HTML: Introduksjon til Web

Skrevet av: Oversatt fra Code Club UK (//codeclub.org.uk)

Oversatt av: Arve Seljebu

Kurs: Web
Tema: Tekstbasert, Nettside
Fag: Programmering, Teknologi
Klassetrinn: 8.-10. klasse, Videregående skole

Introduksjon

Målet med dette oppgavesettet er å gi deg en enkel introduksjon til internett og hvordan nettsider og HTML fungerer. I denne oppgaven er det viktig at du tar deg god tid og leser oppgavene nøye.

Har du noen gang lurt på hvordan internett virker? Klart du har det! I dag kommer du til å lære hvordan man lager nettsider slik at du også kan hjelpe til med å bygge det. Nettsider blir skrevet ved å bruke **HTML**, som er en forkortelse for **HyperText Markup Language** på engelsk. Du kommer til å finne ut mer om dette ettersom du bygger siden din.

Filene som du bes åpne i denne oppgaven finner du i internett.zip (internett.zip). Last ned filen og pakk den ut før du starter. Om du ikke får pakket ut filen, kan du laste ned programmet 7zip (http://www.7-zip.org/) som pakker ut zip-filer.

Steg 1: Hva er nettsider?

Aktiviteter

- [ ] Åpne en **teksteditor**, for eksempel teksteditoren Atom (http://atom.io), Brackets (http://brackets.io/), NotePad eller

```
[NotePad++](https://notepad-plus-plus.org/)
```

- [ ] Lag et nytt dokument

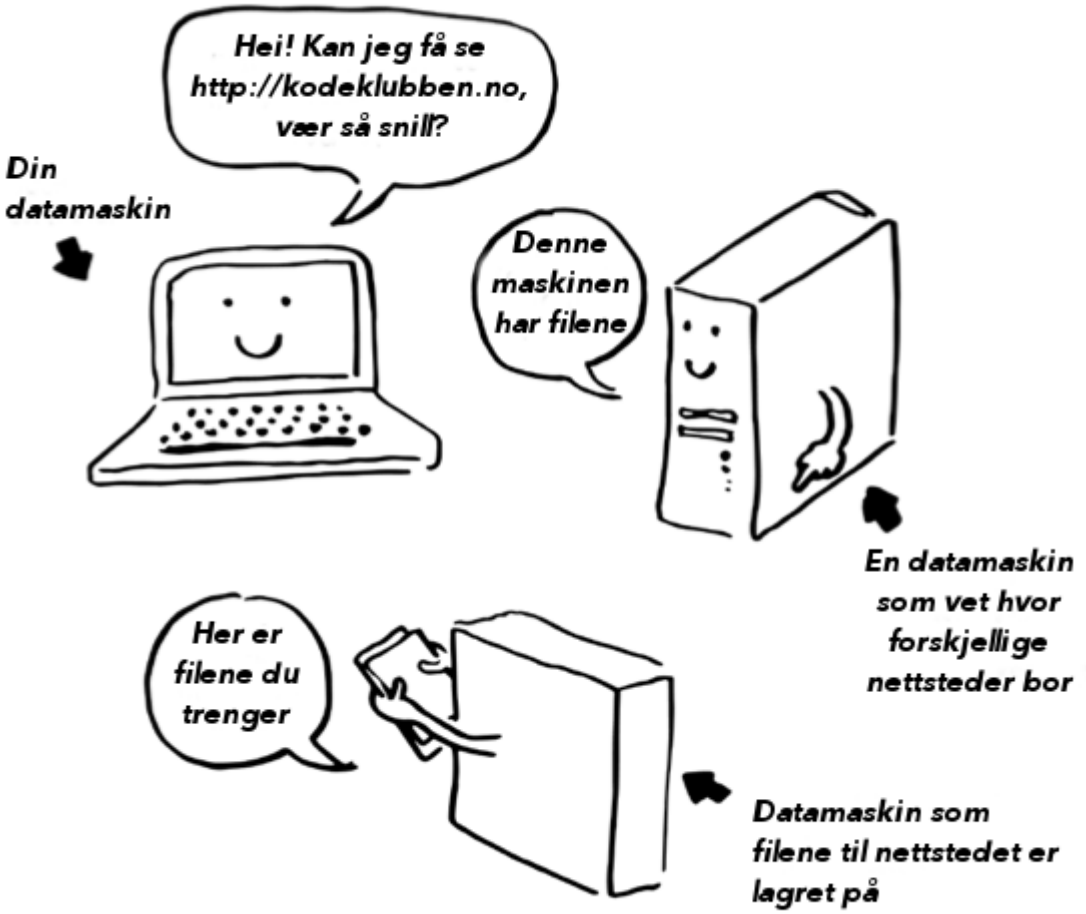
- [ ] Skriv noe! For eksempel: `Hei! Jeg heter ...
- [ ] Lagre filen til filnavnet `hei.txt`
- [ ] Finn filen din i filbehandleren og åpne den. Den åpnes i et tekstprogram, og det er jo ikke så gøy
- [ ] Lagre nå filen på nytt til filnavnet `fil.html` med *Lagre som*
- [ ] Åpne filen på nytt.

Hvilket program ble filen åpnet i denne gangen? Nettleseren er et spesielt program som vet hvordan man skal tolke tekstfiler som er skrevet ved hjelp av HTML. Vi har ikke laget noe HTML enda, bare en HTML-fil. I HTML-filen har vi kun tekst, men det bryr ikke nettleseren seg noe om! Så lenge du gir den en `.html`-fil, kommer den til å gjøre sitt beste for å vise deg hva som er i filen. Det samme gjelder når en HTML-fil inneholder feil, da vil nettleseren prøve å finne ut av hvordan den skal vises fram uansett.

Hvordan kan vi se disse filene?

Når du skriver en adresse inn i nettleseren din, blir forespørselen din sendt avgårde til en datamaskin som alltid står på og er innstilt for å la deg se nettsidene som lever inni den. Denne datamaskinen kalles en *server*. Når den mottar en forespørsel fra din datamaskin, ser den etter alle de nødvendige filene, som for eksempel `.html`-filen, og sender deg den sammen med alt det andre nettsiden trenger, for eksempel bilder og videoer.

Kan jeg få denne siden takk?



Steg 2: Hva er HTML?

HTML er et **markeringsspråk** - det betyr at det brukes til å beskrive hva ting er. Selv om nettleseren prøver å gjøre sitt beste for å vise ting, hjelper det at den vet hva disse tingene er. For å fortelle nettleseren det, bruker vi **tagger**. Tagger ser sånn ut:

```
<p>Dette er litt tekst.</p>
```

`<p>` er en forkortelse for **paragraf**, som er et annet ord for *avsnitt*. Den har en åpning, som er `<p>`, og en tilsvarende avslutning med skråstrek, `</p>`. Nettleseren vet da at alt imellom de to taggene er en paragraf med tekst.

Tagger kan også ha **attributter**, som er informasjon om elementet. La oss se på link-taggen:

```
<a href="http://kodeklubben.no/">Besøk nettsiden til Kodeklubben</a>
```

`<a>` betyr **anker**, som er det linker ble kalt før. Den har åpningen `<a>` og avslutningen `</a>`. I åpningstaggen la vi til attributten `href` med verdien `http://kodeklubben.no/`. `href` står for *hypertekst referanse*. En tekst som linket til andre tekster ble en gang kalt *hypertekst*, fordi den kunne peke til bilder, lyd og andre tekster. Det gjorde *anker* annerledes enn annen vanlig tekst. `href` forteller nettleseren hvor linken skal føre deg, og teksten i mellom taggene vil bli synlig som en link.

Aktiviteter

- [ ] Installer **X-Ray Goggles** ved å trekke linken X-Ray Goggles til din bokmerkerad
- [ ] Åpne filen `side.html` fra internett.zip (internett.zip)
- [ ] Trykk på *X-Ray Goggles* fra bokmerkeraden din. Om ingenting skjer, spør en CodeMaster/veileder om hjelp til å bruke utviklerverktøyet i din nettleser
- [ ] Beveg musen rundt på siden. Da kan du se deler av siden lyse opp og se hvilke tagger delene er laget av
- [ ] Trykk på noe for å se kodesnutten det er laget av
- [ ] Trykk på knappen **ESC** når du er ferdig.

Steg 3: Flere tagger

Vi har allerede nevnt taggene `<p>` og `<a>`, men det finnes mange flere. Her er noen tagger som ofte brukes:

- `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`: Overskrifter med ulik størrelse.
- `<ol>`: Sortert liste.
- `<ul>`: Usortert liste.
- `<li>`: Et punkt i en liste.

- `<hr>` : Horisontal linje.
- `<div>` : En boks for å gruppere ting.
- `<img>` : Et bilde. Bilder er litt spesielle, fordi i motsetning til de fleste andre tagger så har **ikke** `<img>` en avslutning `</img>` . For å vise et bilde bruker vi et attributt som heter `src="lenke til bildet"` , men dette skal vi se nærmere på i en senere oppgave.

Det finnes også noen tagger som vi alltid må ha med i HTML dokumenter:

- `<html>` : Forteller nettleseren at her kommer det HTML-kode.
- `<head>` : Inne i `<head>` skriver vi ting som er nyttig for nettleseren, men som ikke vil dukke opp som tekst på selve nettsiden. For eksempel kan vi bruke taggen `<title>` :

  ```
  <head>
    <title>Arves hjemmeside</title>
  </head>
  ```

  `Arves hjemmeside` vil da benyttes som tittel til hjemmesiden og vises i vinduet til nettleseren.
- `<body>` : Her putter vi det som skal dukke opp på nettsiden.

Aktiviteter

- [ ] Åpne filen `side.html` i teksteditoren din
- [ ] Legg merke til hvordan tagger kan stå på innsiden av andre tagger. Vi har `<a>` -taggen, som er inni `<p>` -taggen, som igjen er inni `<div>` , som er plassert i `<body>` . Når en tagg er på innsiden av en annen sier vi at taggen som er inni er **barnet** og taggen som er rundt er **forelder**. Det er nesten som et slektstre

- [ ] For nettleseren er tagger av samme type like, men du kan skille de fra hverandre ved å bruke klasser. For eksempel kan noen paragrafer være introduksjoner, og da kan vi bruke klassen `introduksjon` for å skille disse paragrafene fra andre paragrafer. Finn taggene som har klasser i filen `side.html`.

- [ ] ID-er brukes for å markere unike elementer på siden din. Finn `div`-taggen som har `id="kattunge"`

- [ ] Hva skjer hvis du flytter ting rundt? Finn en `<ol>` tagg i koden og velg den og alt som er inni den, slik som dette:

```
<ol>
  <li>Kattunger</li>
  <li>Universet</li>
  <li>Å sove lenge</li>
  <li>Å spille spill</li>
</ol>
```

- [ ] Kopier teksten og flytt den til et annet sted

- [ ] Lagre siden og åpne den i nettleseren

- [ ] Hvordan påvirker rekkefølgen av koden rekkefølgen på det som vises i nettleseren?

Ting du kan prøve

- [ ] Lag din egen paragraf med tekst

- [ ] Lag en link som peker til en annen del av siden. **Hint:** Det har noe med ID å gjøre, se etter en link som peker til katten

- [ ] Legg til dine egne overskrifter der du syns de kan passe. Hva skjer hvis du endrer tallet i overskrift-taggen, for eksempel fra `<h3>` til `<h4>`

- [ ] Hva må du gjøre for å linke til en annen side

- [ ] Bruk utviklerverktøyet og dobbeltklikk på kode som ser interessant ut. Endre koden. Du får da en forhåndsvisning på hva som skjer, uten at du trenger å bytte mellom nettleseren og teksteditor. Kult, ikke sant? Oppdater siden. Hva skjedde? Når du redigerer kode på denne måten blir det ikke lagret, så du kan teste hva som skjer uten å ødelegge filen. Sånn kan du eksperimentere masse, men alltid ha muligheten til å gå tilbake.

Lisens: Code Club World Limited Terms of Service (https://github.com/CodeClub/webdev-curriculum/blob/master/LICENSE.md)