

HTML





HTML5

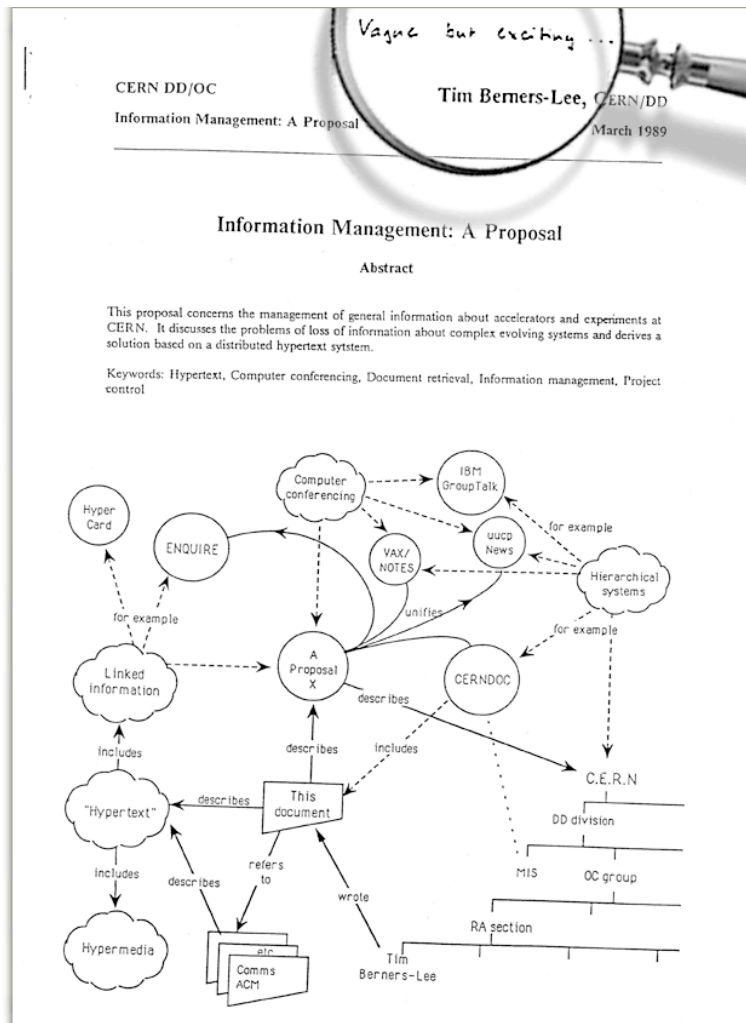
Tekst og layout, af både dette materiale og det tilhørende digitale opgavemateriale:
Hasse Sørensen/Nordisk Grafik for MicroWorld A/S

Alle rettigheder tilhører MicroWorld A/S © 2012.

Kopiering og distribution kun tilladt efter forudgående aftale med MicroWorld A/S.

HTML5 koden og CSS3 udarbejdes og vedtages af W3C, The Internet Consortium. Læs mere på w3.org.

HTML, CSS, FTP og HTTP – og dermed internettet – er oprindeligt udviklet af Tim Berners-Lee i marts 1989.





Indhold

| | |
|---|----|
| Forord | 4 |
| Fremtidens kode | 5 |
| Før vi går i gang..... | 6 |
| De nye strukturkoder..... | 7 |
| Indsatte medier | 9 |
| WebFonts | 10 |
| Rotation – første forsøg..... | 11 |
| Kompatibilitet med gamle browsere | 12 |
| Chrome Frame | 13 |
| Afrundede hjørner..... | 14 |
| Background property | 15 |
| RGB-farver og gennemsigtighed | 16 |
| HSL-farver og gennemsigtighed | 17 |
| Farveforløb | 18 |
| Skygger..... | 19 |
| Tekst i spalter | 20 |
| Animation | 21 |
| Keyframe animation | 22 |
| 3D transformation..... | 23 |
| Canvas | 25 |
| SVG | 26 |
| Formularer | 27 |
| GeoLocation..... | 29 |
| AppCache | 30 |
| App's..... | 31 |
| Links om HTML5, CSS3 og jQuery | 32 |
| Kodeoversigt (engelsk) | 33 |



Forord

Da begreberne *nutid* og *fremtid* bruges flittigt i dette materiale er det vigtigt at gøre opmærksom på at det er forfattet i perioden februar-marts 2012. I en ikke-så-fjern-fremtid vil det opfattes som *latterligt længe siden*.

Materialet kan bruges sammen med dine egne websites i en simpel teksteditor eller et avanceret CMS, men for at få det fulde udbytte af materialet bør det bruges sammen med den tilhørende opgavemappe på computeren.

Desuden bør du benytte

- Adobe Dreamweaver CS 5.5 eller nyere, gerne med både iPhone og Android mobile application installeret (Site > Mobile Application > Configure Application Framework...)
- En Webkit browser, fx Google Chrome, eller – helt optimalt – Apple Safari
- Adobe Device Central til mobiltest





Fremtidens kode



HTML5 er et “working draft”, altså stadig blot et udkast. Meningen med HTML5 er en langt bedre strukturering af koden, og en imødekommenhed overfor hvordan HTML benyttes i virkeligheden. De fleste websider har en top og en bund med et tydeligt differencieret design, med det resultat at webdesignere har navngivet div’er med “top”, “bund”, “bottom”, “footer” etc. De har nu fået entydige koder: `<header>` og `<footer>`.

Samtidig er det en imødekommenhed overfor brugen af eksterne mediefiler, som i stedet for at ‘kalde’ eksterne afspillere skal kunne afspilles af browseren selv, med en enkelt kode: `<video>` eller `<audio>`. – og så fremdeles...

Udkastet til HTML5 standarden fylder pt. 900 sider, og det er hensigten at alle browsere skal kunne afvikle alle websites helt ensartet, og samtidig være bagud-kompatible. Det er altså et klart brud med de principper der ligger bag udviklingen af XHTML, som kunne betragtes som en hård disciplinering af kodeudviklingen. Derfor begynder dokumentet ikke længere med `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">` men helt enkelt med `<!DOCTYPE html>`.

Men hvis HTML5 endnu ikke er færdigudviklet, og browserne endnu ikke understøtter HTML5, hvorfor skal vi så overhovedet beskæftige os med HTML5?

– fordi de andre børn heller ikke kunne vente med at pakke gaver ud!

De dele af HTML5 der ér vedtaget understøttes i en lang række browsere, og selv Adobe Dreamweaver understøtter en lang række HTML5 koder. Google har i vidtstrakt grad implementeret HTML5 og CSS3 i både deres browser (Google Chrome) og i deres webmail (gmail). Derfor!



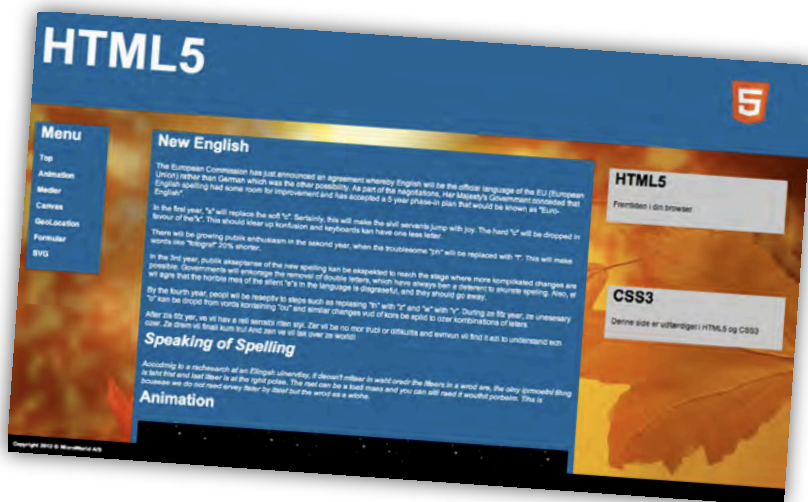


Før vi går i gang...



Opgaverne i dette materiale *kan* løses med en teksteditor, fx Notepad, men det optimale vil være at bruge Dreamweaver 5.5 eller senere.

1. Hent opgavemappen fra adx.dk/html5/opgavemappe.zip
2. Udpak zip-filen til en mappe på skrivebordet
3. Opret et Site i Dreamweaver
4. Knyt sitet til opgavemappen (*Local Site Folder* i *Site Setup*)
5. Hent Apples browser Safari fra apple.com/safari og vælg at den skal være standardbrowser
6. Åben index.html i Dreamweaver





De nye strukturkoder



Helt grundlæggende er sidens struktur ganske som den plejer, men der er både nye koder og gamle koder, der er blevet forenklet, bla. for at afhjælpe inkonsekvent brug af DIV-tags, og for at teksten på et website bedre kan udveksles med andre medier.

| | |
|---|--|
| <code><!DOCTYPE HTML></code> Forenklet | <code><article></code> Ny |
| <code><html></code> | Brødtekst |
| <code><head></code> | <code></article></code> |
| <code><meta charset="UTF-8"></code> Forenklet | <code></section></code> |
| <code><title>HTML5</title></code> | <code><aside></code> Ny |
| <code></head></code> | Sidetekst |
| <code><body></code> | <code></aside></code> |
| <code><header></code> Ny | <code><footer></code> Ny |
| <code><h1>HTML5</h1></code> | <code><h6>Footer tekst</h6></code> |
| <code></header></code> | <code></footer></code> |
| <code><nav>Menu</nav></code> Ny | <code></body></code> |
| <code><section></code> Ny | <code></html></code> |

De enkelte [Section](#), [Article](#) og [Aside](#) tags kan henvise til CSS definitioner, og ved brug af flere kan de udvides med fx `id=aside1`. Det er naturligvis også muligt at oprette CSS definitioner til [Header](#), [Nav](#) og [Footer](#).

```

footer {
    position: fixed;
    bottom: 0px;
    left: 0px;
    width: 100%;
    background-color: #000;
    color: #FFF;
    z-index: 5;
}

```

Vi skal senere i dette kursus kigge på en lang række interessante nyheder i CSS3.

Først skal vi bruge de nye og forenklede koder til at rydde op i `index.html` i opgavemappen.

Kig efter bemærkningerne i koden

```

<!-- HTML bemærkning -->
/* CSS bemærkning */
// JavaScript bemærkning

```





Følgende HTML kode

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<div id="header"> ... </div>
```

```
<div id="navigation"> ... </div>
```

```
<div id="sektion"> ... </div>
```

```
<div id="art1"> ... </div>
```

```
<div id="art2"> ... </div>
```

```
<div id="art3"> ... </div>
```

```
<div id="dims1" class="dims"> ... </div>
```

```
<div id="dims2" class="dims"> ... </div>
```

```
<div id="sidefod"> ... </div>
```

Erstattes med

```
<!DOCTYPE HTML>
```

```
<meta charset="UTF8">
```

```
<header> ... </header>
```

```
<nav> ... </nav>
```

```
<section> ... </section>
```

```
<article id="art1"> ... </article>
```

```
<article id="art2"> ... </article>
```

```
<article id="art3"> ... </article>
```

```
<aside id="aside1"> ... </aside>
```

```
<aside id="aside2"> ... </aside>
```

```
<footer> ... </footer>
```

Følgende CSS kode

```
#header
```

```
#sektion
```

```
#navigation
```

```
#header, #navigation, #sektion
```

```
.dims
```

```
#dims1
```

```
#dims2
```

```
#sidefod
```

Erstattes med

```
header
```

```
section
```

```
nav
```

```
section, nav, header
```

```
aside
```

```
#aside1
```

```
#aside2
```

```
footer
```




Indsætte medier



Vi skal indsætte en lydfil og en lille film i dokumentet. HTML5 kompatible browsere kan afspille disse filer uden plug-in's, så vi skal bruge enkle koder til at indsætte dem:

`<audio>` og `<video>`.

Audio koden ser sådan ud, i den mest enkle form:

```
<audio>
  <source src="media/cph.mp3" type="audio/mpeg" />
</audio>
```

Der er dog et par ekstra koder der med fordel kan indsættes, sådan at browseren viser kontrolknapper til afspilningen, så browseren kan vælge imellem flere formater af lydfilen, og så der fremkommer en tekst til dem der ikke kan høre og se afspilleren:

```
<audio controls="controls">
  <source src="media/cph.ogg" type="audio/ogg" />
  <source src="media/cph.mp3" type="audio/mpeg" />
Din browser understøtter ikke audio-elementet
</audio>
```



Sæt lyden ind under den sidste artikel.

Derefter indsætter vi en lille video:

```
<video width="640" height="360" controls="controls" poster="images/poster.jpg">
  <source src="media/road.ogg" type="video/ogg" />
  <source src="media/road.mp4" type="video/mp4" />
  <source src="media/road.flv" type="video/flv" />
Din browser understøtter ikke video-elementet.
</video>
```

“poster” betyder et billede der bliver vist indtil filmen startes.

Både video og audio koden kan udvides med følgende funktioner:

| | |
|--|--|
| <code><video autoplay="autoplay"></code> | Afspil når siden er hentet. |
| <code><video muted="muted"></code> | Filmen er lydløs. |
| <code><video preload="auto"></code> | Filmen loades automatisk, klar til afspilning. |

Læs mere på w3schools.com/html5/tag_video.asp



WebFonts



Hidtil har vi været voldsomt begrænset i forhold til brugen af typografi på websites: Times New Roman, Arial og Verdana...

Årsagen er jo at skrifttypen skal være installeret på den besøgendes computer for at virke. Derfor har vi enten brugt billeder eller Flash, når vi ville bruge en alternativ skrifttype.

I HTML5/CSS3 er det muligt at benytte skrifter der ligger på en webserver beregnet til formålet.

Der findes en række betalingservices, bla. *webfonts.fonts.com*, der hoster over 8.000 fonte, men i dette tilfælde vil vi benytte en gratis: *www.google.com/webfonts*, der hoster et lidt mere begrænset – men voksende – antal fonte. Der kan du finde og afprøve fonte, og finde koderne til at benytte fontene på dit website.

Lige under `<style type="text/css">` skriver vi

```
@import url(http://fonts.googleapis.com/css?family=Lobster);
```

– hvorved skrifttypen Lobster bliver hentet sammen med siden.

I sidens interne stylesheet retter vi font-informationen i h1 og h2:

```
font-family: 'Lobster', arial, sans-serif;
```



På *www.google.com/webfonts* er det muligt at hente skrifttyperne som opentype til computeren, så du kan se dem imens du arbejder med udviklingen af siden på din computer. Du kan naturligvis vælge også at bruge dem i tryksager, så du sikrer ensartetheden.

Lobster

Josefine Sans

Over the rainbow



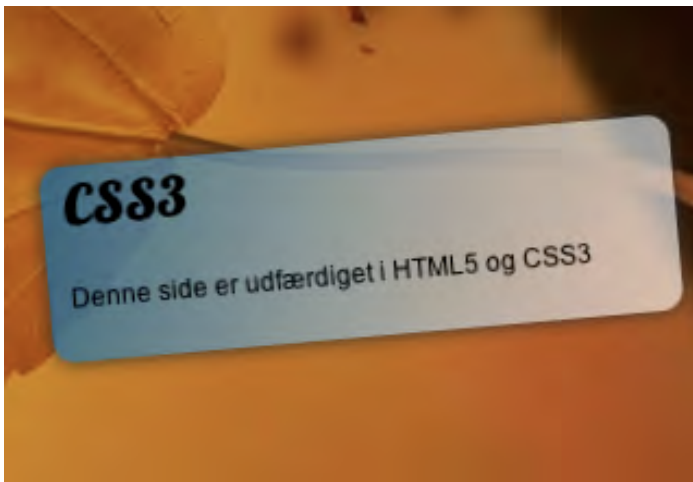
Rotation – første forsøg...



En funktion der i HTML4/XHTML krævede billeder eller Flash.

I `#aside2` i sidens interne stylesheet indsættes følgende kode:
`transform: rotate(-5deg);`

Dette burde rotere vores `aside2` 5 grader mod uret, men det virker ikke...
– fortsæt derfor til “Kompatibilitet med gamle browsere”.





Kompatibilitet med gamle browsere



Relativt nye browsere er bygget til at kunne forstå og eksekvere HTML5 og CSS3 i en vis udstrækning. Men da mange af de sidste opdateringer er lavet imens mange nye koder var 'eksperimentelle' skal vi gøre browseren opmærksom på at det er en ny 'eksperimentel' kode tiltænkt dén.

Der findes fire forskellige typer browsere:

| | |
|--------|--|
| Moz | Mozilla Firefox |
| O | Opera |
| MS | Microsoft Internet Explorer |
| Webkit | Alle browsere der følger W3C's anbefalinger slavisk, fx Google Chrome, og Safari |

Vi indsætter derfor en undtagelseskode:

```
-webkit-transform:rotate(-5deg);
-moz-transform:rotate(-5deg);
-o-transform:rotate(-5deg);
-ms-transform:rotate(-5deg);
transform:rotate(-5deg);
```



Grunden til at vi indsætter den originale kode til sidst er at CSS læses som et skænderi; det *sidste* udsagn browseren forstår bliver eksekveret. Hvis jeg fx skriver `color:#fff; color:#000;` bliver farven #000 (sort), da dét er det sidste udsagn.

– på den måde kan vi indsætte undtagelseskoder på websites, imens vi venter på at browserne forstår de originale koder, uden at behøve huske på at fjerne undtagelseskoder på et større antal websites i fremtiden.

Ældre versioner af Internet Explorer kan forstå flere CSS3 koder hvis vi indsætter en funktion der kan hentes fra css3pie.com. Dette er allerede gjort, og de hentede filer ligger i mappen *pie*. Derfor indsætter vi følgende kode i slutningen af hver CSS definition der bruger nye funktioner (rgba farver, runde hjørner, skygger, rotation m.v.)

```
behavior: url(pie/PIE.htc);
```

Fx:

```
#apdiv1 {
    position: absolute;
    top: 100px;
    left: 100px;
    width: 300px;
    height: 650px;
    -webkit-transform:rotate(-5deg);
    -moz-transform:rotate(-5deg);
    -o-transform:rotate(-5deg);
    -ms-transform:rotate(-5deg);
    transform:rotate(-5deg);
    behavior: url(pie/PIE.htc);
}
```





Chrome Frame



En stor del af de brugere, der *ikke* jævnligt opdaterer deres browser, benytter en ældre version af Internet Explorer. – *det var den der fulgte med computeren, så det må vel være fint!*

Det er nu muligt at få selv ældre versioner af Internet Explorer til at vise HTML5 og CSS3 med alle de funktioner der understøttes af Google Chrome, takket være et plugin ved navn Google Chrome Frame.

Hvis du indsætter følgende kode på din side vil der ske små mirakler.

I HEAD:

```
<meta http-equiv="X-UA-Compatible" content="chrome=1">
```

I BODY:

```
<script type="text/javascript"
  src="http://ajax.googleapis.com/ajax/libs/chrome-frame/1/CFInstall.min.js"></script>
<style>
/*
  CSS rules to use for styling the overlay:
  .chromeFrameOverlayContent
  .chromeFrameOverlayContent iframe
  .chromeFrameOverlayCloseBar
  .chromeFrameOverlayUnderlay
*/
</style>
<script>
  // You may want to place these lines inside an onload handler
  CFInstall.check({
    mode: "overlay",
    destination: "http://www.waikiki.com"
  });
</script>
```

Helt enkelt vil Internet Explorer loade Google Chrome Frame i Internet Explorer, hvis den er installeret, og hvis den ikke er, vil besøgende blive sendt til en side hvor de kan installere den (google.com/chromeframe). Et sådant plugin kræver ikke administratorrettigheder på computeren for at kunne installeres.

Læs mere på code.google.com/intl/da/chrome/chromeframe og html5rocks.com/en/tutorials/google-chrome-frame



Afrundede hjørner



Endnu en ting der kræver billeder i 'skiver'... – indtil CSS3!

Opret en ny CSS definition med følgende kode:

```
section, aside, nav {  
-webkit-border-radius:10px;  
-moz-border-radius:10px;  
-o-border-radius:10px;  
-ms-border-radius:10px;  
border-radius:10px;  
}
```



Hjørnerne kan naturligvis afrundes mere eller mindre ved at justere 10px til fx 5px eller 15px...

Det er også muligt at tilpasse hjørnerne individuelt, fx:

```
border-top-right-radius:10px;  
border-bottom-left-radius:10px;
```

Skal afrundede hjørner bruges på flere forskellige div'er m.v. kan du alternativt bruge en class:

```
#roundcorners {  
-webkit-border-radius:10px;  
-moz-border-radius:10px;  
-o-border-radius:10px;  
-ms-border-radius:10px;  
border-radius:10px;  
}
```

– og påtvinge denne class i HTML-koden:

```
<div id="apdiv1" class="roundcorners">Bla bla bla</div>
```





Background property



Med CSS3 er det nu muligt at styre baggrunde, samt indsætte flere baggrunde i samme område/ side/div. – baggrunde kan placeres overlappende, skalleres og med en defineret afstand og gentagelse. Et eksempel:

```
apdiv1 {
background:url(smiley.gif) top left no-repeat,
url(sqorange.gif) bottom left no-repeat,
url(sqgreen.gif) bottom right no-repeat;
}
```

| | |
|-----------------------|--|
| background-color | Baggrundsfarve: #fff... |
| background-position | Baggrundsbilledets position: bottom, left, center... |
| background-size | Baggrundsbilledets størrelse: auto, contain, cover, 60% osv... NY |
| background-repeat | Bestemmer om baggrundsbilledet skal gentages: repeat, repeat-y, repeat-x, norepeat |
| background-origin | Baggrundsbilledets afgrænsning: border-box, content-box, padding-box NY |
| background-clip | Baggrundsfarvens afgrænsning: border-box, padding-box NY |
| background-attachment | Bestemmer om baggrundsbilledet skal scrolle eller fryses fast når siden scroller: fixed, local, scroll |
| background-image | Specificerer ét eller flere baggrundsbilleder: url(images/back.jpg) |

Vi skal nu tilpasse baggrunden, så den fylder vinduet ud, og lægger fast når vi scroller:

```
body {
background-color: #FFF;
background:url(images/orange.jpg);
background-size: cover;
background-position: top left;
background-attachment:fixed;
}
```





RGB-farver og gennemsigtighed



"Lad mig se en gange; RGB-farven 25,88,133 på en webside hedder #195885, og det giver mening fordi.... Aaaaargh!!!"

I CSS3 hedder RGB-farven 25,88,133:
`color: rgb(25,88,133)`

– og det bliver bedre endnu, for vi kan tilføje alpha (opasitet):

`background-color: rgba(25,88,133,0.5)`

Nu er den farvelagte dims (div, tabel, tekst...) 50% gennemsigtig!

Opret en ny css definition:

```
section, nav, header {
    background-color:#069;
    background-color:rgba(0,102,153,0.7);
}
```



Nu står farven to gange; først den 'gamle' definition, og derefter den nye. Dette skyldes igen at det sidste forståede udsagn eksekveres. På denne måde har browsere, der ikke kan eksekvere rgba-koden, en 'fall-back' kode.

Find også `#aside1` { og tilføj den nye farve under `background-color:#c8c8c8;`
`background-color:rgba(200,200,200,0.8);`



Husk at dette gør baggrunden halvgennemsigtig. Hvis det hele – incl. teksten – skal gøres gennemsigtigt skal du bruge en anden kode, fx:

```
opacity:0.8; /* moz, o, webkit */
filter:alpha(opacity=80); /* ms */
```

Forestil dig fx en div der skal være 50% opasitet indtil musen holdes over, hvor den skifter til 90% opasitet:

```
#apdiv1 {
    opacity:0.5;
    filter:alpha(opacity=50);
}
#apdiv1:hover {
    opacity:0.9;
    filter:alpha(opacity=90);
}
```

Det samme kan du nøjes med at gøre på baggrunden:

```
#apdiv1 {
    background-color:#c8c8c8;
    background-color:rgba(200,200,200,0.5);
}
#apdiv1:hover {
    background-color:#c8c8c8;
    background-color:rgba(200,200,200,0.9);
}
```

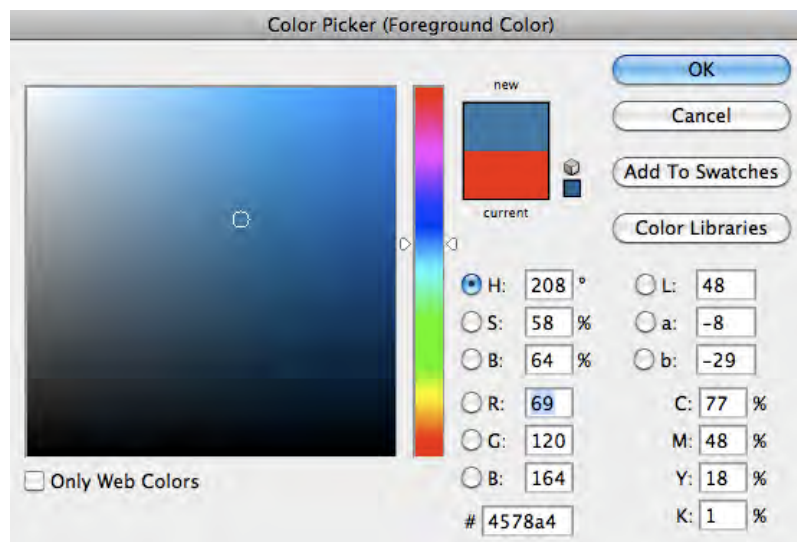



HSL-farver og gennemsigtighed



Som alternativ til RGB kan du også bruge HSL og HSLA (Hue/Saturation/Luminance/Alpha), fx: `background-color: hsla(200,100%,60%,0.7);`

Bemærk: HSL er det samme som Photoshop kalder HSB (Hue/Saturation/Brightnes).





Farveforløb



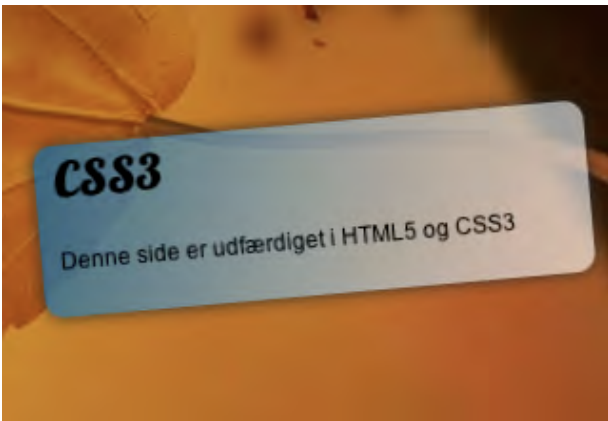
Find `#aside2` { og tilføj et farveforløb:

```
background: -webkit-gradient(linear, left top, right bottom, from(#006699), to(#c8c8c8));  
background: -moz-gradient(linear, left top, right bottom, from(#006699), to(#c8c8c8));  
background: -o-gradient(linear, left top, right bottom, from(#006699), to(#c8c8c8));  
background: -ms-gradient(linear, left top, right bottom, from(#006699), to(#c8c8c8));  
background: gradient(linear, left top, right bottom, from(#006699), to(#c8c8c8));
```

Du kan naturligvis også bruge `rgba`- eller `hsla`-farver:

```
background: gradient(linear, left top, right bottom,  
from(rgba(0,102,153,0.8)), to(rgba(255,255,255,0.6)));
```

– prøv at rette koden til ovenstående, så du kan se hvordan det ser ud.



I resten af materialet nøjes vi med `-webkit`, og beder dig huske på at der skal suppleres med `-moz`, `-o`, `-ms` og den originale kode



Skygger



Skriv en ny CSS definition:

```
header, nav, section, aside, footer {  
-webkit-box-shadow: 0px 0px 10px #333;  
-moz-box-shadow: 0px 0px 10px #333;  
-o-box-shadow: 0px 0px 10px #333;  
-ms-box-shadow: 0px 0px 10px #333;  
box-shadow: 0px 0px 10px #333;  
}
```



Første `0px` er forskydning vandret (mod højre).

Anden `0px` er forskydning lodret (nedad).

`10px` er skyggens udtonede størrelse.

`#333` er en mørk grå. – du kan naturligvis også bruge en rgba-farve:

```
box-shaddow: 0px 0px 10px rgba(51,51,51,0.8);
```

På tekst bruges en anden fremgangsmåde:

Vælg `h1` { og tilføj følgende kode:

```
text-shadow: 3px 3px 6px #000;
```

(`text-shaddow` er ikke ny, og kræver derfor ikke `-webkit-` m.fl.)





Tekst i spalter



Tekst i spalter har ikke rigtigt været en mulighed i HTML4/XHTML, medmindre vi fordelte teksten i tabeller eller div manuelt, hvilket var et helvede når teksten senere blev redigeret.

Med koden `column-count:2;` kan vi fordele teksten i den første artikel i to spalter, og med koden `column-gap:10px;` kan vi give spalterne en afstand på 10 pixels.

I HTML koden har vi givet første `<article>` kode tilføjelsen `id="art1"`:

```
<article id="art1">
```

Derfor kan vi oprette en ny CSS definition:

```
#art1 {
-webkit-column-count:2;
-moz-column-count:2;
column-count:2;
-webkit-column-gap:10px;
-moz-column-gap:10px;
column-gap:10px;
}
```



Det er også muligt at tilføje en lodret skillestreg imellem spalterne. Her et eksempel på en grim én:

```
-webkit-column-rule: #0F0 dashed 3px;
```

New English

The European Commission has just announced an agreement whereby English will be the official language of the EU (European Union) rather than German which was the other possibility. As part of the negotiations, Her Majesty's Government conceded that English spelling had some room for improvement and has accepted a 5 year phase-in plan that would be known as "Euro-English".

In the first year, "s" will replace the soft "c". Certainly, this will make the sivil servants jump with joy. The hard "c" will be dropped in favour of the "k". This should klear up konfusion and keyboards kan have one less letter.

There will be growing publik enthusiasm in the sekond year, when the troublesome "ph" will be replaced with "f". This will make words like "fotograf" 20% shorter.

In the 3rd year, publik akseptanse of the new spelling kan be ekspekted to reach the stage where more komplikated changes are possible. Governments will enkorage the removal of double letters, which have always ben a deterrent to akurate speling. Also, al wil agre that the horrible mes of the silent "e"s in the language is disgraceful, and they should go away.

By the fourth year, peopl wil be reseptiv to steps such as replasing "th" with "z" and "w" with "v". During ze fifz year, ze unesesary "o" kan be dropd from vords kontaining "ou" and similar changes vud of kors be aplid to ozer kombinations of letters.

After zis fifz yer, ve vil hav a reli sensibl riten styl. Zer vil be no mor trubl or difikultis and evrivun vil find it ezi to understand ech ozer. Ze drem vil finali kum tru! And zen ve vil tak over ze world!

Speaking of Spelling

Aoccdmrig to a rscheearch at an Elingsh uinervtisy, it deosn't mitaer in waht oredr the ltteers in a wrod are, the olny ipmoetnt tihng is taht frist and lsat ltteer is at the rghit pclae. The rset can be a toatl mses and you can sitti raed it wouthit porbelm. Tihis is bcuseae we do not raed evvey lteer by itslef but the wrod as a wlohe.



Animation



Flash® er blevet et problembarn, når Apple på deres 'iGadgets' ikke vil tillade flash-afspillere... Men hvorfor også bruge flash når browseren fint selv kan eksekvere animationer skrevet i CSS?

Først skal vi kigge på en kode der fortæller en enkelt historie:

```
#apdiv1 {background-color: #fff;}
#apdiv1:hover {background-color:#000;}
```

En Div med id="apdiv1" har hvid baggrundsfarve, men får sort baggrundsfarve når musen holdes over. Enkelt.

To nye koder – Transition og Transform – gør det muligt at ændre placering, vinkel og størrelse. Nu får vi mere komplekse muligheder.

Find definitionen `nav {` og tilføj følgende kode i bunden:

```
-webkit-transition: all 1s ease-in-out;
```

`all` = Alle dimensioner. Alternativt: `width`, `height`, `none`

`1s` = 1 sekundt

`ease-in-out` = acceleration og deceleration. Alternativt: `linear`, `ease`, `ease-in`, `ease-out`



Nu skal vi beskrive hvilken position, størrelse m.v. denne transition skal føre til. Opret derfor en ny definition ved navn `nav:hover`:

```
nav:hover {
    left:70px;
    background-color: #f30;
    -webkit-transform: rotate(360deg) scale(2);
}
```



– prøv det. Det er sjov og ballade, og værd at huske på et godt dansk designprincip: *Jo flere effekter, jo grimmere er det!*

Nu skal vi prøve en lidt mere kompleks animation. En Div inde i en anden Div, hvor den 'indre' Div bevæger sig når musen holdes over den 'ydre' Div.

Den 'ydre' og aktiverende Div er rummet, og den indre, og aktiverede, Div er rumskibet.

Find koden `div.rocket {` og skriv følgende kode under:

```
-webkit-transition: all 5s ease-in;
```

(Alle dimensioner, 3 sekunder, accelerer)

Find koden `div.rocket img {` og skriv følgende kode under:

```
-webkit-transition: all 2s ease-in-out;
```

(Alle dimensioner, 2 sekunder, decelerer)

Find koden `#outerspace:hover div.rocket {` og skriv følgende kode:

```
-webkit-transform: translate(1500px,-550px);
```

(Flyt elementet 1500 pixels og -550 pixels)

Og find koden `#outerspace:hover div.rocket img {` og skriv følgende kode:

```
-webkit-transform: rotate(70deg);
```

(Roter objektet 70 grader)





Keyframe animation



Find definitionen `#logo {` og skriv følgende kode under:

`-webkit-border-radius:50px;`

(skaber runde hjørner på 1/4 af div'ens størrelse; altså en cirkel)

`-webkit-animation-name:logoanimation;`

(Et navn vi selv finder på, men som skal refereres)

`-webkit-animation-delay:3s;`

(Forsinkelse før animationen starter)

`-webkit-animation-duration:6s;`

(Animationens varighed)

`-webkit-animation-iteration-count:2;`

(Dét antal gange animationen skal køre. "infinite" får den til at loope uden ende)

`-webkit-animation-timing-function:ease-in-out;`

(Accelererer og decelerer)



Herefter skrives en ny funktion ind i CSS:

`@-webkit-keyframes logoanimation {`

`0% {background:rgba(0,0,0,0.0); right:50px; top:15px; -webkit-transform:rotate(0deg);}`

`25% {background:rgba(0,102,153,0.7); right:50px; top:400px; -webkit-transform:rotate(0deg);}`

`50% {background:rgba(255,51,0,0.7); right:400px; top:400px; -webkit-transform:rotate(180deg);}`

`75% {background:rgba(255,255,255,0.7); right:400px; top:15px; -webkit-transform:rotate(0deg);}`

`100% {background:rgba(0,0,0,0.0); right:50px; top:15px; -webkit-transform:rotate(0deg);}`



Procenterne henviser til varigheden, som i dette tilfælde jo er 6 sekunder. Dermed er 50% altså efter 3 sekunder, hvor baggrundsfarven er 0,0,255 (blå) og 0.7 alpha (70% opasitet), Div'en befinder sig 400 pixels fra højre og 400 pixels fra toppen, og rotationen er 180 grader (den vender på hovedet).

I stedet for 0, 25, 50, 75 og 100 procent kan vi sagtens nøjes med 0, 50 og 100, eller udvide med 0, 5, 10, 15, 20, 25.....



3D transformation



I CSS3 kan vi benytte rotation og vridning (“skew”) på X, Y og Z-aksen i forbindelse med [transform](#).

Prøv at finde `#aside1` definitionen og skriv følgende kode:
`-webkit-transform:rotateX(35deg) rotateY(10deg) skew(20deg,0deg);`

En anden tilgang er `rotate3d`, som giver mulighed for at tilpasse x, y, z og vinkel.
Erstat ovenstående kode med denne:
`-webkit-transform:rotate3d(1, 1, 0, 55deg);`



Prøv evt. at sætte koderne på én ad gangen, og se resultatet i browseren.

| Kode | Forklaring |
|--|---|
| <code>transform</code> | Lægger en 3D transformation på et element |
| <code>transform-origin</code> | Ændrer positionen af et transformeret element |
| <code>transform-style</code> | Specificerer hvordan et indbygget element er renderet i 3D |
| <code>perspective</code> | Specificerer 3D elementers perspektiv |
| <code>perspective-origin</code> | Specificerer hvordan bunden af et 3D element fremtræder |
| <code>backface-visibility</code> | Definerer hvorvidt et element skal være synligt når det ikke vender mod skærmen |
| <code>matrix3d</code> (n,n,n,n,n,n,n,n,n,n, n,n,n,n,n) | Definerer en 3D transformation, ved at benytte en 4x4 matrix på 16 værdier |
| <code>translate3d(x,y,z)</code> | Flytter hhv. x, y og z |
| <code>translateX(x)</code> | Flytter x |
| <code>translateY(y)</code> | Flytter y |
| <code>translateZ(z)</code> | Flytter z |
| <code>scale3d(x,y,z)</code> | Skalerer x, y og z i 3D |
| <code>scaleX(x)</code> | Skalerer x |
| <code>scaleY(y)</code> | Skalerer y |
| <code>scaleZ(z)</code> | Skalerer z |
| <code>rotate3d(x,y,z,angle)</code> | 3D ved at tilpasse x, y, z og vinkel |



[rotateX\(angle\)](#)

Roterer x

[rotateY\(angle\)](#)

Roterer y

[rotateZ\(angle\)](#)

Roterer z

[perspective\(n\)](#)

Perspektiv

Læs evt. mere på cubiq.org/building-a-pure-css-3d-city og w3schools.com/css3/css3_3dtransforms.asp





Canvas



Canvas er opfundet for at skabe rum til alle de funktioner der ikke understøttes naturligt i HTML, navnlig JavaScripts.

Canvas bruges til fx:

- Diagrammer
- Animationer
- Grafer
- Tegneapplikationer
- At omgå begrænsninger i CSS

Canvas er et lærred der giver mulighed for at placere tegninger på pixel-niveau, og beziér-kurver (vektorgrafik). – men brugt på den rigtige måde kan det bruges til nogle ret imponerende ting.

I HTML koden indsættes helt enkelt koden `<canvas></canvas>`, evt: `<canvas id="...">`. Derefter indsættes en kode der henter et JavaScript med de ønskede funktioner, fx: `<script src="drawingtools.js"></script>`

Vi skal her blot se på tre forskellige måder canvas kan udnyttes til tegneapplikationer.

Kig under overskriften Canvas, og prøv de forskellige tegneapplikationer: *Skitse*, *Farvelæg* og *Helt tegneprogram*. Hvis du er interesseret i at dykke ned i JavaScriptet kan du se de vedhæftede .js filer i opgavemappen.



Læs mere om Canvas HTML og CSS: w3schools.com/html5/tag_canvas.asp

Læs mere om udviklingen af en tegneapplikation: williammalone.com/articles/create-html5-canvas-javascript-drawing-app





SVG



SVG står for *Scalable Vector Graphics*.

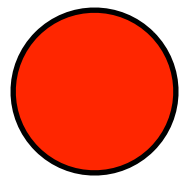
SVG definerer grafik i XML format.

SVG kan bruges til at indsætte en vektortegning i ren kode. I modsætning til Canvas kan SVG ikke bruges til bruger-interaktion.

Det er muligt at lave en tegning i Adobe Illustrator og vælge Gem Som > SVG > Show SVG Code

En SVG kode kan se sådan ud:

```
<svg>
  <circle cx="100" cy="50" r="40" stroke="black"
    stroke-width="2" fill="red" />
</svg>
```



Men vi skal prøve med en langt mere kompleks kode.

1. Åben dokumentet *html5badge_svg.txt* i Dreamweaver eller Notepad

```
<svg viewBox="0 0 512 512">
  <path fill="#E34F26" d="M71,460 L30,0 481,0 440,460 255,512"/>
  <path fill="#EF652A" d="M256,472 L405,431 440,37 256,37"/>
  <path fill="#EBEBEB" d="M256,208 L181,208 176,150 256,150 256,94 255,94 114,94 115,109
129,265 256,265z M256,355 L255,355 192,338 188,293 158,293 132,293 139,382 255,414
256,414z"/>
  <path fill="#FFF" d="M255,208 L255,265 325,265 318,338 255,355 255,414 371,382 372,372
385,223 387,208 371,208z M255,94 L255,129 255,150 255,150 392,150 392,150 392,150 393,138
396,109 397,94z"/>
</svg>
```



2. Kopier hele teksten
3. Find koden `` i *index.html* og erstat den med den kopierede kode

Læs mere om SVG på tutorialspoint.com/html5/html5_svg.htm





Formularer



HTML5 indeholder adskillige nye faciliteter i formularer.

Lad os lige prøve et par af dem. Der findes en enkelt formular i bunden af siden. Ret koden med de viste koder (i forskellige farver):



```

<form action="xxx.asp" autocomplete="on">
  Fornavn:<input type="text" name="fnavn" autofocus="autofocus" placeholder="Fornavn">&nbsp;
  Efternavn:<input type="text" name="enavn" placeholder="Efternavn"><br>
  E-mail: <input type="email" name="email" autocomplete="off"
placeholder="navn@firma.dk">&nbsp;
  Antal kager<input type="number" name="kager" step="3"><br>
  <input type="submit" />
</form>

```

Nye koder til <form>-koden:

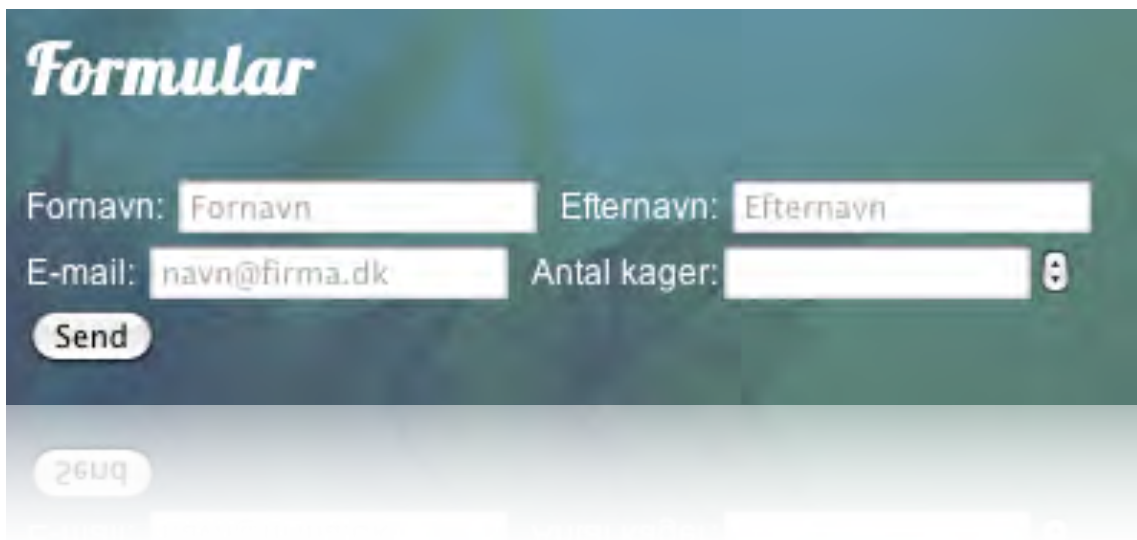
| | | |
|--------------|--|---|
| autocomplete | <form autocomplete="on"> | Formularen tillader at browseren bruger gemte data, hvis "off" tillades det ikke. |
| novalidate | <form formnovalidate="formnovalidate"> | Formularen sendes uden at data valideres før afsendelse. |

Nye koder til <input>-felter:

| | | |
|----------------|---|--|
| autocomplete | <input autocomplete="on"> | Feltet tillader at browseren bruger gemte data, hvis "off" tillades det ikke. |
| autofocus | <input autofocus="autofocus"> | Cursoren står i dette felt når siden loades |
| form | <input form="form1"> | Når flere formularer bruges på samme side kan felter 'tagges' til en specifik form |
| formaction | <input type="submit" formaction="xxx.asp"> | Vælger hvilken URL formularen skal sendes til. Kan benyttes ved flere forsendelsesmuligheder: "Send som ..." |
| formenctype | <input type="submit" formenctype="multipart/form-data"> | Bestemmer hvilken encoding der skal benyttes ved forsendelse (kun ved "post") |
| formmethod | <input type="submit" formmethod="post"> | Bestemmer hvilken forsendelsesform der skal bruges på en alternativ send-knap |
| formnovalidate | <input formnovalidate="formnovalidate"> | Formularen sendes uden at data i dette felt valideres før afsendelse. |



| | | |
|------------------|--|--|
| formtarget | <code><input type="submit" formtarget="_blank"></code> | Bestemmer hvor formularen skal sende brugeren hen: <code>_blank</code> , <code>_self</code> , <code>_parent</code> , <code>_top</code> , <code>ramenavn</code> |
| height and width | <code><input type="image" src="img_submit.gif" alt="Submit" width="48" height="48"></code> | Højde og bredde når billede anvendes som knap |
| multiple | <code><input type="file" name="img" multiple="multiple"></code> | Tillader brugeren at indsætte flere værdier i ét felt, fx vedhæftninger |
| placeholder | <code><input placeholder="Efternavn"></code> | Sætter en standardtekst ind i et tekstfelt i svag grå, så brugeren kan se hvad der forventes. |
| required | <code><input required="required"></code> | Når et felt SKAL udfyldes |
| step | <code><input type="number" name="points" step="3"></code> | Sætter intervaller på type="number" funktionen |





GeoLocation



Geolocation (automatisk placering på et kort) giver ikke rasende meget mening på den stationære computer jeg sidder foran dag efter dag. Men på en iPad eller smartphone, når jeg står et sted i Rom, kan det være en praktisk del af en mobil webside eller HTML5 App'. Derfor er GeoLocation et glimrende eksempel på en jQuery der gør HTML5 spændende på bærbare enheder.

Først skal vi knytte jQuery til siden. Derfor indsætter vi følgende kode i HEAD:

```
<script src="js/jquery-1.4.2.min.js"></script>
```



Bemærk at vi i HTML5 ikke længere skal beskrive arten af script!

Det er på dette kursus ikke meningen at vi skal sidde og skrive lange koder, så derfor åbner vi filen geolocation.txt i opgavemappen og copy-paster teksten ind i HEAD, under den forgående kode.

Det kan dog betale sig at nærlæse kommentarerne i koden, som forklarer hvilke dele af koden der 'vækker' hvilken funktion. Kommentarer i scripts begynder med //

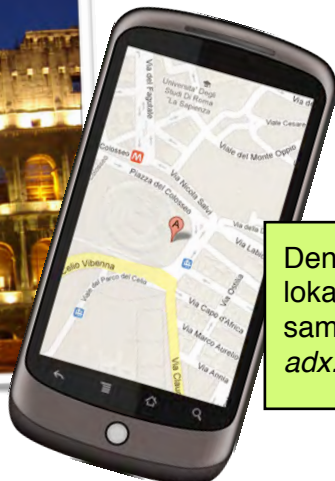
Til sidst skal vi indsætte selve funktionen på siden:
Skriv følgende kode lige over </section>:



```
<div>
  <button id="btnInit" >Find mig!</button>
  <button id="btnStop" >Stop positionering</button>
</div>
```

Husk på at stationære computere på et kabelnetværk vil vise nærmeste forbindelse til internettet som *location*. – potentielt flere kilometer væk fra hvor du sidder...

Positionering kan naturligvis ikke kun bruges sammen med kort, men også i forbindelse med søgning på trafik-, hotel- og personsøgninger.



Denne funktion fungerer ikke ved lokal test på computeren, men den samme kode kan ses i funktion på adx.dk/html5/geolocation.html



AppCache



Tidligere var websites beregnet til at blive hentet fra internettet hver gang vi skulle se dem, med mulighed for at gemme særlige præferencer i små cookies. Cache var en forældet idé, fra dengang internettet var dyrt tilgængeligt. Men på iPads og smartphones er cache kommet i fokus igen, og W3C har udviklet en kodefunktion til at beslutte hvad der skal gemmes i cache og hvad der skal hentes hver gang.

Vi skal udvide vores `<html>` tag så det indeholder en reference til en AppCache fil.

```
<html manifest="offline.appcache">
```



Herefter skal vi oprette en ny tekstfil i Dreamweaver (File > New > Other > Text) eller Notesblok.

I filen skriver vi (udelad bemærkninger i parantes):

```
CACHE MANIFEST
# 2012-03-08 v1.0.0
(Dato og version)
images/logo.jpg
images/bg.jpg
js/jquery-1.4.2.min.js
*
```

(Filer der skal lagres i cache. * betyder "og alle andre")

NETWORK:

```
index.html
```

(Filer der skal hentes fra nettet hver gang)

FALLBACK:

```
offline.html
```

(Erstatter filer der skal hentes fra nettet, hvis forbindelsen er afbrudt. Fx en 'skrabet' kopi af index.html)

Det kan umiddelbart ikke ses at et website indeholder en appcache funktion, men hentes det i Firefox vil browseren spørge om tilladelse til at gemme data på computeren.





App's



Med AppCache koderne på plads er et website faktisk en færdig App. Brugeren skal blot gemme en genvej til webadressen imellem deres app'-ikoner på deres smartphone. – men måske er dét for meget forlangt...

For at skabe en lettilgængelig App' skal vi stadig pakke den ind i en særlig 'ramme', med et tilhørende ikon, der er forskellig for hver type smartphone/tablet, selvom indholdet af App'en er den samme HTML5 kode. – *Andoid, iOS, Windows 8, BlackBerry...*

Dette kræver et SDK (*Software Development Kit*) for hver platform, og optimalt også en API (*Application Programming Interface*) til at teste app'en inden distribution. Disse kan hentes på fx sencha.com/products/touch eller phonegap.com

Med Dreamweaver 5.5 (eller senere) er det muligt at pakke et HTML5 website som en App', og teste den:

1. Opret et Site til din app'.
2. Byg din app' som et lille website, med tilhørende CSS, JavaScripts, jQueryes, billeder etc.
3. Site > Mobile Applications > Configure Application Framework
4. Følg anvisningerne for at installere *Android SDK (Easy Install)* og *iOS Developer Tools* (Kun på Mac!)
5. Site > Mobile Applications > Application Settings > udfyld formularen med *Budle ID, Application Name, Author Name, Application PNG* (ikon), *Startup Screen PNG* (startbillede) og *Taget Path* (hvor skal app'en gemmes)
6. Site > Mobile Applications > Build and Emulate > Android/iOS (iOS kun på en Mac!)

Med Dreamweaver 5.0 eller tidligere kan dette ikke lade sig gøre. Benyt i stedet en alternativ service fra fx phonegap.com

Husk at HTML5 app's kan distribueres uden om Apples App Store, men vil du distribuere dem dér koster det US\$ 99,- at blive registreret som udvikler, hvorefter dine app's skal godkendes. På Android Market skal du blot registrere dig gratis.

En alternativ metode er at benytte en online SDK, med de begrænsninger dette medfører:

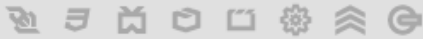
- appsgeyser.com – gratis, kun til Android, indhold skal først være uploadet til en webserver
- widgetbox.com – 14 dage gratis, derefter mod betaling, til Android og iPhone

Se også lungojs.com – et HTML framework til touchscreen app's.





Links om HTML5, CSS3 og jQuery



- w3.org/html
- w3.org/html/logo
- w3schools.com/html5
- html5rocks.com/en
- adobe.com/devnet/html5.html
- google.com/webfonts
- jquery.com
- html5demos.com
- phonegap.com
- atomurl.net/qrcode
- css3.info
- apple.com/html5



Effects



Connectivity



Device Access



Multimedia



Offline Storage



Performance



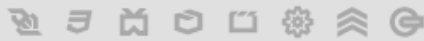
Semantics



Styling



Kodeoversigt (engelsk)



| Tag | Description |
|-----------------------|---|
| <!--...--> | Defines a comment |
| <!DOCTYPE> | Defines the document type |
| <a> | Defines a hyperlink |
| <abbr> | Defines an abbreviation |
| <acronym> | Not supported in HTML5 |
| <address> | Defines contact information for the author/owner of a document/article |
| <applet> | Not supported in HTML5 |
| <area> | Defines an area inside an image-map |
| <article> NEW | Defines an article |
| <aside> NEW | Defines content aside from the page content |
| <audio> NEW | Defines sound content |
| | Defines bold text |
| <base> | Specifies the base URL/target for all relative URLs in a document |
| <basefont> | Not supported in HTML5 |
| <bdi> NEW | Isolates a part of text that might be formatted in a different direction from other text outside it |
| <bdo> | Overrides the current text direction |
| <big> | Not supported in HTML5 |
| <blockquote> | Defines a section that is quoted from another source |
| <body> | Defines the document's body |
| | Defines a single line break |
| <button> | Defines a clickable button |
| <canvas> NEW | Used to draw graphics, on the fly, via scripting (usually JavaScript) |
| <caption> | Defines a table caption |
| <center> | Not supported in HTML5 |
| <cite> | Defines the title of a work |
| <code> | Defines a piece of computer code |
| <col> | Specifies column properties for each column within a <colgroup> element |
| <colgroup> | Specifies a group of one or more columns in a table for formatting |
| <command> NEW | Defines a command button that a user can invoke |
| <datalist> NEW | Specifies a list of pre-defined options for input controls |
| <dd> | Defines a description of an item in a definition list |
| | Defines a text that has been deleted from a document |
| <details> NEW | Defines additional details that the user can view or hide |
| <dfn> | Defines a definition term |
| <dir> | Not supported in HTML5 |
| <div> | Defines a section in a document |
| <dl> | Defines a definition list |
| <dt> | Defines a term (an item) in a definition list |
| | Defines emphasized text |
| <embed> NEW | Defines a container for an external application or interactive content (a plug-in) |



| Tag | Description |
|-------------------------|--|
| <fieldset> | Groups related elements in a form |
| <figcaption> NEW | Defines a caption for a <figure> element |
| <figure> | Specifies self-contained content |
| | Not supported in HTML5 |
| <footer> NEW | Defines a footer for a document or section |
| <form> | Defines an HTML form for user input |
| <frame> | Not supported in HTML5 |
| <frameset> | Not supported in HTML5 |
| <h1> to <h6> | Defines HTML headings |
| <head> | Defines information about the document |
| <header> NEW | Defines a header for a document or section |
| <hgroup> NEW | Groups heading (<h1> to <h6>) elements |
| <hr> | Defines a thematic change in the content |
| <html> | Defines the root of an HTML document |
| <i> | Defines a part of text in an alternate voice or mood |
| <iframe> | Defines an inline frame |
| | Defines an image |
| <input> | Defines an input control |
| <ins> | Defines a text that has been inserted into a document |
| <keygen> NEW | Defines a key-pair generator field (for forms) |
| <kbd> | Defines keyboard input |
| <label> | Defines a label for an input element |
| <legend> | Defines a caption for a <fieldset>, <figure>, or <details> element |
| | Defines a list item |
| <link> | Defines the relationship between a document and an external resource (most used to link to style sheets) |
| <map> | Defines a client-side image-map |
| <mark> NEW | Defines marked/highlighted text |
| <menu> | Defines a list/menu of commands |
| <meta> | Defines metadata about an HTML document |
| <meter> NEW | Defines a scalar measurement within a known range (a gauge) |
| <nav> NEW | Defines navigation links |
| <noframes> | Not supported in HTML5 |
| <noscript> | Defines an alternate content for users that do not support client-side scripts |
| <object> | Defines an embedded object |
| | Defines an ordered list |
| <optgroup> | Defines a group of related options in a drop-down list |
| <option> | Defines an option in a drop-down list |
| <output> NEW | Defines the result of a calculation |
| <p> | Defines a paragraph |
| <param> | Defines a parameter for an object |
| <pre> | Defines preformatted text |
| <progress> NEW | Represents the progress of a task |
| <q> | Defines a short quotation |
| <rp> NEW | Defines what to show in browsers that do not support ruby annotations |
| <rt> NEW | Defines an explanation/pronunciation of characters (for East Asian Typography) |
| <s> | Defines text that is no longer correct |



| Tag | Description |
|------------------------|---|
| < samp > | Defines sample output from a computer program |
| < script > | Defines a client-side script |
| < section > NEW | Defines a section in a document |
| < select > | Defines a drop-down list |
| < small > | Defines smaller text |
| < source > NEW | Defines multiple media resources for media elements (< video > and < audio >) |
| < span > | Defines a section in a document |
| < strike > | Not supported in HTML5 |
| < strong > | Defines important text |
| < style > | Defines style information for a document |
| < sub > | Defines subscripted text |
| < summary > NEW | Defines a visible heading for a < details > element |
| < sup > | Defines superscripted text |
| < table > | Defines a table |
| < tbody > | Groups the body content in a table |
| < td > | Defines a cell in a table |
| < textarea > | Defines a multiline input control (text area) |
| < tfoot > | Groups the footer content in a table |
| < th > | Defines a header cell in a table |
| < thead > | Groups the header content in a table |
| < time > NEW | Defines a date/time |
| < title > | Defines a title for the document |
| < tr > | Defines a row in a table |
| < track > NEW | Defines text tracks for media elements (< video > and < audio >) |
| < tt > | Not supported in HTML5 |
| < u > | Not supported in HTML5 |
| < ul > | Defines an unordered list |
| < var > | Defines a variable |
| < video > NEW | Defines a video or movie |
| < wbr > NEW | Defines a possible line-break |

Kilde: http://www.w3schools.com/html5/html5_reference.asp

Se også <http://www.w3schools.com/cssref/default.asp> for komplet CSS3 kodereference



Ekstraopgaver – hvis du keder dig...



Animation

Se adx.dk/html5/flip.html og prøv selv at konstruere en side med forskellige figurer der reagerer ved at dreje rundt, på hovedet etc. ved *hover*, samt figurer der bevæger sig ved hjælp af en *keyframe animation*.

Prøv evt. at lave en side med figurer der skifter form, farve og størrelse. Se adx.dk/html5/size.html. Bemærk at siden indeholder en QR-kode indsat som SVG, hvilket skal opfattes som en udfordring.

Hvis du trænger til mere animations-udfordring kan du tegne en flyver der bevæger sig hen over siden, imens propellen drejer rundt. Se fx adx.dk/html5/flyvendepolse.html

Canvas

Læs og prøv øvelserne på w3schools.com/html5/html5_canvas.asp, og opret herefter et *canvas* på en ny side, og tegn et møblement eller lign.



Drop Down Menu

Vi kender alle sammen den traditionelle JavaScript menu, vandret, placeret i toppen af siden, med drop down undermenuer. Men mængden af JavaScripts/jQueries der kan loades på én side er begrænset. Så kan man lave den traditionelle drop down menu alene af HTML og CSS? Ja... Måske ikke helt så 'smooth', men det *kan* lade sig gøre. Kig og på selv: adx.dk/html5/menu.html