

Nivå

1

Nätet

{code
club}

Uppgiften

Har du någon gång undrat hur Internet fungerar? Klart du har! Idag kommer du få lära dig hur du gör webbsidor så att du också kan bidra till allt som finns på nätet. Webbsidor skapas med hjälp av något som heter HTML, vilket betyder HyperText Markup Language. Du kommer få veta mer när du börjar komma igång med att bygga din sida.



Activity Checklist

Follow these **INSTRUCTIONS** one by one



Test your Project

Click on the green flag to **TEST** your code



Save your Project

Make sure to **SAVE** your work now

Steg 1: Vad är webbsidor?

✓ Checklista

1. Öppna ett textredigeringsprogram. ☐
2. Skapa ett nytt dokument. ☐
3. Skriv något! Till exempel: "Hej! Jag är en webbsida." ☐
4. Spara filen. Kalla den "hej.txt". ☐
5. Leta upp filen och öppna upp den igen. Den öppnas i ett textredigeringsprogram, och det är ju inte så himla kul. ☐
6. Byt ut filändelsen (delen efter punkten) till ".html" så att filen istället heter "hej.html". ☐
7. Öppna filen igen. ☐

Vilket program öppnades filen i nu? Webbläsaren är ett särskilt program som vet hur man läser textfiler skrivna med språket HTML. Vi har inte lagt till någon HTML än, vi la bara in lite text, men webbläsaren bryr sig inte om det! Så länge du ger den en fil som slutar på ".html" kommer den göra sitt bästa för att visa dig filen på ett snyggt och bra sätt.

Det här är väldigt användbart: även om en webbsida innehåller fel kommer webbläsaren att försöka visa dig den.

Hur kan vi se de här filerna?

När du skriver in en adress i din webbläsare skickas den till en dator som alltid är påslagen och inställd på så sätt att du tillåts se webbsidorna som finns inuti den. En sådan dator kallas för en server. När den får en begäran från din dator (adressen du bad att få surfa in på) letar den upp alla nödvändiga filer: ".html"-filen, och kanske bilder och videor som finns på sidan.

Kan jag få se den här sidan, tack?



Varsågod.

Steg 2: Vad är HTML?

HTML är ett markup-språk. Det betyder att det används till att beskriva olika saker.

Även om webbläsaren kommer göra sitt bästa för att visa allt hjälper det att veta mer om vilka saker den kan tänkas beskriva.

För att berätta för webbläsaren vilka saker vi vill se på sidan använder vi "taggar".

Taggar ser ut såhär: `<p>Här är lite text.</p>`

`<p>` betyder paragraph (paragraf eller stycke).

Det finns en öppningstagg som ser ut såhär: `<p>` och en matchande stängningstagg med ett slash inuti: `</p>`. Webbläsaren vet om att allting som kommer mellan de två taggarna ska tolkas som ett stycke text.

Taggar kan ha attribut, som är små bitar av användbar information om elementet. Vi tittar på en länktagg:

```
<a href="http://codeclub.org.uk">Besök CodeClubs webbsida</a>
```

"" betyder anchor, eller "ankare" på svenska, och det kallade man länkar för länge sen.

Ett länkelement har också öppningstaggen `<a>` och stängningstaggen `` men vi behöver även ett attribut till öppningstaggen: ``

"href" är attributet, och "http://codeclub.org.uk" är dess värde.

"href" står för *hypertext reference*. En text som länkad till andra texter kallades en gång i tiden för hypertext, för att den kunde ha bilder och ljud och ta en vidare till andra sidor. Det gjorde den lite annorlunda än en vanlig text.

"href" berättar för webbläsaren vart länken ska leda, och texten inom de två taggarna kommer bli synlig på sidan som en länk.



1. Öppna filen "page.html".
2. Fråga en ledare om hon eller han kan visa ett sätt att titta på koden i webbläsaren.



Granska en webbsida

1. Prova att inspektera alla delar av sidan. Kan du komma på vad alla de olika taggarna betyder?



Vi känner redan till `<p>` och `<a>`.

`` - ordered list (sorterad lista)

`` - unordered list (osorterad lista)

`` - list item (listelement)

`<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>` - headings (rubriker)

`<hr>` - horizontal rule (horisontell linje)

`<div>` - en box som innehåller saker

`` - ett element som är speciellt på så vis att det inte har en stängningstagg. Vi använder det för att lägga in bilder.

Det finns också några taggar som vi alltid använder i HTML-dokument, och de är:

`<html>` - berättar för webbläsaren var den hittar all kod

`<head>` - inuti `<head>` stoppar vi grejer som webbläsaren kan behöva veta, men som inte syns på sidan. Här till exempel, la vi in en `<title>` för att visa sidnamnet högst upp i webbläsarfönstret.

`<body>` är där vi stoppar allting vi vill ska synas på sidan.



Checklista

1. Lägg märke till hur taggar kan nästlas inuti varandra. Vi



har en `<a>`-tagg som står inuti en `<p>`-tagg, vilken i sin tur är inuti en `<div>` som är inuti en `<body>`. När det här händer säger vi att taggen som är inuti en annan tagg är ett *barn* och taggen som är runt omkring den är en förälder. Det är lite som ett familjeträd!

2. För webbläsaren är alla taggar av samma sort desamma, men du kan identifiera dem genom att använda klasser och ID:n. Till exempel kanske några av dina paragrafer är inledningar på texter, och då kan du ge dem klassen "inledning". Kolla om du kan hitta några klasser inuti "page.html".



+Id:n används för att markera unika element på din sida. Kolla om du kan hitta någon -tagg med id:t "kitten" på sidan.

3. Vad händer om du flyttar runt grejer? Vi går tillbaka till kodredigeraren. Leta upp en ``-tagg i koden och markera den med allt som finns inuti, såhär:



```
<ol>
  <li>Kittens</li>
  <li>Cake</li>
  <li>Lie-ins</li>
  <li>Playing games</li>
</ol>
```

Kopiera den nu och flytta den någon annanstans. Spara sidan och uppdatera den i webbläsaren. Hur påverkar din kods ordning hur sidan ser ut i webbläsaren?

Pröva också

- ☐ Skapa din egen textparagraf.
- ☐ Skapa en länk som pekar ut en annan del av sidan (ledtråd: det har att göra med id - kolla efter en länk som tar dig till kattungen ("kitten" på engelska).
- ☐ Lägg till dina egna rubriker där du tycker de kan passa in. Vad händer om du ändrar siffran i rubriktaggen, till exempel från `<h3>` till `<h4>`?
- ☐ Vad skulle du behöva göra för att länka till en helt annan sida?