

Grundläggande instruktioner för att publicera webbsidor på webbservern.se

Av Julia Klingvall
i samarbete med webbservern.se.

Version 1.1, januari 2005.

*I denna guide kommer du att lära dig de viktigaste grunderna
för att skapa och redigera webbsidor på webbservern.se*

Innehåll

1 Inledning.....	1
1.1 Förändringar från förra versionen.....	1
2 Att ansluta till webbservern.se.....	1
2.1 SSH-anslutning.....	1
2.2 Drall-anslutning.....	3
3 Skapa en webbsida.....	3
3.1 I Emacs via SSH.....	3
3.2 I Drall.....	4
4 HTML-grunder.....	4
4.1 Grundläggande taggar.....	4
4.2 Radbrytningar och kommentarer.....	4
4.3 Formatera innehållet.....	4
4.4 Länkar.....	5
4.5 Tabeller.....	6
4.6 Ramar (frames).....	6
4.7 Bilder.....	6
4.8 Inkludera filer.....	7
4.9 Formatera taggar.....	7
4.10 Validera din webbsida.....	8
5 Vanliga Linuxkommandon.....	8
5.1 Visa aktuell katalog.....	8
5.2 Visa filer i katalog.....	8
5.3 Byta katalog.....	8
5.4 Kopiera filer eller kataloger.....	9
5.5 Flytta eller byta namn på filer och kataloger.....	9
5.6 Ta bort filer eller kataloger.....	9
5.7 Skapa ny katalog.....	9
6 Vanliga Emacskommandon.....	9
7 CSS-formatering.....	10
7.1 Typsnitt.....	10
7.2 Teckenstorlek.....	10
7.3 Färg.....	11
7.4 Marginaler.....	11
7.5 CSS-boxar.....	11
7.6 Exempelformateringar.....	11

1 Inledning

Den webbsserver som denna guide avser är <http://webbservern.se>. Det som är specifikt för den webbsservern är hur man ansluter till den och vilka program som körs på den med mera. HTML-guiden kan användas för webbsidetillverkning även på andra webbservrar.

Mer information om webbsservern finns på <http://webbservern.se>, se speciellt handboken.

Jag rekommenderar att du själv provar det som beskrivs i detta häfte. Det är meningen att det ska gå att läsa det från början till slut och samtidigt göra sin webbsida.

Mycket nöje med webbsidetillverkningen!

1.1 Förändringar från förra versionen

Förändringar i version 1.1 jämfört med den ursprungliga versionen är att instruktioner för Drall har tillkommit, samt att smärre formförändringar genomförts. Bland annat används numera endast en kolumn, för att underlätta för läsare av digitala versioner. Dokumentet är anpassat för dubbelsidig utskrift.

Eventuella fel rapporteras vänligen till författaren med e-postadressen [julia \[snabel-a\] klingvall.se](mailto:julia@snabel-a.klingvall.se).

2 Att ansluta till webbservern.se

Denna information gäller endast när du skall ansluta till just webbservern.se.

Det finns flera sätt att komma åt webbservern.se. Om du vill redigera din webbsida direkt på webbservern finns huvudsakligen två sätt: SSH och i webbläsaren med hjälp av Drall. Båda sätten beskrivs nedan. Om du använder det ena sättet behöver du inte läsa instruktionerna för det andra sättet för att ändå förstå hur du ska gå tillväga.

2.1 SSH-anslutning

SSH står för "secure shell" och används bland annat för att logga in på en dator i nätverket, exempelvis webbservern.se. När det står att du ska skriva nåt i "skalet" är det alltså det textläge du hamnar i efter att du loggat in på webbservern med exempelvis programmet Putty.

Fördelen med SSH är att du slipper ladda ner den fil du vill redigera till din egen dator och redigera den för att sedan ladda upp den igen, utan filen ändras direkt på webbservern.

Nackdelarna är att det man gör syns direkt på webbsidan och att om du till exempel raderar en fil på webbservern blir det lurigare att få tillbaka den. Hjälp finns dock, eftersom webbservern säkerhetskopierar filerna med jämna mellanrum. Om du vill ha en äldre version av din fil ska du använda formuläret som finns på http://webbservern.se/handbok/req_backup.shtml.

SSH är ett kraftfullt sätt att fjärrstyra webbservern, men jag kommer bara att gå in på de funktioner som jag tror att du kommer att behöva för webbsidetillverkningen.

Program som krävs för SSH

Om du använder Windows som operativsystem är programmet Putty en bra SSH-klient. Om du inte redan har det på datorn måste du ladda ner det från till exempel <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Att ansluta med Putty

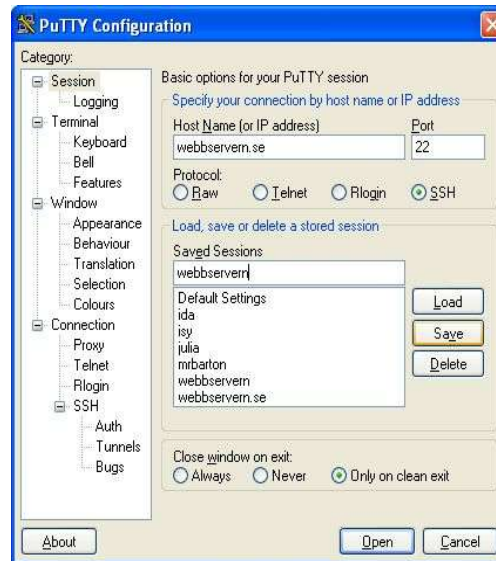
Datorn som du ska ansluta till heter "webbservern.se" (utan citationstecken) och det skriver du i rutan för "Host name". Porten ska vara 22 och du ska välja protokollet SSH. Om du vill spara dessa inställningar för senare användning skriver du ett namn på den i rutan för "Saved Sessions" och trycker sedan på "Save". Starta SSH genom att trycka på "Open". Nästa gång du ska ansluta till webb-

servern markerar du namnet i listan med namn under "Saved Sessions" och trycker på "Load", så laddas inställningarna in. Bilden *Hur Putty konfigureras* nedan på sidan 2, visar hur Putty ser ut och vilka inställningar du ska göra.

Att logga in

När du har startat Putty är du ansluten till webbservern, som har Linux som operativsystem. Därför måste du kommunicera med webbservern med hjälp av Linuxkommandon.

När du har tryckt på "Open" i Putty öppnas en svart ruta där du ska ange användarnamn, och sedan trycka på Enter. Därefter ska du ange ditt lösenord, och om det var rätt hamnar du i din hemkatalog på webbservern. Det syns inget när du skriver in lösenordet och om du blir osäker på om du skrev rätt kan du ta bort alla bokstäver med kommandot Ctrl+u (håll ner Ctrl-tangenten på tangentbordet och tryck på "u") och börja om från början. Du har bara två chanser på dig att skriva rätt lösenord, sedan måste du starta om Putty. *Illustration 2 Inloggningsryn vid SSH* visar hur det ser ut när jag loggar in med användarnamnet "julia".



På tredje raden kommer det upp

```
[julia@webbservern.se julia] $
```

och det talar om vem jag är inloggad som, på vilken dator och i vilken katalog (mapp) jag står nu, alltså:

```
[användarnamn@server katalog] $
```



Illustration 2 Inloggningsryn vid SSH

Efter \$ kan du skriva egna kommandon. För enkelhetens skull skriver jag bara det som du ska skriva och \$ i mina exempel.

I stycke 5 *Vanliga Linuxkommandon* finns de vanliga kommandon som kan vara bra att använda när du navigerar i filstrukturen och till exempel byter namn på filer. Om du vill kan du läsa om dem, eller så börjar du direkt att skapa en webbsida.

Textläge

Om du ansluter på ovan angivna sätt kommer du att kommunicera med webbservern endast med text. Textläge innebär att det inte finns några knappar eller menyer (grafik) och ofta kan inte heller musen användas för att flytta markören runt i fönstret. Textläget gör att det går fortare att arbeta direkt mot webbservern eftersom webbservern annars måste processa mycket mer information som berör till exempel knappar och menyer.

Grafiskt läge (ev. överkurs)

I Putty finns det stöd för så kallad X11-forwarding vilket innebär att du kan visa grafik lokalt som generas på servern. I Putty aktiverar du X11-forwarding när du ansluter till webbservern och låter en X-server (till exempel X-Win32, som du måste ladda ner) köras i bakgrunden lokalt på din dator. Grafiska program som du startar i skalet (i Putty) kommer då att visas lokalt på din dator. Detta är som sagt långsammare även om du har en snabb internetuppkoppling, och denna guide kommer främst att behandla det textbaserade läget.

2.2 Drall-anslutning

Istället för via SSH kan filerna på webbservern redigeras i webbläsaren med hjälp av Drall. Detta sätt kan för nybörjaren vara enklare, mycket på grund av det grafiska gränssnittet.

För att komma åt Drall måste man logga in på `<http://webbservern.se>` genom att trycka på “Logga in” i nedre högra hörnet på webbservern.se:s startsida.

Välj sedan *filhantering med en kolumn*, varefter dina filer visas i webbläsaren. De filer som ligger i `public_html`-katalogen är åtkomliga via internet.

3 Skapa en webbsida

Även i detta kapitel delar jag upp instruktionerna i två delar, en för SSH och en för Drall.

3.1 I Emacs via SSH

I detta stycke går jag igenom hur du skapar en webbsida i textredigeraren Emacs.

Byt namn på indexfilen

Innan du börjar göra din egen webbsida rekommenderar jag att du byter namn på filen `index.html` (även på `index`-filer som har ändelsen `.htm`, `.shtml` och `.php`) till något annat, till exempel `gammal_index.html`. Detta gör du genom att stå i din `public_html`-katalog och skriva

```
$ mv index.html gammal_index.html
```

Kommandot `mv` förklaras ytterligare i stycke 5.5 *Flytta eller byta namn på filer och kataloger* på sidan 9.

Skapa en fil i Emacs

Jag rekommenderar textredigeraren Emacs för att skapa och redigera dina webbsidefiler. Börja med att gå in i den katalog som heter `public_html`, det är där de filer som ska visas på internet ska ligga. Om du står i din hemkatalog går du till `public_html` genom att skriva

```
$ cd public_html
```

Om du är osäker på detta förfarande kan du läsa stycke 5.3 *Byta katalog* på sidan 8.

Sedan låter du Emacs öppna och skapa en fil som heter `index.html` genom att skriva kommandot

```
$ emacs index.html -nw
```

Flaggan `-nw` betyder att Emacs inte ska öppnas som ett eget program med grafik, utan köras i textläge i skalet. Om du inte har aktiverat det grafiska läget (se stycke 2.1 *Grafiskt läge (ev. överkurs)* på sidan 2) behöver du inte skriva `-nw`.

Det är filen `index.html` som besökarna på din webbsida kommer att mötas av, alltså den fil som automatiskt läses in när adressen `http://anvamn.webbservern.se` anropas, till exempel `http://julia.klingvall.se`.

Prova att skriva vad som helst i filen och spara den sedan genom att hålla ner `Ctrl`-tangents och trycka på tangents `x` följt av `s`. Detta förfarande kommer jag i fortsättningen att beteckna `Ctrl+x+s`. Öppna nu en webbläsare och skriv in adressen till din webbsida (på formen som ovan) och se om din text syns!

Med kommandot `Ctrl+x+c` stängs Emacs av. Om du inte har sparat innan du försöker stänga frågar Emacs om du vill spara, genom att ställa frågan längst ner i fönstret. Inom parenteserna efter frågan presenteras din möjliga svarsalternativ där `y` betyder ja och `n` betyder nej. Skriv direkt efter vilket alternativ du väljer, förslagsvis `y`, följt av `Enter`. Då sparas din fil och Emacs avslutas.

En lista över vanliga kommandon i Emacs presenteras i stycke 6 *Vanliga Emacskommandon* på sidan 9.

3.2 I Drall

Eftersom Drall har ett bra grafiskt gränssnitt är det i stort sett självförklarande. Det du behöver göra är att antingen ta bort eller döpa om dina eventuellt befintliga filer som heter `index.html`, `index.htm` eller `index.php`.

Sedan skapar du en fil som heter `index.html` och skriver några tecken i den. Spara filen och surfa till din webbsida, antagligen har den adressen `http://dittnamn.webbservern.se`

Det är filen `index.html` som besökarna på din webbsida kommer att mötas av, alltså den fil som automatiskt läses in när adressen `http://anvnamn.webbservern.se` anropas.

4 HTML-grunder

I detta stycke ska jag gå igenom hur en webbsida kan skrivas och hur du formaterar din text. Samma instruktioner gäller oavsett om du använder Drall eller SSH.

HTML betyder *HyperText Markup Language* och är i princip ett sätt att göra layout till webbsidor.

Öppna filen `index.html` och prova att skriva de olika kommandon som jag går igenom. Titta på webbsidan på internet ofta, så att du ser vilket resultat de olika kommandona ger. Glöm inte att uppdatera sidan i webbläsaren för att se ändringarna!

4.1 Grundläggande taggar

En HTML-sida skrivs mellan taggarna `<html>` och `</html>`, där den senare är en avslutande tagg. Mellan dessa taggar ska *head*- och *body*-taggar finnas. Det man skriver inom *body*-taggarna är det som syns i webbläsaren. Inom *head*-taggarna kan man lägga information om webbsidan, till exempel vilken titel sidan har. Titeln är det som visas högst upp i webbläsarens fönsterkant, samt blir namnet på bokmärket om sidan läggs som bokmärke.

```
<html>
<head>
<title>Min webbsida</title>
</head>
<body>
<!-- innehåll -->
</body>
</html>
```

4.2 Radbrytningar och kommentarer

Du kan göra radbrytningar, tabbar (tabulatorsteg) och mellanslag godtyckligt i koden, fast inte inom en tagg (mellan `<` och `>`). Kommentarer, som inte syns i webbläsaren, skrivs mellan `<!--` och `-->`, se mellan *body*-taggarna i exemplet ovan.

Om du skriver flera mellanslag i koden kommer webbläsaren ändå bara att visa ett. Om du vill ha flera mellanslag skriver du ` `; istället för varje mellanslag, direkt i koden.

En radbrytning utan att använda styckefunktionen åstadkoms genom att använda taggen `
` direkt i koden, och den ska inte avslutas.

4.3 Formatera innehållet

Det finns en mängd taggar som formaterar innehållet på din webbsida.

Några vanliga taggar:

<i>Funktion</i>	<i>Tagg</i>	<i>Exempel</i>
Stycke	<code><p></code>	<code><p>Mitt stycke.</code>

<i>Funktion</i>	<i>Tagg</i>	<i>Exempel</i>
Rubriknivå 1	<h1>	<h1>Rubrik</h1>
Rubriknivå 2	<h2>	<h2>Rubrik</h2>
Fetstil		fetstil
Kursiv stil	<i>	<i>kursiv</i>

Prova dessa taggar på din webbsida, till exempel så här:

```
<html>
<head>
<title>Min webbsida</title>
</head>
<body>
<h1>Min webbsida</h1>
<p>Detta är <i>ett första utkast</i> till en <b>webbsida</b>.
<h2>Om mig</h2>
<p>Jag heter si och så och ligger bakom denna webbsida.
</body>
</html>
```

Här har du inte själv fått vara med och bestämma hur till exempel rubriknivåerna ska se ut, utan webbläsaren använder sin standardmall för det. Men du kan bestämma hur den text som står inom taggarna ska visas, se stycke 4.9 *Formatera taggar* på sidan 7.

Det finns många fler taggar som du kan testa, till exempel:

Listor

En punktlista görs så här:

```
<ul>
<li>Första punkten i en punktlista
<li>Andra punkten, osv.
</ul>
```

För att få en numrerad lista byter man ut mot , och förstås även mot . Dessa taggar betyder *unordered list* respektive *ordered list*.

Listorna kan även länkas, till exempel så här:

```
<ul>
<li>toppnivå
  <ul>
    <li>undernivå
    <li>undernivå
  </ul>
<li>toppnivå
</ul>
```

Justera innehållet

Hur innehållet ska justeras kan du ange i din css-fil eller direkt i html-filen.

Till exempel ett stycke kan centreras genom att lägga till ett attribut till taggen tabell:

```
<p align="center">
```

Förutom att centrera kan man vänster- eller högerställa (*left*, *right*).

4.4 Länkar

En länk skrivs så här:

```
<a href="http://www.dn.se" alt="beskrivning av länken">Dagens Nyheter</a>
```

Om du vill att länken ska öppnas i ett nytt fönster eller i ett fönster med ett visst namn, skriver du

```
<a href="http://www.dn.se" alt="beskrivning av länken" target="fönsternamn">Dagens Nyheter</a>
```

4.5 Tabeller

Taggen `<table>` påbörjar en ny tabell. `<tr>` gör en ny rad och `<td>` gör en cell. Ett exempel på en enkel tabell:

```
<table>
<tr>
<td>cell 1</td>
<td>cell 2</td>
</tr>
<tr>
<td>cell 3</td>
<td>cell 4</td>
</tr>
</table>
```

Denna tabell kommer att ha två celler, nummer 1 och 2 på översta raden, och sedan de två andra cellerna på nästa rad.

Webbsidelayout med tabeller

Tabeller är ett enkelt och vanligt sätt att bygga upp webbsidor på.

Till exempel kan hela din sida bestå av en tabell med två celler, en med meny i och en som visar innehåll.

I body-taggen skriver du då:

```
<table>
<tr>
<td>Meny här</td>
<td>Innehåll här</td>
</tr>
</table>
```

Tabellrubrik

Istället för `<td>` kan man använda `<th>`, *table header*, på första raden. Denna tagg kan man formatera så att den har ett annorlunda utseende.

Formatera tabellen

Taggarna `table`, `tr`, `th` och `td` kan formateras med hjälp av CSS, se stycket 4.9 *Formatera taggar* på sidan 7.

4.6 Ramar (frames)

Ramar innebär att webbläsaren kan visa flera filer i samma fönster, bredvid varandra i olika konstellationer. Detta rekommenderar jag inte eftersom det då finns en risk att besökarna inte hittar alla sidorna samtidigt, utan kanske hamnar på en av dem, till exempel via en sökmotor, och på så sätt missar en viktig del av webbsidan. Jag rekommenderar att du istället använder SSI, *Server Side Include*, se stycket 4.8 *Inkludera filer* på sidan 7.

Ett annat vanligt sätt att bygga upp en sidlayout är *iframes*. Här placeras en fil i en annan till exempel i en tabell. Även här är tabeller ett bättre alternativ eftersom de sitter ihop med övriga sidan.

4.7 Bilder

De bilder som ska synas på webben måste du ladda upp om de inte redan ligger där. Detta kan du till exempel göra genom att surfa till `<http://webbservern.se>` och logga in, och sedan välja att titta på "mina filer".

I din html-fil skriver du följande:


```

```

Den alternativa texten syns i de fall bilden inte visas.

4.8 Inkludera filer

Om webbplatsen består av flera filer är det vanligt att en del av innehållet återkommer på alla sidorna, till exempel en titel på webbsidan och en meny. För att detta ska vara lätt att ändra delar man upp filen i flera delar, och vanligtvis kallar man dem något i stil med `sidhuvud.html` och `sidfot.html`.

Den enkla exempelsidan vi har tittat på förut skulle kunna delas upp i tre filer: `sidhuvud.html`, `index.html` och `sidfot.html`.

Sidhuvudfilen innehåller då:

```
<html>
<head>
<title>Min webbsida</title>
</head>
<body>
```

och `index`-filen innehåller:

```
<!--#include file="sidhuvud.html"-->
<h1>Min webbsida</h1>
<p>Detta är <i>ett första utkast</i> till en <b>webbsida</b>.
<h2>Om mig</h2>
<p>Jag heter si och så och ligger bakom denna webbsida.
<!--#include file="sidfot.html"-->
```

och `sidfot`-filen innehåller:

```
</body>
</html>
```

Jag har alltså lagt till två rader i `index`-filen som anger var och vilken fil som ska sättas ihop med `index`-filen när en besökare vill titta på webbsidan.

Det här är ett mycket smidigt sätt att bygga upp en webbsida på och rekommenderas starkt.

Mer om SSI finns på [webbservern.se](http://webbservern.se/handbok/wwwSSI.shtml): `<http://webbservern.se/handbok/wwwSSI.shtml>`

4.9 Formatera taggar

Taggarna formateras enklast genom att förklarar hur de ska visas i en separat fil, med hjälp av CSS (Cascading Style Sheet).

För att göra detta skapar du en fil i din `public_html`-katalog som heter till exempel `formatering.css`. I Emacs gör du så här: Tryck `Ctrl+x+f` och skriv därefter filnamnet och avsluta med `Enter`.

Här finns en mängd egenskaper som du kan tilldela dina taggar. Ett exempel på formatering av `<h1>`:

```
h1 {
font-family: verdana, arial;
font-size: x-large;
color: #000000;
margin-top: 3px;
margin-bottom: 12px;
text-decoration: none;
font-weight: normal;
border-bottom: 4px double #CCCCCC
}
```

Du måste även berätta för din `index`-fil att den ska titta i din formateringsfil. Detta gör du genom att skriva följande mellan `head`-taggarna:

```
<link rel="stylesheet" href="formatering.css" type="text/css">
```

Skriv in formateringsanvisningarna i din formateringsfil, `formatering.css` och referensen ovan i din index-fil, `index.html`, och testa hur det ser ut i webbläsaren!

På detta sätt kan du formatera flera taggar. Det är viktigt att du avslutar varje egenskap med ett semikolon samt att du skriver alla taggens egenskaper inom mäsvingar:

```
taggnamn {
egenskap: värde;
}
```

En utförligare lista över hur du kan formatera taggarna finns i stycke 7 *CSS-formatering* på sidan 10.

4.10 Validera din webbsida

Du kan validera din HTML-kod, det vill säga kontroller att den är korrekt, på w3c:s webbplats `<http://validator.w3.org/>`.

5 Vanliga Linuxkommandon

I detta stycke presenteras och förklaras några vanliga kommandon för att navigera i filsystemet och kopiera, flytta och byta namn på filer och kataloger.

5.1 Visa aktuell katalog

Kommandot `pwd` (*print working directory*) visar hela sökvägen till den katalog du står i nu. Prova!

```
$ pwd
/home/julia
```

5.2 Visa filer i katalog

Kommandot `ls` (*list*) listar alla filer och kataloger som finns i den katalog du nu står i. I exemplet nedan finns alltså två kataloger som heter `backup` och `public_html` och en fil som heter `links.html`.

```
$ ls
backup          links.html
public_html
```

Det är de filer som ligger i katalogen `public_html` som är åtkomliga via internet.

Om du vill veta utförligare information om filerna lägger du till flaggan `-l`:

```
$ ls -l
```

5.3 Byta katalog

För att utforska vilka filer som ligger i `public_html` kan du gå in i den katalogen genom att skriva

```
$ cd public_html
```

Du kan förenkla skrivsättet genom att bara skriva till exempel `cd pu` och sedan trycka på tabbtangenten. Då fyller webbservern ut med det resterande katalognamnet och du kan avsluta med Enter. Så här kan du alltid prova att göra så föreslår webbservern olika möjliga fortsättningar på ditt påbörjade ord. Kommandot `cd` är en förkortning av *change directory*.

Om du vill komma direkt till din hemkatalog skriver du

```
$ cd
```

Om du vill gå upp ett steg i katalogstrukturen skriver du

```
$ cd ..
```

5.4 Kopiera filer eller kataloger

Kommandot `cp` (*copy*) används för att kopiera en eller flera filer eller kataloger. I exemplet kopierar jag filen `fil.html` som ligger i den katalog som jag står i till en underkatalog som heter `mapp`.

```
$ cp fil.html mapp/
```

Om du ska kopiera en katalog måste du lägga till flaggan `-a` till ditt kommando:

```
$ cp -a mapp mapp2
```

Här kopieras det som ligger i katalogen `mapp` och läggs i en ny katalog som får namnet `mapp2`.

5.5 Flytta eller byta namn på filer och kataloger

För att flytta en fil används kommandot `mv` (*move*).

```
$ mv fil.html mapp/
```

Här flyttas filen `fil.html` in i katalogen `mapp`. Den finns alltså inte kvar i den ursprungliga katalogen.

Detta kommando kan även användas för att döpa om en fil:

```
$ mv fil.html fil2.html
```

Kommandot `mv` fungerar även för att flytta en hel mapp:

```
$ mv mapp public_html/
```

5.6 Ta bort filer eller kataloger

Kommandot `rm` (*remove*) tar bort en fil. Om du vill ta bort en hel katalog med innehåll måste du lägga till en flagga, `-R`.

```
$ rm fil.html
```

```
$ rm -R mapp
```

En tom katalog kan även tas bort med kommandot

```
$ rmdir mapp
```

Detta kommando bör användas med varsamhet.

5.7 Skapa ny katalog

En ny katalog skapas med kommandot

```
$ mkdir ny_mapp
```

som är en förkortning av *make directory*. Katalogen hamnar i den katalog du står i när du utför kommandot. Du kan även skapa en katalog i en annan katalog än den du står i:

```
$ mkdir public_html/ny_mapp
```

Här skapas katalogen `ny_mapp` i katalogen `public_html` som ligger i den katalog som du står i.

6 Vanliga Emacskommandon

Starta Emacs i textläge med kommandot

```
$ emacs -nw
```

i skalfönstret. Du kan även öppna Emacs med en fil:

```
$ emacs index.html -nw
```

Vanliga kommandon i Emacs är:

<i>Funktio n</i>	<i>Ko m m a n d o</i>
Avsluta Emacs	Ctrl+x+c
Spara fil	Ctrl+x+s
Öppna fil eller skapa ny fil	Ctrl+x+f
Spara fil som	Ctrl+x+w
Ångra	Ctrl+_
Klipp ut	Ctrl+w
Kopiera	Esc+w
Klistra in	Ctrl+y
Sätt markör	Ctrl+mellanslag
Gå till slutet av raden	Ctrl+e
Gå till början av raden	Ctrl+a
Sök och ersätt	Esc+x replace-string

För att kunna klippa ut eller kopiera en text måste du markera den först. Om detta inte går att göra med musen måste du göra så här:

1. Sätt markören i ena änden av det stycke som du ska markera.
2. Håll nere Ctrl-tangenten och tryck på mellanslag på tangentbordet.
3. Stega med piltangenterna till slutet av det stycke du vill markera.
4. Nu är stycket markerat och du kan utföra din operation.

Mer om Emacs finns på <http://webbservern.se/handbok/progEmacs.shtml>

7 CSS-formatering

Detta kapitel är förhållandevis fåordigt, men innehåller många exempel.

7.1 Typsnitt

Exempel:

```
p {
font-family: verdana, arial, sans-serif;
}
```

Många olika typsnitt kan anges, och flera alternativ kan anges genom att de separeras med komma-tecken. Denna egenskap är bra att använda i alla taggar som ska innehålla text, det vill säga nästan alla.

7.2 Teckenstorlek

Exempel:

```
p {
font-size: xx-small;
}
```

Möjliga värden är till exempel:

```
xx-small, x-small, small, medium, large, x-large, xx-large
```

men storleken kan även anges till exempel i punkter: 12pt.

7.3 Färg

Färgerna anges hexadecimalt med sex "siffror". På varje av de sex platserna kan du använda någon av tecknen 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, där 0 betyder mest färg och f minst. Till exempel är #000000 helt svart och #ffffff är helt vitt. Det går alltså att göra många olika färger, men man brukar rekommendera att bara använda de färger som bildas av talen 0, 3, 6, 9, c och f eftersom de anses kunna visas av de flesta webbläsare och datorer.

Rubriker inom taggen <h1> får nu färgen svart:

```
h1 {
color: #000000;
}
```

Det går även ofta att skriva endast tre tecken, och då menar man att de ska tolkas som grupper om två. #993300 kan alltså även skrivas #930.

7.4 Marginaler

Marginalerna runt om en text eller en tabell kan anges genom

```
margin-top: 3px;
margin-bottom: 3px;
margin-left: 3px;
margin-right: 3px;
```

7.5 CSS-boxar

CSS-boxar är som textutor. De kan även få en bläddringslist om texten inte får plats. Dessa kan formateras på många sätt, till exempel storlek och position. Ett exempel hämtat från exempelformateringarna nedan:

```
.min_scrollbox {
padding: 5px;
height: 60px;
border: 1px dotted #000000;
overflow: auto;
background-color: #FFFFFF;
margin-bottom: 12px;
}
```

I html-filen skriver man

```
<div class="min_scrollbox">Här står texten som hamnar i boxen.</div>
```

7.6 Exempelformateringar

För att inspirera följer nu några exempelformateringar.

Bodytaggen

```
BODY {
text-decoration: none;
scrollbar-face-color: #D5D5D5;
scrollbar-shadow-color: #000000;
scrollbar-highlight-color: #D5D5D5;
scrollbar-3dlight-color: #000000;
scrollbar-darkshadow-color: #D5D5D5;
scrollbar-track-color: #D5D5D5;
scrollbar-arrow-color: #000000;
}
```

De sista egenskaperna beskriver hur rullgardinslisten ska se ut, men det syns inte i alla webbläsare.

Länkar

```
A:link, A:visited {
color: #666666;
text-decoration: none;
border-bottom: 1px solid #CCCCCC;
}

A:active, A:hover {
color: #000000;
}
```

Här har jag delat upp länkarna i två kategorier, *link* och *visited* bestämmer det utseende länkarna har när de är besökta eller inte, och *active* och *hover* bestämmer länkarnas utseende när de är aktiva eller när muspekaren hålls över dem.

Understrykningar

```
h1 {
font-family: verdana, arial;
font-size: x-large;
color: #000000;
margin-top: 3px;
margin-bottom: 12px;
text-decoration: none;
font-weight: normal;
border-bottom: 4px double #CCC;
}
```

Här har jag använt en dubbelunderstrykning under rubriknivå 1. `text-decoration` kan till exempel ha värdet `underline`, och då blir texten understruken. Men för att göra understrykningen dubbel, tjockare eller ha annan färg än texten krävs att man använder en `border-bottom`. Som attribut kan den ha `double`, `dotted` eller `solid`, förutom tjockleken och färgen.

Rubriknivåer

Här följer ytterligare två rubriknivåer. Marginalerna kanske kan vara av intresse.

```
h2 {
font-family: verdana, arial;
font-size: medium;
font-weight: bold;
color: #000000;
margin-top: 32px;
margin-bottom: 12px;
border-bottom: 1px solid #CCCCCC;
}

h3 {
font-family: verdana, arial;
font-size: small;
font-weight: bold;
color: #000000;
margin-top: 12px;
margin-bottom: 2px;
}
```

Stycke och lista

```
p {
margin-top: 0px;
margin-bottom: 12px;
font-family: verdana, arial;
font-size: x-small;
color: #000000;
}

li {
margin-top: 0px;
margin-bottom: 3px;
font-family: verdana, arial;
color: #000000;
}
```

```
font-size: x-small;
}
```

Celler i tabell

```
td {
margin-top: 0px;
margin-bottom: 24px;
font-family: verdana, arial;
font-size: x-small;
color: #000000;
}
```

```
th {
margin-top: 0px;
margin-bottom: 24px;
font-family: verdana, arial;
font-size: medium;
text-align: center;
color: #000000;
background: #EEEEEE;
font-weight: bold;
}
```

CSS-box

Samma exempel som ovan.

```
.scrollbox {
padding: 5px;
height: 60px;
border: 1px dotted #000000;
overflow: auto;
background-color: #FFFFFF;
margin-bottom: 12px;
}
```