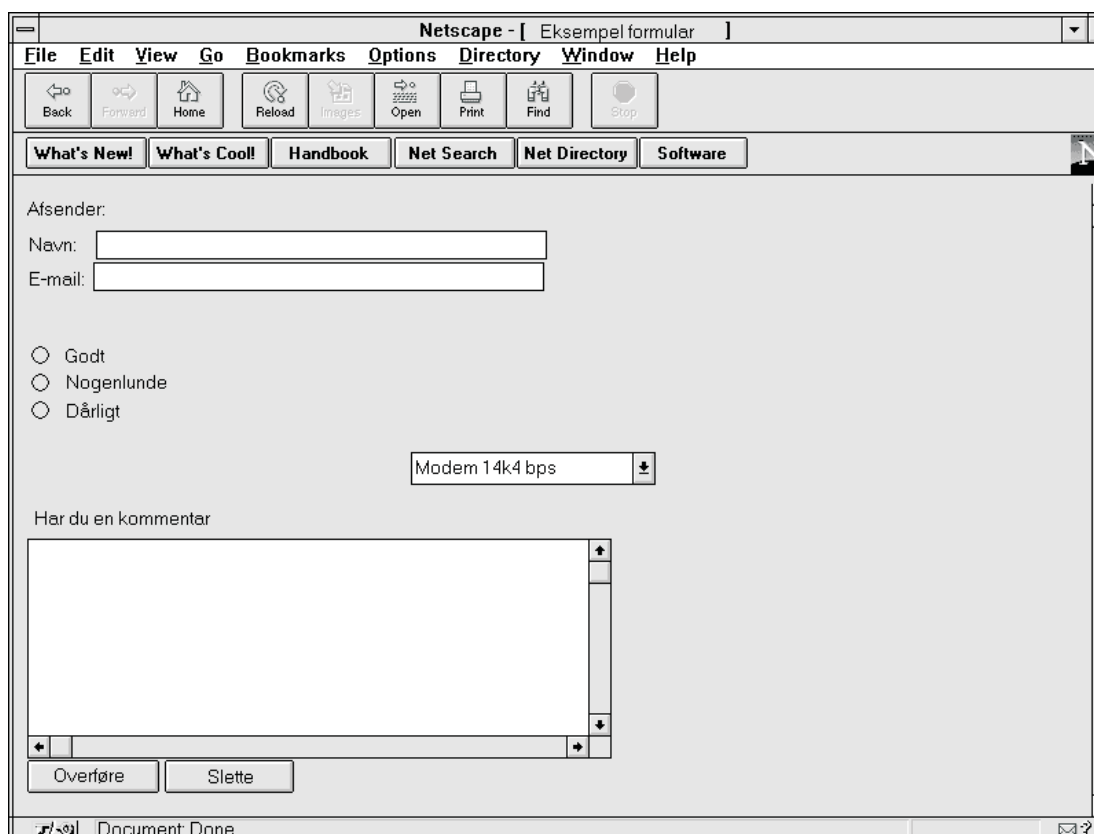


28,-

Internet / WWW & Co.

# HomePages

## - lav dem selv



Introduktion til HTML

Achim Schmidt

[www.KnowWare.dk](http://www.KnowWare.dk)

1. Udgave

## Acrobat Reader: tips ...

**F5/F6** åbner/lukker **Bogmærker**

I Menu **AVis** **sindstiller du, hvordan filen vises på skærmen**

**CTRL+0** = Hele siden **CTRL+1** = Originalstørrelse **CTRL+2** = Vinduesbredde

I samme menu kan du osse sætte: **Enkelt side**, **Fortløbende** eller **Fortløbende - Dobbelsider** .. Prøv, saa ser du forskellen.

### Navigation

**Pil til højre/venstre**: fremad/tilbage en side

**Alt+Pil Højre/Venstre**: som i Browser: fremad/tilbage

**Ctrl+ +** forstørrer og **Ctrl+ -** formindsker

<http://www.knowware.dk>

## Forord

World Wide Web, som vi i det følgende vil kalde WWW, spiller en stadig større rolle i Internettets verden. Så godt som alle Internet-providere, altså firmaer, der sælger tilslutningen til Internettet, tilbyder nu om dage en vis mængde plads på serveren, hvor man kan publicere sine egne informationer i form af en homepage

Men skal disse informationer kunne læses i WWW, må de være skrevet i et specielt kodesprog, der kaldes Hypertext Markup Language – forkortet HTML. Vi vil her se på dette sprogs grundelementer. Har du arbejdet dig igennem hæftet, burde du altså kunne skrive din egen homepage, din hjemmeside, med de informationer, som du måtte ønske at viderebringe – denne side kan du så sende til din provider via FTP, og så er du præsent i WWW.

Selvfølgelig findes der også andre muligheder i HTML – muligheder, der går langt ud over, hvad vi kan nå at se på indenfor dette hæfte. Hvis du kombinerer dine sider med et Common Gateway Interface (CGI), kan du således fremstille særdeles interaktive sider – altså sider, der reagerer på et input fra læseren. Det skal dog nævnes, at et sådant CGI-skript kræver tæt samarbejde med din server, da det skal køres fra serverens programmel – og altså også skal programmeres specielt for dette programmel.

Af pladshensyn vil vi som sagt ikke se nærmere på disse muligheder. Men hvis du kan læse tysk, finder du flere informationer om HTML og CGI samt en del links om dette tema på min homepage: <http://www.saar.de/~as/>

Nogle praktiske tips: Addresser i Internettet skrives med **Arial**, som du kan se i sidste afsnit. HTML-koderne samt den tekst, som de definerer, skrives med **Courier**. Det lønner sig at læse hæftet “foran skærmen” – du lærer unægteligt hurtigst, når du afprøver alle anvisninger på din egen computer.

En tak til min veninde, som var fuld af forståelse for min forfattergerning. Desuden takker jeg især Claudia Bender <[cb@wusel.saar.de](mailto:cb@wusel.saar.de)> og Holger Kenn <[kenn@mpi-sb.mpg.de](mailto:kenn@mpi-sb.mpg.de)>, som har læst hæftet, før det blev trykt, og bidrog med mange tips og forslag til, at det blev mere forståeligt. En tak går også til alle, der læste forud-versionen af hæftet på min homepage og bidrog med gode ideer. Endelig takker jeg alle HTML-forfattere, der har stillet deres samlede viden til rådighed på Nettet og løbende deltager i diskussionerne i Internet News.

Saarlouis, i maj 1996  
Achim Schmidt  
[as@Saar.DE](mailto:as@Saar.DE)

## Grundbegreber

### World Wide Web

World Wide Web, som vi som sagt vil kalde WWW, er ikke en traditionel del af Internettet som Email, altså elektronisk post, eller FTP, altså filoverførsel. WWW består af mange delkomponenter og koncepter. Disse koncepter sammenfattes i én brugeroverflade, der netop ved sin alsidighed er et fremragende værktøj.

### Todelt grundopbygning

Grundlaget i WWW er et Klient-Server-koncept. Det betyder, at en server eller webserver, så at sige en "tilbyder-computer", stiller informationer til rådighed i Nettet. Disse informationer modtages af klient-programmet i den anden ende af ledningen, hvor de omsættes til det ønskede skærbillede. En sådan klient kaldes også browser – på engelsk betyder dette ord, at man bladrer sig igennem noget.

En WWW-server er en computer, der normalt fungerer under et styresystem, der tillader multitasking. Det betyder, at systemet kan håndtere flere programmer og brugere samtidig. En sådan server har to opgaver. For det første stiller den informationer i form af HTML-dokumenter til rådighed; for det andet administrer den klient-programmernes aktiveringer af informationerne – det vil sige den sender dem på opfordring.

I den anden ende af ledningen finder du som sagt WWW-klienterne, altså de programmer, der kontakter den udvalgte server, modtager de ønskede informationer og fremstiller dem ved hjælp af de inkluderede koderinger på skærmen. Den type programmer findes for alle vanlige styresystemer – de hedder Netscape, Mosaic, Lynx, for blot at nævne nogle få.

Udseendet af en WWW-side på skærmen bestemmes altså udelukkende af det anvendte klientprogram. Desværre findes der store forskelle mellem programmerne – og det fører til, at et og samme dokument kan se meget forskelligt ud, alt efter, hvilken klient der trykker det på skærmen.

### WWW-dokumenter vises via HTML

De dokumenter, der anvendes i WWW, skrives normalt i HTML-kodesproget .HTML står for Hypertext Markup Language. Det er ikke noget programmeringssprog i vanlig forstand, men nærmere et tekstbeskrivelses- og layoutsprog for de informationer, der skal vises – altså et sprog, der ud over selve teksten indeholder koder, der bestemmer sidelayout og fremvisningsform.

Dette sprog blev udviklet specielt for WWW, hvor det er det klassiske dokumentformat. Det kendetegnes især ved de såkaldte hyperlinks – henvisninger til andre dokumenter, der kan ligge på andre computere hvorsomhelst i verden.

I grunden er HTML en underafdeling af SGML, Standard Generalized Markup Language, der blev udviklet af Dr. Charles F. Goldfarb. Den defineres i ISO-normen 8879.

Selve HTML-dokumenterne skrives i ren ASCII-tekst – medsamte koderne. De kan altså fremstilles og redigeres i en hvilken som helst teksteditor. Der findes dog nu om dage specielle HTML-editorer, programmer, der letter fremstillingen af HTML-dokumenter.

### **Kommunikationsproget: HTTP**

En speciel protokol, HyperText Transport Protocol (HTTP), definerer, hvordan WWW-server og -browser kan tale med hinanden. En sådan protokol er i grunden et regelsæt for udvekslingen af informationer. HTTP-protokollen er udformet relativt simpelt, for at trække minimal arbejdskraft fra serveren. Kommunikationen forudsætter en yderligere protokol, der definerer den basale forbindelse computerne imellem. I Internettet anvender man TCP/IP (Transmission Control Protocol / Internet Protocol)-protokollen.

En kommunikation via HTTP er bygget op på følgende måde:

- Via TCP/IP etableres forbindelse til den ønskede server,
- hvorpå de ønskede informationer bestilles med en request.
- Serveren reagerer med en response, et svar, idet den sender de ønskede informationer.
- Efter dataoverførslen afbrydes forbindelsen.

Overførslen gennemføres principielt med 8bit – altså volder en overførsel af binærfiler ingen problemer.

### **Ensformet adressering: URI**

Forkortelsen URI (Uniform Resource Identifier) står for et bredt grundkoncept, der definerer en ensformig adressering af vilkårlige ressourcer i Internettet. En sådan ressource er et objekt, som kan ligge hvorsomhelst i Nettet. Det kan være en fil, en database, en newsartikel eller en Telnet-session på en bestemt computer, for blot at nævne nogle eksempler.

Dette grundkoncept definerer hele filosofien bag WWW: alt, hvad der eksisterer, kan adresseres og fremvises.

Dette fleksible koncept betyder, at WWW ikke bare tilbyder isolerede løsninger for enkeltproblemer, men tværtimod er en samling af alle aktuelle Internet-tjenester – og på den måde er det en slags universalinstrument for alle Internettets muligheder.

Tilbage til den tekniske beskrivelse: URI's underdeles i URNs (Uniform Resource Names) og URLs (Uniform Resource Locators). I praksis vil du dog formodentlig kun møde URLs i WWW. En sådan URL er sat sammen af en protokoldel og en almen del. Protokoldelen angiver, hvilken tjeneste eller hvilket protokol i Internettet det aktuelle objekt benytter – for eksempel HTTP (Hypertekst), FTP (filoverførsel), TELNET (arbejde på en anden computer i Nettet) eller andet. Den almene del indeholder objektets adresse.

## HyperText Markup Language (HTML)

HTML er altså standardsproget for dokumenter i WWW. Denne standard overvåges af en arbejdsgruppe, som alt efter behov definerer den på ny. Hidtil var det version 2.0, der var den vedtagne form. Først i maj 1996 blev en ny standard vedtaget under nummeret 3.2. Version 3.0 var et forslag; men i mellemtiden havde flere klientprogrammer indført programspecifikke koder – blandt dem Netscape, der for tiden (september 1996) har en markedsandel på cirka 75%. Den nye standard, HTML 3.2, understøtter så godt som alle de udvidelser, der blev tilføjet til version 2.0 af Netscape, Microsoft Explorer og andre. Det er der ikke noget overraskende i, al den stund disse firmaer deltog i arbejdet med defineringen af den nye standard. Dette hæfte blev skrevet, før den nye standard trådte i kraft – men alle her viste koder følger denne standard.

HTML er ikke noget vanligt programmeringssprog. Det er nærmere et “tekstlayout-sprog”, altså et sprog, der indeholder visse bestanddele eller koder, som definerer en sides udseende. Desuden har sproget den specielle evne at kunne forarbejde henvisninger til andre dokumenter, de såkaldte hyperlinks. Faktisk var det sidste den egentlige mening med dette sprog, da det blev udviklet i videnskabelige cirkler.

**Et hyperlink er en henvisning til et andet dokument eller en ressource i Internettet, der er fast bundet til det første dokument. Klikker man på et sådant hyperlink, aktiveres det andet dokument eller ressourcen.**

### Et HTML-dokumentets struktur

HTML-dokumenter bygges op med en klart defineret struktur. De er delt op i et dokumenthoved, som på engelsk hedder Head, og en brødtekst, på engelsk Body.

I dokumenthovedet finder du de koder, der gælder for hele dokumentet, for eksempel dets titel. Brødteksten, *Body*, indeholder foruden selve teksten, altså informationerne, også de HTML-koder, der sørger for layoutet og definerer hyperlink'ene. Disse HTML-koder har altså en styrende funktion, mens den egentlige tekst findes mellem dem som klartekst.

Et HTML-dokument er et rent ASCII-dokument, der indeholder teksten samt de koder, der definerer dens udseende. Altså kan et sådant dokument fremstilles med en hvilken som helst ASCII-editor. Som sagt findes der dog også specielle HTML-editorer, der letter arbejdet med layoutet ved at tilbyde de ønskede anvisninger i menuer, der automatisk fremstiller de nødvendige koder. Desuden er der på det sidste fremkommet programmer, der direkte viser HTML-kodernes resultat på skærmen – et eksempel er Adobe's PageMill. I disse programmer undgår du enhver berøring med HTML-koderne.

Alligevel burde du stifte bekendtskab med grundlaget for HTML-sproget. Kun på denne måde kan du nemlig checke dine HTML-dokumenter nøjere.

### HTML-koders struktur

Også HTML-koderne har en simpel og klart defineret struktur. De kaldes for tags, det engelske ord for kendemærke. Med få undtagelser er de sat sammen af et starttag og et sluttag.

Et HTML-tag fungerer som en slags kommando. Det indrammes af spidsklammer (<>). *Starttag'et* indeholder selve koden mellem klammerne; *sluttag'et* derudover en skråstreg, en slash / før selve koden.

En HTML-kode har altså principielt følgende udseende:

```
<KODE> ... TEKST ... </KODE>
```

Som du ser, sættes teksten mellem de start- og sluttags, der definerer, hvordan den skal formateres. Start- og sluttag har altså principielt kun betydning for den tekst, der står mellem dem.

Nogle få tags fungerer uden sluttag, fordi deres logiske funktion er umiddelbart indlysende. I det følgende vil vi specielt gøre opmærksom på, om en kode tilhører denne gruppe.

### Grundlaget i et HTML-Dokument

Grundlaget i et HTML-dokument defineres ved nogle få koder, der skal findes i ethvert HTML-dokument.

Først skal nævnes tag'et `<HTML>`: det omfatter hele dokumentet og kendetegner det som en fil i HyperText Markup Language. Dets syntaks lyder sådan:

```
<HTML> ... dokument ... </HTML>
```

Et eksempel er:

```
<HTML>Flere HTML-koder</HTML>
```

Næst på dagsordenen står tag'et `<HEAD>`. Denne kode betegner dokumenthovedet med dets start- og sluttag. Alt, hvad der indesluttet af dette tagpar, relaterer til hele det følgende dokument. Syntaksen lyder:

```
<HEAD> ... Anvisninger ... </HEAD>
```

Et eksempel:

```
<HTML>
<HEAD>
Hoved-Anvisninger
</HEAD>
</HTML>
```

Det tredje grundelement i strukturen er tag'et `<BODY>`. Denne kodes start- og sluttag indbefatter hele brødteksten – det vil sige hele den tekst, der skal fremtræde på skærmen, samt de HTML-koder, der angiver, hvordan den skal fremtræde.

Syntaksen er:

```
<BODY> ... Broedtekstanvisninger ... </BODY>
```

Et eksempel:

```
<HTML>
<HEAD>
... Hoved-Anvisninger ...
</HEAD>
<BODY>
... Broedtekstanvisninger ...
</BODY>
</HTML>
```

I dette eksempel har du grundopbygningen af et HTML-dokument. Denne opbygning er altid den samme. I det følgende skal vi se på de enkelte anvisninger, der kan benyttes i dokumenthovedet og brødteksten.

Hvis du forresten undrer dig over stavemåden *Broedtekst*, så er forklaringen, at HTML-sproget skal forstås af alle mulige forskellige computere. Men da forskellige computere bruger forskellige koder for de særlige tegn, der findes i forskellige lande, må man enten omskrive disse særtegn, som vi har gjort det i dette eksempel; eller også må man benytte en speciel HTML-kode for disse særtegn, der forstås af alle computere. Senere i hæftet (på side 19) finder du en liste over disse særtegn, deres internationale betegnelse og deres HTML-kode.

## Basale HTML-koder

Langt de fleste HTML-filer er, som sagt, rene ASCII-filer, der indeholder HTML-koder, som også skrives i ASCII. Disse koder bestemmer tekstens udseende og definerer henvisningerne. Lad os nu se på de grundlæggende styringskoder.

### Overskrifter

Overskrifter kendetegnes i HTML med `<H>`-tag'et, som står for *Headline* – det engelske ord for overskrift. De inddeles i flere niveauer, der kendetegnes ved hvert sit nummer, alt efter prioriteten.

HTML kender til seks overskrifter, der indstilles med tag'ene `<H1>...</H1>` til `<H6>...</H6>` oppefra og nedefter – altså betegner `<H1>` det øverste niveau.

Hvis du sætter det følgende dokument op og læser det i din browser, vil du forstå, hvad der menes:

#### Eksempel Overskrifter:

```
<HTML>
<HEAD><TITLE>Overskrifter i HTML</TITLE></HEAD>
<BODY>
<H1>Overskrift Nr. 1</H1>
<H2>Overskrift Nr. 2</H2>
<H3>Overskrift Nr. 3</H3>
<H4>Overskrift Nr. 4</H4>
<H5>Overskrift Nr. 5</H5>
<H6>Overskrift Nr. 6</H6>
</BODY>
</HTML>
```

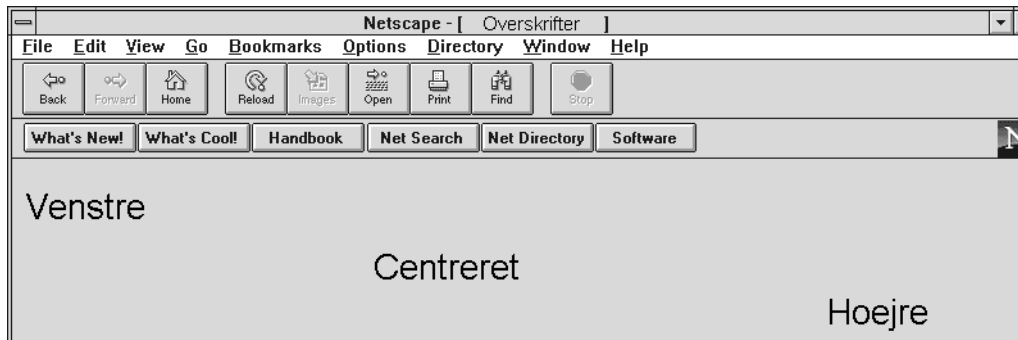
Som du ser, indrammes den tekst, der skal vises, af tag'ene `<Hx> ... </Hx>`.

Lige som ethvert andet tag kan det blotte `<H>`-tag udvides med flere angivelser, de såkaldte attributer, som indføres i det egentlige tag indenfor dets spidsklammer. Et sådant attribut tilføjer mere præcise informationer om fremvisningen af det aktuelle element. I vores eksempel bestemmer attributet `align` sammen med en af de tre retningsbetegnelser, hvor i billedet overskriften skal placeres:

- `align=left` sætter overskriften til venstre,
- `align=center` sætter den i midten,
- `align=right` sætter den til højre.

## Eksempel Overskrifter 2:

```
<HTML>
<HEAD><TITLE>Overskrifter</TITLE></HEAD>
<BODY>
<H1 align=left>Venstre</H1>
<H1 align=center>Centreret</H1>
<H1 align=right>Højre</H1>
</BODY>
</HTML>
```



## Linieskift

Når ASCII-tekst fremvises i HTML-dokumenter, tages der principielt intet hensyn til mellemrum og linieskift. Derfor er det nødvendigt at indføje en kode, der får browseren til at indsætte et linieskift.

Den rette HTML-kode til dette formål er tag'et `<BR>` – et af de førnævnte tags, der ikke behøver et afsluttende tag.

## Eksempel Linieskift:

```
<HTML>
<HEAD><TITLE>Linieskift</TITLE></HEAD>
<BODY>
<H1>Dette er et eksempel paa linieskift</H1>
Denne tekst staar i grunddokumentet i en linie,
mens denne tekst staar i naeste linie. Men som du
ser, indfoejes der udelukkende automatiske linieskift.
Hvis du oensker at indsaette et fast linieskift, maa
du anvende det korrekte tag, som her: <BR>
Som du ser, foerte det til et linie-<BR>
skift, lige som ved denne orddeling.
</BODY>
</HTML>
```



## Undgå et linieskift

Hvis du nu ønsker at *undgå* et lineskift, anvender du tag'et `<no>br</no>`. Sætter du et helt afsnit ind i dette tag, kan du i dette afsnit kun indsætte linieskift ved hjælp af tag'et `<br>`.

### Eksempel Undgå linieskift

```
<HTML>
<HEAD><TITLE>Undgaa et linieskift</TITLE></HEAD>
<BODY>
<NOBR>
Denne tekst vises uden automatiske linieskift. Resultatet er, at
linierne kan opnaa en saadan laengde, at de straekker sig ud over
skaermens rand og kun kan goeres synlige ved at scrolle dem.<br>Vil du
fremtvinge et linieskift i et saadant afsnit, maa du indsaeette det
specielle tag <br>.
</NOBR>
</BODY>
</HTML>
```

## Afsnit

Et afsnit markeres i et HTML-dokument ved tag'et `<P>...</P>`. Afsnittets egentlige tekst står som sædvanlig mellem start- og sluttag. Efter afsnittets afslutning indføres der automatisk et linieskift samt en ekstra linie.

### Eksempel Afsnit:

```
<HTML>
<HEAD><TITLE>Afsnit</TITLE></HEAD>
<BODY>
<P>
Denne tekst staar i foerste afsnit og indeholder et<BR>
linieskift. Efter afsnittet indfoejes automatisk et linieskift samt en
ekstra lnie.
</P>
<P>
Og her ser du naeste afsnit, som optisk er klart adskilt fra foerste
afsnit.
</P>
</BODY>
</HTML>
```

Lige som overskrifter kan også afsnit formateres ved justering. Også her anvendes attributet `align`.

### Eksempel Justerede afsnit

```
<HTML>
<HEAD><TITLE>Justerede afsnit</TITLE></HEAD>
<BODY>
<p align=right>
Dette afsnit vises hoejrejusteret, da det blev saadan defineret med et
tag.</p>
<p align=center>
Her har vi aendret attributet. Altsaa vises dette afsnit
centreret.</p>
<p>Saettes et afsnits-tag uden attribut, vises det tilhoerende afsnit
automatisk venstrejusteret.</p>
</BODY>
</HTML>
```

### Centreret visning

Hvis du vil vise flere elementer og afsnit *centreret*, kan du gøre det ved hjælp af tag'et `<center> ... </center>`. Al den tekst, der indeslutes af start- og slut-tag'et, vil blive vist centreret i browseren.

### Eksempel Centreret visning

```
<HTML>
<HEAD><TITLE>Centreret visning</TITLE></HEAD>
<BODY>
<CENTER>
<H2>Centreret visning</H2>
<P>
Denne tekst skal i sin helhed vises centreret, med samtlige
overskrifter og afsnit.
</P>
<H3>Andet afsnit</H3>
<P>
Ogsaa dette andet afsnit skal vises centreret i browseren.
</P>
</CENTER>
<H3>Tredje afsnit</H3>
<P>
Tredje afsnit derimod er ikke centreret, da slut-tag'et for
centreringen staar umiddelbart foer det.
</P>
</BODY>
</HTML>
```

## Listeformater

HTML-sproget byder på flere muligheder, hvis du ønsker at opstille lister. Vi vil se på dem i rækkefølge.

### Simple lister

Den simpleste form for liste er netop en *simpel liste*. Dens enkelte punkter stilles unummereret op efter hinanden. En sådan liste defineres med tag'et `<ul> ...</ul>`, som omslutter alle listens punkter. Indenfor dette omsluttende tag kendetegnes listens enkelte punkter hvert for sig med tag'et `<li>`, som ikke behøver noget slut-tag.

### Eksempel Simple liste

```
<HTML>
<HEAD><TITLE>Simple lister</TITLE></HEAD>
<BODY>
<P>
Dette eksempel viser en unummereret eller simpel liste:
<br>
<ul>
<li>Dette er listens foerste punkt,
<li>dette dens anden
<li>og dette dens tredje.
</ul>
</BODY>
</HTML>
```

Ud over simpelt at angive listens elementer kan du også kendetegne dens enkelte punkter ved en bestemt markering. Det gør du ved hjælp af attributet `type`. Der er tre mulige værdier for dette attribut. De ser således ud:

- `<ul type=circle>` markering ved en cirkel
- `<ul type=square>` markering ved en firkant
- `<ul type=disc>` markering ved en skive

### Nummererede lister

*Nummererede lister* anvendes, når man vil opremse informationer og argumenter. En sådan nummereret liste defineres i HTML med tag'et `<ol> ... </ol>`, der ligeledes omslutter hele listen. De enkelte punkter betegnes lige som ved en simpel liste med tag'et `<li>`.

### Eksempel Nummererede lister

```
<HTML>
<HEAD><TITLE>Nummereret liste</TITLE></HEAD>
<BODY>
<P>Dette eksempel fremviser en nummereret liste:</P>
<OL>
<LI>Foerste punkt på listen forsynes med "1." i starten
<LI>Andet punkt på listen forsynes med "2." i starten
<LI>Tredje punkt på listen forsynes med "3." i starten
</OL>
</BODY>
</HTML>
```

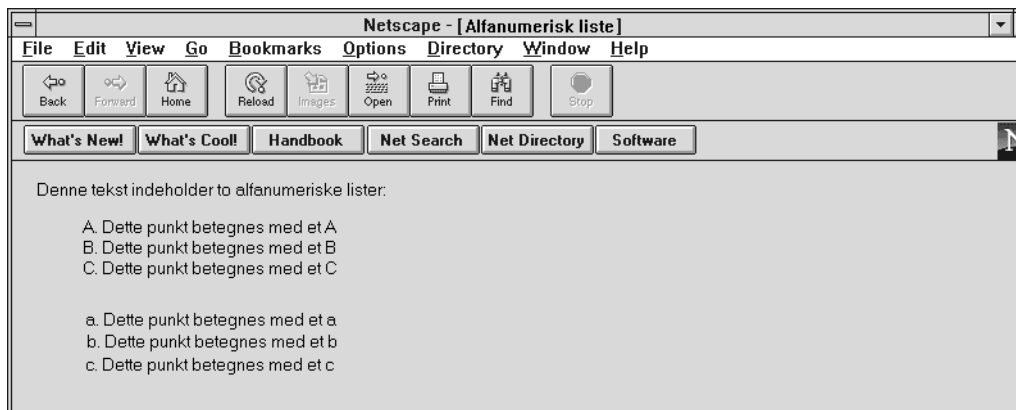
Denne almindelige nummerering bliver indsat automatisk, når du anvender tag'et `</ol>`. Ønsker du en anden form for nummerering, kan du opnå det ved et attribut, som indføres efter tag'et `<ol>`.

Du har følgende muligheder:

- `<ol type=i>` markering med små romertal
- `<ol type=I>` markering med store romertal
- `<ol type=a>` markering med små bogstaver
- `<ol type=A>` markering med store bogstaver

### Eksempel Alfanumerisk liste

```
<HTML>
<HEAD><TITLE>Alfanumerisk liste</TITLE></HEAD>
<BODY>
<P>
Denne tekst indeholder to alfanumeriske lister:
<ol type=A>
<li>Dette punkt betegnes med et A
<li>Dette punkt betegnes med et B
<li>Dette punkt betegnes med et C
</ol>
</P>
<ol type=a>
<li>Dette punkt betegnes med et a
<li>Dette punkt betegnes med et b
<li>Dette punkt betegnes med et c
</ol>
</BODY>
</HTML>
```



Selvfølgelig kan du også begynde en liste med et andet nummer end 1. Det gør du ved hjælp af attributet `start`, som tilføjes til tag'et `<ol>`. En liste, der indledes med tag'et `<ol start=4>`, vil altså vise sit første punkt med nummeret 4.

## Kataloglister

*Kataloglister* kan for eksempel anvendes til opstillinger af fillister, directories eller enkeltelementer. Kataloglister defineres ved tag'et `<dir> ... </dir>`. De enkelte led betegnes atter med tag'et `<li>`.

### Eksempel katalogliste

```
<HTML>
<HEAD><TITLE>Katalogliste</TITLE></HEAD>
<BODY>
Dette er et eksempel på en katalogliste:
<BR>
<DIR>
<li>foerste punkt
<li>andet punkt
<li>tredje punkt
</DIR>
</BODY>
</HTML>
```

## Glossar

Et glossar er en anden form for liste: den indeholder på den ene side betegnelser, der skal defineres, og på den anden side disse definitioner.

Vil du opstille et sådant glossar i HTML, anvender du tag'et `<dl> ... </dl>`, der skal omslutte hele glossarlisten. Til de enkelte punkter i glossaret behøver du to yderligere tags. Det ene, `<dt> ... </dt>`, omslutter den betegnelse, der skal defineres. Det andet, `<dd> ... </dd>`, betegner selve definitionen. Det bliver klarere i det følgende eksempel:

### Eksempel Glossar

```
<HTML>
<HEAD><TITLE>Glossar</TITLE></HEAD>
<BODY>
I dette dokument vil vi opstille et glossar.
<br>
<dl>
<dt> 1. betegnelse </dt>
<dd>definition af foerste betegnelse</dd>
<dt> 2. betegnelse </dt>
<dd> definition af anden betegnelse </dd>
</dl>
</BODY>
</HTML>
```

## Kommentarer

Ud over de egentlige styreanvisninger kan du selvfølgelig også indsætte *kommentarer* i dit HTML-dokument. Sådanne kommentarer omsluttes af tag'et `<!-- -->`.

Det ser ud som følger:

```
<!-- Dette er en kommentar -->
```

En sådan kommentar ignoreres af browseren, det vil sige den bliver ikke vist. Men du kan se den, hvis du åbner en HTML-fil med en normal teksteditor.

## Yderlige anvisninger til formatering

Ved siden af de allerede nævnte styreanvisninger kender HTML også andre anvisninger, der regulerer visningen af den egentlige tekst. Det er indirekte anvisninger, som ikke angiver, hvordan den tekst, som de relaterer til, skal vises, men nærmere definerer, hvilken teksttype denne tekst hører til. Hvordan den så skal vises, det bestemmes af det enkelte browserprogram.

### Citater

Udsagn, der citeres fra andre kilder, kendetegnes normalt ved en speciel typografi. I et HTML-dokument opnår du det ved at bruge tag'et `<blockquote> ... </blockquote>`.

Som altid indleder starttag'et et citat, mens slutttag'et afslutter det. Den omsluttede tekst vises af den anvendte browser på den måde, som den anvender til citater.

### Eksempel citat

```
<HTML>
<HEAD><TITLE>ZITATE</TITLE></HEAD>
<BODY>
Og der loed en roest:
<blockquote>Der blive lys ...</blockquote>
</BODY>
</HTML>
```

Udover tag'et `<blockquote>` kan der også benyttes tag'et `<cite> ... </cite>` for citater.

### Adresser

Adresser betegnes ligeledes ved et specielt tag, nemlig `<address> ... </address>`. Teksten mellem start- og slutttag vises på den enkelte browsers adresse-måde – for det meste er det kursiveret.

### Eksempel Adresse

```
<HTML>
<HEAD>
<TITLE>Adresser</TITLE>
</HEAD>
<BODY>
Du kan finde os under foelgende adresse:<BR>
<ADDRESS>
Moenstermand GmbH<BR>
Moenstergade. 10<BR>
66666 Moensterby<BR>
</ADDRESS>
</BODY>
</HTML>
```

## Præformateret tekst

Hvis du ønsker at vise et afsnit præcist som du selv har skrevet det, altså med samtlige linieskift, mellemrum og styretegn, kan du gøre det ved hjælp af tag'et `<PRE> ... </PRE>`.

Teksten mellem start- og sluttag vil i browseren blive vist på præcist samme måde, som den blev skrevet. Du må dog være opmærksom på, at du ikke kan fastlægge, hvilken skrifttype den enkelte browser anvender. En præformateret tekst vises principielt i ikke-proportional skrift, så at sige skrivemaskine-skrift. Men den enkelte browsers indstilling bestemmer, hvilken skrifttype der anvendes.

### Eksempel præformateret tekst

```
<HTML>
<HEAD>
<TITLE>Præformateret tekst</TITLE>
</HEAD>
<BODY>
Den efterfølgende tekst skal vises præformateret:<br>
<PRE>
Dette er et eksempel paa præformateret tekst.
```

```
Den gengives med alle mellemrum og
linieskift.
</PRE>
</BODY>
</HTML>
```

## Eksempler og kildetekst

*Eksempler* og *kildetekst* behandles på lignende måde som præformateret tekst – de vises af browseren med ikke-proportional skrift.

*Eksempler* vises ved hjælp af tag'et `<tt> ... </tt>`.

*Kildetekst* er tekst, der indeholder HTML-tekst i ubehandlet form, altså med samtlige koder.

Den type tekst kendetegnes ved tag'et `<code> ... </code>`.

## Tekstfremhævelser

I visse situationer findes der en grund til at fremhæve et tekstafsnit visuelt på en eller anden måde. HTML-sproget kan også her tilbyde valget mellem flere muligheder.

### Fed tekst

Den mest almindelige måde at fremhæve en tekst på er at skrive den **fed**. Det opnås ved hjælp af tag'et `<strong> ... </strong>`. Tekst, der omslutes af dette tag, vises af browseren som fed.

### Eksempel fed tekst

```
<HTML>
<HEAD>
<TITLE>Fed tekst</TITLE>
</HEAD>
<BODY>
Denne tekst indeholder <strong>med fed satte</strong> elementer.
</BODY>
</HTML>
```

Forøvrigt findes der endnu en måde at sætte tekst som fed – nemlig ved hjælp af tag'et `<b> ... </b>`

De nu følgende formateringsanvisninger indsættes på samme måde som i det foranstående eksempel.

### Kursiveret

Tekst, der omslutes af tag'et `<i> ... </i>`, vises *kursiveret*.

### Understreget

Nogle browsere kan ud over de nævnte formateringer også vise understreget tekst Det opnår du ved hjælp af tag'et `<u> ... </u>`.

### Blinkende tekst

Vil du fremhæve en tekst yderligere, kan du få den til at blinke ved hjælp af tag'et `<blink> ... </blink>`. Men vær varsom – effekten mister hurtigt sin virkning og bliver let til en irritationskilde.

### Sænket skrift

Vil du sænke tekst indenfor en linie, for eksempel til en anmærkning, kan du gøre det med tag'et `<sub> ... </sub>`

### Hævet skrift

Tilsvarende kan du også definere en hævnig af teksten, for eksempel til eksponentialer. Det gør du ved hjælp af tag'et `<sup> ... </sup>`.

### Skriftstørrelse

Du har selvfølgelig også indflydelse på skriftstørrelsen ved hjælp af en HTML-kode. Der findes ialt 7 forskellige skriftstørrelser – standardstørrelsen blandt dem er 3.

Skriftstørrelsen defineres med tag'et `<font> ... </font>` sammen med attributet `size` – sidstnævnte skal forsynes med en værdi mellem 1 og 7. Modsat overskriftsnummereringen er 1 her den mindste og 7 den størst5e værdi.

### Eksempel skriftstørrelse

```
<HTML>
<HEAD>
<TITLE>Skriftstoerrelse</TITLE>
</HEAD>
<BODY>
Dette er normalskrift<BR>
<font size=2>Skriftstoerrelse 2</font><br>
<font size=5>Skriftstoerrelse 5</font><br>
<font size=7>Skriftstoerrelse 7</font><br>
</BODY>
</HTML>
```

Udover at sætte en fast værdi kan du også indsætte en relativ værdi – her udgår du fra en basisstørrelse, der defineres ved tag'et `<basefont size=x>`, og modificerer den ved angivelse af +(plus) og -(Minus).

### Eksempel relativ skriftstørrelse

```
<HTML>
<HEAD>
<TITLE>Relativ skriftstørrelse</TITLE>
</HEAD>
<BODY>
<basefont size=2>
Denne tekst har størrelsen 2,<br>
<font size=+2>mens denne er to klasser større<br>
</font>
</BODY>
</HTML>
```

### Større skrift

Ønsker du at vise en tekstpassage større end den omgivende tekst, kan du gøre det med tag'et `<big> ... </big>`.

### Mindre skrift

Tilsvarende kan du vise en tekstpassage mindre ved hjælp af tag'et `<small> ... </small>`.

### Farvet skrift

Selvfølgelig kan du også vise tekst i forskellige farver. Egentlig hører dette punkt til i et senere afsnit i hæftet, hvor vi vil beskæftige os nærmere med definitionen af farve i HTML – se side XX. Men da skriftfarven defineres ved et attribut til tag'et `<font>`, vil jeg lige nævne det.

Farver tildeles til tekstafsnit ved hjælp af tag'et `<font>` med attributet `color`, der tildeles den ønskede farveværdi.

### Eksempel Farvet tekst

```
<HTML>
<HEAD>
<TITLE>>Farvet tekst</TITLE>
</HEAD>
<BODY>
<font color=#ffffff>Hvid tekst</font><br>
<font color=#c00000>Roed tekst</font><br>
</BODY>
</HTML>
```

## Specialtegn i HTML-dokumenter

Mange sprog har specialtegn, der skal defineres på en særskilt måde i HTML. På de følgende sider finder du en tabel, der anfører specialtegnene og viser den måde, som de skal skrives på i HTML.

Ønsker du, at de danske særtægn skal gengives korrekt af den anvendte browser, skal du altså sætte de her viste skrivemåder i din fil, istedet for den vante stavemåde.

### Eksempel specialtegn

```
<HTML>
<HEAD>
<TITLE>Specialtegn</TITLE>
</HEAD>
<BODY>
&Oslash;nsker du at vise de danske særtægn korrekt i en HTML-
tekst, må du tage disse gådeagtige koder til hjælp;lp.
</BODY>
</HTML>
```



Specialtegn	Int. betegnelse	HTML-skrivemåde
Æ	capital AE diphthong (ligature)	&AElig;
Á	capital A, acute accent	&Aacute;
Â	capital A, circumflex accent	&Acirc;
À	capital A, grave accent	&Agrave;
Å	capital A, ring	&Aring;
Ã	capital A, tilde	&Atilde;
Ä	capital A, dieresis or umlaut mark	&Auml;
Ç	capital C, cedilla	&Ccedil;
Ð	capital Eth, Icelandic	&ETH;
É	capital E, acute accent	&Eacute;
Ê	capital E, circumflex accent	&Ecirc;
È	capital E, grave accent	&Egrave;
Ë	capital E, dieresis or umlaut mark	&Euml;
Í	capital I, acute accent	&Iacute;
Î	capital I, circumflex accent	&Icirc;
Ì	capital I, grave accent	&Igrave;
Ï	capital I, dieresis or umlaut mark	&Iuml;
Ñ	capital N, tilde	&Ntilde;
Ó	capital O, acute accent	&Oacute;
Ô	capital O, circumflex accent	&Ocirc;
Ò	capital O, grave accent	&Ograve;
Ø	capital O, slash	&Oslash;
Ö	capital O, tilde	&Otilde;

Ö	capital O, dieresis or umlaut mark	&Ouml;
Þ	capital THORN, Icelandic	&THORN;
Ú	capital U, acute accent	&Uacute;
Û	capital U, circumflex accent	&Ucirc;
Û	capital U, grave accent	&Ugrave;
Ü	capital U, dieresis or umlaut mark	&Uuml;
Ý	capital Y, acute accent	&Yacute;
á	small a, acute accent	&aacute;
â	small a, circumflex accent	&acirc;
æ	small ae diphthong (ligature)	&aelig;
à	small a, grave accent	&agrave;
å	small a, ring	&aring;
ã	small a, tilde	&atilde;
ä	small a, dieresis or umlaut mark	&auml;
ç	small c, cedilla	&ccedil;
é	small e, acute accent	&eacute;
ê	small e, circumflex accent	&ecirc;
è	small e, grave accent	&egrave;
ð	small eth, Icelandic	&eth;
ë	small e, dieresis or umlaut mark	&euml;
í	small i, acute accent	&iacute;
î	small i, circumflex accent	&icirc;
ì	small i, grave accent	&igrave;
ï	small i, dieresis or umlaut mark	&iuml;
ñ	small n, tilde	&ntilde;
ó	small o, acute accent	&oacute;
ô	small o, circumflex accent	&ocirc;
ò	small o, grave accent	&ograve;
ø	small o, slash	&oslash;
õ	small o, tilde	&otilde;
ö	small o, dieresis or umlaut mark	&ouml;
ß	small sharp s, German (sz ligature)	&szlig;
þ	small thorn, Icelandic	&thorn;
ú	small u, acute accent	&uacute;
û	small u, circumflex accent	&ucirc;
ù	small u, grave accent	&ugrave;
ü	small u, dieresis or umlaut mark	&uuml;
ý	small y, acute accent	&yacute;
ÿ	small y, dieresis or umlaut mark	&yuml;
©	copyright	&copy;
®	registered trademark	&reg;
<	left angular bracket	&lt;
>	right angular bracket	&gt;
&	ampersand	&amp;

## Henvisninger i HTML-dokumenter

Henvisninger er kernen og hovedkendetegnet i HTML-dokumenter. Deres opgave er at definere tværforbindelser til andre dokumenter, der er lagret på en hvilken som helst anden WWW-server, eller til et andet sted i samme dokument.

Disse tværforbindelser, der også kaldes hyperlinks, er, hvad der gør hypertextdokumenter virkelig spændende – ved deres hjælp kan du springe fra et dokument til et andet, uden hensyn til, hvor i verden dette andet dokument måtte befinde sig.

Vil du indsætte en henvisning til et andet dokument, må du altså definere, hvilket dokument der skal aktiveres. Det gør du ved hjælp af tag'et `<A> ... </A>`. A står her for 'anchor', altså forankringspunkt.

Og nu bliver sagen lettere kompliceret. Du kan nemlig, som sagt, ikke blot aktivere et helt dokument på en anden server, men også et andet sted i det foreliggende dokument eller et bestemt sted i et andet dokument på samme server – ud fra web'ets grundsætning: "Alt, hvad der kan adresseres, kan også vises". Derfor skal tag'et `<A> ... </A>` i hvert enkelt tilfælde forsynes med et attribut, der tydeliggør, om der her skal aktiveres et *forankringspunkt* eller et *målpunkt*. Lad os se nærmere på det. Og ingen panik – i praksis er sagen nemmere end det umiddelbart ser ud til.

### Muliggøre en henvisning

Vil du aktivere et andet dokument, gør du det ved hjælp af tag'et `<A> ... </A>` med det tilføjede attribut `HREF`. Denne forkortelse står for **H**ypertext **R**Eference og udtrykker en reference til en anden tekst. Som værdi tildeles dette attribut adressen af den tekst, der skal aktiveres. Er det et dokument på en anden server, angives hele dette dokumentets adresse – dets URL.

Lad os antage, at du på din hjemmeside vil henvise til min hjemmeside, der har URL'en `http://www.saar.de/~as/`. Så må du anvende følgende kode i din tekst:

```
<A HREF="http://www.saar.de/~as/">Homepage Achim Schmidt</A>
```

Teksten "Homepage: Achim Schmidt" kendetegnes som hyperlink, når den vises i en browser – alt efter browserens indstilling betyder det en anden tekstfarve og/eller understregning. Klikkes der nu på denne tekst, bestiller browseren det pågældende dokument fra dets server.

Dokumentet sendes over til browser-computeren og vises på skærmen – i dette tilfælde er det min hjemmeside.

### Henvisninger indenfor samme server

Selvfølger kan du også aktivere et dokument på samme server som det første dokument. I dette tilfælde skal adresseringen dog formuleres lidt anderledes, som det fremgår af følgende eksempel.

Lad os antage, at vi i et dokument vil henvise til filen "kapitel9.html", der ligger på samme server, i Public-WWW-directory for brugeren as, altså Achim Schmidt.

Denne fil adresseres normalt som følger:

```
http://www.foo.bar/~as/kapitel9.html
```

Ligger nu det HTML-dokument, hvorfra du vil henvise til denne fil, også på serveren `www.foo.bar`, kan du aktivere den ønskede fil med følgende adressering:

```
<A HREF="/~as/kapitel9.html">Gaa til kapitel 9</A>
```

Efter attributet `HREF` ser du den del af URL'en, der betegner det egentlige dokument. Informationerne om protokolform og server undlades ganske enkelt. Lige som ved en

henvisning til et dokument på en anden server formateres teksten mellem `<A HREF ...>` og `</A>` sådan, at den kan klikkes på, hvorved den fil, som der henvises til, aktiveres.

### Definere et målpunkt

Men hvad nu, hvis du ønsker at definere et sted i samme dokument som målpunkt? Vel, du skal definere et *forankringspunkt*. Der gør du ved at tilføje attributet `NAME` til tag'et `<A > ... </A>`. Som værdi tildeler du attributet et reference-ord. Et hvilket som helst sted indenfor et dokument kan defineres som målpunkt. Skematisk bygges det op som følger:

```
<A NAME="m&aring;lpunkt1">Dette tekststed defineres som
m&aring;lpunkt1</A>
```

Sagt på en anden måde definerer du et tekststed som målpunkt, som forankringspunkt, ved at omslutte den tekst, der skal henvises til, med tag'et `<A NAME="reference-ord"> ... </A>`.

Vil du nu henviser til dette sted, skal du, lige som ved henvisninger til andre dokumenter, bytte tag'et `<A> ... </A>` med attributet `HREF`. Blot får attributet denne gang som værdi ikke en filadresse, men det engelske tegn for nummerering `#` med det tilføjede reference-ord.

### Eksempel Lokal henvisning

```
<HTML>
<HEAD>
<TITLE>
Lokal henvisning
</TITLE>
</HEAD>
<BODY>
Her kan du springe til n&oslash;jagtigt definerede steder i samme dokument:
<BR>
<BR>
<A HREF="#m&aring;lpunkt1">M&aring;lpunkt 1</A><BR>
<A HREF="#m&aring;lpunkt2">M&aring;lpunkt 2</A><BR>
<A HREF="#m&aring;lpunkt3">M&aring;lpunkt 3</A><BR>
<BR>
<A NAME="m&aring;lpunkt1">
Her st&aring;r den tekst, der h&oslash;rer til m&aring;lpunkt 1
</A>
<A NAME="m&aring;lpunkt2">
Her ser du den tekst, der passer til m&aring;lpunkt 2 passende Text
</A>
<A NAME="m&aring;lpunkt3">
Og her ser du teksten til m&aring;lpunkt 3
</A>
</BODY>
</HTML>
```

Desuden kan du som sagt også aktivere et forankringspunkt i et andet dokument på samme server – blot skal du i dette tilfælde først angive dokumentet og siden tilføje forankringspunktet som lige vist.

Vil du altså henviser til "henvis1" i det ovenfor nævnte dokument "kapitel9.html", definerer du først dette punkt ved `<A NAME="henvis1"> ...</A>` i dokumentet "kapitel9.html" – og sætter derpå henvisningen i det andet dokument med koden

```
<A HREF="/~as/kapitel9.html#henvis1">Gå til kapitel 9, henvis1</A>
```

## Henvisninger af speciel karakter

Udover de hidtil nævnte typer af henvisninger muliggør tag'et `<A HREF= ... > ... </A>` også andre former for henvisning, der har at gøre med andre filtyper eller protokoller. Lad os se nærmere på nogle af disse henvisningstyper.

### Aktivere en email

Du kan indsætte en henvisning i et HTML-dokument, der får browseren til at sende en email, altså et elektronisk brev. De fleste browsere er udstyret med en egen mailfunktion – en editor, hvori du kan skrive og arkivere eller sende en mail.

Vil du gøre det, skal du benytte tag'et `<A HREF= > ... </A>` – blot skal du her som værdi til attributet ikke anvende en URL, men kodeordet `mailto:` samt den email-adresse, som det færdige brev skal sendes til. Vil du sende en email til mig direkte fra en WWW-side, skal du altså benytte følgende tag:

```
<A HREF=mailto:as@saar.de>Mail til Achim Schmidt</A>
```

`as@saar.de` er altså min Email-adresse. Teksten "Mail til Achim Schmidt" vises atter som et link, som man kan klikke på. Derved aktiveres browserens email-funktion, der åbner et nyt brev, som allerede er udstyret med den adresse, der står bag attributet `mailto:`

### Visning af billeder

Ud over HTML-filer kan der selvfølgelig også vises billeder og grafik, da jo som bekendt en hvilken som helst type filer kan aktiveres ved hjælp af en henvisning. Blot skal billedet været formateret som GIF eller JPG, da browsere kun kan tolke disse to billedformater. Faktisk er det ikke engang alle browsere, der kan tolke JPG-formatet, så hvis du vil være helt sikker, bør du kun benytte GIF-filer. Lad os nu antage, at du vil aktivere billedet `bild1`.med et enkelt klik. Altså sætter du følgende tag ind i dit dokument:

```
<A HREF="bild1.gif">Vis billede 1</A>
```

Vi går her ud fra, at billedet ligger i samme directory på din computer som den egentlige HTML-fil. Ligger det i et andet directory, skal du tilføje stien, sådan som vi så det i afsnittet om "Henvisninger indenfor samme server" (se side XX). Ligger billedet på en anden server, skal du angive dets fuldstændige URL.

## Adressere andre ydelser

Ved siden af Email og adresseringen af grafiske dokumenter eller lignende kan du også aktivere andre protokolformer i Internettet. Hvis vi vil forstå, hvordan det er muligt, skal vi lige se en gang til på strukturen i attributet **HREF**.

Dette attribut er principielt sat sammen af to dele, der adskilles ved et kolon og to skråstreg. Første del betegner protokolformen, mens anden del angiver adresseringen. Komplet ser sagen altså ud som følger:

```
protokol://adresse
```

Drejer det sig om WWW-sider, er protokoltypen altid **http** – det står for Hypertext Transfer Protokoll. Men på samme måde kan du også anvende alle andre protokolformer i Internettet. Lad os nu se på de vigtigste protokoller.

Vil du *overføre en fil af vilkårlig karakter*, kan du benytte File Transfer Protocol, forkortet FTP, der blev anvendt i Internettet til dette formål længe før udviklingen af WWW og Hypertext. I det tilfælde hedder protokolbetegnelsen **ftp**. Altså aktiverer den følgende anvisning en overførsel af filen `/pub/doc/html.ps` på serveren `ftp.saar.de`:

```
<A HREF=ftp://ftp.saar.de/pub/doc/html.ps>Download teksten her</A>
```

Vil du starte en *terminalemulering* eller *telnet-session* på en vilkårlig computer i Nettet, anvender du telnet - protokollen. I så fald skal du brige en henvisning som **telnet://computeradresse**. Vil du starte en telnet-session på computeren `bellona.wg.saar.de`, skal du altså integrere følgende URL i dit dokument:

```
<A HREF=telnet://bellona.wg.saar.de>Telnet p&aring; Bellona</A>
```

Er din browser indstillet på rette måde – for Netscape's vedkommende betyder det, at der er angivet et Telnet-program som hjælper –, åbnes der en telnet-session ved den adresserede computer, når du klikker på denne tekst. Det betyder, at der i dit telnetvindue åbnes computerens Login, hvorpå du som altid ved Telnet kan gøre brug af alle de tjenester, som denne computer tilbyder.

Vil du indsætte en henvisning til en Gopher, skal du angive **gopher://** og den ønskede computers adresse. Vil du altså integrere en henvisning til gopherserveren `gopher.saar.de` i dit dokument, gør du det med følgende kode:

```
<A HREF=gopher://gopher.saar.de>Saar.DE Gopher</A>
```

Et klik på **Saar.DE Gopher** i browseren åbner denne gopherserver.

Endelig kan du i en browser også aktivere en vilkårlig HTML-fil eller en anden fil, uden at du er koblet til Internettet. I så fald benytter du som protokol **file** og angiver stien til filen på din computer som adresse. Desuden skal du benytte hele tre skråstreg ved en **file**-adressering, ikke blot to som ellers:

```
<A HREF=file:///Harddisk/Internet/Netscape/homepage/fil.html>Min fil</A>
```

## Det grafiske design af websider

HTML-sproget har gennemgået en enorm udvikling på bare to år. Oprindeligt blev det udviklet med det enkle formål at muliggøre henvisninger til andre dokumenter. Men de nye grafiske browser stiller helt anderledes krav til layout og design af en HTML-fil. Og HTML-sprogets muligheder bilver hele tiden tilsvarende udviklet. Lad os nu se på nogle af de fundamentale teknikker til at designe en HTML-side, så den bliver interessant for øjet.

### Skillelinier

Det simpleste grafiske udtryksmiddel er anbringelsen af *skillelinier* i din tekst. Det er ikke blot meget let gjort, men gør samtidig din tekst meget mere overskuelig.

Der findes flere forskellige versioner af sådanne skillelinier. Lad os se nærmere på dem.

#### Simpel skillelinie

En *simpel skillelinie* defineres ved tag'et `<HR>`, der ikke behøver noget sluttag.

#### Eksempel Simpel skillelinie

```
<HTML>
<HEAD>
<TITLE>Simpel skillelinie</TITLE>
</HEAD>
<BODY>
Dette er første afsnit, som skal skilles fra det næste
ved en simpel skillelinie:
<HR>
Andet afsnit vises under skillelinien.
</BODY>
```



```
</HTML>
```

#### Ikke-skyggede skillelinier

Mange browsere viser skillelinier principielt tredimensionalt, altså med en skygge. Denne effekt kan dog ofte miste sin virkning ved baggrundsfarven.

Vil du undgå denne tredimensionale effekt, kan du opnå det ved hjælp af attributet `noshade`.

#### Eksempel Ikke-skygget skillelinie

```
<HTML>
<HEAD>
<TITLE>Ikke-skygget skillelinie</TITLE>
</HEAD>
<BODY>
Dette afsnit efterfølges af en ikke-skygget skillelinie:
<hr noshade>
```

```

som igen efterfølges af det andet afsnit.
</BODY>
</HTML>

```

### Skillelinie med fast længde

Almindeligvis gengives skillelinier over hele skærbilledets bredde. Men det er muligt at få browseren til at vise en skillelinie med fast længde og tykkelse.

Du opnår dette med attributterne `width` og `size`. `width` angiver her længden, enten i pixels eller i procent af vinduet, mens `size` definerer bredden eller tykkelsen i pixels. Grundindstillingen til tykkelsen er 2 pixels.

#### Eksempel Fast skillelinie

```

<HTML>
<HEAD>
<TITLE>Fast skillelinie</TITLE>
</HEAD>
<BODY>
Den følgende skillelinie skal fylde 40 procent af skærbilledet
<hr width=40%>
hvorpå følger en skillelinie med en tykkelse af 3
pixels og en længde af 100 pixels
<hr width=100 size=3>
</BODY>
</HTML>

```

### Justering af skillelinier

Anvender du faste skillelinier, kan du ovenikøbet justere dem – det vil sige, at du kan sætte dem centreret eller højre- eller venstrejusteret.

Det gør du med attributten `align`. Dette attribut kan forsynes med tre værdier:

- `align=right` betegner en *højrejusteret* skillelinie
- `align=left` betegner en *venstrejusteret* skillelinie
- `align=center` betegner en *centreret* skillelinie – sættes der ikke noget `align`-attribut, anvendes denne indstilling

#### Eksempel Justerede skillelinier

```

<HTML>
<HEAD>
<TITLE>Justerede skillelinier</TITLE>
</HEAD>
<BODY>
Dette er et eksempel på en venstrejusteret skillelinie:
<hr width=50 align=left>
Derefter følger en højrejusteret:
<hr width=30 align=right>
og endelig en centreret:
<hr width=50>
</BODY>
</HTML>

```

## Farver og baggrundsbilleder

Skillelinier er det simpleste middel til at formgive en side i WWW. Andre grafiske elementer, såsom farver og baggrundsbilleder, byder på flere muligheder.

### Farver i HTML-dokumenter

Forgrunds- og baggrundsfarver har alene gyldighed for det dokument, som de defineres i. Denne definition fastlægges ved et attribut i tag'et `<BODY>`. Hver farveværdi kræver her sit eget attribut.

Det enkelte attributs farveværdi angives i hexadecimal skrivemåde, det vil sige som kode for andelen af hver af de tre grundfarver Rød, Grøn og Blå. En farve defineres altså ved en sekscifret tildeling med værditegnet # foran. Da enhver værdi repræsenteres af et tocifret hexadecimaltal, kan der angives værdier mellem 0 og 255 for hver grundfarve.

*Baggrundsfarven* på en side angives ved det tilsvarende talværdi til attributet `bgcolor`. Selvfølgelig kan du også tildele en farve til sidens *forgrund*, det vil sige selve *teksten* – det gør du med attributet `Text`.

Ud over forgrunds- og baggrundsfarve kan du også bestemme *farveangivelsen i links, altså henvisningerne* – og her kan du ovenikøbet lave forskel mellem ikke besøgte, allerede kendte og lige nu aktive links. Det opnår du med attributterne `LINK`, `VLINK` og `ALINK` og den ønskede farveværdi.

### Eksempel farvedefinitioner

```
<HTML>
<HEAD><TITLE>Farver i HTML</TITLE></HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF" VLINK="#FF0000"
ALINK="#FFCC33">
Dette er sidens egentlige tekst.<BR>
Desværre kan jeg ikke demonstrere link-farverne for dig, da de
er afhængige af, om du allerede har aktiveret et link eller ej.
<BR>
Men for sjovs skyld kan du jo selv sætte nogle links ind i denne
linie, såsom som du har læst det i afsnittet om
henvisninger, og så eksperimentere lidt med sagen.
</BODY>
</HTML>
```

### Baggrundsbilleder

Der findes endnu en mulighed for at peppe et HTML-dokument op: du kan indsætte et *baggrundsbillede*. Et baggrundsbillede er et lille billede i GIF- eller JPG-format, som ligger bag teksten. Dette billede reproduceres igen og igen, sådan at hele sidens baggrund så at sige tapetseres eller "kakles" med billedet.

Et sådant baggrundsbillede defineres for et dokument ad gangen. Lige som farvedefinitionen bestemmes det ved et attribut til `<BODY>`-tag'et, der som værdi får billedfilens navn.

### Eksempel Baggrundsbillede

```
<HTML>
<HEAD>
<TITLE>
Baggrundsbillede
</TITLE>
<BODY background="bild.gif">
Denne HTML-sides tekst sv&aelig;ver s&aring; at sige ovenover
baggrundsbilledet, der p&aring; bedste kakkelmaner gentages igen og
igen.
</BODY>
</HEAD>
```

## Integrere billeder i HTML-dokumenter

Vil du peppe HTML-dokumenter optisk op, har du mulighed for at integrere billeder: Sådanne billeder skal foreligge i GIF- eller JPG-format, da det fleste browsere umiddelbart kun kan vise disse formater – der findes ovenikøbet browsere, der kun kan klare GIF-formatet.

GIF-filer er særligt egnede, da et specielt GIF-format, GIF89a, byder på yderligere muligheder, såsom transparente baggrunde eller GIF-animationer. Sådanne filer kan fremstilles med populære shareware-programmer – under Windows for eksempel Gif Construction Kit eller Paintshop Pro 3.11, mens du på Macintosh kan bruge GraficConverter.

Desuden skal du være opmærksom på billedfilens størrelse. Bliver den for stor, kræves der under overførslen megen båndbredde og mere overførselstid, og det er et problem, hvis man er tilsluttet Internettet via modem. Derfor bør du arbejde præcist og ressourcesparende, når du benytter grafik.

### Indsætte grafik

Vil du integrere et billede i dit dokument, gør du det med tag'et `<IMG ...>`. Dette tag indsættes i dokumentet på det sted, hvor grafikken skal vises.

Vil du altså integrere billedfilen `bild1.gif` i dit HTML-dokument, ser det nødvendige tag ud som følger:

```
<IMG SRC="bild1.gif">
```

**SRC** står for Source, hvilket på dansk betyder kilde. Attributet får som værdi den ønskede billedfilens navn.

Denne kode forudsætter imidlertid, at billedfilen ligger i samme directory som HTML-filen. Er det ikke tilfældet, skal du også indføje billedfilens sti. Ligger den for eksempel i subdirectory `img`, som findes nedenunder dokument-directory, lyder koden således:

```
<IMG SRC="img/bild1.gif">
```

Du kan også integrere en fil, der ligger på en anden server i Internettet. I så fald skal du blot sætte denne fils URL som værdi til attributet **SRC**. Til selve overførslen benyttes protokollet `http`. Ønsker du altså at integrere filen `bild1.gif` i subdirectory `img` af useren `as` på computeren `www.saar.de`, benytter du denne billedfilens URL og skriver følgende:

```
<IMG SRC="http://www.saar.de/~as/img/bild1.gif">
```

## Alternativ tekst

Men hvad nu, hvis modtageren eller læseren benytter en browser, der arbejder ASCII-orienteret og følgelig kun kan vise tekst – som for eksempel LYNX under Unix? Vel, HTML byder på muligheden for at tildele en alternativ tekst til billedet, som dukker op i tekstorienterede browsere. Kan en browser vise billeder, dukker denne alternative tekst slet ikke op.

Du sætter en sådan tekst ved at tilføje endnu et attribut til tag'et `<IMG ...>` – nemlig ganske enkelt `ALT`. Som værdi får dette attribut den ønskede tekst i anførelsestegn. Det ser ud som følger:

```
<IMG SRC="bild1.gif" ALT="Vores logo">
```

Browsere, der kan vise billeder, viser med denne kode billedfilen `bild1.gif` – tekstorienterede browsere nøjes med teksten “Vores logo”.

## Layout af grafik

Billeder kan ikke blot vises simpelt af browseren – der er også mulighed for at forsyne dem med en *ramme*. Det opnås med attributet `BORDER` indenfor tag'et `<IMG ...>`. Ønsker du at fastlægge rammens tykkelse, tilføjer du som værdi til dette attribut den ønskede tykkelse i pixels. Koden `<IMG SRC="bild1.gif" BORDER=10 ALT="Logo">` fører til, at billedet `bild1.gif` forsynes med en ramme af en tykkelse på 10 pixels. En ren tekstbrowser vil istedet vise teksten “Logo”.

Billeder kan også *justeres* til venstre eller højre i forhold til teksten. Det opnår du ved at indsætte attributet `align` med den ønskede justering – `left` eller `right` – i tag'et `<IMG...>`. Koden `<IMG SRC="bild1.gif" ALIGN=left>` viser billedet venstrejusteret, mens koden `<IMG SRC="bild1.gif" ALIGN=right>` justerer billedet mod højre.

Henviser du til billeder med tag'et `<IMG ...>`, har du også indflydelse på *visningens størrelse*. Det er forøvrigt et meget vigtigt punkt. Mange browsere, heriblandt også Netscape, har mulighed for først at opbygge en sides grundlayout, for siden at indsætte billederne. Men det kræver, at billedernes størrelse angives – magler denne angivelse, opbygges billederne samtidig med sidens grundlayout, og alt efter billedernes størrelse kan det tage meget lang tid. Du indsætter størrelsesangivelser med attributterne `WIDTH` og `HEIGHT`, der som værdi får billedets bredde og højde i pixels.

Koden `<IMG SRC="bild1.gif" WIDTH=100 HEIGHT=150>` viser altså billedfilen `bild1.gif` med en højde på 150 og bredde på 100 Pixels.

Ud over *visningens størrelse* kan du også fastlægge *afstanden* mellem et billede og dets omgivelser. Det gør du ved hjælp af attributterne `VSPACE` og `HSPACE` til tag'et `<IMG ...>` – og igen definerer du afstanden i pixels. Attributet `VSPACE` betegner den *vertikale* afstand, altså afstanden til foranstående og efterfølgende tekst, mens attributet `HSPACE` betegner den *horisontale* afstand, altså afstanden til venstre margin. Det ser ud som følger:

```
<IMG SRC="bild1.gif" HSPACE=50 VSPACE=80>
```

## En grafik, to versioner

Vil du loadere en grafik hurtigt, mens selve siden opbygges, kan du aktivere den i to versioner . Den første version vises som vanlig umiddelbart mens siden opbygges, mens den anden version af samme billede aktiveres først, når hele siden er færdig.

Fordelen er her, at den version, der loades først, har en lavere opløsning og altså en mindre filstørrelse, mens den følgende version har en større opløsning og altså en bedre kvalitet.

Det opnår du, idet du tilføjer endnu et attribut til tag'et `<IMG ...>`: ud over det vante attribut `SRC` med filbetegnelsen for den anden, højtopløste grafik sætter du attributet `LOWSRC` med sti og navn for den billedfil, der skal loades først.

### Eksempel Lav/høj opløsning

```
<IMG SRC="bild1.gif" LOWSRC="bild1-l.gif">
```

I dette eksempel vises først billedfilen bild1-l.gif, mens siden bygges op. Er siden færdig, loades til slut billedfilen bild1.gif.

### Et billede som hyperlink

Selvfølgelig er det også muligt at forbinde et hyperlink med et billede og på den måde at gøre billedet "klikbart". Du definerer et sådant link ved ganske enkelt at kombinere tag'et `<IMG>` med tag'et `<A HREF>`.

Kombinationen bliver forståelig, hvis vi ser på det følgende eksempel:

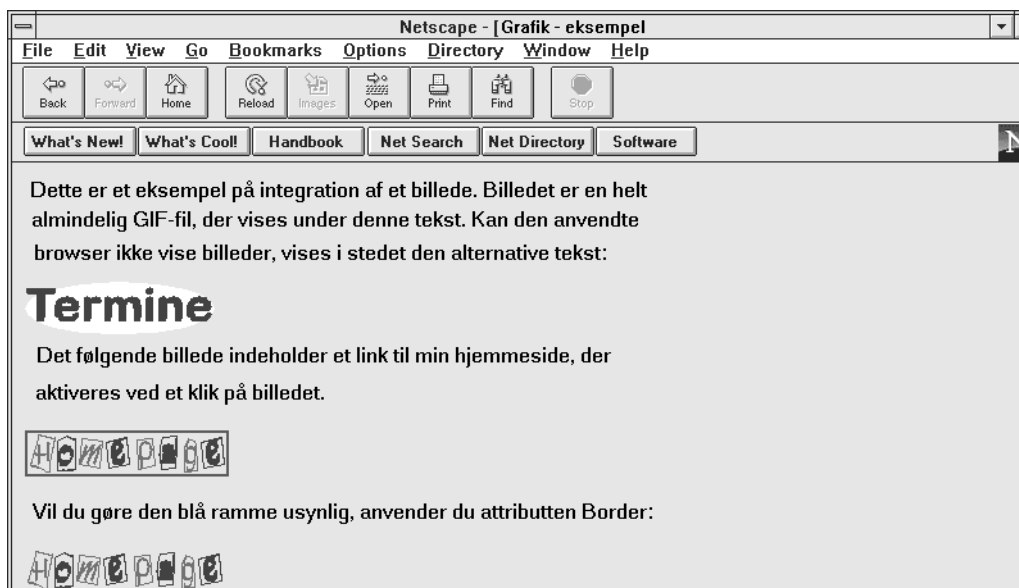
```
<A HREF=http://www.saar.de/><IMG SRC="iplogo.gif" ALT="Saar.DE Server"></A>
```

Dette tag betegner et hyperlink på serveren [www.saar.de](http://www.saar.de). Men istedet for en klikbar tekst vises billedfilen iplogo.gif, som er klikbar og aktiverer det ønskede link – på rene tekstbrowsere vises så istedet teksten "Saar.DE Server", som kan aktiveres ved et klik.

Det viste billede forsynes med en blå ramme for at tydeliggøre, at det er klikbart. Vil du undertrykke denne ramme, kan du gøre det ved at forsyne tag'et `<IMG>` med attributet `BORDER`, som du atter tilføjer værdien 0 (nul). Det ser altså ud som følger:

```
<A HREF=http://www.saar.de/><IMG SRC="iplogo.gif" ALT="Saar.DE Server" BORDER=0 ></A>
```

### Eksempel Billedintegration



```
<HTML>
<HEAD>
<TITLE>Grafik-Eksempel</TITLE>
</HEAD>
<BODY>
Dette er et eksempel på integration af et billede.
Billedet er en helt almindelig GIF-fil, der vises under denne tekst.
Kan den anvendte browser ikke vise billeder, vises istedet den
alternative tekst:
<p>
<IMG SRC="ktermin.gif" ALT="TERMIN-BILLEDE">
<p>
Det følgende billede indeholder et link til min hjemmeside, der
aktiveres ved et klik på billedet:
<p>
<A HREF="http://www.saar.de/~as/"><IMG SRC="hplogo.gif" ALT="Til min
hjemmeside"></A>
<p>
Vil du gøre den usynlige ramme usynlig, anvender du attributen
Border:
<p>
<IMG SRC="hplogo.gif" ALT="Til min hjemmeside" BORDER="0">
<p>
</BODY>
</HTML>
```

## Tabeller i HTML

Tabeller er en relativ ny side af HTML-sproget – faktisk blev de først officielt accepteret i sprogets nyeste version 3.2. Til gengæld byder de på næsten grænseløse muligheder, når det gælder den grafiske udformning af layoutet på en HTML-side. Man kan i sådanne tabeller forbinde celler, der står ved siden af eller oven over hinanden, man kan indføje billeder i celler, ja man kan endog indsætte en tabel i en eksisterende tabel. Lærer du systematisk, hvordan du kan arbejde med tabeller, får du altså muligheden for at udforme en HTML-side på en særdeles fri måde.

### Grundelementer i tabeller

Tabeller defineres ved hjælp af tag'et `<TABLE> ... </TABLE>`, der omslutter tabellen i hele sin udstrækning. Tag'et `<TR> ... </TR>` omslutter en tabel-række. Der findes ikke noget tag for en tabel-spalte – den genereres automatisk ved de lodret stående celler, der konstituerer den. En sådan tabel-celle defineres ved hjælp af tag'et `<TD> ... </TD>`. Lad os nu se på et simpelt eksempel – en tabel med tre rækker og tre spalter.

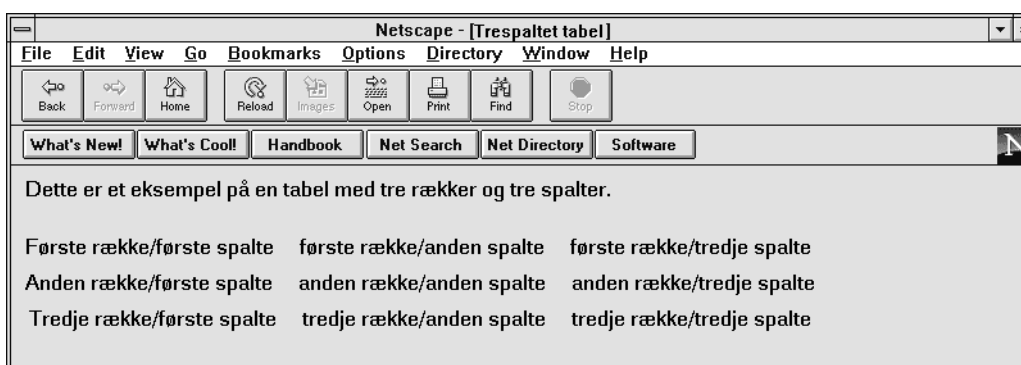
### Eksempel Trespaltet tabel

```
<HTML>
<HEAD>
<TITLE>Trespaltet tabel</TITLE>
</HEAD>
<BODY>
<P>
Dette er et eksempel på en tabel med tre rækker og tre
spalter. </P>
<P>
```

```

<TABLE>
<TR>
<TD>
Første række / Første spalte
</TD>
<TD>
Første række / Anden spalte
</TD>
<TD>
Første række / Tredje spalte
</TD>
</TR>
<TR>
<TD>
Anden række / Første spalte
</TD>
<TD>
Anden række / Anden spalte
</TD>
<TD>
Anden række / Tredje spalte
</TD>
</TR>
<TR>
<TD>
Tredje række / Første spalte
</TD>
<TD>
Tredje række / Anden spalte
</TD>
<TD>
Tredje række / Tredje spalte
</TD>
</TR>
</TABLE>
</P>
</BODY>
</HTML>

```



Som du ser, omslutes hver tabelrække af tag'et `<TR> ... </TR>`, mens de enkelte celler omslutes af tag'et `<TD> ...</TD>` og står indenfor hver sit rækketag.

## Tabeller med ramme, trådnæt og afstandsdefinering

Vil du forsyne en tabel med en synlig ramme, skal du tilføje attributet `BORDER` til starttag'et i tag'et `<TABLE> ... </TABLE>`. Et *trådnæt* opnår du med attributet `CELLSPACING`. Begge disse attributer tildeles en værdi i pixels.

Desuden kan du også definere afstanden fra cellekant til celleindhold. Det gør du med attributet `CELLPADDING` og den ønskede værdi i pixels indenfor tag'et `<TABLE> ... </TABLE>`.

### Eksempel Tabel med ramme, trådnæt og afstandsdefinition

```
<HTML>
<HEAD>
<TITLE>Trespaltet tabel med defineret ramme og afstand</TITLE>
</HEAD>
<BODY>
<P>
Her ser du et eksempel på; en tabel med tre rækker og tre spalter,
samt defineret rammetykkelse og celleafstand. </P>
<P>
<TABLE BORDER=5 CELLSPACING=2 CELLPADDING=7>
<TR>
<TD>
Første række / Første spalte
</TD>
<TD>
Første række / Anden spalte
</TD>
<TD>
Første række / Tredje spalte
</TD>
</TR>
<TR>
<TD>
Anden række / Første spalte
</TD>
<TD>
Anden række / Anden spalte
</TD>
<TD>
Anden række / Tredje spalte
</TD>
</TR>
<TR>
<TD>
Tredje række / Første spalte
</TD>
<TD>
Tredje række / Anden spalte
</TD>
<TD>
Tredje række / Tredje spalte
</TD>
</TR>
</TABLE>
</P>
</BODY>
</HTML>
```

### Definition af højde, bredde og justering

Lige som afstanden mellem cellekant og celleindhold sættes normalt også rækkehøjden og celle- og derved spaltebredden automatisk af browseren. Men selvfølgelig har du også her mulighed for præcist at definere rækkehøjden og celle- og dermed spaltebredden. Blot kan du i så fald ikke benytte tag'et `<TD> ... </TD>` til at definere en celle – du må istedet anvende tag'et `<TH> ... </TH>`, som du så kan forsyne med de nødvendige attributer for at definere højden og bredden.

Disse attributer er `WIDTH` for bredden og `HEIGHT` for højden – og atter angives værdien i pixels.

### Eksempel Tabel med defineret bredde/højde

```
<HTML>
<HEAD>
<TITLE>Tabel med defineret bredde og højde</TITLE>
</HEAD>
<BODY>
<TABLE BORDER>
<TR>
<TH WIDTH=120 HEIGHT=50>
Række 1 / spalte 1
</TH>
<TH WIDTH=100>
Række 1 / spalte 2
</TH>
</TR>
</TABLE>
</BODY>
</HTML>
```

Naturligvis kan du også justere teksten i en tabelcelle. For at gøre dette, skal du forsyne tag'et `<TH> ... </TH>` med attributet `ALIGN` og sætte dens værdi som `LEFT`, `RIGHT` eller `CENTER`, for at justere teksten til venstre, til højre eller centreret.

Endelig kan du ud over spaltebredde og rækkehøjde også definere *hele tabellens bredde og højde*. Også her benytter du attributterne `WIDTH` og `HEIGHT`, som du denne gang tildeler til tag'et `<TABLE>`. Værdien kan du her ikke blot angive i pixels, men alternativt også som procentværdi relativt til vinduets størrelse.

### Eksempel Tabel med defineret størrelse

```
<HTML>
<HEAD>
<TITLE>Tabel med defineret størrelse</TITLE>
</HEAD>
<BODY>
<TABLE BORDER WIDTH=60% HEIGHT=90%>
<TR>
<TD>
spalte 1
</TD>
<TD>
spalte 2
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

## Forbinde celler med hinanden

I en HTML-tabel kan du forbinde flere celler i en række eller spalte med hinanden. Det fører til, at den aktuelle celle udvides henover flere spalter eller indenfor en spalte henover flere rækker.

Vil du *forbinde celler over flere spalter med hinanden*, sætter du i tag'et `<TH> ... </TH>` attributet `COLSPAN` og tildeler det som værdi det ønskede antal celler, der skal forbindes.

### Eksempel Celler over spalter

```
<HTML>
<HEAD>
<TITLE>
Forbinde celler henover spalter
</TITLE>
</HEAD>
<BODY>
<TABLE BORDER>
<TR>
<TH COLSPAN=2 ALIGN=CENTER>
Denne række omfatter to spalter
</TH>
</TR>
<TR>
<TD>
Spalte 1
</TD>
<TD>
Spalte 2
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Tilsvarende kan du selvfølgelig også *forbinde celler henover flere rækker*. Her benytter du atter tag'et `<TH> ... </TH>` – blot tildeler du det dengang attributet `ROWSPAN` med antallet af de ønskede celler som værdi.

### Eksempel Celler over rækker

```
<HTML>
<HEAD>
<TITLE>
Forbinde celler henover rækker
</TITLE>
</HEAD>
<BODY>
<TABLE BORDER>
<TR>
<TH ROWSPAN=2>
Dette er begyndelsen af
</TH>
<TD>
Række 1
</TD>
</TR>
<TR>
<TD>
Række 2
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

### Tabeloverskrift

Efter al denne komplicerede hjernegymnastik tager vi nu en relativ simpel sag: du kan tildele en overskrift til en tabel.

Der gør du ved hjælp af tag'et `<CAPTION> ... </CAPTION>`, som sættes umiddelbart efter tag'et `<TABLE> ... </TABLE>`.

### Eksempel Tabeloverskrift

```
<HTML>
<TITLE>
Tabeloverskrifter
</TITLE>
<BODY>
<TABLE BORDER>
<CAPTION ALIGN=CENTER>Dette er selve overskriften
</CAPTION>
<TR>
<TD>Spalte 1</TD>
<TD>Spalte 2</TD>
<TD>Spalte 3</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

## Yderligere muligheder

### Formularer

Hypertext Markup Language-sproget (HTML) tilbyder muligheden for at fremstille såkaldte *formularer*. En sådan formular er et dokument, der indeholder felter til indtastning af tekst og valgfelter, således at brugeren direkte kan skrive data eller vælge mellem dem. Formularer er altså egentlig en slags input-masker i HTML-dokumenter.

Da du nu normalt er umiddelbart tilsluttet til Internettet, når du står i et HTML-dokument, har du ovenikøbet mulighed for at sende de data, som du indsætter, direkte over nettet. Det kan ske på flere forskellige måder.

Lad os nu se nærmere på, hvordan man opstiller sådanne formularer – hvilke felttyper de kan indeholde, hvordan disse defineres og på hvad måde de indsatte data når frem til målet.

### Formulardefinering

En *formular* defineres ved hjælp af tag'et `<FORM ...> ... </FORM>`, der omslutter det i dets helhed. De enkelte punkter i en sådan formular og dets layout defineres ved yderligere tags, der skal stå indenfor hoved-tag'et på de relevante steder.

For fuldstændigt at kunne definere formularen, behøver serveren visse informationer. Det er til at begynde med *måladressen* i Internettet, altså den URL, som de indsatte data skal sendes til. Dernæst er det defineringen af *overførselsmåden*.

Måladressen betegner den fil, som skal modtage de data, som blev indsat i dit formulardokument. Denne fil kan siden viderebehandle de modtagne data.

Måladressen meddeles serveren ved attributet `ACTION` til tag'et `<FORM ...>`. Starttag'et er altså ud som følger:

```
<FORM ACTION=[INTERNETADRESSE] ...>
```

Måladressen kan være en anden HTML-fil eller et *CGI-Script*. Forkortelsen CGI står for Common Gateway Interface – der er her tale om en programmeringsteknik, der gør det muligt at viderebehandle indleverede data på en server i Internettet. Af pladsmæssige hensyn skal vi ikke komme nærmere ind på det her.

Desuden kan attributet `ACTION` aktivere skrivningen og forsendelsen af en email ved hjælp af værdien `mailto:[INTERNETADRESSE]`.

### Overførselsmetode

Overførselsmetoden beskriver, hvordan de indsatte data overføres til serveren. Metoden kan have sin betydning, når det drejer sig om viderebehandling af de indsatte data. Man skelner her mellem to forskellige måder: den ene kaldes `POST` og den anden `GET`.

Hvilken metode der skal anvendes, det fastlægges indenfor tag'et `<FORM ...>` ved attributet `METHOD` med den ønskede værdi. Vælger du overførselsmetoden `POST`, sendes de indsatte data til serveren på standardmåde. Overføres data med email, er det denne metode, der skal vælges. Attributet `METHOD` får her altså værdien `POST`. `GET`-metoden har en anden fremgangsmåde – her overføres informationerne ved hjælp af en omgivelsesvariabel, nemlig variabelen `QUERY_STRING`.

## Eksempel Formularindledning

Lad os nu se på, hvordan en formulardefinering egentlig ser ud:

```
<FORM ACTION=mailto:as@saar.de method=post>
... formularanvisninger ...
</FORM>
```

Denne kodeblok beskriver et formular, der overfører de ønskede data via email til adressen `as@saar.de`.

## Inputfelt

Et inputfelt er et formularfelt, hvori brugeren direkte kan indsætte data. Man skelner mellem énlinede og flerlinede inputfelter.

Et *enlinet inputfelt* defineres ved hjælp af tag'et `<INPUT>`. Dette tag kan udvides med flere attributer. Det har kun gyldighed indenfor en formulardefinition.

`<INPUT ...>` indleder altså et enlinet inputfelt. De nødvendige præciserende informationer angives med de attributer, som vi nu vil se nærmere på.

Attributet `NAME` betegner navnet for det aktuelle inputfelt. Dette navn benyttes i de scripter, der viderebehandler de indkomne data.

Attributet `TYPE` betegner inputfeltets karakter. Der findes flere typer af input:

<code>text</code>	betegner indsættelse af tekst
<code>password</code>	betegner en skjult password-indsættelse
<code>int</code>	definerer indsættelse af hele tal
<code>float</code>	definerer indsættelse af decimaltal
<code>date</code>	definerer en dato
<code>url</code>	definerer indsættelsen af en internetadresse

Attributet `SIZE` definerer, hvor mange tegn inputfeltet skal råde over på skærmen. Er inputtet længere end denne definition, scrolles den allerede indsatte tekst.

Attributet `MAXLENGTH` definerer den maksimale inputlængde i tegn. På den måde er det muligt at begrænse antallet af tegn, der sættes ind.

## Eksempel `<INPUT ...>`-Tag

```
<FORM action=mailto:as@saar.de method=post>
<INPUT TYPE=TEXT NAME=BRUGER SIZE=30 MAXLENGTH=60>
</FORM>
```

Dette eksempel viser en inputkode i formularet, som tillader indsætning af en tekst på maksimal 60 tegn – på skærmen er feltet højst 30 tegn bred, hvilket fører til, at den øvrige tekst scrolles. Til brug for viderebehandling ved script får feltet navnet BRUGER.

Flerlinede tekstinput defineres med tag'et `<TEXTAREA> ... </TEXTAREA>`. Her kan man for eksempel indsætte kommentarer eller fri tekst. Også flerlinede tekstinput tildeles et navn ved attributet `NAME`.

Endvidere må du her definere tekstfeltets størrelse. Det gør du ved at sætte de tilsvarende værdier til attributterne `ROWS` og `COLS`; `ROWS` angiver antallet af rækker, mens `COLS` betegner antallet af spalter.

### Eksempel Flerlinet tekstinput

```
<FORM action=mailto:as@saar.de method=post>
<TEXTAREA NAME=KOMMENTAR COLS=20 ROWS=10>
</TEXTAREA>
</FORM>
```

Dette formular indeholder et inputfelt med 10 linier á hver 20 tegn.

### Valgfelter

*Valgfelter* er lister med faste svar, som brugeren kan vælge imellem. Overførslen af de valgte data sker ved afsendelse af formularet.

Et valgfelt, som også kaldes selectbox, defineres ved hjælp af tag'et `<SELECT> ... </SELECT>`, der kun har gyldighed indenfor et `<FORM>`-tag. De enkelte valgmuligheder betegnes med tag'et `<OPTION>`, der igen skal stå indenfor tag'et `<SELECT> ... </SELECT>`.

Dette sidste tag, `<SELECT>`, behøver dog flere attributer. For det første er der betegnelsen af det felt, som skal overføres. Feltets navn defineres med attributet `NAME`. Et andet attribut er `SIZE` – dets værdi bestemmer, hvor mange elementer der kan vises.

### Eksempel valgfelt

```
<HTML>
<HEAD>
<TITLE>
Select-eksempel
</TITLE>
</HEAD>
<BODY>
Her kan du vælge dit hjemland: <P>
<FORM ACTION=mailto:as@saar.de METHOD=POST>
<SELECT NAME=land SIZE=2>
<OPTION>Danmark
<OPTION>Frankrig
<OPTION>USA
<OPTION>England
<OPTION>Belgien
<OPTION>Italien
</SELECT>
</FORM>
</BODY>
</HTML>
```

Dette eksempel resulterer i en selectbox, som viser to elementer, mens resten kan gøres synlig ved scrolling. Skal der kunne vælges flere punkter, tilføjer du i tag'et `<SELECT>`attributet `MULTIPLE`.

Er der bestemte værdier, der umiddelbart, altså per default, skal foreslås, kan du indsætte attributet i tag'et `<OPTION>`. Desuden kan du adskille værdioverførslen til formularets resultat fra visningen i formularfeltet, idet du til tag'et `<OPTION>` sætter som værdi til attributet `VALUE` det udtryk, som skal overføres, istedet for det ord, der ses på skærmen.

## Eksempel Valgfelt 2

```
<HTML>
<HEAD>
<TITLE>
Select-eksempel 2
</TITLE>
</HEAD>
<BODY>
<FORM action=mailto:as@saar.de method=post>
<SELECT NAME=default>
<OPTION SELECTED VALUE="0045"> Danmark
<OPTION VALUE="0033"> Frankrig
<OPTION VALUE="001"> USA
</SELECT>
</FORM>
</BODY>
</HTML>
```

Dette eksempel resulterer i et valgfelt, der viser et felt og overfører det valgte lands telefonkode, som er sat som værdi i **VALUE**-attributet, istedet for det synlige landenavn.

## Radio- og checkbuttons

*Radio- og checkbuttons* er andre muligheder, der tilbyder brugeren et valg mellem flere forudgivne muligheder.

Radiobuttons er en række af tekstede knapper, som brugeren kan vælge en og kun en fra.

Navnetildelingen definerer, hvilken gruppe en radiobutton tilhører. Alle definitionstags for radiobuttons med samme navn tilhører samme gruppe – og indenfor denne gruppe kan der altid kun vælges én mulighed.

En radiobutton defineres indenfor tag'et **<INPUT>**, som kun har gyldighed indenfor tag'et **<FORM>**.

En radiobutton defineres ved at tildele **TYPE**-attributet i tag'et **<INPUT>** værdien **RADIO**. Desuden får **NAME**-attributet i tag'et gruppenavnet som værdi, mens **VALUE**-attributet får som værdi det udtryk, som skal overføres til modtageren. Tag'et efterfølges af den tekst, der skal vises på skærmen ved siden af den tilsvarende valgmulighed.

## Eksempel Radiobutton

```
<HTML>
<HEAD>
<TITLE>
Radiobuttons
</TITLE>
</HEAD>
<BODY>
<FORM ACTION=mailto:as@saar.de METHOD=POST>
<INPUT TYPE=RADIO NAME=land VALUE="Danmark"> Danmark
<INPUT TYPE=RADIO NAME=land VALUE="Frankrig"> Frankrig
<INPUT TYPE=RADIO NAME=land VALUE="USA">Forenede Stater
</FORM>
</BODY>
</HTML>
```

*Checkbuttons* er en gruppe af foruddefinerede værdier, som der kan vælges flere fra.

Checkbuttons defineres ligeledes ved hjælp af tag'et `<INPUT>`. Til attributet `TYPE` tildeles her værdien `CHECKBOX`.

Iøvrigt gælder de samme regler som ved radiobuttons. Checkbuttons, som skal sættes på forhånd, altså per default, kan forsynes med det yderligere attribut `CHECKED` i tag'et `<INPUT>`.

### Eksempel Checkbuttons

```
<HTML>
<HEAD>
<TITLE>
Checkbuttons
</TITLE>
</HEAD>
<BODY>
<FORM ACTION=mailto:as@saar.de METHOD=POST>
<INPUT TYPE=CHECKBOX NAME=default VALUE="0045">Danmark
<INPUT TYPE=CHECKBOX NAME=default VALUE="0033">Frankreich
<INPUT TYPE=CHECKBOX NAME=default VALUE="001">USA
</FORM>
</BODY>
</HTML>
```

### Actionbuttons

Actionbuttons er knapper, der udløser en aktion. Der findes to typer af aktionsbuttons – nemlig *Submit-* og *Resetbuttons*.

Aktiveres en `SUBMIT`-knap, fører det til, at de i formularet indsatte data overføres – på grundlag af den modus, der er fastlagt i `ACTION`-attributet i tag'et `<FORM>`.

En `SUBMIT`-knap defineres ved hjælp af tag'et `<INPUT>`. Attributet `TYPE` tildeles til dette formål værdien `SUBMIT`. Du definerer, hvilken tekst der skal vises i knappen, ved at tildele den som værdi til attributet `VALUE`. Desuden kan du tilføje et yderligere navn til knappen ved at sætte det som værdi til attributet `NAME` – dette navn overføres ligeledes sammen med teksten i `VALUE`.

### Eksempel Actionbutton

```
<HTML>
<HEAD>
<TITLE>
SUBMIT BUTTON
</TITLE>
<FORM ACTION=mailto:as@saar.de METHOD=POST>
Kommentar: <INPUT TYPE=TEXT NAME=kommentar SIZE=20>
<INPUT TYPE=SUBMIT VALUE="overf&oslash;re">
</FORM>
</BODY>
</HTML>
```

Dette eksempel resulterer i en lille formular, hvori brugeren kan indsætte en tekstlinje, der sendes vis email til `as@saar.de`, når `SUBMIT`-knappen aktiveres.

En *resetbutton* sletter alle angivelser, som allerede blev sat ind i formularen, og sætter den tilbage til de oprindelige indstillinger. En sådan knap fremstiller du ved at tildele kodeordet **RESET** til attributet **TYPE** i tag'et **<INPUT>**.

Indenfor et formular ser det altså ud som følger:

```
<INPUT TYPE=RESET VALUE="annullere">
```

### Eksempel Formular

Det følgende eksempel resulterer i en kommentarformular, hvis resultater sendes til email-adressen **as@saar.de**, når brugeren klikker på **SUBMIT**-knappen.

```
<HTML>
<HEAD>
<TITLE>
Eksempel Formular
</TITLE>
</HEAD>
<BODY>
<FORM ACTION=mailto:as@saar.de METHOD=POST>
Afsender:<br>
Name: <INPUT TYPE=TEXT NAME=afsender size=40><br>
Email: <INPUT TYPE=TEXT NAME=email size=40><br>
<br>
Hvad synes du om denne side? <br>
<INPUT TYPE=RADIO NAME=synes VALUE="godt"> Godt<br>
<INPUT TYPE=RADIO NAME=synes VALUE="middel">Nogenlunde<br>
<INPUT TYPE=RADIO NAME=synes VALUE="d&aring;rligt">D&aring;rligt<br>
<br>
Hvordan er du tilsluttet til Internettet?
<SELECT name=speed>
<OPTION value="14k4">Modem 14k4 bps
<OPTION value="28k8">Modem 28k8 bps
<OPTION value="64k">Standing line 64 kbps
<OPTION value="net">Standing line &gt; 64kbps
</SELECT>
<p>
Har du en kommentar:<br>
<TEXTAREA NAME=kommentar rows=10 cols=50>
</TEXTAREA>
<br>
<INPUT TYPE=SUBMIT VALUE="overf&oslash;re">
<INPUT TYPE=RESET VALUE="slette">
</FORM>
</BODY>
</HTML>
```

- <!-- -->, 14
- <A HREF=, 21
- <A HREF=mailto  
>, 23
- <A NAME=, 22
- <A> ... </A>, 21
- <ADDRESS>, 15
- <b>, 17
- <basefont size=x>, 17
- <big>, 18
- <blink>, 17
- <blockquote>, 15
- <BODY background>, 28
- <BODY>, 7; 27
- <CAPTION>, 36
- <CENTER>, 11
- <cite>, 15
- <code>, 16
- <DIR>, 14
- <dl>, 14
- <dt>, 14
- <font>, 17; 18
- <FORM ...>, 37
- <H>, 8
- <HEAD>, 7
- <hr align= x>, 26
- <hr noshade>, 25
- <hr size=nn>, 26
- <hr width=nn%>, 26
- <HR>, 25
- <i>, 17
- <IMG ...>, 28
- <li>, 12
- <NOBR>, 10
- <OPTION>, 39
- <P>, 10
- <PRE>, 16
- <SELECT>, 39
- <small>, 18
- <sub>, 17
- <sup>, 17
- <TABLE>, 31
- <TD>, 31
- <TEXTAREA>, 38
- <TH>, 34
- <TR>, 31
- <tt>, 16
- <u>, 17
- <ul>, 12
- <wbr>, 10
- ACTION, 37
- Actionbuttons, 41
- afsnit, 10
- afstanden cellekant til celleindhold, 33
- ALIGN, 34
- ALT, 29
- attribut, 8
- baggrundsbillede, 27
- baggrundsfarve, 27
- billede, afstand, 29
- billede, hyperlink, 30
- billede, to versioner, 29
- Billeder, justere, 29
- billedvisning, størrelse, 29
- Body, 6
- BORDER, 29; 33
- browser, 4
- CELLPADDING, 33
- CELLSPACING, 33
- centreret, 11
- CGI-Script, 37
- CGI-skript, 3
- CHECKBOX, 41
- checkboxbuttons, 41
- CHECKED, 41
- citat, 15
- COLS, 38
- COLSPAN, 35
- Common Gateway Interface, 3
- eksempler, 16
- Email, 4; 23
- farver, 27
- farver i HTML-dokumenter, 27
- fed, 16
- File Transfer Protocol, 24
- forankringspunkt, 22
- forgrundsfarve, 27
- formateringsanvisninger, 15
- formular, 37
- formular, definere, 37
- formular, måladresse, 37
- FTP, 4; 24
  - //, 24
- GET, 37
- GIF, 23
- GIF89a, 28
- glossar, 14
- gopher
  - //, 24
- grafik, integrere, 28
- grafik, vise, 23
- grundlag, HTML-dokument, 7
- Head, 6
- HEIGHT, 29; 34
- henvisning, email, 23
- Henvisninger indenfor samme server, 21
- homepage, 3
- HREF, 21
- HSPACE, 29
- HTML, 3; 4; 6

HTML-koder, 8  
HTTP, 5  
hyperlink, 4; 6  
hyperlink, billede, 30  
Hypertext Reference, 21  
hævet tekst, 17  
inputfelt, 38  
JPG, 23  
kataloglister, 13  
kildetekst, 16  
klient, 4  
kommentarer, 14  
linieskift, 9  
linieskift, undgå, 10  
liste, simpel, 12  
LOWSRC, 29  
LYNX, 29  
mailto:, 23  
markering, simpel liste, 12  
MAXLENGTH, 38  
mindre skrift, 18  
multitasking, 4  
målpoint, 22  
NAME, 38  
nummererede lister, 12  
overføre fil, 24  
Overførselsmetode, 37  
overskrifter, 8  
POST, 37  
præformateret tekst, 16  
QUERY\_STRING, 37  
RADIO, 40  
radiobuttons, 40  
ramme, 29  
ramme, tykkelse, 29  
resetbutton, 41  
Resetbuttons, 41  
ressource, 5  
ROWS, 38  
ROWSPAN, 36  
rækkehøjden, 34  
server, 4  
SGML, 4  
SIZE, 38  
skillelinie, fast længde, 26  
skillelinie, ikke-skygget, 25  
skillelinie, simpel, 25  
skillelinier, 25  
skillelinier, justering, 26  
skriftfarver, 18  
sluttæg, 6  
spaltebredde, 34  
starttag, 6  
strong, 16  
struktur, HTML-dokumenter, 6  
struktur, HTML-koder, 6  
større skrift, 18  
Submitbuttons, 41  
sænket tekst, 17  
Tabel, justering, 34  
tabel, ramme, 33  
tabel-celle, 31  
tabeller, grundelementer, 31  
tabel-række, 31  
tabel-spalte, 31  
tag, 6  
TCP/IP, 5  
tekstfarve, 27  
tekstfremhævelser, 16  
tekstinput, flerlinet, 38  
telnet  
    //, 24  
telnet-session, 24  
terminalemulering, 24  
trådnæt, 33  
TYPE, 38  
URI, 5  
URL, 5; 21  
URN, 5  
valgfelter, 39  
VSPACE, 29  
webserver, 4  
WIDTH, 29; 34  
WWW-server, 4

# WWW

## Homepages - lav dem selv

<b>Forord</b> .....	<b>3</b>	<b>Det grafiske design af websider</b> .....	<b>25</b>
<b>Grundbegreber</b> .....	<b>4</b>	Skillelinier .....	25
World Wide Web .....	4	Simpel skillelinie .....	25
Todelt grundopbygning .....	4	Ikke-skyggede skillelinier .....	25
WWW-dokumenter vises via HTML.....	4	Skillelinie med fast længde .....	26
Kommunikationssproget: HTTP.....	5	Justering af skillelinier .....	26
Ensformet adressering: URI .....	5	Farver og baggrundsbilleder.....	27
HyperText Markup Language (HTML) .....	6	Farver i HTML-dokumenter .....	27
Et HTML-dokuments struktur .....	6	Baggrundsbilleder .....	27
HTML-koders struktur .....	6	Integrere billeder i HTML-dokumenter.....	28
Grundlaget i et HTML-Dokument .....	7	Indsætte grafik .....	28
Basale HTML-koder .....	8	Alternativ tekst.....	29
Overskrifter .....	8	Layout af grafik .....	29
Linieskift .....	9	En grafik, to versioner.....	29
Undgå et linieskift .....	10	Et billede som hyperlink .....	30
Afsnit .....	10	Eksempel Billedintegration.....	30
Centreret visning .....	11	Tabeller i HTML .....	31
Listeformater .....	12	Grundelementer i tabeller.....	31
Kommentarer.....	14	Tabeller med ramme, trådnæt og	
Yderlige anvisninger til formatering .....	15	afstandsdefinering .....	33
Citater .....	15	Definition af højde, bredde og justering.....	34
Adresser.....	15	Forbinde celler med hinanden .....	35
Præformateret tekst.....	16	Tabeloverskrift .....	36
Eksempler og kildetekst.....	16	<b>Yderligere muligheder</b> .....	<b>37</b>
Tekstfremhævelser .....	16	Formularer .....	37
Fed tekst .....	16	Formulardefinering .....	37
Specialtegn i HTML-dokumenter .....	19	Overførselsmetode .....	37
<b>Henvisninger i HTML-dokumenter</b> .....	<b>21</b>	Inputfelt.....	38
Muliggøre en henvisning .....	21	Valgfelter .....	39
Henvisninger indenfor samme server .....	21	Radio- og checkbuttons.....	40
Henvisninger af speciel karakter .....	23	Actionbuttons.....	41
Aktivere en email .....	23	Eksempel Formular .....	42
Visning af billeder .....	23		
Adressere andre ydelser.....	24		