pattern for validering.**
5. **GET sender data i URL, mens POST sender data i forespørselens kropp.**
6. **En submit-knapp sender skjema-data til serveren.**
7. **Du kan bruke JavaScript for å gi tilpassede valideringsmeldinger.**
8. **input type="range" lar brukeren velge en verdi fra et intervall.**
9. **Radio-knapper lar kun én verdi velges, mens avmerkingsbokser lar flere velges.**
10. **Flertrinns skjemaer kan implementeres med JavaScript for å navigere mellom steg**

**HTML-tabeller**

1. **Hvordan kan du lage en kompleks tabell med rowspan og colspan?**
2. **Hva er forskjellen mellom <thead>, <tbody>, og <tfoot>?**
3. **Hvordan kan du stilisere en tabell med CSS for bedre lesbarhet?**
4. **Forklar hvordan du bruker ARIA-attributter i tabeller for tilgjengelighet.**
5. **Hva er <caption>-elementet i tabeller, og hvordan brukes det?**
6. **Hvordan kan du implementere sortering i tabeller med JavaScript?**
7. **Forklar hvordan du håndterer responsivitet i tabeller.**
8. **Hvordan kan du bruke CSS for å lage en "frosset" header i en tabell?**
9. **Hva er en datavisualisering, og hvordan kan tabeller brukes i denne sammenhengen?**
10. **Hvordan kan du bruke <table> til å lage en layout? Hva er ulempene?**

**Svar: - HTML TABELLER**

1. **Bruk rowspan og colspan attributter i <td> for å kombinere celler.**
2. **<thead> er for overskrifter, <tbody> for innhold, og <tfoot> for bunnlinjer.**
3. **Stiliser tabeller med CSS for bedre lesbarhet, som å legge til skygger og marger.**
4. **ARIA-attributter gir kontekst for skjermlesere.**
5. **<caption> gir en beskrivelse av tabellen.**
6. **Sortering i tabeller kan implementeres med JavaScript for å endre rekkefølge.**
7. **Responsivitet kan håndteres med media queries og CSS.**
8. **"Frosset" header kan oppnås med CSS og JavaScript for å holde den synlig.**
9. **Datavisualiseringer bruker tabeller for å presentere data på en strukturert måte.**
10. **<table> kan brukes for layout, men det kan føre til dårlig tilgjengelighet og fleksibilitet.**

**Universell Utforming**

1. **Hva er universell utforming, og hvorfor er det viktig i webdesign?**
2. **Hvordan kan du bruke ARIA for å forbedre tilgjengeligheten på en nettside?**
3. **Forklar hvordan fargekontraster påvirker tilgjengeligheten.**
4. **Hva er "alt"-attributtet, og hvorfor er det viktig for bilder?**
5. **Hvordan kan du lage en tilgjengelig navigasjonsmeny?**
6. **Hva er skjermlesere, og hvordan påvirker de webdesign?**
7. **Forklar prinsippene for tilbakemeldinger og feilbeskjeder i skjemaer.**
8. **Hvordan kan du implementere tastaturnavigasjon på en nettside?**
9. **Hva er tilpasning av innhold, og hvordan kan det implementeres?**
10. **Hvordan kan du teste tilgjengeligheten til en nettside?**

**Svar: UNIVERSELL UTFORMING**

1. **Universell utforming gjør nettsteder tilgjengelige for alle brukere, inkludert de med funksjonsnedsettelser.**
2. **ARIA-attributter gir mer kontekst for skjermlesere.**
3. **God fargekontrast forbedrer lesbarhet og tilgjengelighet.**
4. **"Alt"-attributtet gir tekstbeskrivelser for bilder, viktig for skjermlesere.**
5. **Tilgjengelig navigasjonsmeny bruker klar struktur og ARIA-attributter.**
6. **Skjermlesere konverterer tekst til tale, så design må være strukturelt korrekt.**
7. **Tydelige tilbakemeldinger og feilbeskjeder hjelper brukere forstå hva de gjør galt.**
8. **Tastaturnavigasjon kan implementeres med tabulatornavigering og ARIA.**
9. **Tilpasning av innhold kan implementeres med brukerpreferanser og ARIA.**
10. **Tilgjengeligheten kan testes med verktøy som WAVE og skjermlesere.**

**Søkemotoroptimalisering (SEO)**

1. **Hva er SEO, og hvordan fungerer det?**
2. **Forklar hvordan metatags påvirker SEO.**
3. **Hva er viktige SEO-faktorer for on-page optimalisering?**
4. **Hvordan kan du bruke nøkkelord effektivt i innhold?**
5. **Hva er forskjellen mellom organisk og betalt søk?**
6. **Hvordan kan du bruke Google Analytics for å overvåke SEO-resultater?**
7. **Forklar hva backlinks er, og hvorfor de er viktige for SEO.**
8. **Hva er "robots.txt", og hvordan påvirker det søkemotorer?**
9. **Hvordan kan du optimalisere bilder for SEO?**
10. **Hva er "canonical tags", og hvordan brukes de?**

**Svar: SEO**

1. **SEO (Search Engine Optimization) er prosessen med å forbedre en nettsides synlighet i søkemotorer.**
2. **Metatags gir informasjon om siden til søkemotorer, som beskrivelse og nøkkelord.**
3. **Viktige faktorer inkluderer innholdsstruktur, bruk av nøkkelord, og intern lenking.**
4. **Nøkkelord bør brukes naturlig i titler, overskrifter og innhold.**
5. **Organisk søk er gratis, mens betalt søk krever betaling for annonsering.**
6. **Google Analytics gir innsikt i trafikk og brukeratferd, nyttig for SEO.**
7. **Backlinks er lenker fra andre nettsteder, og de øker autoriteten til din side.**
8. **"robots.txt" styrer hvilke sider søkemotorer kan indeksere.**
9. **Optimaliser bilder ved å bruke beskrivende filnavn og alt-attributter.**
10. **Canonical tags forhindrer duplisert innhold ved å indikere den originale kilden.**

**Flashcards for Webserver**

1. **Hva er forskjellen mellom HTTP og HTTPS?**
2. **Hvordan fungerer en webserver, og hva er dens hovedoppgaver?**
3. **Forklar hva en reverse proxy er, og hvordan det brukes.**

**Svar:**

1. **HTTP (HyperText Transfer Protocol) er en protokoll for overføring av data over internett uten sikkerhet. HTTPS (HTTP Secure) er en sikker versjon av HTTP, som bruker SSL/TLS for å kryptere dataoverføringen, noe som beskytter mot avlytting og datamanipulasjon.**
2. **En webserver er en programvare eller maskinvare som lagrer, håndterer og leverer nettsider til brukere via internett. Hovedoppgavene inkluderer å motta HTTP-forespørsel fra klienter, prosessere forespørslene, hente nødvendige ressurser, og sende tilbake HTTP-respons med innholdet.**
3. **En reverse proxy er en server som står mellom klientene og en eller flere backend-servere. Den håndterer forespørslene fra klientene, distribuerer dem til de riktige serverne, og returnerer svaret til klienten. Reverse proxy brukes ofte for lastbalansering, caching, og forbedret sikkerhet.**