

Introduktion til HTML

```
<html>  
  
<head>  
<title> Introduktion til HTML</title>  
</head>  
  
<body>  
<H1>Dette er ægte hjemmesidedesign</H1>  
</body>  
  
</html>
```



Hvad er HTML

Eksempler og forklaringer

Indhold

| | |
|--|----|
| Introduktion til HTML (Kilde: HTML.dk)..... | 4 |
| Lektion 1: Det praktiske..... | 5 |
| Hvad skal der da bruges? | 5 |
| Skal jeg være online? | 6 |
| Hvad så nu?..... | 6 |
| Lektion 2: HTML..... | 7 |
| Hvem eller hvad er HTML så? | 7 |
| Hvad kan jeg så bruge HTML til? | 7 |
| Okay, men hvad betyder HTML?..... | 8 |
| Lektion 3: Tags..... | 9 |
| "Tags"?..... | 9 |
| Hvad med et par eksempler? | 9 |
| Fint nok, men hvor skriver jeg så mine tags? | 10 |
| Lektion 4: Dit første website..... | 11 |
| Lektion 5: Hvad du har lært | 15 |
| Lektion 6: Lidt flere tags..... | 17 |
| Hvad så nu?..... | 17 |
| Kan jeg bruge to typer tags samtidig? | 18 |
| Hvad med de sidste 6 tags? | 18 |
| Lektion 7: Attributter..... | 21 |
| Hvad skal jeg bruge det til?..... | 21 |
| Og hvordan var det lige siden blev rød? | 21 |
| Lektion 8: Links | 23 |
| Og hvad med links internt på en side | 24 |
| Kan man også linke til en e-mail adresse? | 25 |
| Er der flere attributter, jeg bør kende? | 25 |
| Lektion 9: Indsættelse af billeder | 27 |
| Hvor får jeg nogle andre billeder fra? | 28 |
| Er der mere? | 28 |
| Er der flere attributter, jeg bør kende? | 29 |
| Lektion 10: Tabeller | 32 |
| Er det svært?..... | 32 |



| | |
|---|----|
| Er der ingen attributer? | 33 |
| Flere attributter? | 36 |
| Hvad kan jeg sætte ind i mine tabeller? | 37 |
| Lektion 11: Flere tabeller | 38 |
| Hvad er der da tilbage? | 38 |

Introduktion til HTML (Kilde: HTML.dk)

Mange tror fejlagtigt, at det er umådeligt svært og derfor forbeholdt de få at lave sit eget website / hjemmeside. Sådan forholder det sig imidlertid ikke. Alle kan lære at lave et website! Hvis du vælger at læse videre, har du lavet et om 1 time.

Andre tror - ligeså fejlagtigt - at man laver websites med avancerede programmer. Der findes ganske rigtigt programmer, der påstår, at de kan lave dine sider for dig, og nogle kommer tættere på end andre, men hvis du vil have det gjort på den rigtige måde, må du gøre det selv. Heldigvis er det ret nemt, og du har allerede de programmer, der skal bruges.

Målet med denne tutorial er at give dig en let, men alligevel grundig og korrekt indføring i det at lave websites. Tutorialen starter helt fra bunden og kræver absolut ingen forhåndskendskab til programmering.

Tutorialen kan dog ikke vise dig alt, og der er derfor lagt op til, at du efter hvert modul selv prøver at "lege" lidt med de ting, du lærer, og et vist engagement og gåpåmod er derfor påkrævet.

God fornøjelse.

Lektion 1: Det praktiske

I denne første lektion får du en kort præsentation af de værktøjer, du skal bruge for at lave dine egne websites.

Hvad skal der da bruges?

Højest sandsynligt har du allerede alt det du skal bruge for at komme i gang. Hvis du læser dette, har du f.eks. allerede noget af det vigtigste - en browser. En browser er det pro-gram, du bruger, når du surfer på Internettet og som gør det muligt at se websites.

Det er ikke så vigtigt, hvilken browser du bruger. De mest almindelige er Microsoft Internet Explorer, Safari, Opera og Mozilla Firefox, og fælles for dem alle er, at de kan bruges her. Så du skal altså nok få set dine kommende sider, men først skal de jo lige laves.

I stedet skal du have fat i programmet Notepad eller Notepad++



Hvis du ikke bruger Windows, skal du bruge en tilsvarende simpel teksteditor: f.eks. Simple Text, hvis du er Mac-bruger.

Notepad er blot et simpelt tekstbehandlingsprogram, men genialt til at lære HTML, da det

'holder fingrene' fra det du skriver. Problemet med mange af de programmer der påstår, at de kan lave dine sider for dig, er, at de kan en masse standardfunktioner, men samtidig skal de have alt til at passe ind i disse funktioner. Derfor kan de ofte enten ikke lave siderne, som du vil have dem, eller også laver de ligefrem om i det du allerede har lavet (meget irriterende!). I Notepad har du kun dig selv at takke for både succeser og fejl.

En browser og Notepad er alt du behøver for at gennemgå dette kursus og lave dit eget website. Du kan også vælge at bruge Dreamweaver's split eller code mode.

Skal jeg være online?

Du behøver ikke være online - hverken mens du læser denne tutorial, eller mens du laver dine sider.

Hvis du vil undgå at være online, mens du gennemgår denne tutorial, kan du enten udskrive lektionerne eller blot logge af, mens du læser.

Også når du laver et website (og det gør du snart), kan du være offline. Normalt vil du først lave siderne på din egen computer, og derefter lægge dem ud på Internettet, når de er færdige.

Hvad så nu?

Hop videre til den næste lektion og læs lidt om hvad HTML egentlig er, inden det for alvor går løs i lektion 3.

Lektion 2: HTML

Denne anden lektion har til formål at give dig en kort præsentation af din nye ven - han hedder HTML

Hvem eller hvad er HTML så?

HTML er browserens 'modersmål' og fundamentet for alle sider på Internettet.

For at gøre en lang historie kort var det en forsker ved navn Tim Berners-Lee, der tilbage i det herrrens år 1990 (eller cirka deromkring) opfandt HTML. Målet var at gøre det nemt for forskere på forskellige universiteter at få adgang til hinandens resultater. Projektet lykkedes nok bedre end Tim Berners-Lee havde regnet med, for med HTML skabte han også meget af det Internet, vi kender i dag.

HTML er altså et 'sprog', der gør det muligt at præsentere materiale (f.eks. forskningsresultater) på Internettet. Når du går ind på et websted, er det du ser en 'oversættelse' af HTML - det er din browser, der oversætter HTML. Klik på "View" (Vis) i menuen i din browser og vælg "Source" (Kilde) for at se HTML-koden, der ligger bag denne side.



For almindelige dødelige ser HTML enormt kompliceret og forvirrende ud. Men bare rolig. Når du har læst denne tutorial vil det hele give mening.

Hvad kan jeg så bruge HTML til?

Siden du læser disse sider, er du vel interesseret i at lave din egne websteder, og hvis man vil lave websteder, er der ingen vej udenom HTML. Men frygt ikke for HTML er både let at lære og bruge. Allerede i lektion 4 lærer du at lave din første side.

Så HTML skal bruges til at lave sider på Internettet - så simpelt er det!

Okay, men hvad betyder HTML?

HTML er en forkortelse af HyperText Mark-up Language.

Hyper er det modsatte af lineær. I gamle dage - hvor mus var noget katten legede med - kørte et computerprogram lineært: Dvs. når det havde udført en handling gik det automatisk til den næste linie i rækken og derfra til næste osv. Med HTML er det anderledes. Her kan man gå hvorhen man vil, hvornår man har lyst (det er f.eks. ikke nødvendigt at besøge Politikens websted før du besøger HTML.dk)

Text er bare "tekst".

Mark-up er det man gør med teksten: man markerer den op, på samme måde som man sætter et dokument op i et tekstbehandlingsprogram med overskrifter og f.eks. fed tekst.

Language betyder "sprog". HTML er et næsten ganske almindeligt sprog (engelsk) - HTML er ikke et programmeringssprog men derimod en opmærkning (kodning), hvor du fortæller hvordan browseren skal vise indholdet.

Lektion 3: Tags

Efter disse to indledende lektioner er du nu parat til at lære de første tags at kende.

"Tags"?

"Tags" er det browseren læser og oversætter inden den viser en hjemmeside for dig. Tags er engelsk (udtales også på engelsk) og betyder egentlig etiket eller mærkat. Det er altså en slags mærkater med forskellige kommandoer, du bruger til at fortælle browseren, hvordan du vil have din side til at se ud.

Alle tags har det samme format:

De starter med et mindre-end tegn "<" og slutter med et større-end tegn ">".

Generelt findes der to slags tags: Åbningstags: <kommando> og lukketags: </kommando>. For at lave et lukketag skal du blot tilføje en "/" i åbningstaget. Al information, du putter mellem et åbningstag og lukketag, vil være underlagt den kommando, der står i taget.

Ingen regel uden undtagelse. Der findes også tags, der både åbnes og lukkes i samme tag. Disse tags indeholder kommandoer, der ikke er knyttet til et specielt stykke tekst, men er en enkeltstående kommando - f.eks. et linjeskift.

HTML er ikke andet end tags. At lære HTML er at lære at kende tags og bruge dem.

Hvad med et par eksempler?

Taget fortæller f.eks. browseren at alt der står mellem og skal skrives med fed skrift

Eksempel 1:

Dette skal stå med fed

Vil se sådan ud i browseren:

Dette skal stå med fed

Tagsene <h1>, <h2>, <h3> osv. fortæller browseren at den skal skrive en overskrift (h står for "heading", der er engelsk for overskrift), hvor <h1> er den største, <h2> den næststørste osv.

Eksempel 2:

```
<h1>Dette er en stor overskrift</h1>
```

```
<h2>Dette er en lidt mindre overskrift</h2>
```

Vil se sådan ud i browseren:

Dette er en stor overskrift

Dette er en lidt mindre overskrift

Skal man skrive et tag med stort eller småt?

De fleste browsere er som udgangspunkt ligeglade om du skriver dine tags med stort, småt eller begge dele. <KOMMANDO>, <kommando> eller <KomMaNDO> vil normalt give det samme resultat i browseren. MEN det rigtige er at skrive tags med småt, så det kan du lige så godt gøre det til en vane at skrive dine tags med lille.

Fint nok, men hvor skriver jeg så mine tags?

Dine tags skriver du i et HTML-dokument. Når du surfer rundt på nettet åbner du blot filer i form af HTML-dokumenter.

Forvirret? Så hop videre til den næste lektion, for om ti minutter har du lavet dit første website.

Lektion 4: Dit første website

Ud fra det du har lært i de foregående lektioner, er du nu kun minutter fra at lave dit første HTML-dokument.

Hvordan?

I lektion 1 fik du af vide, hvad du skulle bruge for at lave HTML-dokumenter: en browser og Notepad (eller en lignende teksteditor).

Browseren har du - siden du læser dette - allerede åben, men åben et ekstra vindue (åben browseren en gang til), så du både kan læse videre i tutorialen og se dine egne nye sider.

Åben også Notepad i startmenuen under Programs (Programmer) under Accessories (Tilbehør):



Så er vi klar!

Hvad kan jeg lave?

Jeg tror vi starter med noget enkelt - måske en side der skriver "Dette er min første side".

HTML virker på en meget simpel og logisk facon. Browseren læser HTML, ligesom du selv læser: fra toppen til bunden og fra venstre mod højre. En HTML-side starter derfor med det der skal komme først, og afsluttes med det der skal komme sidst.

Det du allerførst skal gøre, er at fortælle browseren, at du har tænkt dig at skrive HTML. Det gør du med taget `<html>`

Gå derfor ind i Notepad og skriv `<html>` øverst på siden.

Som du måske husker fra den forrige lektion, er `<html>` et åbningstag, og skal derfor lukkes med et `</html>`, når du er færdig med at skrive HTML. Indsæt derfor et `</html>` et par linjer nede, og skriv resten af siden mellem `<html>` og `</html>`.

Dernæst skal din side have et hoved og en krop.

Meget logisk sidder hovedet `<head>` og `</head>` ovenpå kroppen `<body>` og `</body>`.

Nu skulle din side se sådan ud:

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

Hvis den gør det, har du lavet dit første HTML-dokument. En usandsynlig kedelige side og sikkert ikke det du har drømt om, men alligevel et slags websted. For det du har lavet, er grundskabelonen for alle dine fremtidige sider.

Din side har altså to dele: et hoved og en krop.

I 'hovedet' kan du gemme oplysninger om siden, mens 'kroppen' indeholder det, der udfører siden. Hvis du vil oplyse om sidens titel, ville det derfor være naturligt at gøre det i 'hovedet'. Taget for en titel er `<title>`:

```
<title>Noget om html</title>
```

Som lovet skal vi have siden til at skriv "Dette er min første side". Det er altså det, der skal udføres, og hører derfor til i 'kroppen' (body'en):

```
<p>Dette er min første hjemmeside</p>
```

P'et i <p>-taget står for "Paragraph", som oversat betyder "afsnit", og det er jo præcis hvad det er.

Sådan skulle det gerne se ud:

```
<html>

<head>
<title>Noget om html</title>
</head>

<body>
<p>Dette er min første side</p>
</body>

</html>
```

Færdigt arbejde. Du har lavet din første rigtige side!

Nu skal den bare gemmes, så den kan åbnes i din browser. For nemhedens skyld nøjes vi i første omgang med at gemme Notepad siden på din egen computer og ikke på en server:

Vælg "Save As..." ("Gem som...") under "File" i Notepad.

Vælg "All Files" ("Alle Filer") i boksen "Save as type" ("Gem som type") (Dette er meget vigtigt, da du ellers gemmer din side som et tekstdokument og ikke som en HTML-fil).

Gem nu din side som f.eks. "side1.html" (endelsen ".html" fortæller, at det er en HTML-fil, men endelsen ".htm" giver det samme resultat, så du kan selv bestemme, hvilken endelse du vil bruge. ").



Gå nu over i det ekstra browservindue:

Vælg "Open" ("Åben") under "File" ("Fil")

I boksen der kommer til syne klikkes på "Browse" ("Gennemse") Find nu din side og åben den.



Tillykke! Nu skulle der meget gerne stå "Dette er min første side" i din browser.

Hvis du absolut vil have hele verden til at se din første hjemmeside, kan du allerede nu hoppe til lektion 13 og lære at uploade din side på Internettet.

Lektion 5: Hvad du har lært

Den eneste måde at lære HTML, er at prøve sig frem - du kan hverken ødelægge din computer eller Internettet.

Hvad skal det nu betyde?

Ingen bliver dygtig til at lave hjemmesider ved at lave de eksempler, der er givet her i tutorialen. I stedet må du selv ud på det dybe vand og stå på egne ben

Start altid med den grundskabelon du lavede i det forrige modul:

```
<html>

  <head>
  </head>

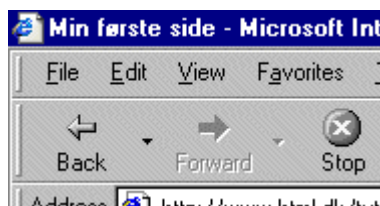
  <body>
  </body>

</html>
```

I princippet er det ligegyldigt, hvordan du sætter din HTML op, men for at bevare overblikket kan det kraftigt anbefales, at du altid sætter siderne pænt og overskueligt med linjeskift og indryk (brug tabulator-tasten), ligesom i ovenstående eksempel.



I head'eren bør du altid skrive en titel: `<title>Titel</title>`. Hvis du ligger mærke til det, bliver titlen på en side vist øverst til venstre i din browser:



Titlen bruges også af søgemaskiner som f.eks. Google, når de indekserer dit website og vises i søgeresultatet.



I body'en formaterer eller koder du sidens egentlige indhold.
Til dette brug kender du allerede en række tags:

`<p>`Bruges ved afsnit. Når du starter et nyt afsnit hopper du automatisk to linjer ned.`</p>`
``fed tekst``
`<h1>`Overskrift`</h1>`
`<h2>`Underoverskrift`</h2>`
`<h3>`Mindre underoverskrift`</h3>`

Hvad er det, du vil have mig til?

Prøv nu selv at sætte et par sider op. Lav f.eks. en side med titel, overskrift, lidt tekst, en underoverskrift og lidt mere tekst. Det er helt okay at kigge på denne side, mens du laver den første side, men prøv senere om ikke du kan lave en "uden at kigge efter".

Lektion 6: Lidt flere tags

Fik du lavet din egen side? Ellers er her et eksempel:

```
<html>

  <head>
    <title>Min side</title>
  </head>

  <body>
    <h1>En overskrift</h1>
    <p>tekst, tekst tekst, tekst</p>
    <h2>Underoverskrift</h2>
    <p>tekst, tekst tekst, tekst</p>
  </body>

</html>
```

Hvad så nu?

Nu er det tid til at lære 8 nye tags.

På samme måde som du laver din tekst fed med ``, kan du sætte din tekst i kursiv med ``. "em" er en forkortelse for "emphasized" ("kursiv").

Eksempel 1:

```
<em>Dette skal stå i kursiv</em>
```

Vil se sådan ud i browseren:

Dette skal stå i kursiv

Med `<small>` ("lille") kan du gøre din tekst mindre:

Eksempel 2:

```
<small>Dette skal være småt</small>
```

Vil se sådan ud i browseren:

Dette skal være småt

Kan jeg bruge to typer tags samtidig?

Du kan sagtens bruge flere typer tags på samme tid, bare du undgår overlappende tags. Dette er bedst illustreret med et eksempel:

Eksempel 3:

Hvis du vil skrive en tekst med fed og i kursiv, skal det gøres sådan:

```
<strong><em>Tekst i fed og kursiv</em></strong>
```

Og IKKE sådan:

```
<strong><em>Tekst i fed og kursiv</strong></em>
```

Forskellen er, at jeg i det første eksempel lukker det tag, jeg har åbnet først, til sidst. På den måde undgår jeg at forvirre både mig selv og browseren.

Hvad med de sidste 6 tags?

Kan du huske, at jeg i lektion 4 fortalte, at der findes tags, der åbnes og lukkes i samme tag. Disse tags indeholder kommandoer, der ikke er knyttet til et specielt stykke tekst, men er en enkeltstående kommando.

Et eksempel på et sådan tag er `
`, som skaber et tvunget lineskift:

Eksempel 4:

```
Lidt tekst <br />og lidt mere tekst på en ny linie
```

Vil altså se sådan ud i browseren: Lidt tekst
og lidt mere tekst på en ny linie

Bemærk at der er indsat et mellemrum og en skråstreg tilside i taget, på den måde åbnes og lukkes det på samme tid.

Et andet tag, der også åbnes og lukkes i samme tag, er `<hr />`, som bruges til at tegne en horisontal streg.

"hr" står for "horizontal rule" = "horizontal streg" (jo, "rule" kan godt betyde "streg"):

Eksempel 5:

```
<hr />
```

Vil altså se sådan ud i browseren:

Et tag, hvor der både skal bruges et åbnings- og et lukningstag (som der oftest skal), er `<blockquote>` ("citattag"). Taget laver et indtryk i din tekst:

Eksempel 6:

```
<p>Pludselig pegede Bilbo:</p>
```

```
<blockquote><p>Der er den gamle drossel igen!</p></blockquote>
```

```
<p>Det var ganske rigtigt den gamle drossel...</p>
```

Vil altså se sådan ud i browseren: Pludselig pegede Bilbo:
Der er den gamle drossel igen!

Det var ganske rigtigt den gamle drossel ...

En anden måde at lave et indtryk på, er ved at bruge ``: "unordered list", hvilket sådan cirka kan oversættes til "liste uden rækkefølge". Eller ``: "ordered list" = "liste med rækkefølge". Til at lave punkter i listen bruges `` ("list item" = "listepunkt"). Forvirret? Se eksemplerne:

Eksempel 7:

```
<ul>
  <li>Her står punkt A</li>
  <li>Her står punkt B</li>
</ul>
```

Vil se sådan ud i browseren:

Her står punkt A
Her står punkt B

Eksempel 8:

```
<ol>
  <li>Her står punkt et</li>
  <li>Her står punkt to</li>
</ol>
```

Vil se sådan ud i browseren:

1. Her står punkt et
2. Her står punkt to

Puha, var det alt?

Det var alt for nu. Prøv nu igen at lave dine egne sider, hvor du bruger nogle af de 8 nye tags:

```
<em>Kursiv</em>
<small>Mindre tekst</small>
<br /> linieskrift
<hr /> Horisontal linie
<blockquote>indrykning</blockquote>
<ul>Liste</ul>
<ol>Ordnet Liste</ol>
<li>Listepunkt</li>
```

Lektion 7: Attributter

Udover at et tag i sig selv er en kommando, kan du også indsætte ekstra information i en række tags.

Undskyld, kan jeg lige få den én gang til?

Et tag er, som du nok husker, en kommando til browseren (f.eks. er `
` en kommando om at lave et lineskift). I visse tags kan du præcisere kommandoen ved at knytte mere information til kommandoen inde i selve taget. Sådanne informationer kaldes attributter.

Hvad skal jeg bruge det til?

Der findes mange forskellige attributter. Den første attribut du skal stifte bekendskab med er `style`. Med `style`-attributten kan du føje layout til dine sider - f.eks. en baggrundsfarve:

Eksempel 1:

```
<html>

  <head>
  </head>

  <body style="background-color:#ff0000;">
  </body>

</html>
```

Vil i browseren være en helt rød side ("`background-color`" betyder "baggrundsfarve"). Prøv selv.

Og hvordan var det lige siden blev rød?

I eksemplet ovenfor satte jeg baggrundsfarven til at være "`#ff0000`", som er en farvekode med den tekniske betegnelse Hexadecimaltal.

Til hver farve er der tilknyttet et Hexadecimaltal. Her er nogle eksempler:

Hvid: `#ffffff`
Sort: `#000000` (nuller) Rød: `#ff0000`
Blå: `#0000ff`
Grøn: `#00ff00`
Gul: `#ffff00`

Et Hexadecimaltal består af et `#` og seks cifre eller bogstaver, og der findes flere tusinde HEX-koder, der giver hver sin farve. Du kan også nøjes med at skrive `#f00`, dette vil også give en rød farve. Værdierne bliver altså automatisk dubleret.

I stedet kan man dog også bruge den engelske betegnelse for de mest almindelige farver (white, black, red, blue, green og yellow).

Men nok om det. Lad os komme tilbage til attributterne.

Okay, i hvilke tags kan der benyttes attributter?

Der kan bruges attributter i alle tags.

Men hvilke attributter og hvor ofte er meget forskelligt.

F.eks. vil man ofte indsætte attributter i `<body>`, for derved at give nogle ekstra informationer om en sides 'krop'. Omvendt vil man meget sjældent bruge attributter i `
`, da et lineskift normalt bare er et lineskift.

Ligesom det findes mange forskellige tags, findes der også mange forskellige attributter. Nogle attributter kan kun bruges i én slags tags, andre kan bruges i mange forskellige. Nogle tags kan kun indeholde en slags attributter, andre kan indeholde mange. Det lyder måske forvirrende, men ofte vil det være de samme attributter man bruger igen og igen, og man opdager, hvor lette de er at bruge, og hvor mange muligheder de giver.

Ligesom du lærer de vigtigste og mest brugte tags, lærer du også de væsentligste attributter i denne tutorial.

Hvad hedder den del af taget, som ikke er attributter?

Grunddelen af et tag, f.eks. `p` i `<p>`, hedder et element. Så et tag består altså af enten bare ét element f.eks. `<p>` eller af ét element og en eller flere attributter f.eks. `<h2 id="headings4">`.

Lektion 8: Links

Velkommen til lektion 8. I denne lektion lærer du at lave links mellem sider.

Hvad skal jeg bruge?

Til at lave links skal du bruge det, du altid bruger når du koder HTML: et tag. Et simpelt lille tag med ét element og én attribut, og du kan linke til alt og alle. Her er et eksempel på hvordan et link til html.dk ville se ud:

Eksempel 1:

```
<a href="http://www.html.dk">Her er et link til html.dk</a>
```

Ville se sådan ud i browseren: [Her er et link til html.dk](http://www.html.dk)

Elementet a står for "anchor", som er engelsk for "anker". Attributten href er en forkortelse af "Hypertext REFERENCE", hvilket betyder "Det er her linket går til"

href sættes lig "<http://www.html.dk>", hvilket er den fulde adresse på html.dk, og kaldes en URL, Uniform Resource Locator, bemærk "[http://](http://www.html.dk)", der også skal medtages. Sætningen "[Her er et link til html.dk](http://www.html.dk)" er det der ses i browseren som et link. Husk at lukke taget med et ``.

Hvad med links mellem mine egne sider

Hvis du linker mellem sider der deler placering, behøver du ikke skrive hele adressen (URL'en). Hvis du f.eks. har lavet to sider (side1.htm & side2.htm) og gemt dem i samme folder (f.eks. under "Dokumenter" på harddisken), kan du linke fra den ene side til den anden med kun navnet på siden. Et link fra side1.htm til side2.htm ville derfor se sådan ud:

Eksempel 2:

```
<a href="side2.htm">Et link til side 2</a>
```

Ligger side 2 i en underfolder (navngivet "underfolder"), kan linket se sådan ud:

Eksempel 3:

```
<a href="underfolder/side2.htm">Et link til side 2</a>
```

Et link den anden vej fra side 2 (i underfolderen) til side 1 vil da se sådan ud:

Eksempel 4:

```
<a href="../side1.htm">Et link til side 1</a>
```

"../side1.htm" gør at der hoppes et niveau tilbage før der kigges efter filen. Man kan også hoppe to niveauer tilbage, så ser det sådan ud: "../..side1.htm".

Forstod du systemet? Ellers kan du selvfølgelig altid bruge den fulde adresse (URL'en).

Og hvad med links internt på en side

Vil man have links internt på sine sider, kan det også sagtens lade sig gøre. F.eks. en indholdsfortegnelse øverst med links til hvert afsnit nedenfor. Du skal blot bruge attributten id. Med id-attributten kan du give tag en unik identifikation.

Eksempel 5:

```
<h1 id="overskrift1">Overskrift 1</h1>
```

Du kan nu linke til det specifikke tag. I linket skrives blot et "#" (så ved browseren at den skal blive på samme side) efterfulgt af det id du vil linke til:

Eksempel 6:

```
<html>
  <head>
  </head>
  <body>
    <p><a href="#overskrift1">Link til overskrift 1</a></p>
    <p><a href="#overskrift2">Link til overskrift 2</a></p>
    <h1 id="overskrift1">Overskrift 1</h1>
    <p>Tekst tekst tekst tekst</p>
    <h1 id="overskrift2">Overskrift 2</h1>
    <p>Tekst tekst tekst tekst</p>
  </body>
</html>
```

Vil se sådan ud i browseren (prøv at klikke på de to links):

Link til overskrift 1

Link til overskrift 2

Overskrift 1

Tekst tekst tekst tekst

Overskrift 2

Tekst tekst tekst tekst

Kan man også linke til en e-mail adresse?

Ja, man kan godt linke til en e-mail adresse.
Måden at gøre det på er den samme, som når du linker til en side:

Eksempel 7:

```
<a href="mailto:info@html.dk">Send en e-mail til html.dk</a>
```

Vil se sådan ud i browseren:

[Send en e-mail til html.dk](mailto:info@html.dk)

Forskellen er, at du i stedet for at skrive adressen på en side, skriver et "mailto:" efterfulgt af e-mail adresse. Når du klikker på linket, åbner du automatisk dit e-mail program med en ny besked adresseret til info@html.dk. Denne funktion vil dog kun virke, hvis der er installeret et e-mail program på computeren. Prøv.

Er der flere attributter, jeg bør kende?

For at lave et link skal du altid bruge href.
Herudover kan du bl.a. også sætte en titel på dit link:

Eksempel 8:

```
<a href="http://www.html.dk" title="Link til html.dk">Her er et link til html.dk</a>
```

Ville se sådan ud i browseren:

[Her er et link til html.dk](http://www.html.dk)

Attributten title kan bruges til at angive en klar beskrivelse af linket.
Hvis du, uden at klikke, holder musen henover linket, vil du se at teksten "Link til html.dk" kommer frem.

Title er en meget vigtig funktion når vi snakker SEO, search engine optimization, hvilket er udtryk for en proces, hvormed man forsøger at ranke eller gøre sitet mere synligt i en søgemaskine. Jo højere eller jo oftere et website bliver vist i en søgemaskine f.eks. Google, jo flere besøgende forventes sitet at have.
Ordlyden i title er vigtig for om den besøgende, finder beskrivelsen relevant og vigtig nok til at ville klikke på linket.

Lektion 9: Indsættelse af billeder

Ville det ikke være lækkert, hvis man kunne indsætte et billede af Richard Ragnvald lige midt på sin side?

Alt du skal bruge er som sædvanligt et tag: Eksempel 1:

```

```

Vil i browseren se sådan ud:



img er et element og en forkortelse af "image" ("billede" på dansk). src er en attribut og en forkortelse af "source" ("kilde"). Det virker jo meget logisk: Først fortæller man browseren, at man ønsker at indsætte et billede (img) og dernæst hvor det er placeret (src). Læg mærke at img åbnes og lukkes i samme tag, ligesom med
 er der tale om en kommando, der ikke er knyttet til et stykke tekst.

"ragnvald.jpg" er navnet på det billede, du ønsker at indsætte på din side. ".jpg" er den fil-type, dit billede er gemt i (på samme måde som ".htm" fortæller at en fil er et HTML-dokument). Der findes tre billedtyper, man kan indsætte på sine sider:

GIF (Graphics Interchange Format)
JPG eller JPEG (Joint Photographic Equipment Group)
PNG (Portable Network Graphics)

GIF-billeder er bedst at bruge til tegninger, mens JPEG-billeder er bedst til fotografier. Dette skyldes to ting. For det første kan GIF-billeder kun bestå af 256 farver, mens JPEG-

billeder kan indeholde millioner af farver. For det andet er GIF-formatet bedst til at komprimere enkle billeder, mens JPEG-formatet er bedre til de mere komplicerede billeder (jo mindre dine billeder fylder, jo hurtigere bliver din side indlæst i browseren - og der findes ikke noget mere irriterende end unødvendige 'tunge' sider, når man sidder på en langsom forbindelse).

Traditionelt har man primært brugt GIF- og JPEG-formatet, men på det seneste er også PNG-formatet blevet mere populært - især på bekostning af GIF-billeder. PNG indeholder på mange måder det bedste fra både JPEG- og GIF-formatet: millioner af farver og effektiv komprimering.

Hvor får jeg nogle andre billeder fra?

Du kan nemt downloade billeder på Internettet uden problemer (pas dog på copyrights!). Her er en beskrivelse af, hvordan du gør:

Klik på det ønskede billede med højre museknap
Vælg "Save picture as..." ("Gem billede som...") i menuen der kommer frem
Vælg derefter en placering til billedet på harddisken og tryk "Save" ("Gem")

Gør dette med nedenstående billede af html.dk's logo og gem det på harddisken samme sted, som du har dine hjemmesider til at ligge (læg mærke til at logoet er gemt som en GIF-fil: logo.gif):



Du kan nu indsætte billedet på en af din egne sider. Prøv selv.

Er der mere?

Der er et par ting mere, der er handy at vide om billeder. Den ene er, at på samme måde som du kan linke til sider i en underfolder - som du lærte i forrige lektion - kan du også hente dine billeder fra andre foldere - eller endda andre websites:

Eksempel 2:

```

```

Eksempel 3:

```

```

Og så lige en anden ting: Billeder kan godt være links:

Eksempel 4:

```
<a href="http://www.html.dk">  
</a>
```

Vil se sådan ud i browseren (prøv at klikke på billedet når du selv har indsat det):



Er der flere attributter, jeg bør kende?

Du skal selvfølgelig altid bruge attributten `src`, der fortæller, hvor billedet er placeret. Derudover er der ligesom ved links en række andre attributter, der kan være nyttige ved indsættelse af billeder.

Attributten `alt` bruges til at give en alternativ beskrivelse af billedet, hvis det af en eller grund ikke vises for brugeren. Dette er især vigtigt for blinde/synshandicappede og hvis siden loades langsomt eller slet ikke vises. Derfor bør attributten altid bruges:

Eksempel 5:

```

```

Ligesom ved links kan attributten `title` bruges til at skrive et kort beskrivelse af billedet:

Eksempel 6:

```

```



Hvis du, uden at klikke, holder musen henover linket, vil du se at teksten "html.dk's logo" kommer frem.

To andre attributter er width og height:

Eksempel 7:

```

```

Ville se sådan ud i browseren:



width (engelsk for "bredde") og height (engelsk for "højde") kan bruges til at angive et billedes højde og bredde. Værdien der bruges til at angive width og height, er pixels. Pixels er også den måleenhed, der bruges til at angive skærmopløsning (den mest almindelige skærmopløsninger er 1024x768 pixels) men vil i øvrigt hele tiden ændre sig. I forhold til f.eks. centimeter er pixels en relativ måleenhed der afhænger af skærmopløsningen - hos en bruger med høj skærmopløsning vil 25 pixels måske svare til en centimeter på skærmen, mens de samme 25 pixel ved en lav skærmopløsning er 1½ centimeter på skærmen.

Hvis du ikke angiver width og height, bliver billedet bare indsat med den størrelse det faktisk har, men med width og height kan du manipulere med størrelsen:

Eksempel 8:

```

```

Ville se sådan ud i browseren:



Den reelle størrelse på billedet forbliver dog den samme, og det tager den samme tid at hente billedet ned. Derfor bør du aldrig gøre et billede mindre ved at bruge width og height, men i stedet formindske det i dit billedprogram, og derved gøre dine sider lettere og hurtigere.

Alligevel er width og height en god idé at bruge, da browseren på den måde på forhånd ved, hvor meget plads billedet vil tage, og kan sætte din side pænt op allerede inden billedet er hentet ned.

Lektion 10: Tabeller

Tabeller bruges til at præsentere tabulær data, data som bedst vises i rækker og kolonner.

Er det svært?

Opbygningen af tabeller i HTML kan virke kompliceret og uoverskueligt, men hvis man holder hovedet koldt og tungen lige i munden, er det egentlig meget simpelt og logisk.

Eksempel 1:

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

Vil se sådan ud i browseren:

Celle 1 Celle 2
Celle 3 Celle 4
Hmmmm... ?

Jeg sagde jo, at det kunne virke kompliceret og uoverskueligt. Men lad os prøve at gøre det simpelt og logisk:

Der bruges 3 grundlæggende typer af tags i tabeller:

[<table>](#) betyder "tabel". At starte tabellen med dette tag og lukke det ved afslutningen er vel derfor meget logisk.
[<tr>](#) står for "table row" = "Tabel række" på dansk. Hver ny række startes med et [<tr>](#) og afsluttes med et [</tr>](#). Stadig logisk.
[<td>](#) er en forkortelse for "table data" ("tabel data"). Taget indsætter en ny celle i din tabel, og [</td>](#) afslutter den.

Det der sker i Eksempel 1 er altså, at tabellen startes med et `<table>`, efterfulgt af et `<tr>`, der fortæller, at her begynder en række.
I denne række indsættes to celler: `<td>Celle 1</td>` og `<td>Celle 2</td>`. Rækken afsluttes derefter med et `</tr>` og en ny række `<tr>` startes umiddelbart bagefter. Den nye række indeholder ligeledes to celler inden den afsluttes. Tabellen afvikles med `</table>`.

Lige for at slå det fast: rækker er celler horisontalt og kolonner er celler vertikalt:

| | |
|---------|---------|
| Celle 1 | Celle 2 |
| Celle 3 | Celle 4 |

Celle 1 og celle 2 udgør altså en række. Celle 1 og celle 3 er en kolonne.

I ovenstående eksempel havde tabellen to rækker og to kolonner. En tabel kan dog have ubegrænset antal rækker og kolonner:

Eksempel 2:

```
<table>
  <tr>
    <td>Celle 1</td>
    <td>Celle 2</td>
    <td>Celle 3</td>
    <td>Celle 4</td>
  </tr>
  <tr>
    <td>Celle 5</td>
    <td>Celle 6</td>
    <td>Celle 7</td>
    <td>Celle 8</td>
  </tr>
  <tr>
    <td>Celle 9</td>
    <td>Celle 10</td>
    <td>Celle 11</td>
    <td>Celle 12</td>
  </tr>
</table>
```

Vil se sådan ud i browseren:

```
Celle 1 Celle 2  Celle 3  Celle 4
Celle 5 Celle 6  Celle 7  Celle 8
Celle 9 Celle 10 Celle 11 Celle 12
```

Er der ingen attributer?

Selvfølgelig er der attributter. Med attributten `border` kan du f.eks. sætte en ramme om din tabel:

Eksempel 3:

```
<table border="1">
  <tr>
    <td>Celle 1</td>
```

```
<td>Celle 2</td>
</tr>
<tr>
  <td>Celle 3</td>
  <td>Celle 4</td>
</tr>
</table>
```

Vil se sådan ud i browseren:

| | |
|---------|---------|
| Celle 1 | Celle 2 |
| Celle 3 | Celle 4 |

Tykkelsen på rammen angives i pixels

Ligesom med billeder kan du også styre en tabels bredde (width) i pixels - men du kan også angive den i procent af skærmen:

Eksempel 4:

```
<table style="border:1px solid #ff0000;width: 30%;">
```

Vil i browseren svare til 30% af et helt skærmbillede. Prøv selv.

Flere attributter?

Der er andre attributter til tabeller.

```
<table border="1" style="float:right; width: 30%;">
```

Styrer placeringen af hele tabellen, men du kan bedre detaljestyre med margin

```
<table border="1" style="margin-left:200px; width: 30%;">
```

valign: Styrer indholdet vertikalt i den enkelte celle - f.eks. top, middle eller bottom

Eksempel 5:

Først en højde på tabellen

```
<table border="1" style="margin-left:200px; width: 30%;height: 200px">
```

herefter placeres indholdet i cellen

```
<td valign="top">Celle 1</td> (valign kan have værdien: top,middle,bottom)
```

Hvad kan jeg sætte ind i mine tabeller?

Du kan stort set indsætte alt i din tabeller: tekst, links og billeder... MEN tabeller er lavet til præsentere tabel-data , det vil sige data som meningsfuldt kan præsenteres i kolonner og rækker, så undlad at proppe ting ind i tabeller blot fordi du vil have det til at stå ved siden af hinanden.

Prøv nu selv at lave en række tabeller med forskellige størrelser, attributter og indhold. Det kan man sagtens få mange timer til at gå med.

Lektion 11: Flere tabeller

Efter lektion 10 lyder overskriften "Flere tabeller" måske som et mareridt. Det positive er, at når du først mester tabellerne, er der absolut intet i HTML-verdenen, der kan slå dig ud.

Hvad er der da tilbage?

`colspan` og `rowspan` er attributter og meget nyttige, når du skal lave tabeller på den seje måde.

`colspan` er en forkortelse af "column span" - på dansk: "kolonne spændvidde". Attribut-ten bruges i `<td>`-taget til at angive, hvor mange kolonner cellen skal spænde over:

Eksempel 1:

```
<table border="1">
  <tr>
    <td colspan="3">Celle 1</td>
  </tr>
  <tr>
    <td>Celle 2</td>
    <td>Celle 3</td>
    <td>Celle 4</td>
  </tr>
</table>
```

Vil se sådan ud i browseren:

| | | |
|---------|---------|---------|
| Celle 1 | | |
| Celle 2 | Celle 3 | Celle 4 |

Ved at sætte `colspan="3"` spænder cellen i den første række over tre kolonner. Havde jeg sat `colspan="2"` havde cellen kun spændt over 2 kolonner, og det havde været nødvendigt at indsætte yderligere en celle i den første rækken, så antallet af kolonner passer sammen i de to rækker.

Eksempel 2:

```
<table border="1">
  <tr>
    <td colspan="2">Celle 1</td>
    <td>Celle 2</td>
  </tr>
  <tr>
    <td>Celle 3</td>
    <td>Celle 4</td>
    <td>Celle 5</td>
  </tr>
</table>
```

Vil se sådan ud i browseren:

| | | |
|---------|---------|---------|
| Celle 1 | Celle 2 | |
| Celle 3 | Celle 4 | Celle 5 |

Hvad med rowspan?

`rowspan` kan oversættes til "række spændevidde", og som du måske allerede har gættet angiver `rowspan`, hvor mange rækker en celle skal spænde over:

Eksempel 3:

```
<table border="1">
  <tr>
    <td rowspan="3">Celle 1</td>
    <td>Celle 2</td>
  </tr>
  <tr>
    <td>Celle 3</td>
  </tr>
  <tr>
    <td>Celle 4</td>
  </tr>
</table>
```

Vil se sådan ud i browseren:

| | |
|---------|---------|
| | Celle 2 |
| Celle 1 | Celle 3 |
| | Celle 4 |

I eksemplet ovenfor er `rowspan` sat til "3" i Celle 1. Det betyder, at cellen skal spænde over 3 rækker (sin egen række plus to mere). Celle 1 og Celle 2 er altså i samme række, mens Celle 3 og Celle 4 udgør to selvstændige rækker.

Forvirret? Jamen, det er heller ikke helt nemt, og man mister hurtigt overblikket. Det kan derfor være en god idé at tegne tabellen på et stykke papir, før man går i gang med den i HTML.