

38,-

Hjemmesider for alle

# Web Design

Tekst, grafik, [links](#)  
[HTML](#), [browsere](#)

Hvilke programmer  
skal man bruge?  
[Microsoft Frontpage](#)  
[Adobe PageMill](#)  
[Netscape Composer](#)  
[Claris Home Page](#)

Hvad er  
[Java](#) • [JavaScript](#)  
[ActiveX](#) • [CGI](#) •  
[Style Sheets](#) ?

[nyttige adresser](#)  
[ordforklaring](#)

- Lav hjemmesider og websites til [World Wide Web](#) eller til [intranet](#)

- Lav **hurtige** sider, der **virker** på alle browsere

- Lav websites, der ser godt ud og kan bruges: Planlægning, struktur, sidelayout, typografi, navigation, brugerflade.

- Hvordan kommer siderne ud på nettet? Hvordan finder andre dem?

► [Læs mere...](#)  




Olav Junker Kjær

[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>

## Indledning

Webdesign er kunsten at lave en website – en eller flere sammenhængende sider, som andre kan få adgang til via internettet med en webbrowser. Dette hæfte henvender sig til alle, der vil lære noget om at lave webdesign. Emnet behandles fra bunden, og hæftet gennemgår alle aspekter, så både begyndere og viderekommende kan få udbytte.

### Dine forudsætninger

Jeg går ud fra, at du

- har en vis erfaring med webbet som bruger, kan søge oplysninger og er fortrolig med udtryk som *links*, *browser* etc. Hvis du er ny på webbet, kan jeg anbefale hæftet *Ud på World Wide Web*.
- kan finde ud af at bruge en computer, installere programmer, kender begreber som *operativsystem*, *skærmopløsning* etc. og ikke er bange for at eksperimentere og prøve dig frem med programmerne.
- kan læse engelsk, da det meste software, hjælp og dokumentation kun findes på engelsk.
- har adgang til internettet. Det er dog ikke nødvendigt at have en server for at kunne eksperimentere med webdesign.

### Sådan skal dette hæfte bruges

Dette hæfte skal læses foran computeren. Man får mest ud af det ved at lege sig frem og eksperimentere med de ting, jeg beskriver.

Hæftet kan læses fra en ende af som en lærebog, men kan også bruges som opslagsværk. Jeg tager udgangspunkt i webbets muligheder og begrænsninger, ikke i specifikke programmers opbygning. Jeg forklarer dog hvordan man bruger forskellige editorer, især [Adobe PageMill](#), men også [Claris Home Page](#), [Netscape Composer](#) og [Microsoft Front Page](#).

### Disposition

Hæfter er opdelt i fire dele. Første del starter fra grunden og forklarer det nødvendige for at lave en simpel website og få den til at køre. Anden del går i dybden med forskellige aspekter af webdesign. Tredje del omhandler mere komplicerede emner som multimedia og interaktivitet, som jeg ikke forventer alle vil have lyst eller tid til at sætte sig ind i.

Fjerde del – måske den vigtigste del af hæftet – handler om den overordnede planlægning bag en website: planlægning, struktur, navigation, tekstopbygning, sidelayout og typografi.

Afsnittet kommer til sidst, fordi det kræver at man har et overblik over de tekniske muligheder og begrænsninger. Bagest i hæftet findes en ordforklaring.

### Online-supplement

Hæftet har et online-supplement på adressen [www.knowware.dk/dk/webdesign](http://www.knowware.dk/dk/webdesign), hvor der findes opdateringer, uddybninger, ofte stillede spørgsmål, links mv.

### Skriv til mig

og fortæl hvad du synes om dette hæfte, så bliver jeg glad. Min adresse er [olavk@pip.dknet.dk](mailto:olavk@pip.dknet.dk)

*Olav Junker Kjær*

## Del 1 – Webdesign for begyndere

### Hvad er webdesign?

Webdesign er en kunst, der består af følgende discipliner:

**Planlægning og Strukturering** – Man skal gennemtænke, hvorledes websiten skal opbygges og struktureres, så brugere nemt kan finde rundt og kan finde, hvad de søger. Man skal også designe en 'brugerflade' til websiten, dvs. tekst, ikoner eller knapper som hjælper brugeren til at navigere.

**Udarbejdelse af de enkelte sider** – De enkelte sider laves i en webeditor, hvor man sætter tekst op og formaterer den og indsætter billeder. Det er også i webeditoren man laver links mellem de forskellige sider.

**Udgivelse** – Når websiten er færdig, skal den lægges ud på internettet, så andre har adgang til den.

**Vedligeholdelse og opdatering** – en website er sjældent et afsluttet projekt. Ofte skal materialet jævnligt ændres og opdateres.

### Er det svært?

Har man prøvet at bruge et almindeligt tekstbehandlingsprogram, er det ret ligetil at lære at bruge en webeditor. Der er dog nogle forhold, som man skal vænne sig til:

- Websider giver (endnu) ikke den samme kontrol over typografi og layout, som tekstbehandlingsprogrammer. Man må altså affinde sig med mere primitive muligheder.
- Man må tage højde for modtagerens udstyr. Der findes mange forskellige typer computere på internettet, med forskellige skærmstørrelser, grafikopløsninger, skrifttyper etc., ligesom der findes forskellige browsere med forskellige muligheder. Det er altså ikke nok at websiderne ser rigtige ud på ens egen computer, man må tage højde for at de også kan fungere på andre systemer.
- Muligheden for at lave links mellem sider og til andre websites, er noget man ikke rigtigt kender fra tekstbehandlingsprogrammer. Til gengæld er det noget af det mest spændende ved nettet.
- Desuden er udgivelsen af websiten naturligvis ikke noget man kender fra tekstbehandling. Der er dog ikke specielt kompliceret.

Webbet giver desuden mange muligheder for at lave 'effekter' på websites – multimedia og interaktivitet, som kræver programmering og/eller kunstnerisk talent i en eller anden grad for at udvikle.

Ingen af disse effekter er dog nødvendige for at lave en professionel og velfungerende website.

Det sværeste ved webdesign er nok at organisere websitens informationer i en logisk og funktionel struktur, samt at layoute siderne, så de er pæne og læsevenlige. Det kræver planlægning, overvejelser og en forståelse af mediets unikke muligheder og begrænsninger. Husk, at det drejer sig om et nyt medie, som adskiller sig grundlæggende fra de kendte. Frygt dog ikke – det hele er med i dette hæfte.

### Hvordan foregår det?

For at andre kan få adgang til en website over internettet, skal siden ligge på en webserver – en computer med en direkte opkobling til internettet og det nødvendige kommunikationssoftware. Normalt lejer man sig plads på en webserver hos en internetudbyder (se side 20).

Man udvikler og afprøver dog normalt websiten på en almindelig PC eller Mac, og først når siden er helt færdig, udgives den (dvs. flyttes ud på serveren). Derfor kan man også prøve det meste i dette hæfte uden at have en serverplads.

### Hvilken computer har man brug for?

Windows 95/NT eller Macintosh er begge udmærkede systemer til webdesign. Jeg vil ikke anbefale at man udvikler på operativsystemer som Windows 3.1, DOS eller OS/2 – ikke fordi der er noget i vejen med dem, men fordi nærmest al nyt internetsoftware skrives til Windows 95/NT eller Mac.

Der er en tendens til at nye programmer kommer først til Windows 95/NT og med nogle måneders forsinkelse til Mac, så vil man være helt på forkant med udviklingen, er Windows 95/NT ikke til at komme udenom. Dette hæfte tager udgangspunkt i Windows 95, men det meste foregår på samme måde på en Mac. (*Højre museknap* på en PC svarer på en Mac for det meste til at holde museknappen nede i et stykke tid).

Hvor kraftig skal computeren så være? De forskellige programmer, der bruges til webdesign, er ikke så krævende, men man vil ofte have flere forskellige programmer kørende samtidigt. RAM er derfor vigtigere end processorens styrke og hastighed.

Jeg har selv lavet webdesign på professionelt niveau i et års tid på en 486 med 12 Mb RAM. Det er måske nok i underkanten, og jeg vil absolut ikke anbefale mindre, men mere behøver man altså

strengt taget ikke. En god skærm (mindst 15") og et godt tastatur er generelt mere værd end processorstyrke. Det er naturligvis nødvendigt med en internetforbindelse på computeren.

### Hvilke programmer skal man bruge?

Der findes ikke ét program, der dækker alle webdesignerens behov. En webdesigner har en værktøjskasse af forskellige programmer, og kunsten er at vide, hvornår man skal anvende hvilket. De vigtigste programmer er:

- Et par browsere: [Netscape Navigator](#) og [Microsoft Internet Explorer \(MSIE\)](#).
- En webeditor. Webeditoren er det program, der kan skabe websiderne. Gode webeditorer er f.eks. [Adobe Pagemill](#), [Netscape Composer](#), [Claris Home Page](#) eller [Microsoft Frontpage](#).
- Et tekstbehandlingsprogram, f.eks. [Microsoft Word](#), til forbehandling af større tekster.
- Et grafikprogram, f.eks. [Adobe Photoshop](#) eller [Paint Shop Pro](#) til forbehandling af grafik.

## HTML

Websider er filer i dokumentformatet *HTML*. HTML står for *HyperText Markup Language*, er et format for dokumenter, der kan indeholde formateret tekst og hyperlinks og kan have billeder og andre elementer indsat.

Som webdesigner er det vigtigt at være opmærksom på forskellene på HTML og almindelige tekst/grafik dokumenter: Et tekstbehandlingsprogram er lavet til at udvikle dokumenter, som oftest skal printes ud på papir. HTML er derimod specielt udviklet til at overføre information over netværk – information som først og fremmest skal læses og navigeres på computere.

### Internettet & åbne standarder

Styrken ved internettet er, at det er baseret på åbne og platformsuafhængige standarder. Udtrykket *åben standard* er godt nok lidt diffust, men normalt menes:

- standarden er ikke ejet af nogen, og ingen har patent eller ophavsret på den
- standarden koordineres af en ikke-kommerciel organisation
- specifikationerne for standarden er gratis tilgængelig for alle
- alle kan lave software som er kompatibel med standarden – både kommercielt og ikke-kommercielt

*Platformsuafhængig* vil sige: som ikke er knyttet til eller afhængig af bestemt hardware eller et bestemt operativsystem. Internettet er baseret på platformsuafhængige standarder, og kan derfor virke på tværs af PC, Mac, Amiga, eller UNIX. Modsat virker de fleste spil, programmer, CD-ROM'er o.lign. kun på en bestemt platform, eller skal udvikles specielt til hver enkelt platform. Det er værd at bemærke, at det er standarderne, der er åbne og platformsuafhængige – de enkelte programmer, som benyttes i forbindelse med internettet, er normalt både kommercielle og knyttede til en bestemt platform.

### HTML er platformsuafhængigt

Websider skal kunne læses på computere af enhver art – systemer med vidt forskellige operativsystemer, skærmstørrelser og grafiske muligheder. Dette er netop styrken ved webbet – man er ikke længere fanget i inkompatible systemer: alle kan kommunikere med alle.

Men det betyder også, at man aldrig kan være helt sikker på, hvordan en anden computer viser en bestemt side. For eksempel varierer forholdet mellem skriftstørrelser, vinduesstørrelse og skærmopløsning fra system til system. Man vil aldrig kunne få samme kontrol over websiders layout, som man har over papirsider.

Formateringskoderne i HTML angiver derfor ikke tekstdelens eksakte udseende ("denne sætning er 24pkt Helvetica"), men bruger mere overordnede angivelser ("denne sætning er en overskrift på 3. niveau"). I princippet er det op til modtageren, hvordan disse strukturer skal tegnes. I praksis viser de mest udbredte browsere dog HTML-strukturerne nogenlunde ens. Man kan altså godt lave layout i HTML, og de fleste vil se *nogenlunde* det layout, man har lavet.

### HTML er et åbent format

HTML er et åbent format. Organisationen *World Wide Web Consortiet (W3C)* koordinerer HTML-standard, dvs. udgiver officielle specifikationer, og anbefaler hvilke standarder man skal bruge, men deres betydning afhænger helt og holdent af om webdesignere og softwareproducenter vælger at følge deres anbefalinger eller ej. Det er for eksempel lykkedes Netscape at indføre en række selvudviklede udvidelser til HTML, som nu er standard.

HTML er i dag mærket af den såkaldte *browserkrig* – et kapløb mellem to førende udvikle-

re af browsere, Netscape og Microsoft, om at lave den browser, der "kan mest". Dette har medført en række nye, ikke standardiserede, udvidelser til HTML, som måske kun virker på den ene browser eller virker lidt forskelligt på dem. Som webdesigner er det derfor om at holde hovedet koldt, når man eksperimenterer med udvidelser til HTML.

### HTML virker altid

HTML er *degradable*, dvs. man kan tilføje nye koder til HTML, uden at gamle browsere holder op med at fungere. Browsere ignorerer simpelthen HTML-koder, de ikke kender, men behandler resten af HTML'en normalt. På den måde kan en gammel browser godt vise et dokument, der indeholder nyere eller ikke-standardiserede udvidelser til HTML.

## Om browserne

Browseren er så at sige webdesignerens publikum, så det er vigtigt at kende dem:

### Navigator 2

Kan ikke downloades mere, men er stadig meget udbredt. Kan siges at være laveste fællesnævner.

### Navigator 3

Den mest udbredte browser.

### Navigator Gold 3

[Navigator 3](#) udvidet med en webeditor.

### Netscape Communicator

Afløseren til [Navigator 3](#). Egentlig en "suite" af internetsoftware, hvoraf browseren [Navigator 4](#) kun er et enkelt komponent. Et andet komponent er editoren [Composer](#), som er afløseren til [Navigator Gold](#).

Ofte kaldes [Netscape Navigator](#) bare for [Netscape](#). For at undgå forvekslinger skriver jeg dog [Navigator](#) om browseren, [Composer](#) om editoren og [Netscape](#) om firmaet.

Netscape software er shareware (kan bruges gratis i 90 dage) og kan downloades fra [www.netscape.com](http://www.netscape.com)

### Microsoft Internet Explorer (MSIE) 3

Netscapes værste konkurrent. Svarer nogenlunde til [Navigator 3](#). MSIE er freeware og kan fås fra [www.microsoft.com/ie](http://www.microsoft.com/ie)

Efterfølgeren, Microsoft Internet Explorer 4 (MSIE 4) er nov. 97 kommer til PC, kommer senere til Mac.

Der er naturligvis andre browsere i omløb, f.eks. Apples [CyberDog](#), tekstbrowseren [Lynx](#), testbrowseren [Amaya](#), individualistbrowseren [Opera](#), nostalgibrowseren [Mosaic](#) og mange andre. Ingen af disse browsere har dog nogen større udbredelse. Der findes også ældre versioner af [Navigator](#) og [MSIE](#), men de er ved at være ret sjældne.

Enhver webdesigner bør mindst have [MSIE 3](#) og [Navigator 3](#). Også selvom man har en nyere browser som [Navigator 4](#), er det vigtigt at kunne afprøve siderne i [Navigator 3](#) og [MSIE 3](#).

## Editoren

Webdesignerens vigtigste værktøj er *webeditoren*. En webeditor (kaldes også *HTML-editor*) virker umiddelbart som en mellemting mellem en browser og et tekstbehandlingsprogram: man ser sine websider nogenlunde, som de vil se ud i en browser, men man kan skrive tekst ind på siderne, redigere og formatere.

### Hvilken editor skal man vælge?

Der findes en række editorer til forskellige behov:

**Adobe PageMill 2.0** er efter min mening den bedste webeditor til mindre websites. For tiden kan en prøveversion af **PageMill** hentes fra [www.adobe.com](http://www.adobe.com). Programmet koster omkring 900 kr. inklusiv moms.

**Clarish Home Page** er ikke helt så intuitivt som **PageMill**, men alligevel et glimrende program. En prøveversion kan for tiden hentes fra [www.clarishhomepage.com](http://www.clarishhomepage.com)

**Netscape Composer** er et komponent i **Netscape Communicator** (**Netscape Navigator 4**). En tidligere version hed **Netscape Gold**. **Composer** er shareware – man må bruge programmet gratis i 90 dage – og er desuden gratis at bruge for studerende. Kan hentes fra [www.netscape.com](http://www.netscape.com). Programmet er ikke lige så avanceret som **PageMill**, men praktisk, fordi det er tæt sammenbygget med browseren.

**Microsoft FrontPage**, en udmærket editor som er specielt velegnet til at arbejde med større websites, der består af mange sider. Man kan få en prøveversion, hvis man melder sig ind i Microsofts **Site Builder Network** (det er gratis): [www.microsoft.com/sitebuilder](http://www.microsoft.com/sitebuilder). Koster omkring 1500 kr. inklusiv moms.

FrontPage er endvidere kommet i en "light" udgave, kaldet FrontPage Express. FrontPage Express er et gratis komponent til MSIE 4.

Den har ikke så mange faciliteter som storebror, men de simple ting fungerer ens i begge versioner.

Alle fire webeditorer findes til både PC og Mac.

Editorer som disse kaldes til tider for **WYSIWYG**-editorer. **WYSIWYG** betyder "What You See Is What You Get" og angiver, at man i denne type editorer ser siden nogenlunde, som den kommer til at se ud på nettet. Dette er et stort fremskridt fra

den forrige generation af editorer, hvor man arbejdede direkte i HTML-koden og indsatte formateringskoder manuelt. Ville man se, hvordan siden så ud, skulle man indlæse den i en browser.

Den mest populære af denne type editor er Hot Dog ([www.sausage.com](http://www.sausage.com)). Den slags vil jeg kun anbefale til specielt teknisk interesserede eller masochistiske webdesignere.

De nyeste versioner af **Word**, **Wordperfect** og **PageMaker** kan også gemme dokumenter som HTML. De har dog (efter min mening) stadig langt at nå, før de kan måle sig med de "fødte" webeditorer.

Jeg tror dog, at fremtidige versioner af de ovennævnte programmer kan blive velegnede til produktion af websider på mellemniveau (til websider på professionelt niveau vil man nok altid bruge specialiserede webeditorer).

De steder, hvor jeg beskriver, hvordan man konkret udfører noget i editorerne, beskriver jeg det for alle fire ovennævnte editorer. Jeg har dog forsøgt at fokusere på selve webbet og HTML's muligheder og begrænsninger, fremfor på teknikaliteter ved de enkelte browsere. Det gør også, at man kan få udbytte af hæftet, selvom man bruger en anden editor end en af de fire.

## I gang

### Webmappen

Før du laver din første webside, skal du lave en *webmappe*. Det er en mappe, som alle de filer, som din website består af: HTML-filer, grafikfiler mv., skal ligge i. Filerne må gerne ligge i andre mapper under webmappen, men ikke andre steder på computeren. Det er indholdet af webmappen, som flyttes til serveren, så den bør ikke indeholde andet end webdokumenter.

Webmappens navn er egentlig ligegyldigt, så du kan f.eks. kalde den "webmappe". Det er også ligegyldigt, hvor på computeren den ligger, men det nemmeste er at have den liggende på skrivebordet eller i harddisk-rod. Hvis man ændrer et filnavn eller flytter en fil til en anden mappe *efter* at have lavet links til eller fra filen, holder linkene op med at virke. Så lad være med det.

### Regler for filnavne

Der er regler for fil- og mappenavne på nettet:

- Navnene må kun indeholde engelske bogstaver, tal samt bindestreg (-) og underscore (\_). De må altså *ikke* indeholde mellemrum, æ ø å, eller specialtegn (!"#? etc.).
- Filnavnet skal ende med et "efternavn", som angiver filtypen. Efternavnet er adskilt fra selve navnet med et punktum. Efternavnet for HTML-dokumenter skal være `.html`, som i `min_foerste_webside.html`. Det er vigtigt at efternavnene er korrekte. Mappenavne har ingen endelse (og intet punktum).
- Brug kun små bogstaver.

Reglerne for filnavne varierer fra server til server, men de ovennævnte regler er "laveste fællesnævner", så følger du dem, skulle der ikke opstå problemer.

Forsiden til din website skal have et specielt filnavn. Dette er normalt `index.html`, men det kan variere fra server til server, så spørg den systemansvarlige.

Visse ældre servere accepterer kun de gamle 8.3-DOS-filnavne. Så må du bruge efternavnet `.htm` i stedet for `.html`. Hvis det er tilfældet, vil du få det at vide af den systemansvarlige.

### Den første side

Når du starter editoren, opretter den en blank side. Du kan gemme siden på normal vis med **File|Save**. Husk at gemme den i webmappen (eller i en mappe under webmappen), og husk `.html` som efternavn.

Du kan oprette nye blanke sider med **File|New**. I **FrontPage** og **Composer**, kan du også vælge at få en ny side baseret på en skabelon (f.eks. standard personlig hjemmeside, standard klubside mv.), som du så kan modificere. Disse skabeloner er generelt ret kiksede og smagløse.

Når du har oprettet en ny side, ser du på et tomt, gråt eller hvidt papir. Øverst i vinduet er desuden menulinien og et par værktøjsbjælker. Du kan umiddelbart begynde at skrive tekst ind på siden, ganske som i et tekstbehandlingsprogram.

### Specielt for Microsoft FrontPage

Microsoft **FrontPage** har en lidt speciel måde at fungere på. Programmet består af to komponenter: **FrontPage Editor** er en webeditor ligesom de andre beskrevne, mens **FrontPage Explorer** er en "siteeditor", som indkapsler hele ens website, altså webmappen og alt i den. Man skal så finde og redigere de enkelte websider inde fra **FrontPage Explorer**.

Internt i webmappen laver **FrontPage Explorer** en række specielle mapper og holder et underligt dobbeltbogholderi med to udgaver af alle filer, men den skjuler disse finurligheder for brugeren. Dette system gør at **FrontPage** kan have en række faciliteter til administration af web-siten som helhed, som de andre editorer ikke har. Men man skal være opmærksom på dette:

- Man skal ikke manuelt oprette en webmappe. I stedet vælger man **File|New FrontPage Web** i **FrontPage Explorer**.
- Man kan ikke uden videre kopiere filer ind i webmappen. I stedet skal man vælge **File|Import** for at importere filen ind i systemet.
- Man skal ikke kopiere hele webmappens indhold til serveren. I stedet skal man bruge hjælpeprogrammet **FrontPage Publishing Wizard** som kun uploader det nødvendige.

## Tekst

Man skriver tekst som i et tekstbehandlingsprogram. Teksten ombrydes automatisk, når man når til sidens kant. Man laver nyt afsnit med *return*-tasten. Afsnit vises adskilt med en tom linie. Hvert afsnit har en *type*, som afgør afsnittets udseende.

Afsnitstypen vælges på en rullemenu på værktøjsbjælken. Normalt står den på **Paragraph** (eller **Normal**), som er almindelig brødtekst, men ruller du menuen ned, vil de se en række andre indbyggede afsnitstyper. I første omgang er det nok overskriftstyperne (der er seks overskriftsniveauer), du vil have brug for. Vælger du et nyt afsnitsformat, vil hele afsnittet skifte format.

### Afsnitsformater

Afsnitsformateringer omfatter et eller flere afsnit.

#### Paragraph

er afsnit med almindelig tekst. (Hedder også *Normal*)

#### Heading 1-6

er de indbyggede overskrifter, hvor H1 er den største og H6 den mindste. De vises normalt med fed tekst i forskellige størrelser. Lister er blokke som omfatter flere afsnit:

#### Bulleted List

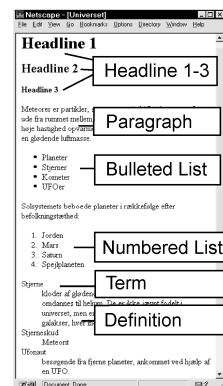
er en liste, hvor de enkelte punkter vises indrykket, og med en kugle ud for hver. **Directory List** og **Menu List** vises ligesom **Bulleted List**.

#### Numbered List

er en liste, hvor de enkelte punkter er nummererede.

#### Term

veksler med **Definition** i definitionslisters. **Term** er venstrestillet og bruges til ordet, der skal defineres. **Definition**-afsnit er definitionerne på ordet, og er lidt indrykket i forhold til **Term**.



Afsnittene/punkterne i lister vises normalt kun adskilt med almindelig linieafstand.

### Address

angiver, at et afsnit er en adresse. Viser normalt kursiv og indrykket

### Preformatted (eller bare Formatted)

er et specielt format beregnet til visning af for eksempel computerkode og modernistisk poesi, hvor indrykning og lineskift har en indholdsmæssig betydning. Modsat de andre afsnitstyper ombrydes **preformatted** tekst ikke efter vinduets bredde; man skal specifikt trykke *return*, hvor man vil have lineskift. Man kan lave indrykning ved at bruge flere mellemrum efter hinanden (hvilket man ikke uden videre kan i almindelige afsnit). Afsnittene vises med skrivemaskineskrift.

Afsnitsformatering	
PageMill	Menuen <b>Format</b>
Composer	Menuen <b>Format</b> eller <b>Paragraph/List Properties</b>
FrontPage	<b>Format Paragraph</b> eller <b>Paragraph Properties</b>
Claris Home Page	Menuen <b>Format</b>

På værktøjsbjælken findes desuden tekstjustering (højre, venstre, centreret) og indrykning (og udrykning, som sådan set bare ophæver en indrykning). Disse formateringer gælder også for hele afsnit af gangen.



## Tegnformatering

Disse formateringer behøver ikke gælde for et helt afsnit, men kan bruges på enkelte ord eller tegn:



<b>Italic (I)</b>	Kursiv
<b>Bold (B)</b>	Fed
<b>Teletype (TT)</b>	Angiver at teksten skal vises med en skrivemaskineskrift, dvs. en skrift, hvor alle tegn er lige brede. (Normalt skrifttypen <i>Courier</i> ).
<b>Font Size</b>	Skriftens størrelse. Kan være fra 1 til 7, hvor 3 er det normale. Nogle editorer tæller fra -2 til +4, men forskellen er ens.
<b>Font Color</b>	Specificerer farven for et stykke tekst

Andre tegnformateringer (ikke så almindeligt brugt):

<b>Big</b>	større tekst; det samme som <b>font size</b> +1
<b>Small</b>	mindre tekst; det samme som <b>font size</b> -1
<b>Subscript</b>	sænket skrift (som i H <sub>2</sub> O)
<b>Superscript</b>	hævet skrift (som i E=mc <sup>2</sup> )
<b>Strike</b>	tekst der er streget ud ( <del>hvad man så end skal bruge det til</del> )
<b>Underline</b>	understreget tekst (NB! Pas på med at bruge den, forveksles let med links)

Der findes også tegnformateringer, som ikke decideret markerer, hvordan ordene skal se ud, men angiver deres *funktion* i teksten. Man kunne kalde dem betydningsmæssige markeringer:

	<i>betydning</i>	<i>udseende</i>
<b>Emphasis</b>	fremhævelse	kursiv
<b>Strong</b>	stærk fremhævelse	fed.
<b>Code</b>	til computerkode	skrivemaskineskrift
<b>Citation</b>	til citerede kilder eller referencer	kursiv
<b>Sample</b>	eksempel på output fra en computer,	skrivemaskineskrift
<b>Keyboard</b>	noget brugeren skal taste ind,	skrivemaskineskrift
<b>Variable</b>	variabler eller argumenter til kommandoer.	kursiv i <a href="#">Navigator</a>
<b>Definition</b>	for et ord der bliver defineret samme sted i teksten	kursiv i <a href="#">MSIE</a>

Ændre skriftformat	
<a href="#">PageMill</a>	Menuen <a href="#">Style</a>
<a href="#">Netscape Composer</a>	<a href="#">Format</a>   <a href="#">Character Properties</a> , fanebladet <a href="#">Character</a>
<a href="#">FrontPage</a>	<a href="#">Format</a>   <a href="#">Font</a> eller <a href="#">Font Properties</a>
<a href="#">Claris Home Page</a>	Menuen <a href="#">Style</a>

## HTML's idehistorie

Som det ses indeholder HTML både markeringer, der beskriver tekstdeles *betydning* (overskrift, adresse, citater mv.) og markeringer, der beskriver tekstdelens *udseende* (skriftstørrelse, farve, mv.). Faktisk er de betydningsmæssige markeringer det oprindelige; der var kun betydningsmæssige markeringer i den tidligste version af HTML. Der er to store fordele ved at basere et dokumentformat på betydningsmæssige markeringer:

1. Betydningsmæssige markeringer er ikke afhængige af et bestemt medie: Markeringen *Italic* (kursiv) giver ikke megen mening på en terminal, hvor alt vises med skrivemaskineskrift, eller hvis teksten skal læses op for en blind. Markeringen *Emphasis* (fremhævet) kan derimod viderebringes i alle medier, f.eks. via store bogstaver på en terminal eller tonefald i en stemmegenerator.
2. Søge- og indekseringsmaskiner kan kortlægge dokumenter efter de indholdsmæssige markeringer – lave indekser genereret ud fra overskrifter eller efter, hvor ord defineres, lave citatlistor mv. Den slags er utroligt nyttigt på større websites, for eksempel universiteter, og den mulighed er netop en af de største styrker ved digital information.

Da webbet begyndte at blive mere udbredt, slog Nescapes browser igennem ved at understøtte formateringer af tekstens udseende: skriftstørrelse, tekst- og baggrundsfarver, mv. Netscape så tidligt, at jo mere udbredt og jo mere kommercielt webbet ville blive, jo større interesse ville der blive for grafiske og layoutmæssige muligheder i HTML.

*World Wide Web Consortiet* (W3C), som fastlægger HTML-standarderne, ville først ikke vide af Nescapes udvidelser, og arbejdede på deres egen HTML 3.0. Men folket (det vil sige webdesignere og browserbrugere) valgte Nescapes vej, og W3C så sig derfor pludselig uden kontrol over HTML's udvikling. W3C fik først indflydelse igen, da de bed i det sure æble og opgav HTML 3.0 og i stedet foreslog HTML 3.2, en HTML standard, der inkorporerede de mest populære og gennemtænkte af Nescapes udvidelser.

W3C prøver nu at bringe HTML tilbage på ret kurs via *Style-sheets*, en udvidelse til HTML, hvor man både får de tekniske fordele ved betydningsmæssige markeringer og udvidet visuel/typo-

grafisk kontrol *uden* dobbelt arbejde. Style sheets er dog endnu ikke bredt understøttede.

## Tekst og tegn

HTML understøtter alle tegn i ISO-8859-1 tegnsættet (indeholder alle bogstaver, som bruges i vesteuropæiske sprog, vistnok bortset fra gælisk), det vil sige blandt andet *ä, Ñ, é, ÿ, Ç*, og ikke mindst *æ, ø* og *å*.

Der findes desværre ikke en pålidelig måde at lave dansk tankestreg på, så man må bruge almindelig bindestreg. Danske anførselstegn („gåseøjne”) er heller ikke understøttet. Man må bruge tommer-tegn (") eller franske anførselstegn (» «). Man kan ikke umiddelbart bruge matematiske symboler eller tegn fra andre alfabeter i HTML.

## Linieskift

Man kan med *Shift+Return* lave et tvunget lineskift, som ikke er et afsnitsskift. Lineskift adskiller sig fra afsnitsskift ved at der ikke kommer den dobbelte linieafstand, og at teksten før og efter lineskiftet stadig tilhører samme afsnit. Man kan altså ikke lave lineskift mellem f.eks. en overskrift og brødtekst, fordi det er to forskellige afsnitstyper.

([Netscape Composer](#) har i øvrigt en særhed: *Return*-tasten indsætter lineskift i stedet for afsnitsskift, når man skriver almindelige tekstafsnit, men indsætter afsnitsskift ved skift til andre afsnitstyper).

## Mellemrum

I HTML kan man ikke bruge mellemrumstasten til at styre tekstindrykning. HTML anerkender kun mellemrum som adskillelse mellem ord: flere mellemrum efter hinanden betragtes som et enkelt, mellemrum først og sidst i et afsnit (altså før eller efter selve teksten) ignoreres, og tomme afsnit ignoreres.

Hvis du alligevel vil bruge mellemrum typografisk, for eksempel til at indrykke den første linie i et afsnit, kan du bruge specialtegnet *nonbreaking space*, som fås med *shift+mellemrum*. ([FrontPage](#) og [Claris Home Page](#) understøtter desværre ikke *nonbreaking space*, dvs det skal indsættes som HTML). Tabulator er ikke understøttet i HTML.

## Øvelser

Du skulle nu være i stand til at skrive et almindeligt tekstdokument. Prøv, for at blive fortrolig

med editoren, at lave en side med overskrifter og brødtekst, nogle lister, indrykket tekst og centreret tekst. Prøv desuden at lave nogle ord fede og kursiv eller med andre størrelser og farver. Prøv at lave nogle tekstafsnit, der fylder flere linier, og se hvordan de ombrydes, når du ændrer på vinduesstørrelsen.

## Elementer og egenskaber

Alle en websides elementer har visse *egenskaber* – et billede har størrelse, siden har baggrundsfarve etc. Alle editorer har en måde at se og ændre elementernes egenskaber, men det foregår ret forskelligt; her viser editorerne deres personlighed.

**FrontPage** og **Netscape Composer** har en række forskellige dialogbokse til forskellige elementers egenskaber, mens **PageMill** og **Claris HomePage** begge har en “flydende” boks, som indeholder det aktuelle elements egenskaber og skifter indhold alt efter det markerede element.

### FrontPage

I **FrontPage** klikker man med *højre museknap* på siden eller på et element på siden. Så kommer en lille menu frem, hvor man kan vælge **properties** for de aktuelle elementer. Der vil normalt være flere at vælge imellem. **Page Properties** er egenskaber for hele siden. **Font Properties** er for skriften (evt. for et markeret område), **Paragraph Properties** er egenskaber for hele afsnit.

Hvis man klikker på et billede, vil **Image Properties** være tilgængelig, klikker man på en celle i en tabel vil **Table Properties** og **Cell Properties** være tilgængelig, og så videre. Når man vælger en af **Properties** punkterne, kommer en dialogboks, hvor man kan se og ændre det valgte elements egenskaber. Properties for det valgte element (billede, formularfelt mv.) ligger også nederst på **Edit**-menuen. Properties for tabeller og tabelceller ligger desuden på **Table**-menuen.

### Netscape Composer

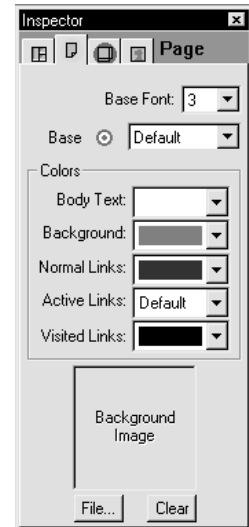
I **Netscape Composer** klikker man med *højre museknap* på siden eller på et element på siden, hvorved en lille menu kommer frem, hvor man kan vælge *properties* for de aktuelle elementer. Der vil normalt være flere at vælge imellem. **Page Properties** er egenskaber for hele siden. **Character Properties** er for skriften (evt. for at markeret område), **Paragraph/List Properties** er egenskaber for hele afsnit.

Hvis man klikker på et billede, vil **Image Properties** være tilgængelig, klikker man i en tabel vil **Table Properties** være tilgængelig, og så videre. Når man vælger en af **Properties** punkterne, kommer en dialogboks, hvor man kan se og ændre det valgte elements egenskaber. De samme *properties*-valgmuligheder som på *højre museknap* er også tilgængelig fra menuen **Format**.

### PageMill

**PageMill** har en “flydende boks”, der hedder *Inspector*. Hvis den er usynlig slås den til med **View|Show Inspector**. I *Inspector* kan man altid se og ændre egenskaberne ved det aktuelle element.

*Inspector* har fire faneblade: **Frame** er egenskaber for den aktuelle frame (er kun aktiv, hvis siden er et *frameset*), **Page** er som sagt egenskaber som gælder for hele siden – baggrundsfarve og den slags, **Form** er egenskaber for den aktuelle formular (er kun aktiv, hvis der er en formular på siden), og **Object** er egenskaber for det aktuelle objekt: billede, tabel, formularfelt eller lignende. Hvis der intet er markeret, står *Inspector* på **Page**. Markerer man et billede, en tabel eller lign. skifter den til det markerede element.



### Claris Home Page

Et elements egenskaber ændres i den “flydende boks” *Object Editor*. Er den usynlig, fås den frem med **Window|Show Object Editor**. *Object Editor* har tre faneblade: **Basic**, med de almindelige egenskaber, **Advanced** med de mere specielle, og **Both** med begge på en gang (for folk med nok plads på skærmen).

## Farver og baggrund

Dette er egenskaber for hele siden. **Background** er baggrundsfarve og **Body text** er tekstens farve (de steder hvor den ikke specifikt er givet en anden farve). **Normal links** er farven på links (normalt blå), **Visited Links** er farven på links, som peger på sider man har besøgt (normalt lilla), og **Active Links** er farven, som links får i det øjeblik man klikker på dem. **Default** betyder, at farven ikke specificeres i dokumentet. Så ser brugeren siden med sin browsers farveindstillinger – normalt sort tekst på grå baggrund, fordi det er den indstilling browseren har fra starten.

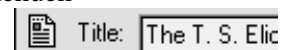
Ændre sidens egenskaber	
Adobe PageMill	I Inspector, på fanebladet <b>Page</b> .
Netscape Composer	<b>Format Page Colors and Properties</b> , eller <b>Page Properties</b>
Microsoft FrontPage	<b>File Page Properties</b> eller <b>Format Background</b> eller <b>Page Properties</b>
Claris Home Page	<b>Edit Document Options</b>

I PageMill kan få et panel med 16 farver frem via **View|Show Color Panel**. Man kan trække farverne fra panelet til felterne i Inspector. Farver kan også trækkes til markeret tekst.

## Titel

Hvert HTML-dokument har en titel. Et dokumentets titel er *ikke* det samme som dets filnavn. Filnavnet er det kun dig selv der ser, titlen er derimod offentlig: den ses i browserens titelbjælke, og hvis man laver et bookmark på siden, er det titlen der kommer på bookmarks-menuen

I PageMill skrives dokumentets titel i feltet **Title**: over vinduet. I de andre editorer indsættes den samme sted som sidens andre egenskaber (se ovenfor).



Titlen kan indeholde mellemrum og specialtegn. Siden med filnavnet `min_foerste_webside.html` kan have titlen `Min første webside!`. Titlen må ikke være længere end 63 tegn. Titlen bør være oplysende og kunne stå alene, da titler optræder i søgemaskiner, indekser og i bookmarks. “Side 1” eller “Indledning” siger ikke så meget som for eksempel “T.S.Eliot forum – indhold”. Brug ikke ordene *website* eller *homepage* i titlen.

## Links

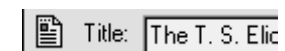
For at lave en link skal du naturligvis have to filer: siden du linker fra, og siden du linker til. Siden, du linker til, skal være gemt med et gyldigt filnavn. Start med at markere den tekst, som skal gøres til en link.

### Link til lokal fil

Træk filen, du vil linke til, fra sin mappe eller fra **Windows Explorer** ind over den markerede tekst, og slip. I **FrontPage** kan du trække en fil fra **FrontPage Explorer**.

Indsætte link til lokal fil	
PageMill	<b>File Place</b>
Composer	<b>Insert Link</b> derefter knappen <b>Choose File</b>
FrontPage	<b>Insert Hyperlink</b> , derefter knappen <b>Browse</b> for at vælge en fil.
Claris Home Page	<b>Insert Link To File</b>

I PageMill kan man desuden lave en link til en anden side, der også er åben i PageMill, ved at trække side-ikonen fra titelbjælken på den side man vil linke til, henover teksten der skal være link, og slippe.



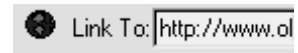
## Links til andre websites

Man kan naturligvis også lave links til sider ude på nettet. Det kaldes *eksterne links*. Det nemmeste er, hvis man har siden, man vil linke til, åben i browseren. Så trækker man bare link-ikonet (den lille kædestump ved siden af **Location**-feltet på **Navigators** bjælke) ind på siden i webeditoren og slipper. Man kan også trække en link fra browservinduet ind i editoren. Man kan også trække et bookmark fra **Navigators** bookmark-vindue (fås via **Windows|Bookmark**) ind i editoren.



## Indtaste URL'en manuelt

Man kan naturligvis også lave en link ved manuelt at indtaste URL'en. (Husk en komplet URL starter med `http://`).



Indsætte URL for en link	
PageMill	klik i feltet <b>Link to:</b> nederst i vinduet, skriv URL'en og tast <i>return</i>
Composer	<b>Insert Link</b>
FrontPage	<b>Insert Hyperlink</b>
Claris Home Page	<b>Insert Link to URL</b>

## Slette en link igen

PageMill	marker hele linkens tekst (kan gøres ved at tripple-klikke på link-teksten), slet indholdet i <b>Link to</b> -feltet, og tast <i>return</i>
Composer	<b>Link Properties</b> og knapper <b>Remove Link</b>
FrontPage	<b>Hyperlink Properties</b> og knappen <b>Clear</b>
Claris Home Page	Gøres i link editoren (som fås frem med <b>Window Show Link Editor</b> ), ved knappen <b>Remove Link</b>

## Relative og absolutte URL'er

En link kan henvise til et andet dokument på to måder: Ved en absolut URL, som er en komplet adresse på dokumentet, eller en relativ URL som fortæller hvor dokumentet ligger i forhold til dokumentet der linkes fra. Absolutte URL'er bruges til at linke til dokumenter på andre servere (for eksempel `http://www.jubii.dk/hurra.html`); relative URL'er bruges til at linke mellem dokumenter indenfor samme website (for eksempel blot `digte/eliot.html`). Absolutte URL'er indeholder både `http:` og servernavn, relative indeholder kun filnavn og eventuelle mappenavne (hvis dokumenterne ligger i forskellige mapper). (Læs i øvrigt mere om webadresser i *Ud på World Wide Web*, afsnittet *Alle dokumenter har en adresse*).

Editoren bruger automatisk relative URL'er, hvis du linker mellem filer i din webmappe.

## Ankre – links internt i en webside

En link går normalt til toppen af en side, men man kan også lade en link gå til et bestemt sted nede på siden. Det kræver at man indsætter et såkaldt *anker* i teksten, der hvor linket skal gå til. Ankeret indstilles ved markøren, ved at:

Indsætte et anker	
PageMill	<b>Edit Insert Invisible Anchor</b>
Netscape Composer	<b>Insert Target</b> (Composer kalder ankre for <i>targets</i> )
FrontPage	<b>Edit Bookmark</b> (FrontPage kalder ankre for <i>bookmarks</i> ).
Claris Home Page	<b>Insert Anchor</b>

Hvert anker skal have et navn. Ankernavne følger samme regler som filnavne: ingen mellemrum, specialtegn mv. Editoren genererer automatisk forskellige navne til ankre, når man sætter dem ind. Navnet kan ændres, når ankeret er markeret

Lave et link til et anker	
PageMill	Træk ankeret hen over den markerede tekst, og slip.
Composer	Insert Link, og vælg Target fra listen Select a named target...
FrontPage	Insert Hyperlink, fanebladet Open Pages, vælg fra listen Bookmark.
Claris Home Page	I Link Editor indsæt #ankernavn i URL'en efter filnavnet (kun #ankernavn hvis det er et anker i samme dokument).

### E-mail link

Du har sikkert set de ”skriv til...” links, som ved klik får browseren til at åbne et e-mail vindue med et tomt brev med modtageradressen udfyldt. En sådan link laves ved at sætte URL'en til at være <mailto:> efterfulgt af e-mail-adressen, for eksempel <mailto:belzebub@underverdenen.dk>. (Bemærk: ingen skråstreger foran e-mail adressen). Det er god skik også at lade linkens tekst være e-mail-adressen, så brugeren kan se det er en e-mail link.

### Preview Mode – afprøvning af links

Editoren har en *preview mode*. Det er en tilstand, hvor man ikke kan ændre i siden, men hvor siden til gengæld vises lidt mere korrekt. F.eks. skjules hjælpelinier ved tabeller, lineskifts-symboler og andre “usynlige” tegn. Desuden kan man afprøve links: klikker man på dem, hentes den nye side ind i editoren (virker ikke på eksterne links).



Skift til <i>preview mode</i>	
Adobe PageMill	Man skifter mellem <i>preview</i> og <i>edit mode</i> med den store knap i højre side af værktøjsbjælkerne.
Netscape Composer	File Browse page
FrontPage	har ingen <i>preview mode</i> , men links kan afprøves ved at holde <i>Ctrl</i> nede og klikke på dem, og hjælpelinier mm. kan slås fra/til med View Format Marks
Claris Home Page	Window Preview Page

## Afprøvning i browser

Jeg vil desuden altid anbefale at hente siderne ind i en browser, for at se hvordan de ser ud. Du kan hente en HTML-fil på din egen harddisk ind i browseren via File|Open File i browseren eller ved at trække filen ind i browservinduet.

Skifter du frem og tilbage mellem editor og browser, så husk at ændringer, du har lavet i editoren, først viser sig i browseren, når du har valgt Save i editoren efterfulgt af Reload i browseren.

Du kan også sende et dokument direkte til en browser, når du har dokumentet åben i editoren:

Sende en side til browseren	
PageMill	View Switch To (det kræver dog at du først har specificeret en eller flere browsere i Edit Preferences Switch To).
Netscape Composer	File Browse page
FrontPage	File Preview in Browser
Claris Home Page	File Preview in Browser

### Øvelse

Prøv at lave to sider med links til hinanden. Prøv at lave en link til <http://www.yahoo.com>. Afprøv de forskellige links først i editoren og så i browseren.

## Billeder

For at indsætte et billede på siden, skal billedfilen

1. ligge i eller under din webmappe.
2. være i grafikformatet GIF eller JPEG. (Hvis du har billedet i et andet format må du hente det ind i et grafikprogram og **Save As / Gem Som** GIF eller JPEG. (Læs i øvrigt mere om grafik side 23).
3. have et gyldigt filnavn (det vil sige ingen mellemrum, specialtegn etc.), og filnavnet skal slutte med `.gif` eller `.jpg` (eller `.jpeg`).

Når et billede bliver indsat på en webside, bliver billedfilen kædet sammen med HTML-filen, men det er stadig to separate filer. Billedfilen må altså ikke slettes, den skal være til stede sammen med HTML-filen.

Indsætte et billede	
Adobe PageMill	File Place
Composer	Insert Image
FrontPage	Insert Image
Claris Home Page	Insert Image

(NB! Netscape Composer har den uvane, at hvis billedet ligger i en anden mappe end den dokumentet ligger i, vil den kopiere billedet til dokumentets mappe. Det er ret forvirrende, men kan heldigvis slås fra ved at afkrydse **Leave image at original location** i **Image Properties**).



Du vil opdage, at billedet ikke kan placeres hvor som helst på siden. I HTML indsættes billedet i teksten og følger tekstens forløb.

Hvis du vil have billedet til at stå alene, så skal det stå i et afsnit for sig selv (altså med *return* før og efter). Billedet kan så flyttes ved at højre-, venstrestille eller centrere afsnittet.

Optræder billedet i en tekstlinie, så kan man justere om billedet er øverst, nederst eller midt på linien i forhold til teksten. Det bruger man mest til små symboler eller ikoner, der optræder som en del af teksten, for eksempel et logo med "nyt". Billedet placeres ved at sætte egenskaben **align** til **top**, **middle** eller **bottom**.

I PageMill benyttes de her viste knapper fra værktøjsbjælken:

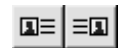


Billeder har egenskaben **alternate text**. Det er en tekst som vil blive vist i stedet for billedet, i en browser der har slået grafik fra. Alternativ tekst er vigtig, især ved billeder der er links eller overskrifter. Flere og flere surfer med grafik slået fra, for at spare tid, og nogle kan kun se tekst, fordi deres computer eller terminal ikke har en grafisk brugerflade, eller fordi de er blinde (og teksten læses op eller konverteres til braille).

Alternativ tekst indekseres desuden i søgemaskiner, hvilket for eksempel overskifter, der er lavet som grafik, naturligvis ikke gør.

### Flydende billeder

Et billede kan også placeres ude ved højre eller venstre kant, så teksten flyder udenom. Dette kaldes at billedet er flydende. Det gøres ved at sætte egenskaben **align** til **left** eller **right**. (I PageMill gøres det via de to viste knapper fra værktøjsbjælken).



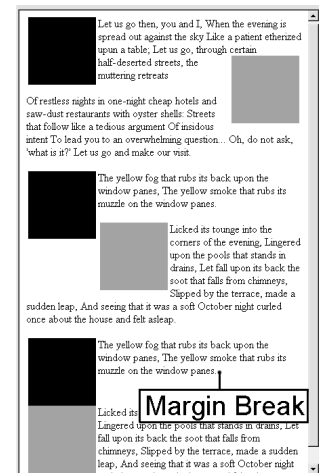
Billedet glider ud til siden på den linie, hvor det indsættes.

Man kan risikere, at billeder støder sammen, hvis der er flere “flydende” billeder i samme side efter hinanden. Dette kan undgås ved at indsætte et *margin break* via [Edit|Indsæt Invisible|Margin Break](#) ([Indsæt|Break](#) i [FrontPage](#)). Et *margin break* laver et ophold, så teksten først fortsætter efter underkanten af eventuelle højre- eller venstreflydende billeder. Hvis der ikke er nogle flydende billeder på linien med *margin break* et virker det bare som et almindeligt lineskift.

Selvom billedet flyder ud til siden, er det stadig sat ind et sted i teksten. Det vil sige: flyttes eller slettes teksten, flytter billedet med.

## Skalering

Billeder kan skaleres direkte i editoren ved at trække i dets hjørner. Gør det ikke! Skaleringen ændrer nemlig ikke selve filstørrelsen, så du får enten et kornet billede eller en for stor billedfil. Skalér i stedet billeder i et grafikprogram.



## Billeder som links

Et billede kan samtidig være et link eller indgå i et link sammen med tekst. Et billede gøres til en link ved at markere det og lave linket på samme måde, som hvis det var et stykke tekst, man havde markeret.

Et billede, der er et link, får en blå ramme. For at undgå denne ikke altid lige klædelige ramme, kan billedets **border**-egenskab sættes til 0. Så kan man bare ikke se, at billedet er en link, med mindre musen føres hen over billedet (hvorved pilen skifter til en hånd). Derfor skal billeder uden ramme, der er links, *signalere* link på en eller anden måde, for eksempel via placering eller motiv.

## Imagemaps

Et billede kan opdeles i områder som er forskellige links. Det kaldes et *imagemap*.

Redigere imagemap	
<a href="#">PageMill</a>	dobbeltklik på billedet, og brug de knapper der bliver aktive på værktøjsbjælken
<a href="#">Netscape Composer</a>	understøtter ikke imagemaps
<a href="#">FrontPage</a>	klik på billedet, og en <a href="#">Image</a> -værktøjsbjælke dukker frem.
<a href="#">Claris Home Page</a>	i <i>Object Editor</i> vælg knappen <a href="#">Client-Side Image Map: Edit</a>

Med disse knapper kan man tegne et firkantet, rundt eller mangesidet område på billedet. Når området er tegnet, kan det laves til en link på almindelig vis.

Ulempen ved imagemaps er, at hvis brugeren har billedlæsning slået fra, kan han ikke se linkene. Man må altså samtidig lave links som indgår i billedet som tekst-links. HTML 3.2 understøtter ganske vist **alternativ tekst** til de enkelte link-områder, men [Netscape Navigator](#) understøtter det ikke, så tekst-links er stadig nødvendige.

Overvej grafikken i imagemaps grundigt. Er det klart for brugeren, hvad de forskellige områder af billedet betyder, og hvor de tilhørende links vil føre hen? Imagemaps bliver nemt uigennemskuelige for brugeren.

Den omtalte form for imagemaps kaldes egentlig *client-side* imagemaps, til forskel fra *server-side* imagemaps, som er en ældre form, der krævede et program på serveren. *Server-side* imagemaps er ikke relevante længere, efter at client side imagemaps er blevet almindeligt understøttet.



## Baggrundsgrafik

Man kan indsætte et billede som baggrundsgrafik. Billedet bliver “fliselagt”, dvs. gentaget vandret og lodret ud til vinduets kanter. Hvis billedet er større end vinduet, bliver det beskåret. Baggrundsbilledet skal være i formatet GIF eller JPEG. Ofte bruges teksturer som baggrund.

Baggrundsbillede indsættes i den samme dialogboks som sidens andre egenskaber.

Vær forsigtig med baggrundsteksturer eller -billeder. Tekst kan være svær at læse på en sådan baggrund. Baggrundsgrafikken skal helst være uden for megen kontrast, men stå i stærk kontrast til teksten. Ellers kan teksten være svær at læse.

Når en webside indlæses, tager det tid, inden et baggrundsbillede indlæses, og indtil da ses siden med dens baggrundsfarve. Man kan gøre indlæsningen blidere for øjet ved at vælge en baggrundsfarve, der ligger tæt på baggrundsbilledets dominerende farver. Når først baggrundsbilledet er indlæst, er baggrundsfarven skjult.

## Øvelser

Lav en side med både baggrundsgrafik og forgrunds-billeder. Hvis du mangler billeder, kan du låne nogle på nettet: find på nettet en side med billeder, klik med *højre museknap* på et billede og vælg *Save Image As...*

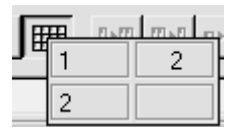
Hent siden ind i browseren for at se, hvordan den ser ud. Ret skærminstillingerne for at se udseendet ved forskellige skærmopløsninger (640x480, 1024x768 m.fl.), og forskellige farveopløsninger (256 farver, 16 bit (*thousands of colors*), 24 bit (*millions*)).

## Tabeller

Tabeller blev indført for at kunne stille tal og data op i skemaer, men bliver nu først og fremmest brugt til at lave mere spændende layout. Uden tabeller er layoutet endimensionalt: afsnit kommer ét efter ét, under hinanden. Vha. tabeller kan man have blokke, der ligger ved siden af hinanden.

En tabel er en ramme inddelt i celler vandret og lodret. De enkelte celler kan indeholde tekst-afsnit, billeder etc. Man kan for eksempel lave et trespaltet layout vha. en tabel, som fylder hele skærmens bredde og er delt i tre celler vandret.

En tabel indsættes via en knap på værktøjsbjælken. Når man indsætter en tabel, bliver man spurgt hvor mange rækker og kolonner, den skal bestå af. I FrontPage og PageMill kan man også “trække” tabel-ikonen ud til det ønskede antal celler.



## Tabellers egenskaber

### Border

angiver i pixel tykkelsen af rammen udenom tabellen. Ved 0 er rammen usynlig.

### Cellspacing

er tykkelsen i pixel af bjælkerne mellem cellerne.

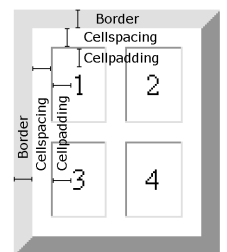
### Cellpadding

er afstanden i pixel mellem cellernes indhold (tekst eller grafik) og bjælkerne.

Hvis **border** er 0, bliver bjælkerne mellem cellerne også usynlige, men afstanden er stadig den samme. Jeg ved ikke, om det er specielt logisk, men sådan virker det altså.

Ved **border=1** og **cellspacing=1** vil rammen og bjælkerne være sorte streger.

Bliver rammen eller bjælkerne tykkere, vil de fremstå som tredimensionale, hvilket ikke er specielt pænt.



**Width** er bredden af tabellen. Den kan angives i procent af skærmens bredde eller i pixel. Problemet ved en tabel, med bredden defineret i pixels, er naturligvis, at hvis brugerens skærm er smallere, skal han scrolle vandret for at se det hele.

Den laveste almindeligt forekommende skærmopløsning er 640x480, så hvis man laver tabeller med pixeldefineret bredde, bør den ikke være mere end 600 pixels. (Til gengæld vil tabellen kun fylde halvdelen af skærmen, på en skærm der er 1200 pixel i bredden!).

**Height** er højden af tabellen. Den kan også angives i pixels eller procent af skærmhøjden. Tabellen vil dog altid være høj nok til dens indhold. **Height** er en lidt problematisk egenskab, fordi den kun forstås af [Netscape Navigator](#), ikke [MSIE](#). Derfor vil jeg fraråde at ændre tabellens **Height**-egenskab.

En tabel kan højre-, venstrestilles eller centreres på siden. Man kan godt lave en tabel inde i en celle.

## Celle

Hver enkelt celle har også sine egenskaber:

**Width** er cellens bredde. Den kan angives i pixels eller i procent af hele tabellens bredde. Cellens bredde kan blive større and angivet, hvis den f.eks. indeholder et billede, der er bredere end cellens definerede bredde.

**Height** cellens højde i pixels eller procent af vinduets højde. Se **Table height** ovenfor.

**Align** angiver den vandrette justering af cellens indhold: højre-, venstrestillet eller centreret.

**VAlign** (*vertical align*) angiver den lodrette justering af indholdet: øverst, centreret eller nederst.

### Cellcolor:

Hver celle kan have sin egen baggrundsfarve. Dette understøttes dog kun af de nyeste browsere ([Netscape Navigator 3.0](#), [MSIE 3.0](#)), så sørg for at have en farvekombination så teksten i cellen kan læses både på cellens baggrundsfarve og på sidens normale baggrundsfarve/baggrundsgrafik.

I [PageMill](#) vælges en celle ved først at dobbeltklikke på kanten af tabellen, så hele rammen vælges, og derefter enkeltklikke i en celle.

Celler kan "smeltes sammen" vandret eller lodret, hvilket f.eks. kan bruges til overskrifter, der strækker sig over flere celler. Det gøres ved at sætte en celles egenskaber **Col Span** og **Row Span** til til så mange kolonner/rækker cellen skal dække. Dette gøres på den celle, der er øverst eller længst til venstre, af de celler der skal smeltes sammen:

Smelte celler sammen	
<a href="#">PageMill</a>	vælg en celle, hold <i>shift</i> nede og vælg endnu en celle. Herefter kan <b>Join cells</b> vælges på værktøjsbjælken
<a href="#">Composer</a>	<b>Table Properties</b> , fanebladet <b>Cell</b> : Sæt <b>Cell spans row(s) and column(s)</b> -egenskaberne.
<a href="#">FrontPage</a>	<b>Cell Properties</b> : Sæt cellens <b>Numbers of rows/columns spanned</b> .
<a href="#">Claris Home Page</a>	I <i>Object Editor</i> , <b>Advanced</b> fanebladet: Cellens <b>Col Span</b> eller <b>Row Span</b> egenskaber.

## Øvelser

Lav to tabeller: én med fast bredde (i pixels) og én med bredde i procent, og fyld noget tekst og grafik ind i cellerne. Hent siden ind i en browser, og se hvordan den opfører sig ved forskellige vinduesstørrelser og skærmopløsninger.

## Afprøvning i browser

Det er vigtigt at afprøve websiten, inden siden uploades.

- Afprøv siden i både [Netscape Navigator](#) og [MSIE](#).
- Afprøv siden uden grafik (vælg [Options|Auto Load Images Off](#) i [Netscape Navigator](#)). Hvis du har lavet meningsfyldte alternativtekster til alle betydningsbærende billeder (og tekstlinks tilsvarende eventuelle imagemaps) skulle siden stadig kunne bruges.

### Øvelse

Prøv at lave en webside med farver, grafik og tabeller. Se siden i begge browsere, om du kan se forskelle.

## Udgivelse

### Plads på en server

Et *Webhotel* er en webserver hos en internetudbyder, hvor man kan leje plads til en website.

Mange internetudbydere giver lidt serverplads, når man får et internetabonnement og en e-mail-adresse, men ellers findes webhoteller i alle prisklasser, alt efter om du er privat eller kommerciel bruger, hvor meget plads du ønsker, hvor meget trafik der er til din webside, om der skal være adgang til CGI-programmer (CGI er forklaret side 47), service mv.

Det billigste jeg har set i Danmark er 150/md, men det normale ser p.t. ud til at være 500/md. Man kan få en gratis homepage på [www.geocities.com](http://www.geocities.com)

Danske webhoteller kan ses på Jubiis oversigt over internetudbydere [www.jubii.dk/EDB/internet/internetudbydere.htm](http://www.jubii.dk/EDB/internet/internetudbydere.htm)

Der er stor forskel på de hotellers båndbredde. Udbyderen kan spare penge ved at have for mange logerende i forhold til servernes kapacitet. Besøg nogle af sitene på webhotellet og læg mærke til indlæsningshastigheden.

### Upload

Man udgiver sin website ved at kopiere indholdet fra webmappen fra egen harddisk til ens mappe på webserveren. Denne proces kaldes at *uplade*. Mange editorer har en indbygget upload-funktion. Ellers kan man bruge et såkaldt *FTP-program*, som kan lave filhåndtering på både din computer og en webserver, herunder kopiere filer fra den ene computer til den anden.

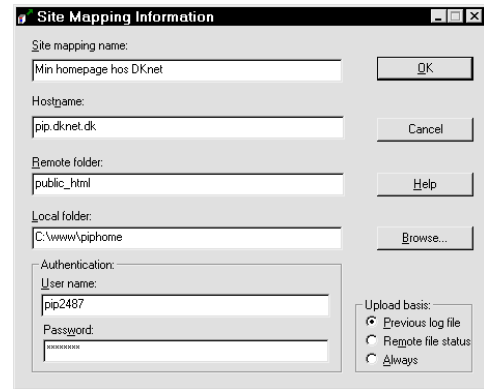
Alle fire omtalte editorer har upload-mulighed. De er dog af svingende kvalitet – [PageMills](#) og [FrontPages](#) er glimrende, mens [Composer](#) og [Claris Home Page](#) er ret svage på dette punkt.

Bruger du et af de to sidste, anbefaler jeg et separat FTP-program. [PageMill](#) har underligt nok heller ingen upload funktion i Macintosh-versionen, så her er det også nødvendigt med et separat FTP-program. Selvom din editor har en indbygget upload-funktion er et FTP-program for øvrigt stadig nyttigt til at rydde op i dine filer på serveren.

## Nødvendig information ved upload

For at uploade skal du have nedenstående informationer af webhotel-udbyderen:

- **Brugernavn.** Det kan du normalt selv bestemme. Ofte vil det være det samme som dit e-mail brugernavn (altså `sofie` hvis din e-mail-adresse er `sofie@hyperweb.dk`).
- **Password.** Det hemmelige kodeord, som gør, at kun du kan ændre i dine websider.
- **Navnet på serveren** du skal have FTP-adgang til, for eksempel `ftp.hyperweb.dk`. Servernavnet ved FTP behøver ikke at være det samme som servernavnet i webadressen, for eksempel kan det have `ftp` i starten af navnet i stedet for `www`.
- **Navnet på mappen**, hvor din website skal placeres. Denne mappe ligger i din brugermappe og hedder oftest `public_html`. Kun du kan få adgang til selve brugermappen, da den indeholder forskellige private ting, for eksempel din postkasse. Mappen `public_html` kan derimod (som navnet antyder) indeholde de dokumenter, der skal være offentligt tilgængelige via nettet. Mappen kan naturligvis indeholde yderligere mapper.
- **Webadressen** på din mappe. Det kan for eksempel være `www.webastoria.dk/~sofie/`. Webadresser behøver ikke at starte med `www`, selvom det er det normale.



Bølge linien (også kaldet en *tilde*) er normal i webhotel-adresser og betyder, at mappen tilhører en bruger på serveren. Faktisk kan man ikke vide, hvor ens brugermappe ligger i serverens interne filstruktur, men det gør ikke noget. Brugernavnet sender dig til den rigtige mappe. Du skal i øvrigt ikke være bange for at komme til at slette andres filer eller formatere serverens harddisk: dit brugernavn og password giver dig kun adgang til din egen brugermappe.

## Upload med PageMill

Vælg **File|Upload Page**. I Upload-dialogen skal du oprette en *site mapping*, som er upload-specifikationerne for en website. Du kan oprette flere site-mappings, hvis du arbejder med flere websites. Når du opretter (eller ændrer) en site-mapping, skal du angive dels din lokale webmappe, dels upload-adressen samt password mv. Desuden skal du vælge **Upload Basis** – om alt i webmappen skal uploades, kun udvalgte filer, eller alt ændret siden sidst. Til sidst vælges **Upload** for at kopiere filerne.

## Upload med FrontPage

I **FrontPage** hedder upload-programmet **Microsoft Web Publishing Wizard**, og startes med menu-punktet **File|Publish FrontPage Web**. Første gang du “publisher”, fører wizarden dig gennem de nødvendige konfigurationer. Når du herefter kører programmet, skal du kun vælge, hvad du vil uploade: 1) alt, 2) udvalgte filer eller 3) kun filer ændret siden sidste upload.

## Upload med Composer

**Composer** kan desværre kun uploade filer fra en enkelt mappe af gangen, ikke medregnet undermapper. Vælg **File|Publish** for at uploade den aktuelle side. Den side, der var åben i editoren, er automatisk valgt. Herudover kan du vælge andre filer i den samme mappe ved at klikke i **All files in page's folder** og markere de filer, der skal uploades.

## Upload med Claris Home Page

**Claris Home Page**'s indbyggede uploader kan kun uploade en enkelt side (inklusive grafik) ad gangen, så jeg anbefaler at man bruger et separat FTP-program. Vælg **File|Remote|Remote** for at uploade den aktuelle fil.

## Upload via et FTP-program

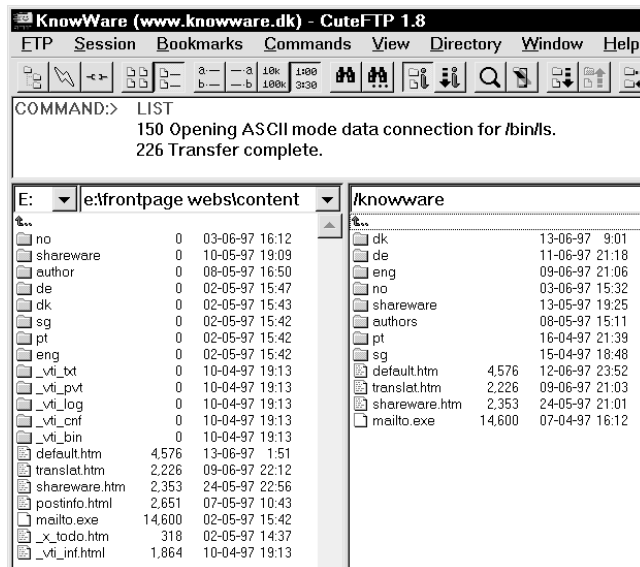
Udmærkede FTP-programmer er:

**CuteFTP** til Windows [www.cuteftp.com/](http://www.cuteftp.com/)

**Fetch** til Mac [www.dartmouth.edu/pages/softdev/fetch.html](http://www.dartmouth.edu/pages/softdev/fetch.html)

Når du starter programmet, skal du oprette en *connection*. Du skal give et navn til forbindelsen, så du kan skelne evt. flere forbindelser. Navnet kan fx være "Min website hos WebHilton". Du skal give de før nævnte nødvendige informationer for en FTP-forbindelse. FTP-programmet opretter forbindelsen, og du får et filhåndteringsvindue hvor du ser indholdet af din mappe. Du kan nu manuelt trække indholdet af din lokale webmappe ind i din webmappe på serveren.

Nogle FTP-programmer spørger om filerne skal uploades som tekst (ASCII), eller binært (også kaldet *Raw Data*). Grafik og anden data skal uploades binært. HTML kan uploades på begge måder – hvis du er i tvivl, så vælg binært. Du skal *ikke* vælge mystiske formater som *uuencoding* eller *macBinary*. Nogle FTP programmer vil desuden tilbyde at tilføje *.bin* eller *.txt* efter filnavne. Det skal du naturligvis sige nej til. (**Fetch** vil f.eks. gerne kode dine filer i *MacBinary* og tilføje *.bin* til navnet. Slå det fra i **Preferences**). FTP-programmet giver desuden mulighed for at omdøbe og slette filer i din brugermappe på serveren.



## Kontrollér

Når du har uploadet, så check i browseren (på web-adressen, ikke FTP-adressen), om det hele virker, som du forventede. Symptomer:

- I stedet for din forside får du en oversigt over filer i mappen: Check om forsiden har det rigtige navn – *index.html*, *default.html* eller hvad din webserver forlanger. Husk: der er forskel på store og små bogstaver, og der er forskel på *.htm* og *.html*
- Links virker ikke, og i stedet for billeder kommer et billedikon med et spørgsmålstegn: Check at filerne er uploadet, at navnene ikke er ændrede, og at links til dem er relative og ikke for eksempel indeholder henvisninger til din harddisk. Check også at der ikke linkes til filer eller mapper, der ikke er uploadet – hvis du for eksempel er kommet til at lave en link til et dokument, der ikke ligger under din webmappe, men ligger et andet sted på din harddisk.
- I stedet for billeder kommer en knækket billedikon: Billederne er ikke uploadet rigtigt. Check at de bliver uploadet binært og ikke som andet.

Når filerne i din website er uploadet og ser korrekt ud i din browser, når du går ind på webadressen (husk Reload!), så er den online.

## Del 2 – Alt om webdesign

Efter første del skulle du være i stand til at skabe og uploade en simpel website. Nu følger en grundigere gennemgang af forskellige aspekter ved webdesign. De forskellige afsnit *kan* læses i vilkårlig rækkefølge.

### Billedbehandling

#### Billedformater

GIF og JPEG er de to billedformater, der bruges på nettet. De er blevet standard, fordi de er platformsuafhængige, og fordi de komprimerer effektivt (dvs. filerne bliver ikke så store). Hvert format har sin anvendelse:

**GIF** egner sig bedst til tegninger, logoer og andre billeder med få farver, ensfarvede flader og enkle former. Desværre kan GIF-billeder højst indeholde 256 farver. Et GIF-billede kan indeholde gennemsnitlige områder. Det kan bruges i irregulære billeder som skal bruges på forskellige baggrundsfarver eller baggrundsgrafik.

**JPEG**-komprimeringen fungerer modsat GIF's bedst på fotografier, malerier og andre billeder med mange nuancer og glidende farveovergange. JPEG billeder er i 24-bit farver (altså millioner af farver). JPEG komprimeringen er yderst effektiv og kan give ret dramatiske reduktioner af filstørrelsen. Til gengæld er den tabsgivende – det vil sige at noget at billedkvaliteten går tabt, når man konverterer billeder til JPEG.

#### Grafik udefra

Når man får grafik af andre, vil man normalt få det i TIFF-format. Måske vil leverandøren tilbyde selv at konvertere filerne til JPEG eller GIF. Sig nej tak! Det vil du selv. Filerne skal nemlig først konverteres *efter* at være skaleret til den størrelse, de skal have på web-siten. Generelt vil jeg anbefale at bruge TIFF til al udveksling. Få hellere billedet for stort end for lille – man kan altid gøre billeder mindre, men man kan ikke gøre dem større uden at forringe billedkvaliteten.

#### Scanning

Scan billedet større end du skal bruge det, så kan du skalere ned i grafikprogrammet. Når du scanner skal du angive hvor mange DPI du vil scanne i. DPI står for *dots (pixels) per inch*. Dvs. scanner du et billede på 5x5 tommer (*inch*) i 150 DPI, får du en billedfil på 750x750 pixels.

#### Tekst, illustrationer og *anti-alias*

Importerer du illustrationer fra et tegneprogram (*Illustrator* eller lignende) eller laver tekst i billedbehandlingsprogrammet, så husk at gøre kanterne *anti-aliased*. Anti-aliased er en teknik, der udglatter hakkede kanter ved hjælp af farver. Det får illustrationen til at se meget mere naturlig ud og giver en illusion af en større skærmløsning. Dog kan tekst i små størrelser blive sværere at læse, hvis den bliver anti-aliased.



#### Transparens

Områder i et GIF-billede kan gøres gennemsnitlige. Det gøres ved at give områderne en bestemt farve, og så, når billedet gemmes som GIF, fortælle at denne farve skal være gennemsnitlig. Desværre kan man ikke lave mere eller mindre gennemsnitlige pixels – de er gennemsnitlige eller ej. Man kan altså ikke “fade” eller anti-alias kanterne af et billede på en gennemsnitlig baggrund. Hvis du vil fade, så gør det ud i en farve og sørg for at websiden, hvor billedet skal ligge, har samme baggrundsfarve.

#### Skalering

Skalér billedet til den rigtige størrelse. Lav al nødvendig retouchering, inden du skalerer billedet. Bemærk at det er størrelsen i *pixels*, der tæller – størrelse i cm/tommer og DPI har kun betydning for billeder, der skal printes på papir.

Jeg vil anbefale at gemme en backup af billedfilen, *inden* den skaleres. Hvis man senere vil lave ændringer i billedet eller skal bruge det i en anden størrelse, kan man starte med backup-filen igen.

## Konvertering til JPEG

Inden du konverterer et billede til JPEG, så kontroller at det er i RGB-mode. Billeder modtaget fra andre kan være gemt i et format kaldet CMYK (et farvesystem, der bruges til grafik beregnet til udskrivning på papir). Er dette tilfældet, så konverter billedet til RGB.

Konverteringen sker ved at vælge **Save As: JPEG**. Man kan vælge, hvor meget kvalitet man vil ofre, fra “ganske lidt” til “ganske meget”. Jo større kvalitetstab, desto mindre filstørrelse. Jeg vil anbefale, at du prøver at gemme filen i forskellige kvaliteter, og derefter sammenligne forholdet mellem billedkvalitet og filstørrelse.

Kvalitetstab er sjældent synligt, dog får større ensfarvede flader generelt misfarvninger, og skarpe kanter bliver uldne. Jo færre detaljer i billedet, jo mere kan man ofre. Et billede af en solnedgang kan komprimeres voldsomt, mens et portræt ikke kan komprimeres så meget.

NB: JPEG-filformatet hedder egentlig *JFIF* (JPEG er nemlig ikke navnet på filformatet, men på komprimeringsmetoden). Normalt siger man JPEG [djæ-pæg] alligevel, men sommetider kaldes formatet *JPEG/JFIF*.

## Progressive JPEG

JPEG findes i en variant, der hedder *Progressive JPEG*. I en progressive JPEG fil ligger billedinformationen i en anden rækkefølge, som gør at billedet under indlæsningen kommer hurtigt frem i en uskarp version for derefter i flere omgange at blive skarpere. Filen fylder det samme som en almindelig JPEG, og tager også den samme tid om at blive hentet ind, men man får tidligere et indtryk af hvordan billedet ser ud. Ikke alle grafikprogrammer kan gemme i progressive JPEG. **Photoshop 3.x** kræver en plug-in for at gøre det, for eksempel den ved navn **Pro JPEG**.

NB! **PageMill** understøtter ikke progressive JPEG – det vil sige man kan godt sætte progressive JPEG-billeder ind med **place**, men **PageMill** viser ikke billedet.

## Konvertering til GIF

For at et billede kan konverteres til GIF, skal billedets farvedybde reduceres til “8-bit” (hvilket betyder maksimalt 256 farver).

Det gøres i **Paint Shop Pro** ved at vælge **Colors|Decrease Color Depth|256 Colors (8 bit)** (I **Photoshop**: **Image|Mode|Indexed Color**)

Man skal så tage stilling til, hvorledes farvedybden skal reduceres. I **Paint Shop Pro** vælges **Palette: Optimized** (i **Photoshop**: **Palette: Adaptive**), for at genererer en palette, der indeholder de 256 vigtigste farver i billedet. Hvis billedet indeholder flere end 256 farver, skal man desuden tage stilling til, hvad der skal ske med de farver, som der ikke er plads til i paletten:

**Reduction Method: Nearest Color (Dither: None** i **Photoshop**) betyder at farver, som ikke er med i paletten, bliver ændret til den nærmeste farve, som er med i paletten.

**Reduction Method: Error Diffusion (Dither: Diffusion** i **Photoshop**) betyder at programmet prøver at “simulere” den manglende farve, ved at lave et mønster af punkter med nærliggende farver fra paletten – billedet bliver altså prikket

Hvilken metode der er bedst afhænger af billedets art. Farverne bevares bedre med Diffusion, men billedet bliver mere “grumset”. Prøv dig frem på forskellige billeder for at se forskellen.

Hvis billedet i forvejen kun indeholder 256 eller færre farver, er der naturligvis ikke disse problemer. **Photoshop** viser dette ved at give valgmuligheden **palette: exact** som betyder at alle billedets farver bevares. Dither-valget er slået fra, fordi der ikke er nogle “manglende farver”.

I **Paint Shop Pro** viser dialogen ikke om billedet indeholder flere eller færre end de 256 farver, men inden man reducerer farvedybden, kan man vælge **Colors|Count Colors Used** for at se, om der er flere end 256 farver i billedet. Hvis ikke, er **Reduction Method**-valgmuligheden irrelevant.

Der er en fare ved at reducere et billedes farver: de mest anvendte farver bliver forstærkede, mens farver, der forekommer ganske lidt i billedet, måske forsvinder helt. Da det tit er disse “kontrastfarver” der giver billedet dets karakter, bør man måske forstærke disse farver manuelt.

Når billedet har en farvedybde på 8-bit, kan det gemmes som GIF med **File|Save As**.

## Interlaced GIF

GIF-billeder kan gemmes på en speciel måde der hedder *interlaced*. Det betyder at informationen i billedfilen gemmes i en anden rækkefølge, så billedet ikke indlæses ovenfra, men i stedet indlæses hurtigt i en grov opløsning og så gradvist bliver mere detaljeret. Interlacede billeder fylder det samme og tager den samme tid at læse ind.

I **Photoshop** vælger du om billedet skal være interlaced i en dialogboks, som kommer op efter du har valgt **Save As... GIF**

I **Paint Shop Pro** er der en ekstra valgmulighed i **Save As**-dialogen, som hedder **Sub type**. Her kan du vælge mellem versionerne 87a eller 89a, i interlaced eller ikke interlaced. Vælg 89a, som er den meste moderne form for GIF.

## 256 farvers skærme og Den Sikre Webpalette

Visse computere kan kun vise 256 farver på skærmen. Her bruger browseren en fast palette på 216 farver. Denne palette er den samme i browsere på alle 256-farvers platforme. At der kun er 216 farver i paletten skyldes, at nogle farver reserveres af systemerne til vinduer, knapper etc.

Farver, der er med blandt disse 216 farver, bliver vist korrekt, mens de andre farver simuleres af computeren ved enten at lave *dither* eller ved at ændre farven til en farve indeholdt i paletten. I mange situationer vil man gerne undgå sådanne uforudsigelige ændringer, for eksempel i ikoner, logoer, overskrifter lavet som grafik, og andre motiver, der består af rene farver samt større, ensfarvede flader. Forestil dig at et ensfarvet logo bliver nålestribet eller småtneret på 256-farvers skærme. For at undgå dette, må man justere billedets farver, så der kun indgår farver fra 216-farvers paletten.

Nyere grafikprogrammer som **Photoshop 4** har 216-farvers paletten (eller "den sikre webpalette", som den også kaldes) indbygget. I **Photoshop 4** kan man f.eks. vælge **Palette: Web** når man reducerer farvedybden.

I **Paint Shop Pro** er det mere kompliceret, fordi man først skal oprette web-paletten. Det gøres således:

1. Gå til en website, hvor der er et billede af paletten, for eksempel <http://www.dsiegel.com/tips/wonk10/images.html>
2. Download billedet af paletten (dvs. højreklik på det, og vælg **Save Image As...**)
3. Åbn billedet i grafikprogrammet

4. Vælg **Colors|Save Palette** for at gemme paletten, f.eks. under navnet webpalette.

Når du så har et billede i grafikprogrammet, som du vil tilpasse til webpaletten, vælger du **Colors|Load Palette** og henter webpaletten.

I begge tilfælde skal man vælge, hvad der skal ske med de farver, der ikke er med i webpaletten. Husk at vælge **Nearest Color (Dither: None i Photoshop)** – pointen med det hele var jo at undgå dither. Billedet bliver nu tilpasset den sikre palette, og du kan gemme det som GIF

Nuancer, som kun kan ses på bedre skærme, går tabt, når man reducerer et billede til den rigide 216-farvers palette. Derfor skal du kun bruge den, hvis det er vigtigt at undgå dither. Billeder med mange nuancer og farvetoner – for eksempel malerier eller fotografier – må ikke tilpasses webpaletten, men skal i stedet gemmes i JPEG. Hvis du er i tvivl, om et billede ser ordentligt ud på en 256-farvers skærm, kan du prøve at ændre din skærmopløsning til 8 bit (256 farver).

Der er i øvrigt også forskel på platformenes "lysstyrke" – de samme billeder ser normalt lysere ud på en Mac end på en PC. Dette kan man ikke gøre noget ved. Man må blot passe på ikke at lave formørkede billeder på en Mac, da de vil være endnu mørkere på en PC. Og omvendt.

## Billedbehandling i PageMill

**PageMill** indeholder en simpel billededitor, som kommer frem, når du åbner billedfilen gennem **File|Open**. I billededitoren kan du gøre filen *interlaced* og vælge gennemsigtige områder. Dette er fint og lige så nemt at gøre som i et grafikprogram. Man kan også skalere billedstørrelsen, hvilket ikke er godt! Editoren ændrer nemlig ikke selve billedfilen, kun på de mål, der angives i dokumentet, hvilket betyder, at billedet enten bliver for kornet, eller at billedfilen bliver større end nødvendigt og siden derved unødigt langsom.

## URL

*URL* er det tekniske udtryk for internetadresse. Som almindelig surfer mærker man kun en URL, når man skal indtaste en adresse, men faktisk indholder HTML-dokumenter URL'er internt i alle links og hver gang, der er indsat et billede eller lignende. En URL er opbygget af fire ting: *protokol*, *servernavn*, *sti* og *filnavn*. For eksempel:

```
http://www.inferno.dk/pandæmonium/limbus/elysium.html
```

Protokollen angiver med hvilken metode filen skal hentes. Så længe man er på webbet er det `http`. Protokolnavnet efterfølges af et kolon. Servernavnet starter med to skråstreger.

Når man skriver en URL i de førende browseres adressefelt, kan `http://` undlades, fordi disse browsere selv tilføjer det. `http://` kan *ikke* udelades i en URL i HTML.

Stien angiver filens placering på serveren. Eksemplet viser en fil i mappen `limbus`, som ligger i mappen `pandæmonium`, som ligger i serverens web-rod. Stien starter med en skråstreg, og der er desuden en skråstreg efter hvert mappenavn. Hvis du er bekendt med DOS, så bemærk, at en skråstreg i en URL er en almindelig skråstreg, ikke en backslash. Den kortest mulige sti er `/`, som angiver at filen ligger i web-roden.

Der skelnes ikke mellem store og små bogstaver i servernavnet – `www.lego.com` er det samme som `WWW.LeGo.Com` – men det gør der i stien og filnavnet: `eksempel.HTML` er *ikke* det samme som `Eksempel.html`. Mange brugere er ikke opmærksomme på dette, og indtaster altid en URL med små bogstaver. Derfor vil jeg anbefale altid at bruge små bogstaver i filnavne.

### Forsidens filnavn

En URL behøver ikke at indeholde et filnavn. Mangler filnavnet, vil serveren kikke i den angivne mappe efter en fil med et bestemt navn, som er serverens "default"-navn. Oftest er dette navn `index.html`, men kan også være `default.html`, `welcome.html` eller andet. Spørg den systemansvarlige.

Findes en fil med "default"-navnet, vil serveren sende den, ellers vil den generere en oversigt over mappens indhold. Med andre ord er `http://www.olav.dk/enmappe/` det samme som `http://www.olav.dk/enmappe/index.html`

Kald derfor websitens forside `index.html`, så URL'en til den bliver så kort som mulig. Man kan i øvrigt også undlade den afsluttende skråstreg, så URL'en bliver `http://www.olav.dk/enmappe`

### Relative URL'er

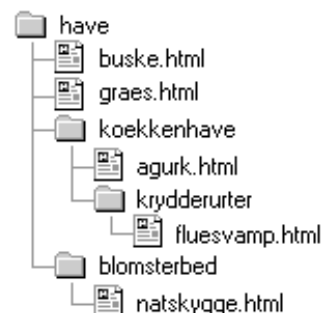
Hvis en link henviser til et dokument på den samme computer, behøver URL'en ikke at være komplet med protokol og servernavn. Man kan nøjes med at beskrive hvor dokumentet ligger i filstrukturen *i forhold til* dokumentet der linkes fra. Det kaldes *relative URL'er*

De forklares nok bedst med nogle eksempler. På billedet ses en filstruktur med websider i forskellige mapper inden i hinanden.

Man henviser til et dokument i den samme mappe blot ved filnavnet.

Hvis `buske.html` skal linke til `graes.html`, er URL'en blot `graes.html`

Man henviser til en fil længere nede i filsystemet, ved at angive stien, dvs. de mapper, man skal ind i for at nå til filen. Hvert mappenavn efterfølges af en skråstreg:



- `buske.html` linker til `agurk.html` med `koekkenhave/agurk.html`
- `agurk.html` linker til `fluesvamp.html` med `krydderurter/fluesvamp.html`
- `buske.html` linker til `fluesvamp.html` med `koekkenhave/krydderurter/fluesvamp.html`

Man henviser “opad”, til den mappe der indeholder den aktuelle mappe, ved at bruge to prikker i stedet for et mappenavn:

- [agurk.html](#) linker til [graes.html](#) med `../graes.html`
- [fluesvamp.html](#) linker til [graes.html](#) med `../../graes.html`

Endelig kan man henvide til dokumenter i andre “grene” ved at gå "op og ned igen":

- [agurk.html](#) linker til [natskygge.html](#) med `../blomsterbed/natskygge.html`
- [natskygge.html](#) linker til [fluesvamp.html](#) med `../koekkenhave/krydderurter/fluesvamp.html`

Relative URL'er er smarte, fordi de stadig virker, når man flytter indholdet af sin webmappe fra sin egen computer til en server.

Starter en URL med en skråstreg, betyder det at stien regnes fra serverens rod, på samme måde som i en absolut URL. På serveren [www.danhave.dk](#) henviser URL'en [/index.html](#) til samme dokument som den absolutte URL <http://www.danhave.dk/index.html>, og URL'en [/billeder/ko.gif](#) til <http://www.danhave.dk/billeder/ko.gif>

Det er en nyttig måde at henvide til et fast sted i filstrukturen, uden at skulle bruge hele servernavnet. Desværre virker det kun over en server, ikke hvis man læser HTML-dokumenter direkte fra sin egen harddisk.

## Ankre

En URL kan henvide til et bestemt anker i et dokument ved at tilføje ankernavnet efter et # sidst i URL'en: [peterplys.html#kapitel2](#). Ankre kan bruges i både absolutte og relative URL'er. Hvis en link henviser til et anker i det samme dokument, behøver URL'en blot at være [#kapitel2](#)

## Frames

Frames gør det muligt at inddele skærmen i flere dele, der hver indeholder et dokument, som kan scrolle uafhængigt. [www.microsoft.com/workshop/prog/ie4](#) er et godt eksempel på, hvad man kan få ud af frames. Frames fungerer således: Man har et “moderdokument”, kaldet et *frameset*, som opdeler skærmen og angiver hvilke dokumenter, der skal hentes ind i de enkelte frames. Selve indholdet i hver enkelt frame er separate dokumenter, som også kunne hentes ind som normale “fuldskærmsider”.

Links får udvidet deres funktionalitet. På en almindelig webside betyder en link jo altid, at en ny side (eller et andet sted på samme side) hentes ind i vinduet. En link i et dokument i en frame kan enten hente den nye side ind i den samme frame, i en anden frame i framesettet, eller i hele vinduet. Dette styres ved at links får en ny egenskab: **target**, som angiver, hvor dokumentet, som linkene peger på, skal hentes ind. Hver frame har et navn (som ikke må forveksles med navnet på dokumentet, der er i rammen).

Framesets kan ligge i flere lag. En side kan være opdelt i fire frames via et enkelt frameset, men det kan også være opdelt i to frames, som hver indeholder et nyt frameset, der indeholder to frames. Umiddelbart kan man ikke se hvor mange “lag” af framesets, der er på en side med mange frames.

[PageMill](#) er den eneste editor jeg har prøvet, hvor jeg synes man kan arbejde med frames på en tilfredsstillende måde. [Netscape Composer](#) understøtter slet ikke frames, og i [FrontPage](#) og [Claris HomePage](#) håndteres de på en ret klodset måde.

I [PageMill](#) er det ret enkelt at opdele et dokument i frames: Man vælger [Edit|Split Frame Horizontally](#) eller [Edit|Split Frame Vertically](#). Efter opdelingen af dokumentet, har man tre dokumenter: selve framesettet og et dokument i hver frame. En frame gemmes ved at markere rammen (klikke i den) og vælge [File|Save frame](#). Framesettet gemmes med [File|Save Frameset](#). En frame har disse egenskaber:

**Name** framens navn

**Width/height** bredde/højde i pixel eller i procent af skærmens bredde

**Margin** afstand fra rammens kanter til dokumentets indhold af tekst/grafik

**Scrollbars** angiver om "rullepaneler" skal være synlige eller ej. Er værdien **auto**, er de kun synlige, hvis dokumentet er større end rammen

**Resize** angiver, om rammen kan gives en anden størrelse, for eksempel hvis brugeren ændrer vinduets størrelse

Disse egenskaber ændres i Inspector på fanebladet **Frame**. Inspector har desuden for hver frame den almindelige **Page** faneblad, hvor man kan ændre baggrundsfarver mv. for dokumentet i rammen.

## Targets

Når man har markeret en link i en frame, bliver "mål"-symbolet i højre side af **Link-to** bjælken synlig. Klikker man på den, kan man vælge, hvilken frame/vindue siden skal hentes ind i. Man ser grafisk de frames der er i vinduet, og desuden kan man bl.a. vælge **Parent Window**, som betyder at den nye side hentes ind i hele vinduet i stedet for framesettet.



### Same Window (**\_top**):

siden læses ind i stedet for det øverste frameset-dokument. Sådan laver man en link, der går ud af framesettet.

### Parent Window (**\_parent**):

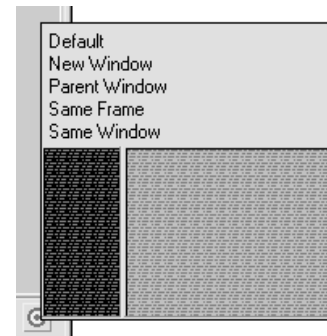
siden læses ind i stedet for det frameset dokument, der indeholder rammen, der linkes fra (det samme som **\_top**, hvis der kun er et lag i framesettet).

### Same Frame (**\_self**):

siden læses ind i den samme frame. Dette er den normale måde, hvis **Base Target** ikke er defineret.

### New Window (**\_blank**):

siden læses ind i et nyt vindue, der oprettes til formålet.



I **FrontPage** og **Claris Home Page** skal man manuelt skrive en links target ind i **Link**-dialogen. Her skriver man navnet på rammen, der skal være mål for linket, eller en af de specielle frame-navne, der starter med underscore. Angiver man som target et navn, som *ikke* er navnet på en eksisterende frame eller vindue, så vil browseren oprette et helt nyt browservindue med det angivne navn, når man klikker på linket. Dokumentet, som linket peger på, vil blive hentet ind i det nye vindue.

En side har egenskaben **base target**, som angiver targetet på alle links på siden, der ikke er specifikt definerede. Med den kan man på én gang sætte targetet for alle links til en anden frame i for eksempel en indholdsfortegnelse.

## Advarsel

Vær forsigtig med links mellem frames. Hvis du i en frame har en link, der går til et nyt frameset, så bliver det nye frameset hentet ind i den enkelte frame, med mindre linkens target er **\_parent** eller **\_top**. Ligeledes: hvis du laver en link fra en frame og ud af din website, skal du huske at sætte linkens mål til **\_top**, ellers kommer brugeren aldrig ud af dit frameset!

## Usynlige rammer

Kanterne mellem frames er ikke særligt kønne. Fra og med **Netscape Navigator 3** og **MSIE** kan man gøre dem usynlige. Dette er dog endnu ikke understøttet af nogle af **WYSIWYG**-editorerne, så koderne for dette skal indsættes manuelt (se side 37). Det gøres ved at indsætte en attribut i **frameset**-taggen, som optræder i frameset-dokumentet. **Netscape Navigator** og **MSIE** bruger forskellige koder til det, men kombinationen af disse tre attributter virker på begge:

```
<FRAMESET
  BORDER=0
  FRAMEBORDER=0
  FRAMESPACING=0>
```

I PageMill er der desuden det problem, at man ikke umiddelbart kan komme til at ændre i frameset-dokumentets kode, på samme måde som man kan med andre dokumenter. Frameset-dokumentet skal åbnes i en ekstern editor som for eksempel Notepad. Det kan gøres nemmere ved at tilføje Notepad til switch to-menuen via Edit|Preferences|Switch to (Notepad.exe ligger i Windows mappen). Herefter kan man vælge Edit|Switch to|Notepad.exe, når framesettet er inde i editoren.

### Links der ændrer indholdet i flere rammer på en gang

Nogen gange kan man i en frame have brug for at lave en link, som, når man klikker på den, ændrer indholdet i to andre frames samtidig. Dette kan uheldigvis ikke laves med en almindelig link, men det kan laves på flere andre måder:

1. Lav et helt nyt frameset med samme opdeling, men med den nye kombination af dokumenter i de enkelte frames, og lav en link til det (med **target** sat til **same window**).
2. Hvis de to frames, hvori indholdet skal skiftes, ligger ved siden af hinanden, så kan du have et underframeset, der indeholder de to dokumenter, og som kan skiftes med et nyt frameset med to nye dokumenter.

3. Brug JavaScript til at lave en "dobbel-link". Lav en link, der skifter den ene frame. Tilføj til link-taggen denne specielle JavaScript-attribut:

```
onClick="top.framenavn.location='filnavn.html' "
```

(Se i øvrigt *JavaScript* side 47 og *Indsætte attributter i WYSIWYG-tilstand* side 40). Eksempel på HTML-koden for linket:

```
<A HREF="side1.html" TARGET=frameNr1
  onClick="top.frameNr2.location='side2.html' ">
```

Mit råd er at undlade links, der skal ændre to frames på én gang. Overvej om du ikke kan lave det samme med færre frames.

### Noframes

Hvis browseren ikke forstår frames, vil den ikke forstå noget af frameset-definitionerne i moderdokumentet. Man må derfor også have et konventionelt webside-indhold i dokumentet til en sådan browser. Denne side kan man arbejde med ved at vælge Edit|No Frames Message. Nogle vælger at lade denne side indeholde beskeden "Se at få dig en ordentlig browser" og en link til Netscapes website, men dette kan næppe anbefales. Klogere er det at lade "No Frames"-siden indeholde en nogenlunde kopi af forsiden eller indholdsfortegnelsen, så man har en frame-løs udgave af websiten.

### Problemer med frames

Før du beslutter dig for at bruge frames, vær opmærksom på disse ulemper:

- Kun den oprindelige frameopstilling har en URL (nemlig framesettets URL). Hvis man har skiftet indhold i de forskellige frames ved at klikke på links, så har man en side, som man ikke kan lave bookmarks på og ikke kan linke til.
- Man kan ikke printe et frameset – kun de enkelte frames hver for sig.
- Det er ærgerligt at man nærmest bliver nødt til at lave to udgaver af en website for at den både kan køre med og uden frames.
- I Netscape Navigator 2.0 virker knapperne frem og tilbage ikke indenfor frames. For at gå tilbage i en frame skal man bruge højre museknap. Det gør navigation i frames forvirrende. I Navigator 3.0 virker frem og tilbage også indenfor frames.

## Misbrug HTML til at lave flot typografi og layout

Du har måske fundet ud af at HTML ikke umiddelbart layouts særligt flot i browseren: dobbelt linieafstand mellem afsnit, overskrifter altid i fed etc. Dette skyldes ikke HTML-specifikationerne, men er konvention fastslået af browserne. HTML har desuden ret primitive muligheder for at styre typografi og layout (i hvert fald indtil style sheets slår igennem, se side 42). Disse muligheder er mest tricks, som udnytter [Netscape Navigators](#) og [MSIE](#) måder at formatere websider på, og de præsenteres her uden ansvar for virkemåden i andre browsere.

### Tekst som grafik

Ved at lave tekst som grafik kan man kontrollere skrifttype, knibning mv. præcist. Dette er metoden til at slippe uden om alle typografiske begrænsninger i HTML. Det bruges især ved forsidesitler og firmanavne, hvor det er vigtigt at teksten ser rigtig ud. Tekst som grafik giver normalt ret små GIF-filer, fordi det indeholder få farver og simple former. Husk at anti-alias tekstens kanter, og at lave en **alternativ tekst** til billedet.

### Kapitæler

Skriv med store bogstaver og gør det første bogstav større end resten via **font-size**.

### Større afstand mellem bogstaver

Indsæt mellemrum mellem hvert tegn, og styr mellemrummets bredde ved at ændre dets skriftstørrelse.

### Mere kompakte afsnit

Browseren viser altid en tom linie mellem afsnit. Det giver et rodet skriftbillede. Det kan undgås ved at lave tvunget lineskift (*shift + return*) i stedet for afsnitsskift (*return*). For at markere nyt afsnit kan den første linie efter lineskiftet indrykkes med et eller flere tvungne mellemrum (*ctrl + mellemrum*). Hele teksten er nu teknisk set et stort afsnit, så man kan f.eks. ikke bruge en afsnitsformatering kun på overskriften. I stedet kan man fremhæve overskriften med skriftformateringer som skriftstørrelse og -farve.

### Større linieafstand

Man kan lave dobbelt linieafstand ved at bruge afsnitsskift efter hver linie. Mere præcis kontrol over linieafstanden kan opnås ved at lave tvunget lineskift (*Shift + Return*) efter hver linie, og sætte et nonbreaking space (*Shift + Space*) ind sidst på hver linie, og gøre mellemrummets skriftstørrelse større end tekstens.

### Marginer

Marginer kan laves med indrykning, men kan kontrolleres mere præcist med tabeller. I en tabel kan man lave en margin med en kolonne, som kun indeholder et passende antal tvungne mellemrum.

### Luft

Luft og afstand er et af de vigtigste redskaber i layout. Tomme felter af kontrolleret størrelse kan laves med en 1x1 pixel stor gennemsigtig GIF. Det giver et usynligt billede, som du kan styre marginen omkring med billedets **vspace** og **hspace** egenskaber. Metoden er mere præcis end at ændre skriftstørrelse på et tvunget mellemrum, men virker ikke, hvis brugeren har billedindlæsning slået fra.

[Netscape Navigator 3](#) har i øvrigt et nyt element, den såkaldte **spacer**, som kan bruges til at lave tomme felter på en nemmere måde, men da det kun er [Navigator 3](#) der understøtter den, vil jeg ikke anbefale den.

### Andre skrifttyper

I HTML 3.2 har man kun to forskellige skrifttyper at arbejde med: standardskriften som normalt er *Times*, og så "skrivemaskineskrift", normalt *Courier*.

Fra [Netscape Navigator 3](#) og [MSIE 2](#) understøttes taggen **font face**, hvor man kan angive, hvilken skrifttype teksten skal vises med. Det lyder smart, men problemet er, at modtageren skal have den angivne skrifttype installeret på sit system før det virker, og ingen har de samme skrifter. Heldigvis giver attributten mulighed for at specificere en række navne på skrifttyper med komma imellem. Browseren bruger så den første skrift på listen, som er installeret på brugerens system.

For eksempel: `FONT FACE="Optima, Futura, Helvetica, Arial"`. Systemer der har installeret Optima vil se Optima, hvis de ikke har Optima, men Futura, vil de se den etc. Har man ingen af skrifterne bruges standardskriften.

Man kan faktisk kun regne med tre skrifter, som alle har. De hedder lidt forskelligt på PC og Mac, så man må angive begge navne (med komme imellem):

#### **Times, Times New Roman**

standardskriften som også dette hæfte er skrevet med. Ifølge eksperter den mest læselige skrift til længere tekst.

#### **Arial, Helvetica**

en groteskskrift, det vil sige en skrift uden "fødder". Bruges ofte til overskrifter – således også i dette hæfte.

#### **Courier, Courier New**

den klassiske skrivemaskineskrift.

NB! Du må aldrig, aldrig prøve at lave matematiske symboler eller græske tegn eller lignende via **font-face**. Ikke bare giver det meningsløse resultater på browsere, der ikke forstår taggen, det giver også inkonsistente resultater på tværs af platforme.

### Tricks med baggrundsgrafik

Baggrundsbilleder bliver gentaget på begge ledter over hele sidearealet. Man kan undgå, at et baggrundsmotiv bliver gentaget ved at lave billedet stort nok. Er det 1600x1200 pixel stort, bliver billedet ikke gentaget på selv de største skærme (med mindre siden er længere). Det egentlige motiv behøver ikke at fylde hele billedfladen, og bør egentlig ikke fylde mere end højst 640x400 pixel. Resten af billedfladen kan så være ensfarvet.

Oftest ser man sider, der vandret er delt ind i flere forskellige baggrundsfarver, for eksempel rød baggrund den første tredjedel af bredden og hvid baggrund på resten. Dette er lavet med et enkelt smalt og lang billede, hvor det første stykke af billedet er rødt og resten hvidt. Billedet behøver ikke at være mere end en pixel højt, men skal være mindst 1600 pixels bredt.

Hvis du støder på baggrundsgrafik og undrer dig over, hvordan det er lavet, så kan du klikke med højre museknap og vælge **Save Background As** og gemme baggrundsbilledet på dit skrivebord og så hente det ind i browseren igen.

I **Netscape Navigator** kan man ikke lave tekst eller forgrundsgrafik, der går helt ud til kanten af vinduet. Endvidere svinger det fra system til system præcis, hvor stor afstanden er fra kanten af vinduet

til indholdet. Man kan altså ikke synkronisere forgrund og baggrund helt. Dog kan man i en frame angive størrelsen af margin fra kanten af ramen til indholdet. Hvis man vil have forgrundsgrafik helt ud til vinduets kant, skal det altså ske i en frame. I **MSIE** kan man angive sidens øverste- og venstre-margin med egenskaberne **body margintop** og **marginleft**, men det virker desværre ikke i **Navigator**.

### Linielængde

Man kan få kontrol over et tekststykkets linjelængde/spaltebredde ved at fylde teksten ind i en celle med bredden defineret i pixel. Det betyder vel og mærke ikke absolut kontrol over linjelængde og ombyrning, for størrelsen af "normaltekst" afviger stadig for eksempel fra PC til Macintosh, men man har dog væsentlig større kontrol end ved tekst uden for tabeller, hvor linjelængden altid tilpasser sig vinduets bredde.

### Spalter

Man laver et spaltelayout ved at dele teksten op i mindre dele, som så fordeles i et antal vandrette celler. Spalterne må helst ikke være mere end omkring 300 pixels høje, ellers skal man på små skærme rulle op og ned for at komme fra bunden af en spalte til toppen af den næste. Ulempen er, at man manuelt skal dele teksten op i afsnit af den rigtige størrelse for at de udfylder cellerne lige meget. Men sådan er dét.

### Tomme celler

Celler uden indhold kan bruges i layout, men de følger ikke altid de pixelstørrelser, man definerer dem til. Celler indeholdende tekst vil for eksempel "presse" tomme celler i bredden. Man kan tvinge en tom celle til at have en mindstestørrelse ved at indsætte en linie med en række *nonbreaking spaces*.

## Underlige inddelinger

### 4. Planlægning & design af en website

Online hypertext er langt fra det ideelle medie. Tekst er sværere at læse på en computerkræm end på papir, og jo mere man skal koncentrere sig for at læse noget, jo mindre får man ud af det man læser. På en skærm mangler man desuden fornuftens af 'hvor man er' og 'hvor langt man er', som man får ved rent fysisk at have en bog eller et blad i hånden. Webbets decentral, kaotiske struktur med hyperlinks på kryds og tværs gør det ikke bedre.

Nem adgang til information er ikke det samme som viden.

I en website må man derfor prioritere læseshed, overskuelighed og klar kommunikation over smukt design, corporate identity mv. Dette er forskelligt fra papirmedier, hvor man sagtens kan tillade sig et ikke helt læseoptimalt layout, hvis layoutet gør læseren mere spændende.

Lærangen mere spændende. Det betyder naturligvis ikke at man ikke må lave flot layout og typografi på en website, kun at layout og design skal være underordnet det der skal kommunikeres - naturligvis med mindre designet er det budskab. På en forside til en website er fængende design og/eller tydelig corporate identity naturligvis en vigtig del af 'budskabet'.

### 4. Planlægning & design af en website

Online hypertext er langt fra det ideelle medie. Tekst er sværere at læse på en computerkræm end på papir, og jo mere man skal koncentrere sig for at læse noget, jo mindre får man ud af det man læser. På en skærm mangler man desuden fornuftens af 'hvor man er' og 'hvor langt man er', som man får ved rent fysisk at have en bog eller et blad i hånden. Webbets decentral, kaotiske struktur med hyperlinks på kryds og tværs gør det ikke bedre.

Nem adgang til information er ikke det samme som viden.

I en website må man derfor prioritere læseshed, overskuelighed og klar kommunikation over smukt design, corporate identity mv. Dette er forskelligt fra papirmedier, hvor man sagtens kan tillade sig et ikke helt læseoptimalt layout, hvis layoutet gør læseren mere spændende.

Bjælkerne mellem celler går gennem hele tabellen, både lodret og vandret. Man kan ikke umiddelbart lave en bredere celle under en smallere, men det kan man slippe udenom ved at sammensmelte celler.

I eksemplet her har jeg lavet et layout ved først at lave en 4x5 cellers tabel, dernæst smeltet celler sammen vandret på forskellige måder, og så indsat tekstdelene i cellerne. For at styre søjlens bredde har jeg fyldt et antal tvungne mellemrum ind i den nederste venstre celle.

Artiklerne i [www.hotwired.com/synapse/](http://www.hotwired.com/synapse/) er et godt eksempel på brug af tabeller i layout.

## Fra tekstbehandling til HTML

Der findes ingen perfekt måde at konvertere tekst fra et tekstbehandlingsprogram til HTML. Dels er der stor forskel på webmediet og papirmediet (som tekstbehandlingsprogrammerne er udviklet til), dels er HTML's typografi- og layoutmuligheder stadig ret primitive i forhold til et almindeligt tekstbehandlingsprogram. Layout (sideopsætning, marginer, tekstfelter, billedplaceringer, spalter) kan ikke overføres til HTML.

Kun få typografiske formateringer som fed, kursiv, skriftførelse (i få trin), indrykning og justering kan overføres. Formateringer som lineafstand, indrykning af første linie i afsnit, tabulator, marginer, afstand før og efter afsnit mv. er ikke understøttet i HTML.

Mange specialsymboler og matematiske tegn er ikke understøttet i HTML.

Billeder kan overføres fra tekstbehandlingsdokumenter til HTML, men oftest er det nemmere at tage den oprindelige billedfil, konvertere den til GIF eller JPEG, og så indsætte den i webeditoren.

Overskrifter, punktopstillinger og tabelopstillinger kan godt overføres til HTML.

## Konklusion

Kun selve teksten og dens funktionelle struktur (markeringer af overskrifter, fed, kursiv, punktopstillinger, tabelopstillinger) kan laves i tekstbehandlingsprogrammet og overføres til HTML. Det visuelle layout, farver, billeder mv., må laves i webeditoren efter at teksten er importeret. Links samt eventuel multimedia eller interaktivitet må også indsættes i webeditoren, da det er tæt knyttet til HTML-formatet.

Der er to muligheder for at overføre tekst fra tekstbehandling til HTML: At lade tekstbehandlingsprogrammet konvertere dokumentet til HTML, eller at indlæse tekstbehandlingsdokumentet i webeditoren og lade webeditoren foretage konverteringen til HTML. I begge tilfælde vil man kunne overføre den grundlæggende struktur, men resultatet skal finpudses og layouts i webeditoren.

PageMill og FrontPage kan læse dokumenter fra de mest almindelige tekstbehandlingsprogrammer, og kan desuden læse RTF (et "udvekslingsformat" for tekstbehandlingsdokumenter). Alle tekstbehandlingsprogrammer kan gemme som RTF. Netscape Composer og Claris Home Page kan ikke læse nogen tekstbehandlingsformater, kun HTML eller ren, uformateret tekst.

Her må man altså enten gemme som HTML fra tekstbehandlingsprogrammet eller som ren tekst (hvorved alle formateringer går tabt).

De nyeste versioner af Word og Wordperfect kan umiddelbart gemme dokumenter som HTML. Knapt så nye versioner kan udvides med muligheden for at gemme som HTML via makroer eller hjælpeprogrammer, f.eks. Microsofts Word Internet Assistant.

## QuarkXPress, Pagemaker etc.

Det er muligt at overføre dokumenter fra sideopsætningsprogrammer som Quark XPress og Pagemaker til HTML, men ligesom med tekstbehandlingsdokumenter vil det meste formatering og layout gå tabt. Det kan altså ikke anbefales at forsøge at lave websider i disse programmer og så bagefter konvertere dem til HTML. Normalt vil det layoutede dokument være lavet med tekst fra et tekstbehandlingsprogram. Det vil derfor være lettere at få den oprindelige tekstbehandlingstekst og billederne for sig, og så selv samle siderne.

## HTML, Browserkrigen og Webbets historie

Som bekendt er det ofte nødvendigt at forstå historien for at forstå nutiden. Således også med HTML. Så her er ganske kort HTML's og webbets historie.

### 1969

Den første internetforbindelse oprettes. De næste mange år anvendes internettet til e-mail, overførsel af filer og programmer, og til terminalarbejde.

### 1989

*Tim Berners-Lee* ved *CERN* (Det Europæiske kernefysiske forskningscenter i Schweiz) fremlægger en ide til at køre hypertext over netværk. Hypertext (altså tekst hvor ord kan være *hyperlinks* til andre steder i teksten) er en idé, der stammer tilbage fra tresserne, og er blandt andet også kendt fra Windows-hjælpesystemet. Det revolutionerende ved Berners-Lee's idé var, at de sammenkædede dokumenter ikke behøvede at ligge på den samme computer, men kunne ligge på forskellige computere på et netværk – distribueret hypertext kaldte man det. Man valgte naturligt nok internettet, som det netværk, systemet skulle køre over.

En sammenhængende samling af hypertext-dokumenter kaldes for et "web", så det var naturligt at kalde det distribuerede hypertextsystem for *World Wide Web*.

Det var centralt for Berners-Lee, at webbet skulle føles som en enhed – det skulle være lige så nemt at følge en link til et dokument på en computer på den anden side af jorden, som en link til et dokument på den samme computer.

### Julen 1990

En prototype på webbet demonstreres på en NEXT-computer. Systemet består af en server, en browser, et hypertext-dokumentformat, og en protokol (HTTP) for kommunikation mellem server og browser. Komponenterne var naturligvis simple end vi kender dem i dag (f.eks. var dokumenterne ren tekst, uden farver eller billeder), men den grundlæggende idé var på plads. For øvrigt havde den første browser indbygget en WYSIWYG-editor, og mulighed for at tilføje kommentarer og noter til eksisterende sider.

### 1991

En simple browser (uden editor) konstrueres til flere systemer. Den første webserver i USA oprettes. *Dan Connoll* udvikler specifikationerne for den tidligste HTML-version. HTML er baseret på SGML som er en overordnet standard for dokumentformater til digital information.

### 1992

Server- og browsersoftware udsendes som *free-ware*. Første browser på en grafisk brugerflade: *Erwise*, udviklet i Finland. 50 webservere i hele verden.

### 1993

Browseren *Mosaic* udvikles på NCSA-universitet af *Marc Andreessen* og *Eric Bina*. *Mosaic* understøtter billeder på siderne, og sendes ud gratis. Dette er webbets gennembrud. *Mosaic* kommer bl.a. til PC og Mac, og får en million brugere i løbet af året. 250 servere i verden. HTML 2.0 specificeres.

### 1994

Marc Andreessen m.fl. forlader NCSA og stifter *Netscape Communications*. Første internationale WWW-konference, senere kendt som "Webbets Woodstock". W3C oprettes med Tim Berners-Lee som formand; formålet er at styre og udvikle standarderne webbet bygger på: HTTP, HTML, URL'er etc., og at udvikle nye standarder som VRML, style sheets og HTML 3.0. CERN må trække sig ud af web-projektet (vil hellere bruge deres penge på partikelacceleratorer). Der er 2500 webservere i verden.

### 1995

Netscapes *Navigator* overhaler *Mosaic* som den mest populære browser. *Navigator 1.1* har udvidelser til HTML, som giver mulighed for tabeller, baggrundsgrafik, farver, skriftstørrelser og blinkende tekst. Webdesignere udnytter Netscapes udvidelser til at lave flotte websites, og opfordrer andre til at bruge *Navigator*, så de kan se, hvor flotte de er. Effekten er selvforstærkende, og Netscape opnår 90% af browsermarkedet. Kritikere anklager Netscape for at korrumpere HTML.

W3C arbejder på HTML 3.0 specifikationerne, som vokser sig stadig mere komplekse. HTML 3.0 omfatter bl.a. style sheets.

Microsoft opdager pludselig, at webbet er ved at blive rigtigt stort, og at de ikke er med. Microsoft beslutter at satse på webbet.

#### Ved årsskiftet til 1996

Ved årsskiftet til 1996 kommer **Navigator 2.0** som understøtter frames, plug-ins, JavaScript, *client-side imagemaps* og Java, men kun få af HTML 3.0 udvidelserne. Navigator 2.0 koster 500 kr., men er stadig shareware, og må bruges gratis af studerende. Mange mener, at Java er en trussel mod selvste Microsofts eksistens.

Microsoft lancerer **MSIE 2.0** som nogenlunde svarer til **Netscape 1.1**. Microsoft har dog (inspireret af Netscape) opfundet nogle udvidelser til HTML, blandt andet rullende tekst og baggrundsmusik. Mange griner af Microsoft. Men Microsoft har et ekstra kort i ærmet: IE er gratis. Der kommer hurtigt at par opgraderinger til IE, som understøtter frames og plug-ins. Det er tydeligt, at Microsoft satser på deres browser.

Webdesignere begynder at lave sites ”optimeret for Netscape” eller ”optimeret for MSIE”. Ingen understøtter HTML 3.0. Jagttagere taler om ”HTML’s død”.

W3C opgiver 3.0 og lancerer HTML 3.2, som grundlæggende er HTML 2.0 plus de mest fornuftige af Netscapes udvidelser (frames og blinkende tekst kommer ikke med). Desuden anbefaler W3C, at browserne begynder at implementere style sheets.

Microsoft sværger, at de fra nu af kun vil implementere HTML, som er specificeret i W3C-standarder, i stedet for at køre deres eget løb som Netscape. Men kritikere kalder det spil for galleriet: Microsoft ved udmærket, at kampen i fremtiden kommer til at stå om scripting (som JavaScript) og interaktive objekter (som Java), og her har Microsoft deres helt egen dagsorden.

Internet-hysteriet når uanede højder – i danske aviser kan man læse artikler som ”Julemanden på internettet”, ”Mode på internettet” mv.

#### August 1996

**MSIE 3** og **Navigator 3** kommer på markedet med en uges mellemrum. **MSIE** overrasker og har stort set alle **Navigators** faciliteter inkl. Java og JavaScript, understøtter desuden ActiveX, VBScript og Style sheets. Overfor dette har **Navigator 3** kun forbedret JavaScript og et par nye (ikke-standard) tags til layout. Det fortælles, at programmørerne hos Netscape hverken har tid til at spise eller sove. **MSIE** og **Navigator** er nu nogenlunde lige gode, og **MSIE** haler ind på Netscapes markedsandel. NCSA opgiver **Mosaic**, og der er ingen reelle konkurrenter til de ”to store”.

#### Vinter og forår 1997

Netscape og Microsoft lancerer begge 4.0 versionerne af deres browsere. De har blandt andet hver deres – inkompatible – måder at lave ”aktivt HTML” på. Det er nu tydeligt, at der står meget mere på spil end bare kontrol over HTML. Det drejer sig om sammensmeltningen mellem computerens brugerinterface og internettet: Microsofts strategi er at integrere browseren helt i Windows-operativsystemet, Netscapes er at gøre browseren til programmet man kan alt igennem, så operativsystemet blive usynligt. Kampen drejer sig også om den underliggende objekt-model på webbet – ”webbets operativsystem” – Netscape holder på de platformsuafhængige standarder CORBA og JavaBeans, Microsoft på deres Windows-baserede COM og ActiveX.

## Hvad understøtter hvilke browsere?

Det er vigtigt at vide, hvilke browsere der understøtter hvilke muligheder. HTML bliver som bekendt jævnlige beriget med nye påfund fra de to førende browserudviklere Netscape og Microsoft. W3C, der udvikler og anbefaler HTML standarder, tænker normalt længere og dybere over tingene, så deres anbefalede standard er normalt et års tid bageude i forhold til den faktiske situation på nettet. Det betyder, at de eksperimentelle udvidelser når at blive prøvet af, før de bliver godkendt.

For eksempel vakte frames stor begejstring blandt webdesignere, da de blev introduceret af Netscape, men siden har der vist sig en del problemer med dem, som gør at de nok aldrig bliver en anbefalet standard. W3C og browserproducenterne eksperimenterer nu med en frame-model lavet via style sheets, som forhåbentligt er bedre. Noget andet er så, at folk naturligvis vil bruge frames, indtil der kommer et alternativ med lignende funktionalitet, uanset hvad W3C siger.

I dag er den anbefalede version HTML 3.2, som indeholder det meste af, hvad man bruger i dag. Hvis man holder sig til tags fra HTML 3.2, kan man regne med, at alle normale browsere kan vise dem. Her følger de udvidelser der ikke er kommet med i standarden, med en angivelse af hvilke browsere der understøtter dem:

### Navigator 2.0 og MSIE 2.x

Disse tags er ikke HTML-standard, men dog nærmest universelt understøttede:

- Frames (**frameset**, **frame**, **target**, **base target**)
- Indsætning af plug-ins (**embed**)
- Animerede GIF.

### Navigator 3.0 og MSIE 3.0 understøtter

Disse er bredt, men ikke universelt understøttede:

- Farvede framekanter eller usynlige frames (Men med forskellige tags!)
- Angivelse af skrifttype (**font face**)
- Baggrundsfarver i tabel-celler (**td/tr/th bgcolor**)

### Kun Navigator

Disse tags er understøttede af den mest udbredte browser, men det er tvivlsomt, om de kommer med andre steder:

- Blinkende tekst (**blink**, fra [Navigator 1.1](#)). Der findes hele websider dedikeret til nedrakning og latterliggørelse af den – efter manges mening –

ret irriterende tag. Netscapes retfærdiggørelse af taggen kan læses ved at skrive `about:mozilla` i Netscapes adressefelt.

- Kolonner (**multicol**, fra [Navigator 3](#)). Det samme kan opnås med tabeller.
- Tomme felter (**spacer**, fra [Navigator 3](#)) Faktisk en ret smart tag, som kan bruges til at lave mellemrumfelter. Det samme kan dog laves med style sheets.
- At sende indholdet af en formular til en e-mail adresse i stedet for til et CGI-program. Det er faktisk ret smart, så det er ærgerligt, at **MSIE** ikke understøtter det.

### Kun MSIE

Disse tags er ikke bredt understøttede og bliver det næppe nogen sinde:

- Baggrundsmusik (**body bgsound**) Plug-ins er en bredt understøttet måde at gøre det samme på.
- Vandmærke (**body bgproperties=fixed**), et baggrundsbillede som ikke ruller med teksten. Det samme er muligt via style sheets (selvom jeg ikke rigtigt kan se pointen med det).
- Sidemargin (**body leftmargin/topmargin**) Angiver afstanden fra vinduets kant til sidens indhold. Det er faktisk smart! Det samme kan specificeres i style sheets.
- Udvidelser til **img** for at vise film. (**img dynsrc**) Plug-ins er en bredt understøttede måde at det samme på.
- Flydende frames (**iframes**): en webside indsat som et vindue på en anden webside. Vinduet scroller sammen med websiden.
- Vandret rullende tekst (**marquee**). Årsagen til at jeg foretrækker [Navigator](#). Med ActiveX-teknologien kan man også lave lodret rullende tekst.
- Udvidelser til tabellers udseende, for eksempel farvede skygger og baggrundsgrafik i celler.

Disse er ikke bredt understøttede, men vil sikkert blive det:

- W3C-udviklede udvidelser til tabeller.
- **Object**-tag
- Style sheets

## Plug-ins, Java, JavaScript, ActiveX

Læs i del 4 om, hvorledes disse teknologier fungerer og er understøttet. Generelt kan man ikke regne med at folk understøtter dem, for selvom browseren har teknologien, er der mulighed for at slå disse ting fra.

## CGI

Formularer og CGI understøttes af alle browsere.

## Hvilke browsere skal man designe til?

Skal man bruge de nyeste udvidelser og opfordre alle besøgende til at downloade den allernyeste browser? Eller skal man designe sine sider, så selv de ældste browsere kan være med? Det er et om-diskuteret spørgsmål. For det første må man huske, at man sagtens kan lave sider, som både udnytter de nyeste muligheder og stadig fungerer tilfredsstillende på ældre browsere. Generelt vil jeg sige, at man skal designe efter ens publikum, og folk har de browsere, de vil have – man får næppe nogen til at downloade en ny browser ved at skrive “denne side ses bedst med...”.

## Browsersnes udbredelse

3 browsere er meget udbredte: [Netscape Navigator 2.0](#), [3.0](#) og [MSIE 3.0](#). Til sammen dækker de formentligt over 95% af markedet. Netscape er langt den mest udbredte. [MSIE](#) har nok omkring 25% og [Navigator](#) resten. Disse tal er dog ret usikre, og ændrer sig hurtigt. 4.0-browsere vil nok brede sig hurtigt i løbet af efteråret, men [Navigator 2.0](#) vil formentligt forblive laveste fællesnævner længe endnu.

[Navigator](#) findes til en lang række platforme, inklusive VMS og OS/2. [MSIE](#) findes til Windows 95/NT, Windows 3.11 og til Macintosh, men er kun en væsentlig faktor på Windows 95/NT.

## Tilbagefald

Hvis man bruger tags, som ikke forstås eller anvendes af alle browsere, så skal man gøre det på en sådan måde, at der ikke opstår fejl eller tab af information:

### Farver

Specifikation af tekst- og dokumentfarver er ganske vist understøttet i alle nyere browsere, men brugeren kan specificere egne farver, som så tager fortrin for farverne specificeret i dokumentet ([Options|General Preferences|Color](#) i [Navigator](#)). Man må altså ikke lave betydningsmæssige markeringer *udelukkende* med farver – for eksempel bruge rød i stedet for fed.

### Skrifttyper

**Font face**-taggen er ikke universelt understøttet. Den skal derfor ikke bruges som eneste markering af noget betydningsmæssigt.

**Font face** må aldrig bruges til at lave andre tegn end det almindelige tegnsæt – den må ikke bruges til at lave græske tegn, symboler eller “wingdings”. Ikke bare virker det ikke i ældre browsere, det virker heller ikke på tværs af platforme!

### Billeder

Mange surfer med grafik slået fra ([Options|Auto Load Images](#) i [Navigator](#)). Angiv altid alternativ tekst, med mindre billedet udelukkende har dekorativ betydning.

### Clientside Imagemaps

Visse ældre browsere understøtter ikke clientside imagemaps. Desuden vil imagemappet ikke kunne bruges, hvis grafik er slået fra. Lav derfor altid de samme links også som tekst-links.

### Frames

Frames er specielt djævelske. Hvis ikke du har lavet en hel alternativ side på framesetsiden, så får folk med en browser, der ikke understøtter frames, en blank side! Frames er dog understøttet af de allerfleste browsere i dag.

### Cellefarve

den er farlig! Hvis man har hvid tekst på en sort cellebaggrund på en hvid side, så er teksten ulæselig i de mange browsere, der ikke understøtter farvede cellebaggrunde.

## Andre dokumentformater

Alle mulige filformater kan overføres via nettet. En link kan gå til et program, der skal downloades, en videofilm, et [Wordperfect 4.1](#)-dokument

Problemet er mest om browseren ved, hvad den skal gøre med den modtagne fil.

Man laver en link til en ikke-HTML-fil ligesom til HTML-filer. Eneste forskel er, at filen vil have et andet efternavn end `.html`

Når serveren sender en fil til en browser, fortæller den først hvilken *MIME-type* filen har. MIME-types er en standardiseret måde at angive dataformater på. De angives med navnene på en hovedgruppe og en undergruppe adskilt af en skråstreg, for eksempel `text/html`, `image/jpeg` eller `world/vrml`

Men hvordan kan serveren vide hvilken MIME-type en fil har? Hvis den ikke er konfigureret til andet, vil den gætte det ud fra filnavnets efternavn. (Det samme gør i øvrigt browseren, hvis den henter filerne fra egen harddisk). Derfor er det vigtigt, at filer har de rigtige efternavne.

Browseren vurderer ud fra MIME-typen, hvordan filen skal håndteres. HTML-sider og GIF/JPEG-grafik vises direkte i browservinduet. Filer, som kan håndteres af en installeret plug-in, bliver "afspillet" af plug-in'en i browservinduet. Visse filtyper sendes til eksterne programmer – for eksempel pakkede filer til et udpakningsprogram. Andre filer downloades til ens harddisk, for eksempel programmer.

Kender browseren ikke det modtagne filformat, spørger den, hvad den skal gøre ved filen. Se i øvrigt hæftet *Ud på Word Wide Web*; afsnittet *Hjælpeprogrammer til særlige formater*. Hvis du ikke vil risikere at en fil åbnes i et eksternt program på klientens computer, så kan du zippe filen.

### Hvilke dokumentformater skal man bruge?

Internettet drejer sig grundlæggende om udveksling af informationer i formater, som både afsender og modtager forstår. En ting er nemlig, at man kan lægge tekst ud på nettet i *Wordstar 2.5.1 b for Atari* formatet, noget andet er om modtageren kan læse dokumentet. Det er klogt at bruge nogle af de mest etablerede dokumentformater, hvis man går udenfor HTML:

### Tekstbehandlingsdokumenter: RTF

*RTF* er et udvekslingsformat til tekstbehandlingsdokumenter. De fleste moderne tekstbehandlingsprogrammer kan åbne og gemme i RTF. RTF kan anbefales når man vil lægge et tekstbehandlingsdokument ud på nettet. RTF bevarer de fleste af de formateringer man kan lave i et tekstbehandlingsdokument, samt specialtegn som *æ*, *ø* og *å*. RTF er udviklet af Microsoft.

### Layoutede dokumenter: PDF

*PDF* er et format, udviklet af firmaet Adobe, beregnet til udveksling af layoutede dokumenter. PDF kan indeholde layout af professionel standard, lige så præcist som man kan lave det i sideopsætningsprogrammer som [Quark XPress](#) eller [Pagemaker](#). PDF er første og fremmest beregnet til overførsel af dokumenter, der skal printes ud på papir, og anvendes også især til distribution af manualer, bøger, brochurer mv.

PDF kræver ikke et layoutprogram for at man kan åbne filerne. Adobe har lavet et mindre, gratis program ved navn [Acrobat Reader](#) til at åbne, læse og udskrive PDF dokumenter. [Acrobat Reader](#) kan hentes fra [www.adobe.com](http://www.adobe.com)

## Ind bagved, til HTML koderne

Hidtil har jeg taget udgangspunkt i det venlige WYSIWYG-interface til at arbejde med websider. Det følgende handler om, hvordan man går ind og arbejder med selve HTML-koden.

I alle editorerne er der mulighed for at skifte over til at arbejde med selve HTML-koden for dokumentet.

Ind til selve HTML-koden	
PageMill	Edit HTML Source
Netscape Composer	Edit HTML Source (men koden sendes til et eksternt program, f.eks. Notepad.)
FrontPage	View HTML
Claris Home Page	Window Edit HTML Source

Men hvorfor overhovedet arbejde direkte i HTML-koden, når man kan lave siden med WYSIWYG? Til almindelige behov er det heller ikke nødvendigt, men for super-webdesigneren er det nødvendigt af flere grunde:

- Webeditorerne vil altid være et skridt bagefter browserne. Når en udvidelse til HTML bliver bygget ind i en ny browsergeneration, vil der normalt gå et halvt år, før der kommer en WYSIWYG-editor, der understøtter faciliteten. Hvis man vil være *cutting edge*, kan man derfor være nødt til at skrive direkte i HTML'en.
- Ingen webeditor er perfekt. For at få noget til at virke optimalt er det nødvendigt selv at trimme det direkte i HTML-koden.

### HTML fra bunden

HTML er tekst med *tags*. Tags er formateringskoder, der markerer et stykke tekst og fortæller noget om det, f.eks. hvordan det skal se ud. En tag består af en kode omsluttet med `<` og `>`. Den markering, en tag starter, afsluttes igen med en tag med samme navn, men med `/` foran navnet. Her er en stump HTML:

Dette er et `<B>fedt</B>` eksempel.

*B* er taggen for fed (*Bold*). Navnet på en tag er altid et enkelt ord uden mellemrum eller specialtegn i. Der er ikke mellemrum mellem de spidse parenteser og tagnavnet.

Tags kan sættes inden i hinanden:

```
<B>Det er fedt at <I>bade</I> i regnvejr.</B>
```

*I* er taggen for kursiv (*Italic*). Afsnit markeres ligeledes med tags:

```
<H1>Om badning i regnvejr</H1>
<P>Paraply er ikke relevant.</P>
<P>Men se efter evt. torden.</P>
```

*P* er taggen for et normalt afsnit (*Paragraph*), *H1* er taggen for *headline 1*. Det er taggene – og ikke lineskiftene – der fortæller hvad der er forskellige afsnit. Lineskift i selve HTML-koden har ingen betydning for hvordan websiden kommer til at se ud.

De fleste tags består af en åbnings-tag og en lukke-tag, men visse tags kan stå alene:

```
Aldrig mere
<BR>Vil jeg se
```

*BR* betyder lineskift (svarer til *Shift + Return*), og skal ikke lukkes igen.

Jeg har skrevet alle tag-navne med store bogstaver. Det er kotume og det mest overskuelige, men er ikke nødvendigt – i HTML-tags skelnes ikke mellem store og små bogstaver.

## Attributter

De fleste tags kan udvides med *attributter* – yderligere specifikationer af taggens funktion. En attribut består normalt af et navn og en værdi med lighedstegn imellem. F.eks. kan en link se således ud:

```
<A HREF="andetsteds.html">Klik ikke her!</A>
```

A betyder *anchor* (link), og attributten HREF betyder *Hyper Reference*, og værdien er en URL (i dette tilfælde en relativ URL). Et billede indsættes med IMG-taggen således:

```
<IMG SRC="/imgs/and.gif" WIDTH=100 HEIGHT=150 ALT="And">
```

En tag's attributter svarer nogenlunde til de egenskaber man kan ændre i [PageMills Inspector](#) eller [Properties](#)-dialogen i [FrontPage](#) og [Composer](#), når det tilsvarende element er valgt i editoren.

Attributnavne er ligesom navne på tags enkeltord uden mellemrum eller specialtegn i. Attributværdier kan derimod indeholde alle mulige tegn, hvis værdien blot er sat i anførselstegn. (Hvis værdien kun indeholder bogstaver, tal eller bindesteg, er anførselstegnet ikke nødvendigt).

En attribut kan også være et navn der står alene: `<BR CLEAR>`.

Der skal være mellemrum mellem tagnavnet og attributterne, og mellem hver attribut, hvis der er flere, men der er ikke mellemrum omkring lighedstegnet.

Hvis der er flere attributter er rækkefølgen ligegyldig. Attributter skal ikke gentages i afsluttende tags:

```
<FONT SIZE="+1">Denne skrift er større</FONT>
```

## Syntaksregler

Tags må ikke overlappe:

Forkert: `<P>normal<B>fed</P><P>fed</B>normal</P>`

Rigtigt: `<P>normal<B>fed</B></P><P><B>fed</B>normal</P>`

## Struktur

Et HTML dokument indeholder to hovedblokke: en HEAD-blok og en BODY-blok. HEAD-blokken indeholder information, som ikke direkte ses i vinduet, for eksempel titel. BODY-blokken indeholder det synlige indhold, dvs. alle tekstafsnit, billeder mv. Hele dokumentet er logisk nok en HTML-blok:

```
<HTML>
  <HEAD>
    <TITLE>Dokumentets titel</TITLE>
    ... yderligere meta-information ...
  </HEAD>
  <BODY>
    ... sidens synlige indhold ...
  </BODY>
</HTML>
```

## Kommentar

Kommentarer er tekst, som skal ignoreres af browsere og editorer. De indsættes mellem `<!--` og `-->`, f.eks. `<!--dette ignoreres af browseren -->`

Kommentaren må ikke indeholde `--` eller `>` (den kan altså ikke indeholde andre tags). Kommentarer vises i [PageMill](#) editoren med dette ikon:



Kommentarer kan indsættes i [PageMill](#) med [Edit|Insert Invisible|Comment](#).

## Mellemrum

HTML "minimerer mellemrum", dvs. mere end ét mellemrum mellem ord ignoreres. Mellemrum først eller sidst i et afsnit ignoreres. Linieskift i HTML-teksten behandles som mellemrum – et linieskift adskiller ord, men flere linieskift eller mellemrum vises stadig kun som et almindeligt ordmellemrum.

Man kan med andre ord frit bruge lineskift og mellemrum i HTML-koden uden betydning for udseendet i browseren.

### Specialtegn

Tegnene fra US-ASCII skrives ligeud i HTML-filen. Æ, Ø, Å og andre europæiske bogstaver skrives med specielle "kodenavne" der starter med et &-tegn og slutter med et semikolon, f.eks. `&oslash;` for ø, `&Aring;` for Å etc.

Teksten "Se så, nu begynder historien" ser sådan ud i HTML: `Se s&aring;, nu begynder historien`. (Bemærk at der ikke er < og > omkring tegnkoder).

Der findes kodenavne (også kaldet *character entity references*) på alle ikke-US-ASCII-tegn i ISO-8859-1-tegnsettet. Der findes desuden kodenavnene `&lt;` for <, `&gt;` for >, `&quot;` for anførelstegn og `&amp;` for &. De anvendes når tegnene skal ses af brugeren i stedet for at tolkes som dele af HTML-koder.

Man kan også referere tegnene med kodenummer i stedet for navn, f.eks. `&#192;` for tegn nr. 192 i tegnsættet ISO-8859-1, men ikke alle browsere forstår dette korrekt.

### Mere om HTML

Der er en oversigt over HTML-tags og attributter i hæftets on-line supplement

**The NCSA Beginners Guide to HTML**: et godt sted at gå videre herfra.

[www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html](http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html)

**Oversigt over HTML 3.2** *Web Design Group* har en god og velskrevet oversigt over

*Wilbur* (HTML 3.2's kælenavn) på [www.htmlhelp.com/reference/wilbur](http://www.htmlhelp.com/reference/wilbur).

De har også lavet oversigten som en Windows hjælpefil, hvilket er ret praktisk.


**De officielle specifikationer** kan naturligvis fås hos W3C på [www.w3.org](http://www.w3.org)


## Indsætte HTML i editoren

### Indsætte tags

I nogle editorer kan man indsætte tags direkte i WYSIWYG-tilstand:

Indsætte tag	
PageMill	Edit Insert Placeholder kan bruges til at indsætte vilkårlig HTML.
Composer	Insert HTML Tag kan bruges til at indsætte en enkelt tag.
FrontPage	Insert HTML Markup kan bruges til at indsætte vilkårlig HTML
Claris Home Page	Det kan man ikke. Øv.

Indsætter man en tag som **PageMill** ikke forstår, så vil editoren vise den som et lille ikon med et spørgsmålstegn i. Markerer man spørgsmålstegnet, vil man kunne se og ændre taggen i Inspector. 

Med **Edit|Insert Placeholder** kan man indsætte vilkårlig HTML-kode i **PageMill**. Et ikon der forestiller en kegle vises, og markerer man den kan man ændre indholdet i **Inspector**. Man kan endda indsætte et billede som vil blive vist i editoren i stedet for kegle-ikonet. Hvad man skal bruge det til, ved jeg ærligt talt ikke. Hvis du finder ud af det, må du gerne sende en e-mail. 

## Indsætte attributter i WYSIWYG-tilstand

Nogle editorer giver mulighed for at indsætte attributter i en tag, selvom editoren er i WYSIWYG-tilstand. Dialogboksen for at ændre egenskaber ved et element (menu via højre museknap) har en knap til at indsætte ekstra attributter. Attributterne bliver indsat i den tag der svarer til det element man ændrer egenskaber på.

Indsætte attribut	
PageMill	Det kan man ikke. Øv.
Composer	Extra HTML i <i>properties</i>
FrontPage	Advanced i <i>properties</i>
Claris Home Page	Extra HTML på <i>Advanced</i> -fanebladet i <i>Object Editor</i>

## Meta-information om dokumentet

Med **Meta**-taggen kan indsættes metainformation, dvs. beskrivelser af et dokument i dokumentet. For eksempel kan man beskrive dokumentet med nogle nøgleord, som søge- og indekseringsmaskiner så kan bruge.

**Meta**-taggen indsætter man i dokumentets hoved og den har attributterne **name** og **value**. F.eks:

```
<META
  NAME="keywords"
  CONTENT="sex, l&oslash;gn, video">
```

I **Name** angives hvad man beskriver, og **Content** er så selve indholdet/beskrivelsen.

Indsætte meta-tags	
PageMill	<b>Meta</b> -tags må indsættes manuelt i <i>Edit HTML source</i> . Husk, at de skal ind i <i>HEAD</i> -markeringen.
Composer	<i>Page Properties</i> , fanebladet <i>META Tags</i>
FrontPage	<i>Page Properties</i> , fanebladet <i>Custom</i>
Claris Home Page	De komplette <b>Meta</b> -tags må indsættes via <i>Edit Document Options</i> , fanebladet <i>HTML Extras</i> , feltet <i>Text in Head Section</i>

HTML-specifikationerne giver ikke nogen regler for, hvilke **Name**-værdier man kan bruge, det er op til folket at finde på anvendelige metabeskrivelser. De mest anvendte meta-navne er **Keywords** og **Description**, som anvendes af flere søgemaskiner, for eksempel *AltaVista* og *Infoseek*. **Description** er en kort beskrivelse af siden:

```
<META
  NAME="Description"
  CONTENT="T. S. Eliot forum. Indeholder samlede digte, biografi,
  bibliografi, analyser samt en statusbjælke-lysavis">
```

Beskrivelsen vil blive vist i *AltaVista* i stedet for – som normalt – de første par linier i teksten. De fleste editorer indsætter automatisk en "signatur" i dokumenter; for eksempel:

```
<META NAME="Generator" CONTENT="Adobe PageMill 2.0 Win">
```

Det er ikke utænkeligt at nogen et eller andet sted indekserer disse signaturer...

Du kan også selv signere de sider du laver:

```
<META NAME="Author" CONTENT="Sofus Clausen">
```

## Style sheets

Med *style sheets* kan man styre typografien på websider. Som du har sikkert har opdaget, har man i HTML kun beskedne kontrol over typografi, og man skal nærmest snyde med HTML for at lave noget, der ser godt ud. Style sheets, (udviklet af W3C) gør det muligt at styre skriftsnit, skrifttyngde, kapitæler, skydning, knibning, ordmelletrum, indrykning, størrelse og farve på tekst, samt marginer, forskellige typer rammer, og baggrundsfarve eller -grafik på alle elementer.

Style sheet-standarden er imidlertid så ny, at den endnu kun er understøttet i [Navigator 4](#), og i en begrænset form i [MSIE 3](#). Den fulde standard vil dog også være understøttet i [MSIE 4](#), så der er ingen tvivl om, at standarden vil slå an og med tiden blive den gældende måde for layout af websider. Derfor vil jeg bruge lidt plads på fænomenet.

Princippet i style sheets er enkelt: I et dokumentets hoved er en blok, hvori man angiver regler for hvordan markeringerne i HTML'en, skal vises typografisk. Man kan for eksempel erklære at alle overskrifter skal vises med 30 punkt Helvetica. På den måde adskilles form og indhold: I selve HTML'en markeres tekstens struktur og tekstelementernes funktion, i style sheetet defineres deres typografiske udseende.

Et style sheet kan indsættes i et dokumentets hoved, men det kan også være et separat dokument, som er linket til siden. Et separat style sheet kan være linket til flere sider.

### Fordelene ved style sheets

Style sheets er geniale af flere grunde:

- Ved kun at ændre en enkelt typografi-specifikation, kan man ændre layoutet af en hel side eller mange sider
- Fordi man ikke har brug for typografiske tags, bliver selve HTML'en enklere
- Fordi reglerne bygger ovenpå de eksisterende tags, kan man linke style sheets til eksisterende sider, blot ved at tilføje en enkelt linie
- Man kan bruge de betydningsmæssige markeringer, som har betydning for søge- og indekseringsmaskiner – og stadig få en flot typografi

Mig bekendt understøtter endnu ingen WYSIWYG-editorer style sheets, men man kan rimeligt nemt arbejde med et style sheet ved at have det åbent i en teksteditor, og samtidig have et dokumentet, der implementerer style sheetet, åbent i browseren. Hver gang man vil afprøve en ændring i style sheetet, gemmes og reloades dokumentet i browseren.

Style sheets bliver naturligvis først for alvor interessante, når browsere med understøttelse bliver almindeligt udbredte. Det tager mindst et halvt års tid. Til den tid vil der sikkert også være kommet nye versioner af editorerne, som understøtter WYSIWYG-arbejde med style sheets.

### Syntaksen

Style sheets skrives i et simpelt typografisk sprog der hedder CSS (Cascading Style Sheets). Et style sheet består af en række såkaldte regler, som beskriver hvordan en tag (eller en kombination af tags) skal se ud typografisk. For eksempel:

```
H1 { font-family: Helvetica }
```

Betyder at alle H1-afsnit (det vil sige de største overskrifter) skal vises med skrifttypen Helvetica. Flere deklARATIONER kan indgå i en regel, f.eks.

```
P { color: blue; font-size: 12pt }
```

Der kan laves regler for alle HTML-tags som indgår i dokumentet. Ved hjælp af en ny attribut, CLASS, som kan bruges i alle tags, kan man underindele tags, f.eks. lave flere forskellige typer *Paragraph*-afsnit, og dermed lave mere præcise regler.

## Indsætning af style sheets

Et style sheet indsættes i et dokumentets hoved via en ny tag ved navn **Style**:

```
<HEAD>
<TITLE>Eksempel på indsætning af style-sheet i dokumenthoved</TITLE>
<STYLE>
<!--
  P { color: blue; font-size: 12pt }
  H1 { font-size: 24pt }
-->
</STYLE>
</HEAD>
```

Style sheetet er indsat i en kommentar-tag, så browsere og editorer, der ikke er forberedt på style sheets, ikke skal kløjes i taggens indhold. Hvis en browser derimod understøtter style sheets, vil den ignorere kommentartegnene i **style**-taggen og læse style sheetet. Hvis style sheetet er et separat dokument, skal der henvises til det via en **Link**-tag i dokumentets hoved:

```
<LINK REL=StyleSheet HREF="dennyestil.css" TYPE="text/css">
```

Style sheetet kan skrives som en almindelig tekstfil, f. eks. i en teksteditor som **Notesblok**. Filen med style sheetet skal have efternavnet **.css**.

## Mere om CSS

Læs mere om style sheets i hæftets online-supplement.

**Brugsanvisning** De komplette specifikationer på CSS samt en god brugsanvisning kan findes hos den tidligere omtalte på **Web Design Group** på [www.htmlhelp.com](http://www.htmlhelp.com).

**Web Style Sheets** alt om style sheets: [www.w3.org/Style](http://www.w3.org/Style)

**CSS Gallery** Eksempler på style sheets: [www.microsoft.com/truetype/css](http://www.microsoft.com/truetype/css)

## Lag og absolut placering

Næste skridt i udviklingen af CSS hedder *Layers and absolute placement*. *Layers* betyder, at man kan lade flere lag af tekst og grafik overlape, og *absolute placement* betyder, at man kan angive et tekst- eller grafikelements præcise placering på skærmen. Til sammen giver det HTML ret avancerede layoutmuligheder.

Specifikationerne findes på <http://www.w3.org/TR/WD-positioning>.

Både **Navigator 4** og **MSIE 4** understøtter disse specifikationer.

## Gør siden hurtigere!

Den rutinerede websurfer har ca. 10 sekunders tålmodighed, længere gider han ikke at vente på at en webside er indlæst. Begyndere har normalt længere tålmodighed, men det går hurtigt over.

Det almindeligste modem for hjemmebrugere kan overføre 28.800 bits i sekundet. Det er ca. tre en halv kilobyte i sekundet eller 35 kilobyte på ti sekunder. Det er ikke særligt meget!

I [PageMill](#) under [Edit|Download Statistics](#) og i [Claris Home Page](#) under [Edit|Document Statistics](#) kan man se en sides størrelse inklusive grafik samt anslået indlæsningsstid. Selve HTML-dokumentet fylder ikke det store, fordi det er tekst, men billeder er tidskrævende.

- Store, tunge billeder øverst på en forside er især farlige. Det, som brugeren egentlig er utålmodig med, er nemlig hvor lang tid der går, før den *synlige* del af dokumentet er læst ind. Hvis dokumentet er længere end skærmens størrelse, så vil det tage lidt tid før brugeren scroller nedad, og i den tid vil tekst og billeder længere nede blive læst ind.
- Tabeller kan først tegnes, når hele tabellen er indlæst. Hvis hele sidens indhold er inde i en tabel, så vil siden være blank indtil det hele er læst ind. Del eventuelt tabellen op i mindre bidder.
- Genbrug billeder. Når først et billede er hentet en gang bliver det liggende i browserens cache, og hentes derfra hvis det bruges igen på andre sider. Hvis du for eksempel har en bjælke på hver side, hvor en del af grafikken skifter fra side til side (for eksempel en sideoverskrift) og en fast del (for eksempel nogle ikoner), så split den op i to billeder, hvor kun det ene skifter.

Billedernes størrelse i kilobytes har ikke særligt meget med deres mål at gøre. Det afhænger i meget større grad af, hvor effektivt de er komprimerede:

- Gem JPEG billeder med så stort et kvalitetstab som æstetisk tilladeligt.
- Ensfarvede flader fylder næsten ingenting i GIF-komprimeringen, det der fylder er "overgange". Jo simplere billedet er, jo mindre fylder det. Jo færre farver der er, jo mindre fylder det. Undlad *dither*, når du reducerer farver i GIF-billeder. Prikkede områder fylder langt mere end ensfarvede.

## Sikkerhed

Websites behøver ikke være tilgængelige for alle og enhver. Man kan lave sider, hvor den besøgende skal indtaste et password for at få siden at se. Dette kan man imidlertid ikke lave i HTML. Det skal konfigureres på serveren og varierer fra server til server, hvorledes (men det er ikke specielt kompliceret). Så tal med serveradministratoren om det. Password-beskyttelse er dog ikke fuldstændig sikkert – en snedig hacker vil stadig kunne opsnappe siderne ved at "aflytte" en bruger, der har adgang til dem.

Skal kommunikationen være helt sikker, skal den ske med en såkaldt SSL-server. SSL står for *Secure Socket Layer* og betyder, at al kommunikation mellem server og bruger er kodet. SSL bruges blandt andet til pengeoverførsler og andre transaktioner, som kræver høj sikkerhed. Igen er det et serverteknisk spørgsmål, om serveren understøtter SSL. Spørg serveradministratoren.

## www.migselv.dk

Adressen [www.etwebhotel.dk/~sofie](http://www.etwebhotel.dk/~sofie) er lidt besværlig, det kunne være nemmere (og sejere), hvis jeg havde adressen [www.sofie.dk](http://www.sofie.dk). Så kan man ikke se at jeg bor på webhotel. Det kræver at jeg ejer *domænenavnet* [sofie](#) under det danske topdomæne [dk](#). Via en internetudbyder kan du købe et sådant domænenavn, hvilket koster ca. 1.000 kr. Man kan i øvrigt også registrere et domænenavn under de internationale topdomæner [com](#), [org](#) m.fl.

Der er ingen teknisk grund til at få sit eget domænenavn. Man får bare en nemmere adresse, der ser mere professionel ud. Folk har lige så nemt (eller svært) ved at finde en website gennem Jubii, Yahoo eller enhver anden søgemaskine, hvad enten websiten har eget domænenavn eller ej. FTP-adgangen foregår i øvrigt på samme måde, selvom du har eget domænenavn.

## DEL 3: Interaktivitet og multimedia

### Objekter

Et objekt er et eller andet medie ud over tekst, som kan optræde på en webside; for eksempel animationer, Java-applets, film, musik etc. Teknisk set er billeder også objekter, selvom man normalt ikke tænker på dem som sådanne. Objekter har en række ting til fælles:

- de er ikke en del af selve HTML-filen, men er data i separate filer, som der så henvises til i HTML-koden med en speciel tag.
- de optager et rektangulært område på websiden (undtaget er dog en musik-plug-in, der kan være usynlig).
- de kræver at browseren kan håndtere det. Visse objekttyper som GIF- og JPEG-billeder er understøttet i alle browsere. Andre såsom Java eller ActiveX er understøttet i nogle browsere, og andre igen såsom som plug-in-objekter kræver, at browseren er udvidet med et hjælpeprogram (en plug-in), som kan håndtere disse data.
- de reagerer på parametre, som indsættes i HTML'en i forbindelse med taggen, der indsætter objektet. For eksempel kan lyd-afspiller plug-in'en forstå parametret **loop=true**. Det er forskelligt fra objekt til objekt, hvilke parametre de reagerer på.

Der bruges forskellige HTML-tags til at indsætte objekter; **img** til billeder, **applet** til Java, **embed** til plug-ins. De har dog nogenlunde de samme egenskaber – URL til filen, objektets bredde og højde mv. Objekter indgår i layoutet på samme måde som billeder, dvs. de følger tekstens flow, kan højre- eller venstrestilles, indsættes i tabeller etc.

W3C har udviklet en **object**-tag, som samler funktionaliteten fra **img**, **applet**, **embed** og også kan bruges til meget andet. Den er dog indtil videre kun understøttet af de allernyeste browsere. ActiveX-komponenter indsættes med **object**-taggen.

Indsætte objekt	
PageMill	File Place. PageMill finder selv ud af objektets art ved filnavnets efternavn
Composer	understøtter ikke indsættelse af objekter (andet end billeder)
FrontPage	Insert Other Component Plug-in / Java Applet / ActiveX Component
Claris Home Page	Insert Plug-in / Insert Applet

### GIF-animationer

De fleste animationer på websider er *GIF-animationer*. GIF-animationer er en udvidelse til GIF-grafikformatet, som gør det muligt at "pakke" en række billeder i en enkelt GIF-fil, og specificere afspilningshastighed mv. GIF-animationer har ligesom almindelige GIF-filer efternavnet **.gif** og indsættes på websiden som et almindeligt billede. Der kræves ingen plug-in eller lignende for at spille GIF-animationer. Ældre browsere vil ikke kunne vise animationen, men vil vise det første billede, og det samme vil en editor formodentlig gøre, når man indsætter animationen.

GIF-animationer kan laves i flere forskellige programmer, for eksempel:

*GifBuilder* til Macintosh: [iawww.epfl.ch/Staff/Yves.Piguet/clip2gif-home/GifBuilder.html](http://iawww.epfl.ch/Staff/Yves.Piguet/clip2gif-home/GifBuilder.html)

*Gif Construction Set* til PC: [www.mindworkshop.com/alchemy/gifcon.html](http://www.mindworkshop.com/alchemy/gifcon.html)

Princippet i disse programmer er, at man henter en række enkeltstående billeder ind i programmet og disse samles så til en enkelt GIF-animations-fil. Man kan angive timingen af billedskiftene, om animationen skal spilles en enkelt gang eller køre i ring, og lignende.

GIF-animationer bliver afspillet af browseren, mens de bliver indlæst, dvs. afspilningshastigheden afhænger af, hvor hurtigt de bliver indlæst. Hvis animationen kører flere gange, vil den køre hurtigere de følgende gange, fordi filen er indlæst. Læs mere om GIF-animationer i hæftets online-supplement.

## Plug-ins

Plug-ins er hjælpeprogrammer, som gør browseren i stand til at håndtere nye dokument- eller medieformater. Læs i *Ud på World Wide Web*; afsnittet *Plug-ins* om installation og brug af plug-ins.

Man indsætter et plug-in-objekt (dvs. en fil, som skal afspilles/fremvises af en plug in), på en side, på nogenlunde samme måde som et billede. Filformatet afgør, hvilken plug-in browseren vil anvende til at afspille/fremvise objektet. Man skal altså ikke specifikt angive et plug-in program, som skal anvendes.

Naturligvis kan man ikke være sikker på at brugeren gider downloade og installere plug-in-programmet. Det er heller ikke alle plug-ins, der findes til alle systemer.

Plug-ins til lyd i MIDI, WAV og AIFF formaterne, og film i QuickTime eller AVI, følger med *Navigator* som standard. Herudover findes der et utal af plug-ins til et utal af forskellige dataformater. De mest populære er VRML (3-d verdner), *Shockwave* (animation og multimedia) og *Realaudio* (realtime lyd, dvs. internet-radio).

Det besværlige er naturligvis at lave selve filen, som skal afspilles/fremvises. Standardformater som MIDI, VRML og Quicktime kan laves i mange forskellige programmer. Andre formater, som f.eks. *Shockwave Flash*, skal udvikles i et specielt program, der også sælges af plug-in udvikleren. Igen andre plug-ins, som f.eks. *Realaudio*, kræver en udvidelse på serveren, for at den kan transmittere lyden flydende ("streaming"). Normalt er plug-ins gratis, mens det er udviklingsprogrammerne eller serverudvidelser, plug-in-producenterne tjener på.

Mine råd angående plug-ins er:

- Brug kun plug-ins, hvis der er en grund til det. Hvis der er en væsentlig grund til at præsentere information i et plug-in format, så er det også sandsynligt at målgruppen vil ulejlige sig med at downloade og installere plug-in'en.
- Lav en link til et sted, hvorfra plug-in-programmet kan downloades.
- Brug de mest udbredte plug-in formater. Undgå underlige og sjældne plug-in formater eller formater, der kun findes til en enkelt platform.

Læs mere om plug-ins i hæftets online-supplement.

## Java-applets

*Applets* er små programmer, lavet i programmeringssprog *Java*, som kan være indsat på en webside. *Java* er et programmeringssprog specielt udviklet til at skrive programmer, der hentes over netværk. Programmer lavet i *Java*, udmærker sig ved at være platformsuafhængige, dvs. det samme *Java*-program kan uændret køre på Windows, Mac, UNIX, OS/2, Amiga m.fl.

Der er indbygget visse sikkerhedsforanstaltninger i *Java*, som gør at en applet ikke kan smitte med virus, stjæle information fra ens computer eller andre farlige ting (med mindre man eksplicit giver den lov). *Java* er udviklet af Sun Microsystems.

Applets bruges mest til småting som animationer, små spil og lignende, eller som brugerinterface til et program, der ligger på en server. *Java* kan dog også bruges til egentlige programmer, f.eks. er man ved at omskrive Wordperfect-tekstbehandlingen til *Java*. Man skal være programmør for at lave *Java*-applets. Derimod er det ret simpelt at indsætte en færdiglavet applet på en webside. Læs mere om *Java* i hæftets online-supplement.

**Java** Javas officielle website: [java.sun.com](http://java.sun.com)

**Tutorial** Suns *Java*-lærebog: [java.sun.com/docs/books/tutorial](http://java.sun.com/docs/books/tutorial)

**JavaWorld** magasin for *Java* samfundet: [www.javaworld.com](http://www.javaworld.com)

## ActiveX-kontroller

ActiveX-kontroller er små programmer som kan indsættes på websider, på samme måde som Java-applets. En ActiveX-kontrol skal dog skrives specielt til en platform, f.eks. Windows95, modsat Java-applets som er platformsuafhængige. Til gengæld er ActiveX-kontroller mere direkte integreret med systemet, hvilket giver mere avancerede muligheder – og større sikkerhedsrisiko.

ActiveX-teknologien har ikke så stor anvendelighed på internettet, da systemet kun fungerer på Windows95/NT, og kun hvis man bruger MSIE 3. I mindre, interne netværk, som f.eks. i en virksomhed hvor alle har Windows95-computere, kan det derimod være et stærkt værktøj.

ActiveX er udviklet af Microsoft, og er en del af deres strategi om at skabe et internet, der er “bedst set med Windows95/NT”. Læs mere om ActiveX i hæftets online-supplement.

## JavaScript

JavaScript er et simpelt programmeringssprog, som kan bruges i HTML-dokumenter. Det kan bruges til at gøre websider mere interaktive. F.eks. kan man

- kontrollere at en formular er udfyldt, før den sendes
- vise rulleaviser eller meddelelser i statusbjælken, f.eks. når musen føres over en link.
- udskifte et billede, når musen føres henover det
- styre mere komplicerede frameset, hvor man for eksempel skal skifte flere frames på en gang
- åbne nye vinduer med en specifik størrelse og evt. uden menu- og værktøjsbjælker

Desuden kan JavaScript sende og modtage beskeder fra objekter på siden som Java-applets, Plug-ins og lignende. Trods navnet er JavaScript ikke i familie med Java. Begge sprog er programmeringssprog specielt designet til webbet, men i deres funktion er de ret forskellige. Java-applets er programfiler i sit eget filformat. De kan optræde på websider, men har i øvrigt ikke noget med HTML at gøre og kan også køre uden en webbrowser. JavaScript er derimod kode, der optræder direkte i HTML-filen og manipulerer de forskellige elementer på websiden.

JavaScript er også simple og mere løst end Java, og nemmere at lære.

Umiddelbart er JavaScript en lovende teknologi, men der er visse problemer. Sproget er udviklet og kontrolleret af Netscape, og er ikke nogen egentlig standard. Microsoft har lavet en kopi af sproget som hedder JScript, men den “laveste fællesnævner”, dvs. den del af sproget der fungerer på både MSIE 3 og Navigator, er ret begrænset, og der er kompatibilitetsproblemer mellem de to versioner. Læs mere om JavaScript i hæftets online-supplement.

### JavaScript 1.0

Det mest komplette materiale om JavaScript findes hos Netscape:

[www.netscape.com/eng/mozilla/Gold/handbook/javascript](http://www.netscape.com/eng/mozilla/Gold/handbook/javascript)

### JavaScript 1.1

også hos Netscape: [www.netscape.com/eng/mozilla/3.0/handbook/javascript](http://www.netscape.com/eng/mozilla/3.0/handbook/javascript)

### En lærebog i JavaScript:

[www.webconn.com/java/javascript/intro/](http://www.webconn.com/java/javascript/intro/)

### Inkompatibiliteter:

En vigtig oversigt over forskelle mellem Netscapes JavaScript og Microsofts JScript:

<http://www.webcoder.com/browsersupport>

## CGI & formularer

CGI muliggør interaktion mellem websider og programmer på serveren. Med CGI kan brugeren via en webside sende information til serveren, og en server kan generere og returnere websider baseret på denne information til brugeren.

CGI adskiller sig fundamentalt fra “mobil kode” som Java, JavaScript og ActiveX, der sendes til brugerens computer og udføres der. CGI programmer udføres derimod på serveren, og outputtet er altid en ny webside, som sendes til brugeren. CGI stiller derfor ikke nogen krav om at brugeren skal have en bestemt browserversion eller lignende, sådan som Java, ActiveX og JavaScript. CGI-programmer arbejder ofte sammen med databaser, der ligger på serveren.

Et CGI-program kaldes enten via en link, eller ved at man trykker på en knap i en *formular*.

Eksempel: Hos [Yahoo](#) skriver man nogle ord ind i søgefeltet og trykker på knappen “search”. Søgeordene bliver så sendt til CGI-scriptet, som foretager en søgning i sin database, og derpå genererer en HTML-side med søgeresultater. Denne returneres derefter til brugeren.

Desværre er det ikke alle web-hoteller, der giver mulighed for CGI-programmer. Det skyldes at der er en vis sikkerhedsrisiko forbundet med CGI, og at det belaster serveren hårdere end almindelige websider. Visse systemansvarlige kræver at gennemse CGI-programmerne for sikkerhedsbrister, før man må anvende dem.

Ofte vil det være en programmør der laver selve CGI-programmet, og webdesigneren skal kun bekymre sig om interfacet, dvs formularen på websiden.

### Formularer

Brugeren sender information til CGI-programmet via en *formular* (på engelsk *form*). En formular består af et eller flere felter (tekstfelter, afkrydsningsfelter el.lign.), samt en *submit*-knap. Når man trykker på submit-knappen, sendes den information, der er udfyldt i felterne, til CGI-programmet.

Alle felter har egenskaben **Name**. Disse er vigtige, for at CGI-programmet kan skelne de tilsendte værdier fra hinanden.

Knapper til indsættelse af formularfelter	
PageMill	Knapper på den nederste værktøjsbjælke.
Netscape Composer	understøtter ikke formularer
FrontPage	View Forms Toolbar
Claris Home Page	Window Show Forms Palette

Der findes følgende formularfelter:

#### Textbox

et felt man kan indtaste en tekst i. Feltet er en enkelt linie højt. Har egenskaberne **Size** og **Maxlength**, som angiver henholdsvis hvor bredt feltet er (i antal tegn), og hvor mange tegn det kan indeholde.

#### Password

ligesom **textbox**, bortset fra at de indtastede tegn vises som stjerner. Bruges når der skal indtastes password. NB! Skal ikke bruges til vigtige passwords – ordet bliver nemlig sendt som klartekst, så en spion kan opsnappe det.

#### Textarea

et felt til indtastning af tekst. Feltet kan være flere linier højt, og have scrollbars.

#### Checkbox

et afkrydsningsfelt. Egenskaben **Checked** angiver at boksen er valgt fra starten.

The image shows a collection of form elements:
 

- Textbox:** A single-line text input field with the placeholder text "Her kan man skrive noget...".
- Password:** A single-line text input field with a password mask (dots).
- Select:** A dropdown menu with "Værid" selected.
- Checkbox:** A group of three checkboxes: "sex" (unchecked), "mad" (checked), and "penge" (unchecked).
- Radio:** A group of three radio buttons: "blæse" (unchecked), "have mel i munden" (unchecked), and "spise" (checked).
- Textarea:** A multi-line text input field with the placeholder text "Her kan man skrive flere linier tekst...".
- Buttons:** Two buttons labeled "Submit" and "Reset" at the bottom.

**Radio**

en rund knap der kan markeres på samme måde som afkrydsningsfelter. Forskellen er, at når radioknapper optræder i grupper, kan kun en af dem markeres ad gangen – ligesom på en transistor-radio, hvor kun en af knapperne kan være trykket ned af gangen. Radio-knapper er placeret i samme gruppe, hvis de har samme navn. **Checked** angiver at knappen er valgt fra starten

**Select**

en liste af elementer der kan vælges. Der findes to typer: en hvor man kun kan vælge et enkelt element, og en hvor man kan vælge flere. Man angiver hvor mange valgmuligheder der skal være, og om man kan vælge flere af gangen. Man skal angive **Name** og **Value** for hver valgmulighed på listen.

**Hidden**

et felt der er usynligt for brugeren, men som alligevel indeholder information der skal sendes til serveren. I [PageMill](#) indsættes et skjult felt med `Edit|Insert Invisible|Hidden Field`.

**Submit** knappen som sender formularen. Skal være til stede i alle formularer.

**Reset**-knap sletter alt indtastet i formularen.

Formularen som helhed har to egenskaber:

**Action** er URL'en på CGI-programmet som informationen skal sendes til. (Et CGI-program har også en URL).

**Method** er den metode, hvorved informationen skal sendes, og den kan være **Get** eller **Post**.

Begge disse værdier bestemmes af CGI-programmøren.

Egenskaber for formularen	
PageMill	i Inspector, på fanebladet <b>Form</b>
Composer	understøtter ikke formularer
FrontPage	<b>Form Properties</b> , knappen <b>Settings</b>
Claris Home Page	<b>Edit Document Options Advanced</b>

[PageMill](#) og [Claris Home Page](#) går for nemheds skyld ud fra, at der maksimalt er én formular på en side, selvom der egentlig kan være mange ifølge HTML-specifikationerne.

**E-mail formular**

Hvis du ikke har adgang til at skrive CGI-programmer, kan du lave en formular, hvor indholdet bliver sendt direkte til en e-mail-adresse, i stedet for et CGI-program.

Det laves ved at sætte `mailto:mig@min.adresse` i **Action**-feltet. **Method** skal være **post**.

NB! Dette virker i [Netscape Navigator](#), men ikke i [MSIE](#).

**GGI programmering**

CGI står for *Common Gateway Interface* og er egentlig kun specifikationer for, hvordan programmet på serveren får sit input, og hvorledes output skal være.

Selve programmet på serveren kan være skrevet i et hvilket som helst programmeringssprog, for eksempel C/C++, Perl, Java, eller Basic. Desuden findes der systemer, hvor man på serveren kan kombinere HTML-sider med JavaScript eller lignende sprog, som bliver udført før resultatet (den modificerede HTML-side) sendes til brugeren. Scriptene har desuden adgang til nogle standardobjekter på serveren, f.eks. databaser. Netscapes [LiveWire](#), Microsofts [Active Server Pages](#) og NeXT's [Webobjects](#) er sådanne systemer. Læs mere om CGI i hæftets online-supplement.

**CGI specifikationerne:** [hoohoo.ncsa.uiuc.edu/cgi](http://hoohoo.ncsa.uiuc.edu/cgi)

**Mere om CGI:** [www.cgi-resources.com](http://www.cgi-resources.com)

## Del 4: Planlægning & design af en website

Nem adgang til information er ikke det samme som viden. – *Terry Pratchett*

Online hypertext er langt fra det ideelle medie. Tekst er sværere at læse på en computerskærm end på papir, og jo mere man skal koncentrere sig for at læse noget, jo mindre får man ud af det man læser. På en skærm mangler man desuden fornemmelsen af “hvor man er” og “hvor langt man er”, sådan som man har ved rent fysisk at have en bog eller et blad i hånden. Webbets decentrale, kaotiske struktur med hyperlinks på kryds og tværs, gør det ikke bedre.

I en website må man derfor vægte læselighed, overskuelighed og klar kommunikation over smart design, *corporate identity* mv. Dette er forskelligt fra papirmedier, hvor man sagtens kan tillade sig et ikke helt læseoptimalt layout, hvis layoutet gør læsningen mere spændende.

Det betyder naturligvis ikke, at man ikke må lave flot layout og typografi på en website, kun at layout og design skal være *underordnet* det der skal kommunikeres – naturligvis med mindre *designed* er dit budskab. På en forside til en website er fængende design og/eller tydelig *corporate identity* naturligvis en vigtig del af “budskabet”.

Mediets begrænsninger gør det nødvendigt med grundig planlægning, inden man går i gang med at lave de enkelte sider. Begynder man med at lave selve siderne, er det nemt at blive fanget i de tekniske muligheder og begrænsninger, og miste overblikket over hvad man vil med websiten.

### Arbejdsplan

En naturlig arbejdsgang for at udvikle en website:

1. Definer formål, målgruppe og indhold.
2. Planlæg hvorledes indholdet skal struktureres og organiseres, og find ud af hvorledes besøgende skal kunne navigere rundt.
3. Design skabeloner til sidelayout og typografi.
4. Lav de enkelte HTML-sider i webeditoren.
5. Afprøvning.

Det er min erfaring at man *ikke* skal blande ovenstående trin sammen. Det er dumt at begynde at lave layout før man har lagt sig fast på en overordnet struktur, ligesom det er dumt at begynde at lave de enkelte sider i editoren, før formålet med websiten er klarlagt.

Man kan ændre og tilføje sider løbende, så længe det sker indenfor rammerne af den overordnede

struktur. Hvis man derimod vil ændre det overordnede design eller formålet med siden, så er det vigtigt igen at planlægge det og tænke alle tre punkter igennem, før man går i gang med at ændre selve siderne. Jo bedre man udfører forarbejdet, jo nemmere går det i sidste ende med at lave siderne og opdatere siden. (Jeg taler her af bitter erfaring).

Inden du annoncerer hvornår din website går i luften, så husk Hofstadters Lov: *Det tager altid længere tid end man regner med – selv hvis man medregner Hofstadters Lov.* (Douglas R. Hofstadter)

Prøv at glemme alt om de tekniske aspekter ved websiten, mens du er i planlægningsfasen. Koncentrer dig om, hvad du vil med siden.

### Slap af

Ingen grund til panik, hvis du ikke har fået oprettet en website endnu. Mange, især virksomheder, har den forestilling at informationsteknologi er som et tog man skal nå – det gælder om at hoppe ombord inden det er kørt. Sandheden er naturligvis, at informationsteknologi er et produkt, som nogen lever af at sælge. Det vil sige, at produktet vil være til at få, så længe der er nogen der er interesseret i at købe.

Faktisk er internettet nok et af de steder hvor man får mindst ud af at være “først med på vognen”. Alt ændres så hurtigt, at det er ganske ligegyldigt om du var med eller ej for et år siden, alt er alligevel anderledes nu.

### Definer dit formål

Gør dig klart hvad formålet med websiten er.

Formålet kan f.eks. være:

- Image – at kunne fortælle venner/forretningspartnere at vi har en website, at kunne skrive URL på visitkort etc.
- Online visitkort – websiten er en måde folk hurtigt kan finde dit firmas eller dit eget navn, adresse telefonnummer, e-mail-adresse mv.
- Feedback – websiten er en nem måde for folk at komme i kontakt med dig, og sende meddelelser til dig.
- Kommunikation – en gruppe mennesker kan bruge din website som forum for kommunikation omkring et emne.
- Information. Du vil fortælle noget, som nogen er interesseret i at høre. Det kan være et nyhedsbrev for din organisation eller en videnska-

belig afhandling, du mener vil have interesse for andre. Det er måske den mest udbredte form for websites.

- Reklame/salg af et produkt. Dette er nok den form for websites, der er sværest at få til at virke efter hensigten.
- Fordi du synes det er spændende.

Anbefalingerne i de følgende afsnit går mestendels ud fra, at formålet med websiten er kommunikation.

## Målgruppe

Hvem kan være interesseret i din website? Hvem skal komme, og hvad skal få dem til at komme? Husk på at "folk" er ligeglade med hvad du vil fortælle/sælge/have dem til, spørgsmålet for dem er, om du har noget *de* gider at komme efter. Hvis du ikke har noget at byde på, så dur webbet ikke. Husstandsomdelte reklamer, avisannoncer, tv-reklamer etc. er alle måder at "fortælle folk noget de ikke vil vide", mens webbet kræver, at folk foretager en bevidst handling, for at komme til din side.

Jo mere specifik og nicheagtig din målgruppe er, jo større chance er der for at du kan ramme den. Hvis du laver en site, der skal tiltrække "alle og enhver", så konkurrerer du med Disney, MTV, Louvre, Playboy etc. Hvis du derimod målretter din site mod dåseåbnerfabrikanter eller samlere af tofarvede otteøresfrimærker, og giver dem noget de kan bruge, så er du sikker på succes.

## Sortering

Husk at lidt information er mere værd end megen information. Jo mere information der er, jo mere tid skal brugeren bruge på at sortere og finde det væsentlige. Jo større kvalificeret sorteringsarbejde, der er foretaget på informationen i forvejen, jo mere værdifuld er den for brugeren.

## Læses on-line eller printes ud?

Overvej om brugerne overhovedet vil læse dit indhold on-line. Store, sammenhængende tekster, som for eksempel en lærebog eller en afhandling, vil de fleste foretrække at downloade for at læse offline eller printe ud. Overvej om ikke RTF eller PDF vil være bedre formater til dit materiale. Det kræver dog stadig en on-line beskrivelse af dit materiale, fordi surferen skal gøres interesseret i det; man downloader ikke store tekster uden videre. Til gengæld læser man offline/udprintede tekster grundigere og med større udbytte.

## Den overordnede metafor

Indholdet skal struktureres. Hypertekst gør det muligt at linke tekst og dokumenter på kryds og tværs i et stort spindelvæv, men det betyder *ikke*, at man ikke mere behøver et tænke på disposition og rækkefølge af information. Tværtimod har det aldrig været vigtigere med klar, logisk og funktionel strukturering af information.

Brugeren skal kunne finde rundt i din website og kunne finde, hvad han skal bruge. Siten skal være struktureret efter en klar plan eller idé, som den førstegangsbesøgende umiddelbart fatter – ingen vil bruge minutter på at sætte sig ind i hvordan en website er opbygget. Det nemmeste er at basere strukturen på en velkendt metafor. Nogle eksempler:

### Avismetaforen

Siten består af en række artikler. Artiklerne skal ikke læses i nogen bestemt rækkefølge, bortset fra at artikler om et emne til en vis grad bygger på information i tidligere artikler om samme emne. Artiklerne er ordnet efter vigtighed (største overskrifter tidligst på oversigter) og aktualitet (nyeste artikler øverst på oversigter).

### Leksikon

Siten indeholder faktuel information i mindre bidder. Artiklerne hører ikke sammen i en bestemt rækkefølge, men henviser til hinanden indbyrdes.

### Bog

Siten er en bog, hvor siderne skal læses i rækkefølge fra ende til anden. Muligvis er der en indholdsfortegnelse og et indeks, der gør det muligt at springe ind forskellige steder i teksten.

Bemærk, at disse tre metaforer alle baserer sig på den overordnede metafor: *websider er papir*. Dette er en metafor som gennemsyrrer hele vores opfattelse af webbet i dag, og er derfor god at bygge på. Den gør udtryk som "forrige side" og "næste side", "hjemmeside" umiddelbart forståelige. Papirmetaforen er især velegnet til organisering af tekst, men virker mere malplaceret jo mere interaktiv siten er.

### Rummetaforen

er ligeså universel som papirmetaforen. Websiten kan være et hus med rum, man kan bevæge sig rundt imellem; en butik, hvor man bevæger sig rundt mellem hylder; en by, hvor man bevæger sig rundt mellem steder og huse.

## Træstrukturen

Man går igennem flere lag af menuer og undermenuer, mere og mere specifikke, indtil man finder det man leder efter. Træstrukturen er meget populær blandt computerfolk, fordi den gør det muligt at organisere og overskue meget store mængder information, hvis man tænker systematisk. Men gennemtænkte træstrukturer har det problem, at brugeren ikke nødvendigvis har samme opfattelse eller forståelse af emnets struktur. Det er farligt for eksempel at strukturere en website efter et firmas interne opbygning, eller strukturere en lærebog efter emnets indre logik – en logik som lærebogens målgruppe endnu ikke har overblik over.

Jo mere overblik en bruger har over et emne, jo mindre lineært søger man information. Som nybegynder læser man lærebøger fra ende til anden, som ekspert kan man dykke ned i store informationsmængder efter enkelte meget specifikke oplysninger.

I et diffust materiale er det ofte nemmere at finde hvad man leder efter i en *spindelvævsstruktur*, hvor der fra hvert afsnit er mange “associative” links til relaterede afsnit. Så kan brugeren søge information på en mere “intuitiv” måde.

## Andre strukturer

Nye strukturer for organisering af information har vundet frem på internettet. I *FAQ*'en er information opbygget som spørgsmål og svar, noget der passer godt til steder, hvor der foregår en kommunikation og udveksling.

*Threading* egner sig specielt godt til diskussioner – indlæggene er organiseret som en flad, forgrenende struktur, ordnet efter hvilke indlæg der er kommentarer til hvilke. En meget levende, men også kaotisk måde at organisere information på. Mest nyttig, hvis der samtidig findes en emneoversigt over de emner der berøres, og i hvilke artikler de kan findes.

*Kronologisk*/"*what's new*" – mange sites har en liste med ændringer og tilføjelser til en website i baglæns kronologisk orden. Via den liste kan faste besøgende holde sig ajour med en site.

## Eksempler

Man kan blande flere metaforer. Microsofts *Sitebuilder Workshop*: ([www.microsoft.com/workshop](http://www.microsoft.com/workshop)) benytter overordnet en planetmetafor for hver af de seks hovedgrupper. Det signalerer at der er forskellige verdner, som ikke hører sammen i en bestemt rækkefølge. Den enkelte “planet” er derimod opbygget som et magasin med

artikler, interviews, nyheder mv. Kombinationer af flere metaforer/strukturer giver flere muligheder for organisering, men er også sværere tilgængelige for udenforstående.

En metafor kan være meget subtil og alligevel meget effektiv. Yahoo ([www.yahoo.com](http://www.yahoo.com)) virker umiddelbart bare som en masse sider med links. Men øverst på hver side er angivet en sti til den side man er på, for eksempel “Top|Computers and Internet|World Wide Web|Page Design and Layout”. Hver overskrift er en link til den pågældende side. På den måde antydes en vej, der er opdelt i en række faser. @-tegnet bruges i links til at vise, at linkene går til en gruppe, der ligger et andet sted i træstrukturen end den pågældende gren.

Bemærk at billederne “træ” og “sti” ikke eksplisit vises, men strukturen får mig alligevel til at opfatte siden som “træer” og “stier”, hvilket gør mig bedre i stand til at finde rundt.

## Metaforenes grænser

Metaforen er ikke bare en knage til at hænge strukturen op på, den er også et redskab man bruger til at se materialet fra nye vinkler. Tænk dit materiale igennem i forhold til flere forskellige metaforer – det giver garanteret mange nye ideer.

Undgå metaforer, som er i strid med andre fastslåede metaforer. For eksempel kunne det egentlig være ganske logisk i en “bog” at sige *ned* til næste artikel, og *op* til forrige. Det er imidlertid allerede blevet fastslået, at næste artikel er *frem* (til højre) og forrige er *tilbage* (til venstre).

Metaforen må naturligvis aldrig blive så bindende, at det gør brugen af websiten sværere. Er websiten eksempelvis opbygget af “rum” efter en “hus”-metafor, vil det være logisk at man skal gå igennem et tredje rum, “hallen”, for at komme fra et rum til et andet. Men det er også unødigt besværligt. I stedet kunne man have en menu, der giver mulighed for, at den besøgende kan springe fra rum til rum. En sådan funktion ville være “magisk”. Magiske funktioner er funktioner, som ikke er “realistiske” i forhold til metaforen, f.eks. interaktive illustrationer i en online lærebog, eller muligheden for at kunne søge efter ord i en on-line bog.

Ofte er det netop de magiske effekter, som udnytter det nye i webmediet, men da de ikke er umiddelbart forståelige i forhold til metaforen, kræver de normalt en forklaring. En website baseret på blad-metaforen og helt uden magiske effekter, kunne lige så godt være et rigtigt blad.

## Søgning

Det kræver adgang til CGI-programmer at lave en søgemaskine, der kan søge efter ord på websiten. Alternativer er et håndlavet indeks (stort arbejde!) eller et indeks genereret af et hjælpeprogram.

## Indholdsfortegnelse

Der bør altid være en indholdsfortegnelse. Mindre websites kan have indholdsfortegnelsen på forsiden. Ved større sites, kan den være på en side for sig, og hovedoverskrifterne fra indholdsfortegnelsen kan så gentages på forsiden.

En kort beskrivelse af sidens indhold under hver overskrift, kan være til stor hjælp.

Indholdsfortegnelsen behøver ikke at være tekst, den kan for eksempel laves grafisk som et kort eller en træstruktur, hvis det kan gøre siden mere overskuelig.

## Kommunikation

Flervejs kommunikation er en af internettets mest spændende muligheder (overfor envejs kommunikation som en almindelig website jo er). Kommunikation kan opdeles i *real-time kommunikation* (chat, telefoni, videokonference) og *forsinket kommunikation* (e-mail, nyhedsgrupper).

Real-time kommunikation er forholdsvist kompliceret, og kræver specielt programmel på både klient og server, så det vil jeg ikke komme nærmere ind på. "Forsinket kommunikation" er derimod indenfor de fleste websites muligheder, og er desuden indholdsmæssigt ofte mere interessant.

Den simpleste måde at køre en diskussion på, er at brugeren kan sende e-mail til websiten, som webmasteren så modtager, sorterer og lægger op på websider. Det kræver meget arbejde af webmasteren, især hvis han får mange breve, men kræver ikke adgang til CGI.

Der findes til gengæld CGI-programmer som automatisk kan modtage og sortere e-mail og lægge brevene ind på websider. Nogle websites har en side med en smart formular, som man kan skrive sit indlæg ind i. Jeg tror dog de fleste foretrækker at bruge almindelig e-mail, som man i forvejen er fortrolig med. Et godt eksempel på et web-diskussionsmiljø er Hotwireds Threads: [threads.hotwired.com/threads](http://threads.hotwired.com/threads)

Browserwatch/news er et eksempel på et en anden måde at køre et "kommunikationsbaseret" website på. Her redigerer og kommenterer webredaktøren alle indlæg: [www.browserwatch.com/news.html](http://www.browserwatch.com/news.html)

## Sidens længde

Hvor lang må en side være? En side skal indeholde en "informationsenhed", det vil sige så meget information, som du regner med læseren vil læse fra en ende af uden at springe. Hvis du har en tekst på 86 sider som er helt lineær, kan du godt have den på en enkelt webside (men måske ville det være bedre at have en sådan tekst i RTF, så man kan downloade den).

Artikler, anmeldelser og lignende er ofte informationenheder – man læser den enkelte artikel lineært, men der er ikke fastlagt i hvilken rækkefølge man læser artiklerne.

## Navigation og brugerflade

En webside skal dels fortælle brugeren hvor i strukturen han er, dels give ham knapper eller links, så han kan komme videre. Denne brugerflade til websiten er ikke bare en "skal" – den skal være en direkte funktion af sitens struktur, og udledt af den samme metafor.

### Faste links

Det er links til en bestemt fast side. For eksempel til forsiden, en søgeside, en feedbackside. Hver eneste side bør have en link til forsiden, til brug hvis den besøgende kommer ind midt i strukturen, for eksempel via et hop fra en søgemaskine.

### Relative links

Det er links med en titel, der skal forstås i forhold til sidens position i strukturen. For eksempel *frem*, *tilbage* eller *op*. Disse bruger en velkendt metafor, men bør normalt ikke stå alene. I forbindelse med knappen "op" bør det f.eks. fremgå, hvor dybt nede i hierarkiet man er, og i forbindelse med frem/tilbage bør angives hvad de går hen til, f.eks. "forrige side: *Ezra Pound*- forbindelsen | næste side: Noter til *The Waste Land*"

Relative links bør kun bruges, hvis der rent faktisk er et entydigt forløb eller rækkefølge i sidernes indhold. I en artikelsamling eller et opslagsværk vil det blot forvirre, hvis man bruger frem/tilbage metaforen.

### Parallelnavigation

Ofte er en website opdelt i en række ligestillede sider eller hovedgrupper. Det er almindeligt at links til sådanne hovedpunkter vises som en vandret bjælke øverst på siden, som faneblade, eller som en *sidebar* i siden af skærmen. Det bør altid fremgå

klart hvilken af siderne eller grupperne man er i. Det skal desuden fremgå, om man er på selve det punkt, der er markeret, eller på sider under punktet.

Fordi punkterne er ligestillede behøver de ikke være på række. [www.naestvednet.dk](http://www.naestvednet.dk) benytter for eksempel en lille tegning af en by i fugleperspektiv, med vigtige bygninger som links til hovedgrupper i websiten.

### Kontekstafhængige links

I en mere diffus ulineær struktur, for eksempel en nyhedstjeneste, vil der normalt ved hver artikel være en række links til artikler om samme eller beslægtede emner. Sådanne sider vil ofte have et stort antal links i forhold til selve artiklens størrelse. Man bruger også mere tid på at søge og klikke rundt sådan et sted (lige som man bruger forholdsvis megen tid på at skimme overskrifter i en avis).

[www.news.com](http://www.news.com) er et eksempel. I forbindelse med hver artikel er tre navigationsfelter: en til højre med alle nyhedstjenestens faste rubrikker, en til venstre med nyeste overskrifter, og en for neden med artikler som har noget med emnet at gøre. Links til relevante historier kan også optræde inde i artiklen, som et ikonagtigt billede med en overskrift og teksten *relatet story*.

### Ikoner

Ikoner bør altid have en forklarende tekst. Husk på at man yderst sjældent kan gætte ud fra en ikons motiv hvad dens funktion er – først når man har lært ikonens betydning, er den nemmere at “læse” end tekst.

Pas på med uklare eller indforståede ikoner eller navne. Ofte ser man ikoner med titler som “pavillion”, “center”, “ekstra” eller andre uigenkendskuelige begreber, som måske giver mening for designeren, men ikke fortæller brugeren noget.

Det er vigtigt at navigationsværktøjerne er ensartede gennem hele websiten. Og hvis en ikon henviser til en bestemt side, bør den samme ikon naturligtvis gå igen i den sides hoved.

### Eksempler

[www.adobe.com](http://www.adobe.com) er et eksempel på en site med en overdesignet navigationsbar, som bare gøre det mere forvirrende: Under hvert produkt er menu-punkter som *Overview*, *Details*, *At work*, *Getting help*. Men hvad betyder *At work* egentlig i den forbindelse? Og hvad er forskellen på *Support* og *Getting help*? Og hvad er forskellen på *Solutions*, *Services* og *Products*? Og hvor fører *Up* hen, når

bjælken giver indtryk af det hele er en flad struktur? Og hvad betyder *Studio*? Desuden giver bjælken ingen indikation af hvor man er.

Naturligtvis er strukturen både logisk og gennemtænkt Problemet er imidlertid, at navigationsbjælken ikke gør dette klart for brugeren. Problemet er også, at de har valgt en meget stram struktur, som alle produkter så skal passes ind i.

Pas på med at benytte for tvingende strukturer! Træk aldrig en skabelon længere end den kan holde.

### Hypertekst

Links, der optræder direkte i teksten, bør give mening som en læsevej, dvs. læseren skal umiddelbart kunne følge linkene og fortsætte med at læse/blade det nye sted. Linken skal altså ikke gå ind midt i en sammenhæng, som man ikke kan forstå uden at gå baglæns (indenfor det nye steds struktur). Man skal naturligtvis heller ikke lave en link til noget information, som ikke kan forstås, hvis man ikke først læser videre i teksten, der linkes fra. Vær opmærksom på, at enhver link indebærer muligheden for at en besøgende følger den, og måske aldrig kommer tilbage til din site.

Links skal indgå i tekstens naturlige forløb, og skal ikke omtales – de er jo ret tydeligt markerede i forvejen. Skriv ikke “klik her for at gå til...”. Hvis aldrig til link ved for eksempel at skrive “nederst på siden er en link til...” eller lignende.

### Henvisninger

Hvis man har links der er relevante, men som ikke direkte kan indgå i tekstens forløb, så placer dem nederst på siden eller på en side for sig som en “litteraturliste”.

Beskriv helst ganske kort under hver link hvad den fører hen til.

Lister over “interessante steder på nettet” er kun værdifulde, hvis de er resultat af et kritisk sorteringsarbejde, og man beskriver *hvorfor* stedet, der linkes til, er interessant og hvad det kan bruges til.

## Tekstens redigering

De vigtigste navigationsredskaber i større tekster – hypertext eller ej – er ikke ikoner eller sidebars, men overskrifter.

Beslut hvor mange overskriftsniveauer du vil arbejde med. Der er sjældent brug for mere end 2-3 overskriftsniveauer. I nogle tekster virker det naturligt med en meget “dyb” struktur med mange overskriftsniveauer. I hypertext er dette dog ofte mere forvirrende end oplysende, fordi man alligevel springer rundt.

Jo mere oplysende overskrifterne er, jo nemmere finder folk hvad de søger efter. *Kapitel 2* eller §4.2.1 er ikke særligt gode overskrifter.

“Underholdene” overskrifter som forsøger at fange læserens interesse frem for at oplyse om afsnittets indhold, er desværre heller ikke så meget værd, hvis man læser teksten ulineært.

Vær nogenlunde konsekvent med, hvor lange afsnittene er, og hvor mange afsnit der er under hver overskrift. Underrubrikker (altså korte beskrivelser af artiklens indhold under overskriften) kender man fra blade og aviser, og er en god hjælp.

Brug lister og opstillinger, hvor det kan gøre teksten mere overskuelig.

### Omtal ikke mediet

Ligesom en avisartikel ikke behøver at sige “dette er en avisartikel”, behøver websites ikke at fortælle at de er websites. Faktisk kan det virke lidt irriterende, når de gør det.

Undgå derfor at skrive *Narnias website, Velkommen til Narnia on-line!* eller lignede formuleringer. En overskrift med *Narnia*, eventuelt *Velkommen!* er nok – brugeren ved godt han er online.

Skriv heller ikke “denne website indholder” – skriv blot indholdet. Skriv ikke “links til andre relevante websites...”, skriv blot “se også:”. Tal ikke om mekanikken bag siderne – kun webdesignere interesserer sig for dét, brugerne interesserer sig for indhold. Hvis det er nødvendigt at forklare hvordan man rent teknisk navigerer rundt eller får siden til at fungere (“--Maksimer skærmen så den har mindst denne bredde--”), så tyder det på, at der er noget galt med designet.

I visse situationer kan det naturligvis være nødvendigt at omtale teknikken – for eksempel hvis du bruger plug-ins, og skal forklare hvilken plug-in folk skal bruge, og hvor den kan downloades. Når man gør det, begrænser man antallet af potentielle

brugere voldsomt. Kun en brøkdel af webbets brugere kan finde ud af at downloade – endelige installere – software. Husk på at webbets publikum er meget bredt.

### Sproget

Sørg for hurtigt og klart at få fortalt, hvad din website kan bruges til. Webbets brugere har generelt en *meget* kort tålmodighed med websites, som de ikke umiddelbart kan se formålet med. Når først målgruppen er “fanget”, behøver du derimod ikke at forenkle indholdet. Tværtimod. Webbets brugere er generelt parat til at bruge tid og energi på at sætte sig grundigt ind i et emne. Ofte bruger man nettet, hvor aviser/tv/det lokale bibliotek ikke har information, der er grundig eller specialiseret nok.

Vær lige så grundig med korrektur mv. på websider, som hvis materialet var udgivet på papir.

Pas på ikke at anglificere sproget. Man kommer meget nemt til at tænke på engelsk, fordi det engelske sprog er så dominerende på internettet.

## Sidelayout

Formålet med sidelayout er, at gøre en sides indhold mere overskueligt ved at opdele det i blokke eller områder. Store mængder information kan gøres overskuelig ved et sidelayout, som inddeler sidens elementer i et hierarki af grupper og undergrupper. (Se for eksempel [www.news.com](http://www.news.com)). En tekst uden overskrifter eller inddelinger, eller med dobbelt linieafstand mellem hver anden linie, fremstår som et stort gråt felt, det er svært for øjet at overskue.

Ser man en side lidt på afstand, vil man iagttage, at hvert felt har en farve og en tyngde. Tekst er grå. Overskrifter står tungere end teksten, så øjet inddeler teksten i overskrifter og brødtekst. Lodret afstand mellem tekstdele opdeler tekstmængden i afsnit. Forskellige mængder af lodret afstand opdeler igen afsnittene i grupper.

Blokke som er udenfor teksthierarkiet – for eksempel sidehoved, navigationsbjælker, links til andre sider – kan adskilles fra teksten og fra hinanden med farver og tyngde.

Afstand er det vigtigste værktøj til at opdele siden. Jo mere luft du skaber omkring et element, jo stærkere står det frem, og jo mere læseligt er indholdet.

Når øjet skal overskue en side, bevæger det sig skråt henover skærmen fra øverste venstre hjørne til nederste højre hjørne. Læg de vigtigste informationer på denne linie, for eksempel sidens overskrift i

øverste venstre hjørne, og dernæst underrubrik/indholdsfortegnelse. Hvis du putter *sitens* overskrift eller logo til venstre, og *sidens* overskrift øverst til højre, vil brugeren bruge længere tid på at orientere sig.

Vær konsekvent med sidelayout! Det er vigtigt at alle sider har nogenlunde samme opbygning, og nogenlunde samme navigationsværktøj. Det hjælper brugeren.

Jeg vil klart anbefale at man skitserer sidelayoutet ned på papir, før man begynder at konstruere det på skærmen. Lav en skitse hvor du tegner felter med forskellige gråtoner, for at finde den rette balance. Prøv også at skitsere kompositionen så abstrakt som muligt, med ganske få liner eller former. Danner sidekompositionen nogle klare linier eller former, som øjet kan følge?

### Funktionelle blokke

De fleste websider er opbygget af funktionelle blokke: Et sidehoved øverst, med identifikation af siden: logo og navn, og sidens overskrift/emne. Nogle faste knapper med "hjem", "søg" "feed-back", "indhold" mv, en navigationsbjælke med sitens eller emnets hovedemner, og måske "frem" og "tilbage".

Navigationsknapper sidder fornuftigt i øverste højre side, fordi man for det meste først skal bruge dem, når man har dannet sig et overblik over sidens indhold.

På længere sider er det også nødvendigt med en sidefod, som gentager de vigtigste navigationsmuligheder (mindst en link til toppen af siden). Desuden er der oftest en e-mail-link til webmasteren på sidefoden, samt eventuel metainformation: copyright, dato, URL (til brug hvis en bruger printer siden ud). Den "naturlige" navigationsknap – frem til næste side eller tilbage til forsiden alt efter sitens struktur – bør gentages i nederste højre hjørne.

### Design efter det synlige felt

De fleste websider scroller man aldrig. Man skimmer det synlige felt, og springer et nyt sted hen. Fortæl derfor altid øverst på siden hvad den indeholder. Hvis den øverste overskrift ikke dækker hele siden, så lav en lille indholdsfortegnelse øverst med alle overskrifter på siden.

640x480 er den mindste, almindeligt forekommende skærmopløsning. Hvis *Navigatør* fylder hele skærmen, og alle tre værktøjsbjælker, statusbjælken, og Windows 95-bjælken er slået til, er selve browservinduet ikke større end 620x290 pixels. Man kan forøge vinduets højde til omkring 350

pixels ved at slå de unødvendige bjælker fra, ellers end man ikke kan regne med at folk har gjort det (eller ved hvordan man gør).

Sidehoved, en angivelse af sidens indhold, og alle væsentlige navigationsknapper skal helst være indefor denne "minimalskærm". Billeder bør heller ikke være større end disse mål.

På en Macintosh fylder browservinduet normalt ikke hele skærmens bredde, der er en stribe i højre side, hvor man kan se skrivebordet. Browservinduet er så nede på 480 pixels i bredden.

### Mådehold

Jo flere forskellige knapper, ikoner, bjælker og siderubrikker der er på en side, jo mere uoverskuelig og jo mindre læsevenlig bliver den. Brug derfor kun det nødvendige.

Undgå grafik som ikke har nogen funktion i sig selv. "Bullets" – stjerner, kugler eller anden grafik der bruges ud for punkter på lister, trækker i virkeligheden bare opmærksomheden fra indholdet i listen. Rammer og kanter omkring tabeller og opstillinger trækker opmærksomhed fra deres indhold. Farvestrålende eller blinkende streger mellem tekstafsnit trækker opmærksomheden fra teksten.

### Typografi

En typografi er (i hvert fald i det følgende afsnit) en kombination af skrifttype, -snit, -størrelse, og -farve, som markerer betydningen af et stykke tekst for læseren. Vær nærig med typografierne! Mange forskellige typografier forvirrer øjet. Jeg vil anbefale én typografi til brødteksten, og en eller to andre til overskrifter. Jo mindre tekst der er, jo mindre krav stiller det til læseligheden. Brødtekst bør være så læselig som overhovedet muligt, mens forsides titlens typografi først og fremmest skal være flot.

### Størrelse

Undgå at ændre brødtekstens størrelse. Brug ikke for store overskrifter, de forstyrrer læsningen. Overskrifter behøver ikke at være større end brødteksten. Vælg i stedet at fremhæve overskriften med fed tekst, eller en anden skrifttype i samme størrelse, det er ofte mere læsevenligt.

### Skriftsnit

STORE BOGSTAVER og KAPITÆLER er ikke så læsevenlige, fordi alle bogstaver har samme højde. Bør kun bruges til titler og korte overskrifter.

*Kursiv* er heller ikke særligt læsevenligt på en computerskærm, fordi de skrå linier er i modstrid

med skærmen pixels. Kursiv bør derfor kun bruges til enkelte ord af gangen, ikke til hele afsnit eller overskrifter. Skal et helt afsnit fremhæves (f.eks. fordi det er et citat) kan det indrykkes.

**Fed** er nemmere at læse end kursiv, men forstyrrer også skriftbilledet mere, så det er mest egnet i tekst man skimmer igennem uden at læse alt (f.eks. til opslagsord i en ordliste).

Det kan normalt ikke anbefales at bruge både **fed** og *kursiv* i samme brødtekst, da det giver et meget rodet skriftbillede.

Understregning bør aldrig bruges som fremhævning, da det nemt forveksles med links.

## Skrifttype

Brug ikke mere end to skrifttyper på samme side, en til overskrifter og en til brødteksten. Flere skrifttyper giver et rodet og forvirret udseende. Visse skrifttyper er specielt optimeret til at være skærmskrifter, for eksempel Times, Arial/Helvetica og Courier. Andre skrifter vil sandsynligvis ikke være så læsevenlige på skærmen, da de fleste skrifter først og fremmest er beregnet til at printes på papir.

## Farver

Sort tekst er det mest læsevenlige. Hvid baggrund giver størst kontrast til teksten, men den rene hvide baggrund er lidt for lysende på mange skærme, og derfor anstrengende for øjnene. En diskret lys farve er mere behageligt.

Farver kan bruges til at fremhæve typografier, for eksempel giver en anden farve overskrifter mere liv og kontrast til brødteksten. Som ved skrifttyper vil jeg højest anbefale to forskellige skriftfarver: en til overskrifter og en til brødtekst.

## Lodret afstand

Browsere viser som standard en tom linie mellem hvert afsnit. Dette er absolut ikke særligt læsevenligt. Luft mellem tekstblokke skal bruges til at gøre sammenhænge og inddelinger i teksten tydeligere. Derfor skal der helst ikke være lodret afstand mellem overskrifter og efterfølgende brødtekst, eller mellem afsnit i brødteksten. Lodret afstand bør i stedet bruges over overskrifter eller til større ophold i teksten.

## Bjælker og vignetter

En vignette eller et symbol kan bruges til at markere et større ophold i teksten i situationer, hvor det ikke vil være passende med en overskrift – for eksempel

til at indikere en pause, et brud eller et skift i tonefald i en novelle eller et essay, hvor man ikke bruger overskrifter.

HTML har et indbygger symbol til det formål: den vandrette bjælke. Vandrette bjælker er desværre ikke særligt pæne (heller ikke selvom de er regnbuefarvede, blinker eller lignende, som man ser det på mange websites). I de fleste tilfælde er en stjerne eller en simpel krusedulle langt bedre.

## Marginer og linielængde

Jo mere luft der er omkring teksten, jo nemmere er den at læse. Den optimale linielængde ligger ifølge eksperter på 40-60 tegn. Hvor meget dette er i pixels er svært at sige; det afhænger af tekstens skrifttype og størrelse. Ved 12 punkt Times New Roman (Som er [Netscape Navigators](#) normale indstilling) får man 50-60 tegn ved en spaltebredde på 400 pixels. Man får lidt flere tegn på en Macintosh, hvor skærmskrifterne er lidt mindre.

Pas på med at centrere overskrifter eller billeder – det opbryder den faste venstrekant, som hjælper læsningen.

## Farver på links

Browsere viser normalt links med blå, og links der peger til steder man har besøgt, som lilla. Eksperter i interface-design påpegede tidligt at det var dumt: Links skal naturligvis træde frem, og bør derfor have den mest iøjnefaldende farve, for eksempel rød. Links man har brugt, bør derimod træde i baggrunden, og have en mere kølig farve, for eksempel blå.

Dette synes mange webdesignere lød fornuftigt, og brugte disse farver i stedet. Herefter var det helt umuligt at gennemskue hvad farverne stod for, fordi det svingede fra sted til sted.

Mit råd angående farver på links er derfor: Det bedste er ikke at ændre dem fra "default". Hvis du alligevel ændrer dem, så gør det, så der ikke er tvivl om, hvad der er hvad. Giv links en klar, ren farve, for eksempel rød eller blå eller grøn, og giv "brugte" links en mindre klar, mørkere eller mere grålig version af farven. Hvis du har mange links i selve brødteksten, skal link farven dog ikke være alt for iøjnefaldende, det gør læsningen sværere.

## Forside

Designet af forsiden er naturligvis yderst vigtig. Indenfor det felt, der er synligt på selv de mindste skærme, uden at man scroller, skal fremgå:

- Hvad. En klar angivelse af formålet med denne site – hvad den indeholder og hvad den kan bruges til.
- Hvem. Hvem der har ansvaret for siden. De fleste webservere udvikler hurtigt en sund kildeskritisk sans, og er ikke tryk ved websites, hvor det er uklart hvem der står bag.

På selve forsiden skal desuden gerne være en e-mail link til webmasteren, så besøgende kan anmelde eventuelle fejl.

Hvis organisationen har en adresse og telefonnummer i ikke-cyber verdenen, skal den være på forsiden, eller der skal være en tydelig link til den fra forsiden.

### Litteratur

Nogle gode websites om webdesign ud fra en designmæssig synsvinkel:

#### Web Pages That Suck

“Learn Good Web Page Design by Looking at Bad Web Pages”

[www.webpagesthatsuck.com](http://www.webpagesthatsuck.com)

#### Yale Web Style Guide

Grundig, omfattende og seriøs gennemgang af alle aspekter af webdesign

[info.med.yale.edu/caim/manual/](http://info.med.yale.edu/caim/manual/)

#### Web Page Design for Designers

fokus på layout og typografi

[ds.dial.pipex.com/pixelp/wpdesign](http://ds.dial.pipex.com/pixelp/wpdesign)

#### Guide to Web Style

Suns vejledning i webdesign, fokus på funktionalitet og navigation. [www.sun.com/styleguide/](http://www.sun.com/styleguide/)

## Afprøvning

Sæt en person, som ikke har haft noget med arbejdet med websiten at gøre, foran din computer, og lad hende afprøve din website. Bed hende om at sige alt hvad der falder hende ind – hvad hendes indtryk er, om hun kan finde rundt etc. – men lad være med at svare på noget af det. Notér i stedet reaktionerne ned. Bagefter kan du udspørge forsøgspersonen om siden fungerede efter hensigten, men tit er det de umiddelbare reaktioner, der fortæller mest om sitens svagheder: “Hov, hvor kom jeg nu hen?”, “Hvordan kommer jeg tilbage til...”, “Under hvilken kategori ligger mon...”.

## Vedligeholdelse af en website

En person skal udpeges til webmaster. Webmasterens opgave er at foretage de nødvendige løbende rettelser og opdateringer på websiten. Webmasteren behøver ikke at være den samme som webdesigneren. Mange virksomheder får deres website designet af et eksternt bureau, mens webmasteren er en internt i virksomheden. Man kan sige at webdesigneren er arkitekten og webmasteren er pedellen.

Webmasteren skal have en kopi af websiten på sin egen PC. Her foretages rettelser og ændringer, og de afprøves inden ændringerne flyttes til serveren, (også selvom webserveren ligger på lokalnettet). Det er vigtigt at visse rutiner bliver indarbejdet i organisationen:

- Websiten skal ind i virksomhedens "informationsstrøm" - det skal afgøres hvilke informationer der løbende skal ud på nettet - pressemeddelelser, kalender, begivenheder, produkter mv. Alle, der leverer materiale, skal have at vide hvilke formater (RTF eller HTML, TIFF eller GIF etc. mv.) webmasteren foretrækker materialet i.
- Alle, der laver sider til websiten, leverer dem til webmasteren – ingen uploader direkte.
- Webmaterialet skal gennemgå samme korrekturproces som trykt materiale. Af en eller anden grund er slåfejl meget tydeligere i en browser end i et tekstbehandlingsprogram.
- E-mail sendt til websiten skal læses og evt. besvares. Brugere opdager måske fejl du ikke selv har set, og deres “dumme” spørgsmål kan måske afsløre, at ens website ikke er så letforståelig og logisk opbygget, som man selv synes.
- Det er også meget vigtigt, at der ikke er forældet information på websiten. Sørg for at holde al information ajour og slet sider, der er forældede.

### Hvor mange har besøgt siden?

Man finder ud af hvor mange der har besøgt ens site, ved at analysere serverens log. Det er ikke noget man umiddelbart har adgang til, hvis man ikke selv ejer serveren, så tal med den systemansvarlige. (Man kan ikke stole på de såkaldte counters – tællere – man ser på mange websider).

## Webdesign for intranet

Et *intranet* er et netværk internt i en virksomhed, som benytter de samme komponenter som på internettet: browsere, HTTP, HTML, e-mail. Tidligere brugte man mange forskellige mere eller mindre inkompatible netværkstyper i virksomhederne, men nu er man ved at finde ud af, at det er både nemmere og billigere at bruge internetstandarderne.

Webdesign for et intranet adskiller sig ikke voldsomt fra webdesign til internettet. Tre vigtige forskelle er:

1. Normalt vil man have både et web udadtil, som alle har adgang til, og et web indadtil, som kun ansatte har adgang til. Disse kan være mere eller mindre sammenvævede.
2. Man kan have et reelt overblik over – og kontrol over – klienternes software: operativsystemer, browserversioner mv. Det gør det muligt at basere funktionalitet på Plug-ins, JavaScript, ActiveX og lignende, fordi man kan sikre sig at brugere har det rette udstyr.
3. Båndbredden på et internt netværk kan være mange gange større end på internettet. Det giver større muligheder for multimedia og andre spændende, men tunge effekter.

## Offentliggør din website

### Hvordan finder andre din website?

For det første kan du jo fortælle dem adressen, skrive den på visitkort etc. Husk at skrive den kortest mulige URL – ingen grund til at skrive <http://www.ozymandias.dk/index.html> hvis du kan nøjes med [www.ozymandias.dk](http://www.ozymandias.dk)

### Hvordan får man sin website med i Yahoo, Jubii, AltaVista etc.?

Man tilmelder sin website. De fleste danskere bruger Jubii til at finde danske websites, så den er specielt vigtig. Gå ind på Jubii's forside [www.jubii.dk](http://www.jubii.dk). Find linken med teksten "Tilføj Link". Her kan du tilmelde din site.

Og til dit internationale publikum: Der findes websites, som kan tilføje dine sider til mange internationale indekser og kataloger som Yahoo og AltaVista på én gang. Prøv for eksempel [www.liquidimaging.com/liqimg/submit](http://www.liquidimaging.com/liqimg/submit)

Husk at lave **meta**-tags, så dine sider bliver indekseret, som du vil have det.

Husk oplysende titler, som kan forstås ude af sammenhæng.

Den bedste måde at blive opdaget på er dog at få andre sites til at linke til dig. Send e-mails til beslægtede sites og meld din ankomst på webbet (foreslå, at hvis de linker til dig, linker du til dem). Hvis din site er interessant, vil de sikkert gøre det (gælder dog nok ikke ved kommercielle sites). Der findes også nyhedsgrupper, hvor man kan annoncere nye, relevante sites. Check dog nyhedsgruppens FAQ først, ikke alle steder accepterer man annoncering af websites (især ikke hvis de er kommercielle). Du kan læse mere om nyhedsgrupper i *Start med Internettet*, afsnittet *Nyhedsgrupperne på USENET*.

## Ophavsret på nettet?

Der er mange misforståelser omkring ophavsret på internettet. Kort fortalt er reglerne de samme som andre steder. Det vil sige:

- Ophavsmanden har automatisk ophavsret (*copyright*) på al originalt 'intellektuelt materiale', dvs. original tekst, grafik, musik, programkode etc. Det er *ikke* nødvendigt specifikt at skrive "©" eller "Copyright by..." i materialet, eller at registrere materialet nogen steder, for at have copyright.
- det er ulovligt at kopiere og anvende materiale, som en anden har copyright på, med mindre man eksplicit har fået tilladelse. Man må dog citere o.lign. "inden for rimeligheden grænser". (Spørg bare Thøger Seidenfaden).
- Copyright bortfalder automatisk efter en periode, f.eks. ved litteratur 70 år efter ophavsmandens død, hvorefter materialet kan bruges helt frit.

Webbet skaber også nye spørgsmål omkring copyright, som ikke er helt afklarede. Reglerne ser ud til at være således:

- Man behøver ikke tilladelse til at linke til andres materiale.
- Det er ulovligt at "indkapsle" materiale der ligger på andre servere, så det ser ud som om der er en del af ens egne sider – f.eks. ved at indsætte et billede, hvor kildefilen ligger på en anden server, eller "frame" et dokument på en anden server, så det ser ud som om det er en del af ens eget materiale.

Læs mere om copyrightregler på: **10 Big Myths about copyright explained;**

[www.clarinet.com/brad/copymyths.html](http://www.clarinet.com/brad/copymyths.html)

## Webbets fremtid

HTML er naturligvis i stadig udvikling. For tiden er W3C ved at færdiggøre specifikationerne på HTML 4.0, som dels indeholder frames, scripting og andet der allerede har vundet indpas, dels gør HTML i stand til at håndtere mange skriftsprog, herunder skrifter som kinesisk og japansk, mulighed for skiftende skriftretning mv. De foreløbige specifikationer kan ses på

<http://www.w3.org/TR/WD-html40/>.

Der sker meget andet spændende, f.eks. *Push*, *PNG*, *XML*, *Desktop integration*, *Dynamic HTML*, *Document Object Model*, *CORBA/IIOP*. Læs mere om det i hæftets online supplement.

## Don't try this at home

Naturligvis bliver man ikke professionel bare af at læse et hæfte, men her er nogle ting man kan gøre, hvis man i det mindste ikke vil give indtryk af at være en amatør:

- Brug ikke baggrundsgrafik.
- Lad være med at bytte rundt på farverne på links og besøgte links. Hvis du ændrer farverne, så gør det sådan, at det stadig er tydeligt, hvad der er hvad.
- Lad være med at bruge unødvendige og irriterende specialeffekter som animerede GIF's eller JavaScript-lysaviser.
- Brug ikke for mange forskellige typografier på samme side. To, højst tre, forskellige kombinationer af skrifttype, størrelse og farve. Vær konsekvent med typografien
- Lav ikke indholdstomme sider, f.eks. sider der kun indeholder et logo eller en titel, eller kun to-tre links til undersider.

- Brug ikke tabeller med tredimensionale rammer. Brug ikke den tredimensionale vandrette streg. Det får siden til at se billig ud.
- Referer ikke til mediet. Skriv ikke "denne website...". Lav ikke links med teksten "...klik her..." eller "...denne link fører til...". Skriv blot linkens indhold. Henvi ikke til links: "nedenfor er en link til...".
- Lav ikke sider der kun virker på en af browserne. Det er ikke svært at lave sider som virker i alle browsere, og man får alligevel ikke nogen til at downloade en ny browser, for at se ens side. Skriv ikke "Denne side ses bedst med x browser".
- Lad være med at sætte reklamer for browsere på din hjemmeside, det virker billigt. (Når andre gør det, er det ofte fordi de bliver bestukket. Microsoft giver for eksempel gratis software, pizzaer mv. til webdesignere, der sætter MSIE-logoet på de websites, de laver).
- Skriv ikke "Denne side under konstruktion", eller "Her kommer snart mere indhold". Websider er altid under konstruktion. Lad være med at uploade sider der ikke er færdige.
- Brug ikke plug-ins, med mindre der er en meget god grund til det.
- Lav ikke brødtekst i en anden størrelse end det normale. Lav ikke brødtekst i farver.
- Brug ikke udtryk som "cyber-", "cyberspace" eller "informationsmotorvejen".
- Fjern forældet materiale. Opdater eller fjern siderne, når de er uaktuelle.
- Brug et pænt sprog. Pas især på med "anglicisering" af sproget. Skriv på dansk, hvis dit publikum er dansk.

## Diverse links

**The Webmaster Reference Library** alt om alt: <http://www.webreference.com/>

**WWW på Yahoo** den mest komplette oversigt over alt webrelateret:

[www.yahoo.com/Computers\\_and\\_Internet/Internet/World\\_Wide\\_Web/](http://www.yahoo.com/Computers_and_Internet/Internet/World_Wide_Web/)

**Browserwatch** dagsaktuelle nyheder om browsere og andet:

[www.browserwatch.com/news/news-current.html](http://www.browserwatch.com/news/news-current.html)

**Tucows** en stor oversigt over internetsoftware incl. anmeldelser [www.tucows.dk/](http://www.tucows.dk/)

**PC Webopædia** Leksikon over computer- og internetudtryk, med links til relevante sites.

[www.pcwebopædia.com/Internet\\_Category.html](http://www.pcwebopædia.com/Internet_Category.html)

## Ordforklaringer

**ActiveX** Microsofts svar på Java og Plug-ins mm., Windows-centreret.

**Alta Vista** en af de stærkeste søgemaskiner;  
[www.altavista.digital.com](http://www.altavista.digital.com)

**anker** et punkt i et HTML-dokument, som er markeret, så man kan lave en link til det.

**animation** en sekvens af billeder over tid; f.eks. en tegnefilm.

**anti-alias** et grafisk trick der gør hakkede kanter glattere ved at bruge farvetoner.

**applet** et Java-program indsat på en webside.

**ASCII** et tegnsæt der indeholder de mest almindelige tegn, bl.a. det engelske alfabet.

**attribut** en udvidelse i en tag.

**browser** klienten som fremviser websider, og gør det muligt at navigere dem.

**browserkrigen** malerisk udtryk for konkurrencen mellem de to førende browserproducenter Netscape (med **Navigator**) og Microsoft (med **MSIE**).

**båndbredde** en forbindelses evne til at overføre data. Jo større båndbredde en forbindelse har, jo hurtigere overføres data.

**cache** i en browser: de nyligt indlæste filer som stadig ligger i hukommelsen.

**cyber-** se Cyberspace

**Cyberspace** (fra science-fiction romanen *Neuromancer* (1984) af *William Gibson*, hvor udtrykket betegner et virtual-reality computernetværk der nærmest er en parallelverden. Forstavelen *cyber-* er taget fra *cybernetics* (Kybernetik). Udtrykket *C.* og forstavelen *cyber-* bruges i dag i flæng om alt vedr. computere, internet, informationsteknologi eller andet sejt.

**cookie** en stump information en server kan gemme i en browser, så den kan genkende browseren igen, næste gang henvender sig.

**CGI** *Common Gateway Interface* et system så man gennem websider kan kalde et program på en server.

**CSS** *Cascading Style Sheets* et typografisk sprog som bruges i style sheets

**dither** simulation af en farve i et billede ved at blande prikker af andre farver.

**domænenavn** del af navnet på en server.

**download** at hente en fil over internettet for at gemme den lokalt på ens computer.

**editor** et program man bruger til at redigere en fil i, f.eks. en webeditor til websider, eller en teksteditor til tekstfiler.

**FAQ** *Frequently Asked Questions*: liste over alm. spørgsmål (og svar) indenfor et givet område.

**flydende billeder** billeder på en webside, der højre- eller venstre-stilles så teksten flyder udenom.

**formular** en samling felter til at fylde information i, som kan sendes til et CGI-program.

**frame** en af delene af et frameset.

**frameset** et dokument der opdeler browservinduet i flere separate dokumenter.

**freeware** gratis software.

**FTP** *File Transfer Protocol* en protokol som bruges til at overføre filer og udføre filhåndtering (oprette mapper, slette filer, ændre filnavne mv.) over internettet.

**FTP-program** til at udføre upload og filhåndtering over internet.

**GIF** et grafikformat brugt på nettet, velegnet til simpel grafik.

**GIF-animation** en variation af GIF-formatet der kan indeholde animationer.

**header** metainformation om et dokument, f.eks. MIME-type, sendt før selve dokumentet i en HTTP-overførsel.

**HTTP** *HyperText Transport Protocol*. Protokollen som bruges til at hente websider fra en server.

**HTML** *HyperText Markup Language*. Det sprog websider laves i. Et HTML-dokument kan indeholde tekst, billeder og links til andre objekter på nettet.

**homepage** se hjemmeside

**hjemmeside** 1. en websites forside 2. den side ens browserprogram starter ved. 3. *vulg.* en webside.

**hyperlink** en aktiv henvisning i hypertext eller hypermedia.

**hypermedia** som hypertext, bortset fra det også kan indeholde billeder, lyd etc.

**hypertext** tekst, der indeholder henvisninger på kryds og tværs i teksten, henvisninger man umiddelbart kan følge, og derved læse teksten på en ikke-liniær måde.

**ISO-8859-1** et tegnsæt som er en udvidelse af ASCII med alle specialtegn fra de vesteuropæiske sprog, herunder *æ, ø, å*.

**informationsmotorvejen** (fra eng. *the information superhighway*) (*vulg.*) den informationsteknologiske infrastruktur, herunder internettet.

**interaktivitet** når et program reagerer på brugerens input. Den simpleste form for interaktivitet på nettet er links. Mere avancerede former kan laves med Java eller JavaScript.

**imagemap** et billede hvor forskellige områder er links til forskellige steder.

**ISDN** en speciel digital telefonlinie med lidt større båndbredde end et almindeligt modem. Populær i erhvervslivet.

**internetudbydere** et firma der tilbyder servicer som internetadgang, servere, webhotel el.lign.

**intranet** et internet netværk i en virksomhed som er kompatibel med og benytter de samme komponenter som internettet, f.eks. e-mail, HTTP, websider i HTML, Java, mv.

**Java** et programmeringssprog specielt udviklet til at køre over netværk. Udmærker sig bl.a. ved at være platformsuafhængigt, dvs. det samme program kan uden ændringer køre på mange forskellige typer computere. Udviklet af Sun Microsystems

**JavaScript** et letvægtssprog som integreret med HTML kan gøre websider mere interaktive. Udviklet af Netscape.

**JPEG** et grafikformat brugt på webbet, velegnet til fotografier.

**JScript** Microsofts udgave af JavaScript.

**klient** et program i brugerens ende af et netværk, f.eks. en webbrowser eller et mailprogram. Kommunicerer med servere.

**komprimering** metode til at få en datafil til fylde mindre.

**link** hyperlink.

**Microsoft** verdens førende softwarefirma, producerer blandt andet Windows-operativsystemerne, kontorprogrammer som **Word** og **Excel** og internetprogrammer som browseren **Internet Explorer (MSIE)** og webeditoren **FrontPage**

**MIME** *Multipurpose Internet Mail Extension* standard hvormed servere på internettet angiver formatet på filer de sender over nettet.

**MSIE** *Microsoft Internet Explorer* den næstmest populære browser.

**multimedia** et lidt diffust udtryk, der normalt betegner kombinationer af interaktive ikke-tekstbaserede medier som lyd eller levende billeder over computer. Bruges ofte om CD-ROM'er.

**Netscape** softwarefirma; har længe været det førende internetsoftwarefirma med både browsere (**Netscape Navigator**) og servere. Har været meget betydningsfuldt i webnets udvikling.

**nyhedsgrupper** diskussionsgrupper på internettet.

**objekt 1.** i HTML: en ekstern fil som indsættes som et felt på en webside, f.eks. et billede, en applet, en plug-in. 2. indenfor programmering: en samling af funktioner og data; et element som har egenskaber og funktioner; et programmodul, der kan kommunikere med andre programmoduler.

**objektorienteret** indenfor programmering: baseret på objekter.

**palette** de farver som indgår i et billede.

**PDF** *Portable Document Format* et format beregnet til overførsel af layoutede dokumenter beregnede til udprintning.

**pixel** et billedpunkt på en skærm.

**platform** kombination af hardware og operativsystem.

**platformsuafhængig** software eller data, der kan forstås af computere på tværs af forskellige hardware- og operativsystemer.

**plug-in** et udvidelsesmodul til browseren, som gør den bliver i stand til at vise/afspille et nyt dataformat, f.eks. film, lyd, VRML mv.

**protokol** nogle regler for en kommunikation over netværk. Der findes forskellige protokoller til forskellige formål, f.eks. HTTP for overførsel af websider, og SMTP for overførsel af post.

**relativ URL** en adresse som fortolkes i forhold til adressen på dokumentet, hvori den optræder.

**RFC** *Request For Comments* dokument, som beskriver standarderne der bruges på internettet. Navnet angiver at r. aldrig er endegyldige eller udenfor diskussion.

**RTF** *Rich Text Format* et platformsuafhængigt filformat beregnet for udveksling af tekstbehandlings-dokumenter mellem forskellige programmer.

**server** et program eller en computer, der leverer en service til klienter over netværk. En webserver leverer for eksempel websider og andre dokumenter til en bruger med en webbrowser.

**SGML** *Standardized General Markup Language* et "metasprog" til udvikling af dataformater. HTML er udviklet af SGML.

**shareware** programmer, som kan prøves gratis i en periode (ofte 30 dage) før man beslutter om man vil købe dem

**site** kort for website

**style sheets** en mekanisme for at tilføje avancerede typografi- og layoutbeskrivelser til HTML

**tabel** en ramme, som kan bruges til opstillinger og layout.

**tag** en af de markeringer eller formateringskoder der bruges i HTML.

**teksteditor** en editor til at redigere ren ASCII eller ISO-8859-1 tekst, f.eks. **Notesblok** i Windows eller **Tekst & Lyd** på Mac.

**upload** at flytte en fil fra ens computer ud til en anden computer på internettet.

**URL** *Uniform Resource Locator* en adresse på et objekt på internettet f.eks. en webside.

**VRML** *Virtual Reality Markup Language* et standardformat for beskrivelse af 3D-modeller eller -verdner. Styres af W3C.

**W3C** fork. for WWW-Consortiet

**web** en sammenhængende struktur af hypertext- eller hypermediadokumenter

**webbet** fork. for World Wide Web

**webdesign** den ædle kunst at lave websider og websites.

**webeditor** et program til fremstilling af websider.

**webhotel** lejet plads på en internetudbyders server, hvor en website kan ligge.

**website** dokument i HTML, som kan hentes over internettet via HTTP.

**website** websted; en sammenhængende gruppe af websider på internettet.

**World Wide Web** et enormt web af hypermediadokumenter på internettet, samt de filformater (HTML m.fl.), protokoller (HTTP m.fl.), og software (servere, browsere) der gør det muligt.

**World Wide Web Consortiet** en ikke-kommerciel organisation som udstikker retningslinier og anbefalinger for standarder der indgår i webbet, f.eks. HTML og HTTP.

**WWW** fork. for World Wide Web

**WYSIWYG** *What You See Is What You Get* bruges bl.a. om webeditorer der giver et nogenlunde realistisk billede af, hvordan websiden vil se ud for brugeren (modsat ældre type editorer, hvor man kun så formateringskoderne).

**Yahoo** det uundværlige altomfattende katalog over websites; [www.yahoo.com](http://www.yahoo.com)

**åben** om dataformater, standarder, protokoller mv.: Lidt diffus betegnelse, men betyder normalt "ikke kontrolleret af en kommerciel virksomhed, specifikationerne tilgængelige for alle, kan benyttes af alle".

- 216-farvers paletten, 25
- absolute placement, 43
- ActiveX, 47
- Adobe PageMill, 7
- afprøvning, 15; 20; 22; 58
- alternativ tekst, 16
- anker, 14; 27
- anti-alias, 23
- applets, 46
- arbejdsplan, 50
- attributter, 39
- baggrundsgrafik, 18; 31
- betydningsmæssige markeringer, 11
- browserkrigen, 33
- celle, 19
- CGI, 48
- character entity references, 40
- Claris Home Page, 7
- client-side imagemaps, 17
- Communicator, 6
- Composer, 6
- copyright, 59
- CSS, 42
- definitionsliste, 9
- Den Sikre Webpalette, 25
- det synlige felt, 56
- domænenavn, 44
- egenskaber, 12
- eksterne links, 14
- farver, 13; 57
- filnavne, 8
- flydende billeder, 16
- formularer, 48
- formål, 50
- forside, 57
- frames, 27
- frameset, 27
- FrontPage, 7
- FTP-program, 20
- GIF, 23
- GIF-animationer, 45
- grafik
  - formater, 23
  - indsætte, 16
- HTML 3.0, 33
- HTML 3.2, 34; 35
- HTML 4.0, 59
- HTML-editor, 7
- hypertekst, 54
- HyperText Markup Language, 5
- ikoner, 54
- imagemap, 17
- index.html, 26
- indrykning, 9
- Inspector, 12
- interlaced, 25
- intranet, 58
- ISO-8859-1, 11
- Java, 46
- JavaScript, 47
- JPEG, 23
- JScript, 47
- kapitæler, 30
- layers, 43
- linieafstand, 30
- linielængde, 31; 57
- lineskift, 11
- links, 13; 57
- lister, 9
- mailto, 15
- margin break, 17
- marginer, 30; 57
- mellemrum, 11
- metafor, 51
- meta-information, 41
- Microsoft FrontPage, 7; 8
- Microsoft Internet Explorer, 6
- Mosaic, 33
- MSIE, 6
- målgruppe, 51
- navigation, 53
- Navigator, 6
- Navigator Gold, 6
- Netscape, 6
- Netscape Communications, 33
- Netscape Communicator, 6
- Netscape Composer, 6; 7
- Netscape Navigator, 6
- nonbreaking space, 11
- Object Editor, 12
- objekter, 45
- offentliggørelse, 59
- online-supplement, 3
- ophavsret, 59
- overskrifter, 9
- Pagemaker, 32
- PDF, 37
- plug-ins, 46
- platformsuafhængig, 5
- preview mode, 15
- Progressive JPEG, 24
- properties, 12
- push, 60
- QuarkXPress, 32
- relative URL'er, 26
- RTF, 37
- server-side imagemaps, 17
- skriftsnit, 56
- skrifttyper, 30
- skærmopløsninger, 56
- spalter, 31
- specialtegn, 40
- style sheets, 42
- tabeller, 18
- tabulator, 11
- tags, 38
- tekstbehandling, 32
- tekstjustering, 9
- titel, 13
- transparens, 23
- teletype, 10
- typografi, 56
- Udgivelse, 20
- upload, 20
- URL, 26
- vedligeholdelse, 58
- W3C, 5; 33
- webeditor, 7
- Webhotel, 20
- webmappe, 8
- Wilbur, 40
- Word, 32
- Wordperfect, 32
- World Wide Web Consortiet, 5
- webdesign, 3
- WYSIWYG, 7
- åben standard, 5

# Web Design

<b>Del 1 – Webdesign for begyndere .....</b>	<b>4</b>	<b>DEL 3: Interaktivitet og multimedia .....</b>	<b>45</b>
HTML .....	5	Objekter .....	45
Om browserne .....	6	GIF-animationer .....	45
Editoren .....	7	Plug-ins .....	46
I gang .....	8	Java-applets .....	46
Tekst .....	9	ActiveX-kontroller .....	47
Elementer og egenskaber .....	12	JavaScript .....	47
Farver og baggrund .....	13	CGI & formularer .....	48
Titel .....	13	<b>Del 4: Planlægning &amp; design af en website... 50</b>	
Links .....	13	Den overordnede metafor .....	51
Afprøvning i browser .....	15	Navigation og brugerflade .....	53
Billeder .....	16	Tekstens redigering .....	54
Tabeller .....	18	Sidelayout .....	55
Afprøvning i browser .....	20	Typografi .....	56
Udgivelse .....	20	Forside .....	57
<b>Del 2 – Alt om webdesign.....</b>	<b>23</b>	Afprøvning .....	58
Billedbehandling .....	23	Vedligeholdelse af en website .....	58
URL .....	26	Webdesign for intranet .....	59
Frames .....	27	Offentliggør din website .....	59
Misbrug HTML til at lave flot typografi og layout 30		Ophavsret på nettet? .....	59
Fra tekstbehandling til HTML .....	32	Webbets fremtid .....	60
HTML, Browserkrigen og Webbets historie .....	33	Don't try this at home .....	60
Hvad understøtter hvilke browsere? .....	35	Diverse links .....	60
Andre dokumentformater .....	37	Ordforklaringer .....	61
Ind bagved, til HTML koderne .....	38		
Indsætte HTML i editoren .....	40		
Meta-information om dokumentet .....	41		
Style sheets .....	42		
Gør siden hurtigere! .....	44		
Sikkerhed .....	44		
www.migselv.dk .....	44		

**KnowWare**

ISBN 87-90027-59-0

