



Webpublisering for nybegynnere

HTML / XHTML / CSS

HTML / XHTML	3
Hvorfor HTML?.....	3
Markeringsspråk	3
XML.....	3
World Wide Web Consortium (W3C)	3
Utviklingen av HTML	4
XHTML	4
XHTML dokumentets bestanddeler.....	4
XHTML-struktur.....	5
XHTML-dokumentets bestanddeler - body	5
Dokumenttyper.....	6
Block- og inline elementer.....	7
Velformet XHTML.....	7
Lister	7
Punktlistor	7
Nummererte lister	8
Definisjonslister	8
Lenker	9
Bokmerker.....	9
Validering	9
Bilder og bildeformater.....	10
GIF	10
JPG (JPEG)	10
PNG.....	10
Bilder og plassering av tekst.....	10
Linjeskift for bildetekst.....	11
Stilark: Cascading Style Sheets (CSS).....	12
Fordeler med stilark	12
Stilregler - syntaks	12
CSS kommentarer	12
Koble stilregler til HTML/XHTML-dokumenter	13
Inheritance - arv	13
Cascading: kombinere stilsett	13
Skriftegenskaper	14
Måleenheter i CSS	14
Farger	15
Bakgrunnsegenskaper	15
Boksegenskaper	15
Borderegenskaper	15
Listestiler.....	15
Lenkestiler - pseudoklasser.....	16
Plassering av bilder - CSS.....	16
class attributten	16
id attributten	17
Selektører i kontekst	17
div-elementet.....	17
XHTML / HTML igjen.....	18
Tabeller	18

CSS-formatering av tabell.....	19
META-tagger.....	19
Frames - Rammer.....	20
Forms – Skjema	21
Tekstbokser	22
Radioknapper	22
Avkryssningsbokser	22
Nedtrekksmenyer	23
Kommentarboks	23
Skjemareultater til epost.....	23
Referanser	24

HTML / XHTML

Hvorfor HTML?

HyperText Markup Language - hvorfor lære det når det er så mange webdesign programmer som produserer websider uten at du behøver å kunne HTML?

- ❑ Du får en bedre forståelse av WYSIWYG editorene
- ❑ Du kan selv se hva som skjer og rette opp feil
- ❑ Nye koder kommer til som du kan sette inn manuelt
- ❑ Du kan sette inn kode du har fått annet steds fra
- ❑ Bruk av stylesheets krever HTML kunnskap
- ❑ Scripting: Javascripts, VBScripts m fl krever HTML kunnskap
- ❑ Kommunikasjon med eksperter du samarbeider med blir bedre
- ❑ Det er lett og morsomt å lære, så hvorfor skal ikke du kunne det!

Markeringspråk

- HTML er et markeringspråk (kodespråk) – en applikasjon av SGML (Standardized Generalized Markup Language)
- SGML (Standardized Generalized Markup Language) er en samling regler for hvordan man lager markeringspråk
- Strukturen markeres med tagger <taggnavn>innhold</taggnavn>
- Et markeringspråk – en applikasjon av SGML – beskrives i en DTD (Document Type Definition)
- HTML er definert, beskrevet, med en DTD
- SGML-baserte dokumenter lagres som ren tekst
- SGML ble utviklet av Charles F. Goldberg m.fl. hos IBM på 1970-tallet og akseptert som en ISO-standard for dokumentutveksling i 1986 (ISO 8879)

XML

- ❑ XML står for Extensible Markup Language
- ❑ XML er en lettversjon av SGML som har høstet erfaringer fra HTML og er tilpasset dokumentutveksling på World Wide Web
- ❑ XML har ikke predefinerte tagger

World Wide Web Consortium (W3C)

- URL: <http://www.w3.org/>
- Opprettet oktober 1994 ved MIT (Massachusetts Institute of Technology)
- Ledes av Tim Berners-Lee
- Skal sikre felles standarder for WWW-teknologien
- Både akademiske og kommersielle institusjoner er medlemmer. For eksempel IBM, Microsoft, Netscape og Sun

Utviklingen av HTML

- HTML: Definert av Tim Berners-Lee i 1990
- HTML 2.0: Publisert av W3C i 1995
- HTML 3.2: W3C-standard i 1997
 - Innførte elementer for presentasjon (farge, skrifttyper og -størrelser m.m.)
- HTML 4: W3C-standard 1999. Tre varianter:
 - HTML 4.01 Strict
 - HTML 4.01 Transitional
 - HTML 4.01 Frameset

Minimum av obligatorisk koding for et html-dokument. Skriv inn i Notisblokk og vis i nettleser (Internet Explorer): Fil > Åpne > Bla gjennom, finn filen og klikk OK.

```
<html>
<head>
<title>Min første HTML-side</title>
</head>
<body>
Her kommer innholdet på siden min.
</body>
</html>
```

XHTML

- XHTML står for eXtensible HyperText Markup Language
- XHTML 1.0: W3C-standard i 2000
- En reformulering av HTML 4 i XML - tre varianter:
 - XHTML 1.0 Strict
 - XHTML 1.0 Transitional
 - XHTML 1.0 Frameset
- XHTML er nesten identisk med HTML 4.01
- XHTML er en strengere og renere versjon av HTML
- XHTML skal erstatte HTML
- XHTML er en applikasjon av XML
- XHTML er kompatibel med HTML 4.01 og støttes av alle nye nettlesere
- XHTML er en kombinasjon av alle elementene i HTML 4.01 kombinert med syntaksen, reglene, for XML
- XHTML er rent tekstformat. Brukes for å angi dokumentets struktur Presentasjon, utseende, angis med stilark (Cascading StyleSheets, CSS)

XHTML dokumentets bestanddeler

- **DOCTYPE**
- **head**
- **body**

XHTML-struktur

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <title>Et XHTML-dokument</title>
  </head>
  <body>
    <h1>Overskrift i XHTML-dokumentet</h1>
    <p>Avsnitt i dokumentet - dokumentinnholdet</p>
  </body>
</html>
```

XHTML-dokumentets bestanddeler - body

- ❑ Elementer
- ❑ Attributter
- ❑ Verdier
- ❑ Kommentarer
- ❑ Spesialtegn

Elementer:

- Eksempel: `<h1>Overskrift</h1>`
 - Dette elementet heter **h1**
 - Elementets **innhold** er **Overskrift**
 - **Start-tag**: `<h1>`
 - **Slutt-tag**: `</h1>`
 - Alle elementer må åpnes med en start-tag og lukkes med en slutt-tag
- Elementer uten innhold kalles **tomme**
 - Det finnes et fåtall tomme elementer i XHTML
 - Eksempel: `
` Lukkes ved å avslutte med en ordskiller og skråstrek i start-taggen

Atributter:

- De fleste XHTML-elementer kan ha ett eller flere **attributter**
- Eksempel: `Lenke til ICONS Management`
 - Attributter er par av navn og verdi
 - Verdien skrives med anførselstegn rundt

Kommentarer:

Kommentarer synes i kodebildet, men ikke i nettleser. Kommentarer er nyttig bl a for å skrive forklaringer til oss selv og andre som skal bidra til dokumentet. Eksempel på en enkel kommentar:

<!--Her kommer en forklaring på noe i kodebildet -->

Spesialtegn - entiteter

Før eller siden trenger du å sette inn et tegn som ikke finnes på tastaturet eller et tegn du ikke kan bruke fordi det er del av markeringsspråket. Du kan sette inn ANSI tegnkoder eller HTML spesialkoder. Begge koder starter med (&) og ender med (;)

Tegn	Beskrivelse	HTML-kode
©	Copyright symbol	© eller ©
®	Registrert varemerke symbol	® eller ®
é	Liten e med aksent	é
<	Mindre enn symbol	<
>	Større enn symbol	>
	Non-breaking space	

”Non-breaking space character” () brukes til å få mere luft. HTML ignorerer mer enn ett mellomrom. Dette kan rettes på ved å sette inn én eller flere ** **;

Dokumenttyper

Kilde: <http://www.w3schools.com/xhtml/>

XHTML 1.0 Strict

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

XHTML 1.0 Transitional

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Use this when you need to take advantage of HTML's presentational features and when you want to support browsers that don't understand Cascading Style Sheets.

XHTML 1.0 Frameset

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Use this when you want to use HTML Frames to partition the browser window into two or more frames.

Block- og inline elementer

Blokkelementer

<code><p></p></code>	”Paragraph”, avsnitt.
<code><h1></h1></code>	”Heading”, overskrifter fra 1-6
<code><blockquote></blockquote ></code>	Sitater. Kan ha attributten "cite" som er en URL til nettstedet sitatet er hentet fra. Mye brukt som innrykk uten at det er korrekt bruk av elementet
<code><hr /></code>	”Horisontal rule”, vannrett strek
<code></code>	”Unordered List”, punktlister. Flere listetyper
<code><pre></pre></code>	”Preformatting”, fast oppstilling på siden
<code>
</code>	”Break”, linjeskift (ikke riktig blokkelement)
<code><div align=’right’ ’left’ ’center’></div></code>	Justering av sideinnhold

Blokkelementer gjelder et helt avsnitt. Vanligvis får blokkelementer ”luft” over og under. Unntaket er `
` som likevel regnes som et blokkelement.

Inlineelementer

Brukes for å merke opp bokstaver, ord eller setninger i blokkelementer. Vises *ikke* på en ny linje, men flyter som en del av teksten i blokkelementet. Kan inneholde ren tekst og andre inline-elementer, men *ikke* blokkelementer.

Eksempel:

```
<p>Her kommer et avsnitt der <em>noen ord </em> vil vises som kursiv og der
<strong>noen andre ord </strong> vises uthevet.
```

em (emphasize) erstatter **i** (italic) i tidligere HTML-utgaver.

strong erstatter **b** for bold i tidligere HTML-utgaver.

Velformet XHTML

- Elementer må være nøstet riktig
- Taggnavn må være med små bokstaver
- Alle elementer må være avsluttet

Lister

Punktlister - Nummererte lister - Definisjonslister

Punktlister

```
<ul>
  <li>Første punkt i listen</li>
  <li>Andre punkt i listen</li>
  <li>Andre punkt i listen</li>
</ul>
```


Punkttyper i vanlig bruk:

```
<li type="circle"> "disc", "square" (CSS stilregler for listeformatering snart!)
```

Nummererte lister

```
<ol>
  <li>Første nummererte punkt</li>
  <li>Andre nummererte punkt</li>
  <li>Tredje nummererte punkt</li>
</ol>
```

Romertall og bokstaver kan brukes i stedet for vanlige tall i en nummerert liste:

```
<ol type="A">      blir store bokstaver
<ol type="a">      blir små bokstver
<ol type="I">      blir romertall
<ol type="i">      blir små romertall
```

Du kan starte en nummerert liste på et annet tall enn 1:

```
<ol start="10">
```

Lister kan settes inn i andre lister, nøstede lister, for å lage en effekt av underpunkter. Man kan også godt blande nummererte lister og punktlistor:

```
<ul>
<li> Første hovedpunkt</li>
<li> Andre hovedpunkt</li>
<li> Tredje hovedpunkt
  <ol>
  <li>Første underpunkt</li>
  <li>Andre underpunkt</li>
  </ol></li>
<li>Fjerde hovedpunkt</li>
</ul>
```

Definisjonslister

Slike lister har et ord eller en setning på én linje, *Definition Term*, og så definisjonen innrykket på linjen under, *Definition Data*.

```
<dl>
<dt>Svelvik</dt>
  <dd>Er verdens sentrum!</dd>
<dt>Oslo</dt>
  <dd>Er landets hovedstad</dd>
</dl>
```

Lenker

Det er `<a>` anker (anchor) elementet og attributten **href**, HyperText Reference, som brukes. I tillegg må det være noe å klikke på som synes på skjermen, og så en slutt `` element. Eksempel på en ekstern lenke (absolutt adressering) og på en lenke som går til en fil på samme nettsted og i samme mappe (relativ adressering):

```
<a href="http://www.icons.no">ICONS Management</a>
<a href="interessert.htm">Til interessesiden</a>
```

Vi har ofte bruk for en lenke som sender en epost. Det blir på samme måte som en ekstern lenke til Internett, men denne gangen med mail-protokollen "mailto":

```
<a href="mailto:ebuntz@icons.no">Send epost!</a>
```

Bilder kan også være lenker. Bilder som er lenker, får en ramme (border) rundt seg. Vi kan kode slik at rammen forsvinner.

```
<a href="index.htm"></a>
<a href="http://www.microsoft.com"></a>
```

Fjerne understreking på lenker, og annen formatering, gjøres med eksterne (helst) eller interne, embedded, stilark, men en enkelt lenke kan vi benytte *inline* stil på:

```
<a href=http://www.icons.no style="text-decoration: none">ICONS</a>
```

Bokmerker

Vi kan også lage lenker som hopper f eks fra en innholdsfortegnelse øverst på siden til et avsnitt lengere nede på samme side. Lenken går til et bokmerke.

HTML 4 versjonens koding: `Kapittel 2`
Attributten name er erstattet med **id** for bokmerker i XHTML.

XHTML:

Bokmerket ``

Lenken til bokmerket: `Til Kapittel 2`

Attributten **id** kan legges til et hvilket som helst element i dokumentet for å sette et bokmerke.

Bokmerke: `<h3 id="metaindexer">Hvordan fungerer metaindexer?</h3>`

Lenken: `Les mer om metaindexer`

Validering

Bruk W3Cs valideringstjeneste for å sjekke at en nettside er korrekt XHTML

URL: <http://validator.w3.org/>

Bilder og bildeformater

Grafikkfiler, bilder, er egne datafiler som lenkes inn i HTML-dokumentet.

```

```

Det bør være en tekstbeskrivelse av bildet for situasjoner der bildet ikke sees.

To bildefilformater er mest brukt på web, **gif**, Graphics Interchange Format, og **jpg**, uttales "jåddpegg", Joint Photographic Experts Group.

GIF

Gif-filformatet brukes for bilder med få farger, inntil 256 farger, større bildeflater med samme farge, strektegninger, tekst som bilder. Gif støtter:

- ❑ **Transparency** – en farge kan gjøres gjennomsiktig slik at bakgrunnen på siden synes gjennom bildet.
- ❑ **Interlacing** – få bildet til å "fade" inn på siden, komme til syne i stadier slik at man ikke behøver å vente til hele filen er lastet ned før vi ser noe på skjermen
- ❑ **Animasjon** – en serie gif-filer kan lage en animasjons-effekt

JPG (JPEG)

Jpg-filformatet brukes typisk for fotografier, bilder med mange og komplekse farger. Jo mer vi komprimerer filene, jo dårligere kvalitet. Komprimeringsmåten er såkalt "lossy", informasjon går tapt ved komprimering.

- ❑ **Progressive passes**– Jpg bilder kan også komme gradvis inn på skjermen, bli mer og mer tydelige, slik at noe synes på skjermen mens bilder lastes inn. Bilder skal være ganske store før man har glede av effekten.

PNG

Portable Network Graphics er en W3C-standard. Komprimerer uten tap av informasjon, kan ha millioner av farger.

Bilder og plassering av tekst

Det er flere måter å plassere bilder på en webside og flere måter å plassere tekst i forhold til bilder. i HTML 4 bl a:

```
  

```

Width, height, hspace, vspace, border er andre attributter vi kan bruke i forbindelse med bildefiler:

```

```

XHTML strict posisjonerer bilder og tekst ved hjelp av stilark.

Linjeskift for bildetekst

`<br clear="left" />` Nyttig linjeskift for bildetekst som får teksten til å stoppe og så fortsette under bildet når vestremargen, alternativt høyremargen "right", er "klar", eller begge margene er klare: `<br clear="all" />`.

Stilark: Cascading Style Sheets (CSS)

Stilark inneholder stilistiske retningslinjer som forteller nettleseren hvordan HTML/XHTML dokumenter skal vises for brukerne. Stilreglene bestemmer utseende på én eller flere HTML/XHTML sider, på ett eller flere elementer.

Stilsettet beskriver eksempelvis hvordan farger, fonter, overskrifter, listepunkter etc skal se ut og hvilke marger, linjeavstander og bakgrunner som skal brukes. CSS kan også kontrollere justering og plassering av bilder, tekst og andre elementer i dokumentene.

CSS er en standard fra World Wide Web Consortium. Det arbeides med en tredje utgave.

<http://www.w3.org/TR/REC-CSS2>

Fordeler med stilark

- Separerer form og innhold
- Gjør det enklere å skape felles utseende på sider
- Gjør det lettere å forandre og oppdatere en stor website
- Bidrar til renere HTML/XHTML-kode
- Bidrar til kortere nedlastningstid
- Bedre tilgjengelighet for ulike typer nettleserutstyr
- Samme dokument kan ha flere presentasjonsmåter

Stilregler - syntaks

En stilregel ser slik ut:

selektor { deklarasjon; deklarasjon; etc... }

Selektoren angir hvilket html-element det skal gjøres noe med, og deklarasjonene sier hva som skal gjøres med elementet. **Deklarasjonene består av egenskap og verdi**

Eksempel på en stilregel: `h1 { color: red; }`

- **selektor** er XHTML-elementet `h1`
- **deklarasjonen** er `color: red`
- **egenskapen** `color` får **verdien** `red`

CSS kommentarer

Vi kan sette forklarende kommentarer inn i stilark på samme måte som for html-dokumenter. En CSS kommentar ser bare litt anderledes ut:

`/* Her kommer en forklarende kommentar til en stilregel */`

Koble stilregler til HTML/XHTML-dokumenter

Stilreglene er ren tekst. Stilreglene kan skrives som en separat fil og lenkes inn (eksternt stilark, **linked**) i flere dokumenter eller stilreglene kan skrives inn i <head> i ett dokument (internt stilark, **embedded**) og vil da gjelde bare for det dokumentet. Stilregler kan også settes til enkelte elementer (**inline** stil).

Linking:

```
<head>
<title> ..... </title>
<link rel="stylesheet" href="lenkestil.css" type="text/css" />
</head>
```

Embedding:

```
<head>
<title> ..... </title>
<style type="text/css">
<!--
      h1 {font-family: arial; color: red;}
-->
</style>
</head>
```

Inline:

```
<a href="http://www.nb.no" style="text-decoration: none">Nasjonalbiblioteket</a>
```

Stilreglene har samme form uansett hvordan de kobles eller settes inn i html-filer!

Inheritance - arv

Stilregler satt for f eks **body-elementet** ”arves” nedover til alle andre elementer. Body-elementet er foreldreelementet og de andre elementene er barnelementer av body. Barnelementer arver egenskaper fra foreldreelementer. Eks: <p>Dette er tekst i et enkelt avsnitt.</p> En stilregel satt for p-elementet vil også gjelde for em-elementet.

Cascading: kombinere stilsett

Man kan godt kombinere flere stilsett og f eks ha ett eller flere eksterne stilsett i en og samme HTML-fil, med i tillegg flere innebygde STYLE-elementer og endatil noen HTML-elementer med STYLE-attributter.

Stilreglene kombineres med hverandre etter kaskadereglene. Det er regler for hva som skal brukes om noe er i konflikt med hverandre. Enkelt sagt, så er "vinner" det mest spesifikke eller det sist deklarererte.

Skriftegenskaper

- **font-family** brukes til å angi skrifttype
- **font-style** kan ha tre verdier: normal, italic (kursiv), oblique (hellende)
- **font-weight** brukes til å angi hvor fet skriften skal være, f eks normal eller bold.
- **font-size** angir hvor stor skriften skal være, enten relativt eller absolutt.
- **text-decoration** angir f eks: none (ingen understreking) eller underline (understreket tekst)

NB!

Når vi fører opp vår foretrukne font, så kan det være fint å føye til en generisk font-familie i tilfellet vår favorittfont ikke finnes på brukerens maskin.

De generiske fontfamiliene er:

- **serif**: fonter med ”seriffer”, f eks Times New Roman, Garamond
- **sans-serif**: fonter uten “seriffer”, f eks Arial, Verdana, Helvetica
- **monotype**: fonter med fast bredde for alle bokstaver, f eks Courier
- **cursive**: fonter med sammenhengende skrift, f eks Comic Sans MS
- **fantasy**: fonter med dekorative formål, f eks Western

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

Måleenheter i CSS

Relative størrelser:

- **em**: Høyden på skriften i elementet. Eks: 2em
PS: ”em” var opprinnelig bredden på bokstaven M for typografer, men brukes nå både horisontalt og vertikalt. Settes H1 til 2em, så blir overskriften dobbel størrelse av brødteksten. ”em” kan brukes til det meste.
- **ex**: Høyden på bokstaven x i elementet. Eks: 2ex. Brukes ikke så mye.
- **prosent**: Størrelsen på skriften i forhold til standard. Eks: 120%
Negative prosenttall er i noen sammenhenger tillatt. Prosent brukes ofte.

Absolutte størrelser:

- **in**: Tommer (en tomme er 2,54 cm)
- **cm**: Centimeter
- **mm**: Millimeter
- **pt**: Punkter (ett punkt er 1/72 tomme, altså 0,0352 cm)
- **pc**: Pica (en pica er 12 punkter, altså 1/6 tomme, eller 0,423 cm)

Det er én systemavhengig verdi:

- **px**: Pixler, punkter på skjermen. Størrelsen vil variere med skjermopløsningen. Fungerer mest som en absolutt verdi.

Farger

Fargekoden kan settes med rgb-verdier enten med desimaltall, hexadesimaltall, fargenavn (noen) eller også med prosentverdier. Eksempler på blå overskrift:

```
h1 { color: #0000ff; } h1 { color: blue; } h1 { color: #00f; }  
h1 { color : rgb (0,0,255); } h1 {color: rgb(0%, 0%, 100% )}
```

Bakgrunnsegenskaper

- **background-color** angir bakgrunnsfargen
- **background-image** angir bakgrunnsbilde:
body { background-image: url('bakgrunn.jpg'); }
- **background-repeat** angir om, eller hvordan et bilde repeteres. Kan ha verdiene repeat, repeat-x (horisontalt), repeat-y (vertikalt), no-repeat

Boksegenskaper

- **margin-left, margin-right, margin-top, margin-bottom** angir størrelsen på margene
- **padding-left, padding-top, padding-right, padding-bottom** angir "luft" mellom innhold og marger eller kanter, tilsvarende cellpadding for tabeller
- **float** angir at, og hvordan, f eks tekst skal flyte rundt elementet. Gyldige verdier er *left, right* og *none*

Borderegenskaper

- **border-style** angir utseende på en ramme rundt elementet. Kan være *none* (ingen ramme), *dotted* (prikker), *dashed* (små korte streker), *solid* (heltrukken linje), *double* (dobbel linje), *groove* (senket linje), *ridge* (hevet linje), *inset* (utstående linje) eller *outset* (senket relieff). En rekke forskjellige stilegenskaper kan angi farge, bredde osv på rammen: h1 {border: black solid thin;}
- **border-left, top, right, bottom** kan ha verdier for farge, linjetype og tykkelse (thin, medium og thick)

Listestiler

Punktlistier:

ul {list-style-type: **square**; }, **disc**: fylt sirkel, klassisk punkt, **circle**: en sirkel uten fyll.
ul {list-type-image: url("bilde.gif"); } et bilde brukt som listepunkt

Nummererte lister:

ol {list-style-type: **decimal**; }, **decimal-leading-zero**: to-sifrede desimaler, dvs 01, 02 ... 98, 99 osv, **lower-roman**: små romertall, **upper-roman**: store romertall, **lower-alpha**: små bokstaver, **upper-alpha**: store bokstaver

Plassering av listepunkter: list-style-position

Angir om listepunkttekst som går over mer enn en linje skal fortsette under i flukt med teksten over eller fortsette under listepunktet. Mulige verdier er:

- **outside:** punktet ligger til venstre for alle tekstlinjer (vanlig)
- **inside:** punktet ligger til venstre for første tekstlinje, andre tekstlinje har samme marg som listepunktet: `li { list-style-type: square inside; }`

Lenkestiler - pseudoklasser

CSS har noen få forhåndsdefinerte klasser, som ikke egentlig er klasser i vanlig forstand, bla *a link*, *visited*, *active* og *hover* som settes til a-elementet for å forandre utseende på de forskjellige stadier av lenker.

```
a:link {color: blue; text-decoration: none;} utseende for ikke besøkte lenker  
a:visited {color: navy; text-decoration: none;} besøkte lenker  
a:hover {background: cyan;} i det musepekeren er over lenken  
a:active {color: red;} i det musetasten trykkes ned over lenken
```

Plassering av bilder - CSS

Boksegenskaper: float, margin

Vi har tidligere brukt HTML-koding til å få tekstflyt rundt bilder, nå skal posisjoneringen skje med css-egenskaper. Vi erstatter align, hspace og vspace med float og margin.

Bildet står til venstre og teksten flyter rundt på høyre side og med litt klaring til teksten:

```
img {float: left; margin-right: 0.8em;}
```

class attributten

Alle elementer i HTML 4 og XHTML har en valgfri attributt **class** som kan settes til ett eller flere elementer. Stiler kan altså defineres som **en klasse "class"**. Du bestemmer hva klassen skal hete og hva den skal inneholde. Defineres f eks slik med "flag character" . (punktum):

```
.getaway {  
  color: white;  
  background-color: red;  
}
```

Brukes slik i avsnitts-taggen: `<p class="getaway">`. Stilen kan også brukes i andre elementer enn p. Hvis stilen bare skal brukes i p-elementet, så kan man kalle klassen for **p.getaway**. Det brukes på samme måte, men kun i p-elementet i dokumentet.

id attributten

id attributten kan også brukes som selektør i CSS-regler. id attributten er som class attributten med en viktig forskjell. id attributten må være unik for en type element i dokumentet. Hvis du skal ha samme effekt, samme id stilregel for forskjellige elementer, så må det settes forskjellig id navn for hvert element. Defineres f eks slik med "flag character" # (hash mark):

```
#intro {
  font-size: 110%;
  font-weight: bold;
  color: #0000ff;
}
```

Brukes slik i f eks p elementet: <p id="intro">

Selektører i kontekst

- Selektører vi har gått gjennom:
 - Elementer h1
 - Liste av elementer: h1, h2, h3
 - Psudoklasser for lenker
 - Klasser
 - id-attributter
- Selektører kan også settes i en spesiell kontekst:
ul li {color: blue;} stilen gjelder for listepunkter i punktlister. Listepunkter i nummererte lister, ol, vil ikke få blå farge.

div-elementet

div-elementet er et blokk element og brukes bl a til layout på sider. Alt innhold kan plasseres ved hjelp av CSS og man unngår å bruke tabeller f eks. Hver div-inndeling identifiseres med en id-attributt. Inndelingene skal deretter posisjoneres i skjermbildet.

- XHTML-elementet `div` brukes for å legge ut sider
- Hver seksjon får sitt eget bokmerke
- Det vil si at `div`-elementet får en *id*-attributt
- Eksempel: `<div id="meny"><p>Her er et avsnitt inne i seksjonen som heter meny.</p></div>`
- Vi kan lage CSS-regler for elementer inne i seksjoner
- `#meny p { color: green; }`
- Regelen betyr at avsnitt i seksjonen med bokmerke *meny* får skriftfarge grønn

XHTML / HTML igjen

Tabeller

Tabeller uten synlige rammer og cellevegger er fortsatt et viktig layout-verktøy på web. Vi kan bruke tabeller slik i HTML 4 og i XHTML transitional, men ikke i XHTML strict.

Koding for en minimalistisk tabell på 2 rader og 3 kolonner:


```
<table>
  <tr>
    <td>celleinnhold</td>
    <td>celleinnhold</td>
    <td>celleinnhold</td>
  </tr>
  <tr>
    <td>celleinnhold</td>
    <td>celleinnhold</td>
    <td>celleinnhold</td>
  </tr>
</table>
```

Element	Tag	Bruk
TABLE	<code><table>...</table></code>	Obligatorisk. Omslutter hele tabellen.
TABLE CAPTION	<code><caption>...</caption></code> <code><caption align="top"></code>	Valgfri. Overskrift "bottom", "right", "left" til tabellen
TABLE ROW	<code><tr>...</tr></code>	Obligatorisk. Omslutter innhold i en tabellrad.
TABLE HEADER	<code><th>...</th></code>	Valgri. Tittelcelle, teksten vil typisk vises uthevet og sentrert.
TABLE DATA	<code><td>...</td></code>	Obligatorisk. Omslutter celleinnhold. (Kan utelates dersom <code><th></code> brukes i stedet).
TABLE BORDER	<code>border="antall pixler"</code>	Valgri. Rammen rundt tabellen. Attributt til <code><table></code> tag.
CELLPADDING	<code>cellpadding="n pxl"</code>	Valgfri. Luft mellom celleinnhold og cellevegger. Attributt til <code><table></code> tag.
CELLSPACING	<code>cellspacing="n pxl"</code>	Valgfri. Tykkelsen på celleveggene inne i tabellen. Attributt til <code><table></code> tag.

```
<table border="5">
<caption>Tabelloverskrift settes her</caption>
<tr>
  <th>Kolonneoverskrift</th>
  <th>Kolonneoverskrift</th>
  <th>Kolonneoverskrift</th>
</tr>
<tr>
  <td>Vanlig celleinnhold</td>
  <td>Vanlig celleinnhold</td>
  <td>Vanlig celleinnhold</td>
</tr>
</table>
```

Tabellceller

Det er også nyttig å kunne styre hvordan innholdet i en celle plasserer seg. Standard er at tekst og annet innhold blir venstrejustert horisontalt og midtstilt vertikalt i cellen; unntaket er <th>, header-cellen som får innholdet helt sentrert. Vi kan styre plassering av innholdet i alle cellene på en rad med <tr> og for en enkelt celle med <td>. Vi setter attributter for "vertical" og "horisontal" justering:

```
<tr valign="top"> (alternativt: "bottom" eller "middle")
<td valign="top"> (alternativt: "bottom" eller "middle")
```

Celler kan slås sammen. Sammenslåing gjøres med td-attributtene **rowspan** og **colspan**.

```
<td colspan="3"> Cellen er tre kolonner bred </td>
```

CSS-formatering av tabell

- For rutenett rundt celler: **th, td** {border: black solid 1px;}
- For å unngå dobbeltstreker: **table** {border-collapse: collapse;}
- For luft mellom innhold og cellevegger: **th, td** {padding: 0.2em;}
- For justering av overskriftscelle: **th** {text-align: left;}
- For litt farger på overskriftscelle: **th** {color: white; background-color: green;}

META-tagger

<meta> tagger plasseres i <head> og er nyttige bl a for at søkemaskiner skal registrere sidene dine. I meta-taggen settes emneord og en beskrivelse av nettstedet.

```
<head>
<meta name="keywords" content="html, htmlkurs, webdesign" />
<meta name="description" content="Nybegynnerkurs i html-koding" />
</head>
```

Frames - Rammer

Frames, rammer, splitter websiden opp i områder, rader og kolonner. Typisk står ett eller to områder fast på skjermen, f eks toppområdet for logo etc og venstresiden for en meny. På en frames-side, som er delt i tre rammeområder, er det fire HTML-sider (filer).

FRAMESET siden som ikke har noe innhold selv, kun definerer rammene som skal fylles med andre sider, og tre **FRAMES** sider som vises i hvert sitt område.

Frameset-sider har ingen BODY tag, den er erstattet av FRAMSET taggen. HTML-sidene som skal vises i rammene er vanlige HTML-sider med BODY tag.

Framsett delt i to kolonner:

```
<html>
<head>
<title>To-delt frames-side</title>
</head>
<frameset cols="20%, 80%">
    <frame src="meny.htm" name="venstre">
    <frame src="velkommen.htm" name="hoyre">
</frameset>
</html>
```

Et klikk på en lenke i menyområdet får en ny side til å vises i hovedområdet f eks. Attributten **NAME** i FRAME taggen angir i hvilket område sidene skal vises. Dersom man skal lenke til og vise en side fra et eksternt nettsted, bør man sørge for at siden vises over hele skjermen og ikke inni vårt eget framesett. Det gjøres med attributten **TARGET=""_top"**. **TARGET=""_blank"** åpner en ny nettleser.

```
<a href="http://www.alltheweb.com" target=""_top">
```

Frameset delt i to rader der rad to er delt i to kolonner ("nested frames"):

```
<html>
<head><title>Tre-delt frames-side</title>
</head>
<frameset rows="15%, 85%">
    <frame src="logo.htm" name="toppvindu">
<frameset cols="20%, 80%"
    <frame src="meny.htm" name="venstrevindu">
    <frame src="velkommen.htm" name="hovedvindu">
</frameset>
</frameset>
</html>
```

Attributter:

```
<frameset frameborder="yes|no" border="0">
<frame scrolling="yes|no|auto" noresize frameborder="yes|no" border="0">
```

På grunn av at noen fremdeles bruker gamle nettlesere som ikke kan vise frames, så er det vanlig å legge inn en alternativ side mellom taggene `<noframes>` `</noframes>` foran avsluttet framestag `</frameset>`:

```
-----  
<noframes>  
  <body>  
    <h1>Framesfritt alternativ</h1>  
    Alternativ side for nettleser som ikke kan vise frames ...  
  </body>  
</noframes>  
</frameset>  
</html>
```

Forms – Skjema

Mange typer skjemafelter kan brukes til å samle input, tekstbokser, avkrysningbokser, nedtrekksmenyer mm. Common Gateway Interface (CGI) eller annen skripting må til for å få skjemaene til å virke.

Et klassisk skjema består av tre elementer: `<form>` tag, input felter, og send-knapp.

```
<form  
  method="post"  
  action="http://www.server.com/cgi-bin/scriptfil">  
</form>
```

METHOD attributten avgjør hvilken metode nettleseren skal bruke for å sende data til web-serveren. METHOD har to verdier "get" og "post". Med "post" blir data fra skjemaet sent til URL'en oppført i ACTION attributten. ACTION attributten spesifiserer navn og lokalisasjon for scriptet som skal behandle informasjonen sendt i skjemaet. Post er den vanligste metoden å sende skjema-informasjon på.

Følgende tagger settes innenfor FORM taggene for å lage de forskjellige typene skjemafelt:

```
<input />, <select> og <textarea>
```

INPUT brukes til tekstbokser, avkryssningsbokser, radioknapper, send og nullstillknapper. SELECT brukes til nedtrekkslister. TEXTAREA brukes til kommentarbokser for lengre meldinger.

INPUT har en attributt TYPE og en verdi for å angi hva slags skjemafelt det skal være "radio", "text", "checkbox", "submit", "reset". Skjemafeltene har også navn slik at vi ser hvilket felt som er fylt ut når vi får svarene fra skjema.

```
<input type="feltype" name="feltnavn" size="bredde" value="tekst som står i  
boksen eventuelt" maxlength="max antall tegn tillatt" />
```

Eksempel: `<input type="text" name="Epostadresse" size="40" />`

Attributten VALUE brukes for å sette tekst på skjemaknappene send og reset slik at de får den teksten vi vil, f eks: "Send skjema" og "Tøm skjema".

Tekstbokser

Tekstbokser brukes mye i skjemaer. Tekstbokser egner seg for utfylling av navn, adresse, epost etc. Et enkelt skjema med tekstskjemafelt kan se ut som følger:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Enkelt skjema</title></head>
<body>
<h1>Enkelt skjema</h1>
<form
  method="post"
  action="http://hoohoo.ncsa.uiuc.edu/htbin-post/post-query">
Skriv ditt navn her: <input type="text" name="Navn" size="40"><br />
Skriv din epost her: <input type="text" name="Epost" size="40">< br />
<input type="submit" value="Send skjema">
<input type="reset" value="Tøm skjema">
</form>
</body>
</html>
```

Radioknapper

Radioknapper er sammen som en gruppe. Du kan bare velge én av to eller flere. Alle skjemafeltene i gruppen har samme NAME attributt. Én av knappene kan merkes som valgt, CHECKED, i <INPUT> taggen:

```
<p>Har du sett filmen "I've got mail?" <br />
<input type="radio" name="settfilm" value="ja" checked />Ja
< input type ="radio" name="settfilm" value="nei" />Nei
</p>
```

Avkryssningsbokser

Med avkryssningsbokser kan man velge å krysse av så mange man vil.

```
<p>Hvilke steder vil du reise til? Velg ett eller flere steder!
<input type ="checkbox" name="Sydney" /> Sydney
<input type="checkbox" name="London" /> London
<input type ="checkbox" name="NewYork" /> New York
</p>
```

Nedtrekksmenyer

Nedtrekksmenyer (Select lists) likner mere på lister der du kan velge fra innførsler i listen. Det er teksten, verdien, til høyre for OPTION som sendes fra skjemaet. Hvis du vil sende en verdi som er forskjellig, så kan du sette til VALUE="verdi" inni én eller flere av OPTION taggene.

```
<form method="post" action="/~brukernavn/cgi-bin/cgiprog.cgi">
<pre>
Fornavn: <input type="text" size="30" name="fornavn" value="" />
Etternavn: <input type="text" size="30" name="etternavn" value="" />
  <select>
    <option value="katt">Katt </option>
    <option value="hund">Hund </option >
    <option value="elefant" selected>Elefant </option>
  </ select >
  <input type="submit" /> <input type="reset" />
</pre>
</form>
```

Hvis du vil lage nedtrekksmenyen til en (Multiple-option select list) og tillate flere valg på én gang fra samme liste, så setter du til attributten MULTIPLE i <SELECT> taggen. Menyene vil da vanligvis se litt anderledes ut, flere eller alle valg vil vises i skjermen.

Kommentarboks

Kommentarboks (Textarea) er beregnet på mere enn én tekstlinje. Her kan du skrive "brev" hvis du vil. <TEXTAREA> må ha en slutttagg </TEXTAREA> i tillegg til starttaggen. Du kan sette til noen attributter og verdier som styrer utseende og måten teksten vises i kommentarboksen.

Attributt	Verdi	Beskrivelse
COLS	Tallverdi	Skjemafeltets bredde i pixler.
ROWS	Tallverdi	Skjemafeltets høyde i antall linjer som vises på skjermen.
WRAP	"none" eller "virtual"	Hvis "none" så kommer teksten på én linje og det må scrolles horisontalt. Hvis "virtual" så bryter teksten til neste linje når kanten er nådd.

```
<p>Vi vil gjerne høre fra deg! Skriv inn dine kommentarer til oss!<br />
<textarea name="Kommentar" cols="50" rows="5" wrap="virtual">
</textarea>
</p>
```

Skjemaresultater til epost

Du kan få skjemaresultater sendt til én eller flere epost-adresser og komme utenom å bruke et script på serveren til prosessering av skjemaresultatet.


```
<form action="mailto:ebuntz@icons.no" method="post">
```

Du kan også spesifisere hva som skal stå i Subject feltet når mailen mottas og også om noen andre skal ha kopi av mailen:

```
<form action="mailto:ebuntz@icons.no"?subject=Websvar cc=ola@firma.no  
cc=kari@firma.no" method="post">
```

Det er ingen spesielle CSS egenskaper for skjemaer – vanlige CSS egenskaper kan brukes, men det varierer en del hvordan nettlesere tolker CSS for skjema.

Referanser

Liste over HTML-tagger:

http://webach.com/barebones/no/barebones_no.html

<http://www.htmlhelp.com/reference/wilbur/list.html>

Kurs

<http://www.w3schools.com/html/default.asp>

<http://www.w3schools.com/xhtml/default.asp>

<http://www.buntz.no/html/>

CSS

Komplett liste over cascading stylesheet spesifikasjoner:

<http://www.htmlhelp.com/reference/css/properties.html>

Ressurssted for stilark;

<http://www.csszengarden.com/>

