



Layout uden tabeller - muligheder og begrænsninger

Tabeller er meget udskældte for tiden. Mange ønsker at erstatte et layout bestående af tabeller med serier af <div> bokse styret med CSS. Det kan godt lade sig gøre, men der er nogle begrænsninger. Her er fire opskrifter.

Skrevet den **03. Feb 2009** af **mik789** | kategorien **(D)HTML / CSS** | ★★★★★

Revideret d.30-12-2005

Layout uden tabeller

Lad os slå fast med det samme: tabeller kan noget helt specielt og meget nyttigt. Med tabeller kan man stille bokse ved siden af hinanden og få dem til at flugte med højden af den boks der er mest indhold i. Man behøver ikke tænke på højde. Boksene aligner sig selv ud fra den højeste boks. Den egenskab er der ofte brug for i et sidelayout.

Tabeller

Traditionelt bruges tabeller til to ting:

- 1) Til det de er tænkt som, nemlig tabeller af flere rækker og kollerter med data der alignes ens.
- 2) Til at lave sidelayout ved at opdele siden i få store felter, f.eks. det populære trefløjs-layout.

Til første opgave ville det være tosset ikke at bruge tabeller. Til den anden opgave opstår der nemt problemer. Så snart der fyldes mere komplekst indhold i de enkelte felter - nemlig ved hjælp af andre tabeller - får man hurtigt et uoverskueligt design med mange tabeller inden i hinanden.

Så lad os se på hvordan man laver et trefløjet layout uden per automatik at gribe til tabellerne. Desværre må man dog erkende, at så forskelligt som browserne understøtter CSS, er det ikke helt enkelt, og der kan ikke opnås samme bagudkompatibilitet som med et traditionelt tabel-layout.

Vi vil lave et design med en central boks der består af en "overligger"-boks og tre bokse der flugter ved siden af hinanden under denne. Altså en boks svarende til:

```
<table cellpadding="0" cellspacing="0" border="0" width="800">
  <tr>
    <td colspan="3">Topfelt</td>
  </tr>
  <tr>
    <td width="200">Venstre felt</td>
    <td width="300">Midterfelt</td>
    <td width="300">Højre felt</td>
  </tr>
</table>
```

Forberedelse

Inden vi går i gang med at erstatte dette med div'er og CSS, skal forudsætningerne på plads: For det første

en korrekt DTD ("doctype declaration") i starten af dokumentet. Hvis siden skal vises i HTML 4.01 transitional, f.eks., ser DTD'en sådan ud:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

```
"http://www.w3.org/TR/html4/loose.dtd">[/div]
```

Vær opmærksom på at der kræves særskilt DTD for en frameset fil. Se de forskellige muligheder i W3C's liste:

<http://www.w3.org/QA/2002/04/valid-dtd-list.html>

Desværre er Internet Explorer indrettet sådan, at hvis ikke denne DTD står som det aller første i dokumentet - uden noget tegn foran - bliver den sat ud af spillet.

Dernæst er det vigtigt hvis vi skal angive relativ højde på de enkelte elementer, at der er noget at udregne den af. Derfor skal der først angives en højde for elementerne html og body i stylesheetet. Altså:

```
html {
    height: 100%;
}
body {
    height: 100%;
    margin: 0;
}
```

Løsning 1

NB! I de følgende kodeeksempler har jeg indskudt et mellemrum mellem "<" og "/" i div-endtag'et. Dette kun af hensyn til Ekspertens CMS, der ellers ville opfatte dem som egentlige tags og dermed afslutte en tidligere div, der ikke skulle afsluttes her. Husk at fjerne disse mellemrum, hvis I kopierer noget af koden.

Her følger nu tre forskellige måder at lave layoutet på. Først den enkleste:

```
CSS
.container {
    position: absolute;
    text-align: left;
    left: 50%;
    margin-left: -401px;
    height: auto;
    width: 800px;
    border: 1px solid black;
}
.top-box {
    position: relative;
    top: 0px;
    left: 0px;
    width: 800px;
    height: 98px;
    background-color: red;
}
```

```

.cell-like {
    display: table-cell;
}
.left {
    width: 200px;
    background-color: yellow;
}
.center {
    width: 300px;
    background-color: white;
}
.right {
    width: 300px;
    background-color: blue;
    color: white;
}

```

HTML

```

<div id="tabel-1" class="container">
<div class="top-box">&nbsp;< /div> <!-- HUSK AT FJERNE MELLEMRUM -->
<div class="left cell-like">
Menu<br>
P u n k t 1<br>
P u n k t 2<br>
[...]
< /div>
<div class="center cell-like">
Bla bla bla bla<br>
Bla bla bla bla<br>
Bla bla bla bla [...]
< /div>
<div class="right cell-like">
Mere tekst<br>
Mere tekst
< /div>
< /div>

```

Den omgivende container boks er centreret på sædvanlig vis: skyd den 50% til højre i vinduet, og flyt den til venstre igen svarende til halvdelen af dens egen bredde (her er der også taget højde for borderen på 1px). Når det gælder de tre bokse ved siden af hinanden er det der gør tricket her, egenskaben "display: table-cell". Normalt kan man ikke stille div'er ved siden af hinanden, for en div er et blok-element, og derfor starter og slutter den med at bryde linjen. Næste div i rækken vil derfor blive stillet under den foregående.

Men "display: table-cell" bryder denne regel. Samtidig gør den at de tre bokse kommer til at flugte i højden med den højeste af dem - netop den egenskab der er særlig for tabelceller. Så det er jo sådan set meget enkelt. Der er bare det alvorlige minus, at denne egenskab kun er understøttet af Gecko browserne (Netscape, Mozilla), ikke af Internet Explorer. I denne vil boksene altså i stedet blive vist under hinanden, og de vil have forskellig højde alt efter deres indhold. Så desværre, ud med den løsning, så længe IE halter bagefter med CSS.

Løsning 2

Løsning 2 benytter sig af en anden mulighed for at stille blok-elementer ved siden af hinanden: "float".

De kommer ganske vist ikke til at flygte i bunden med den højeste boks, som i den forrige løsning, men vi får dem til at se ud som om de gør det på to måder: 1) Container boksen der angiver kanten omkring hele designet, skal udvide sig med den højeste boks; det har vi allerede sikret os ved at denne container boks har egenskaben "position: absolute". 2) Vi styrer baggrundsfarverne sådan at de bokse der ikke fylder hele container boksen ud, flyder i ét med den omgivende boks.

CSS

```
/* Vi genbruger hele det foregående CSS,  
bort set fra at vi erstatter klassen .cell-like med denne: */
```

```
.fl-left {  
    float: left;  
}
```

HTML

```
<!-- Ligesådan her: vi udskifter bare class="cell-like" med class="fl-left"  
-->
```

Men dette design har nogle begrænsninger, for som nævnt kommer de tre bokse under overliggerboksen ikke til at flygte. Hver af dem bliver ikke højere end det indhold der er i dem. Hvis de alle skal have samme farve, gør det ikke så meget. For så giver vi bare alle bokse, inklusiv den omgivende boks, ens farve.

Men hvis vi vil have forskellig baggrundsfarve på boksene er vi nødt til at udvide løsning 2:

Løsning 3

Udvidelsen består i at vi lægger center- og right-boksen ind i deres egen omgivende boks. Og også den får egenskaben "float: left". Vi kan beholde det meste af CSS'en fra løsning 2, men udvider den med denne ekstra omgivende boks og justerer nu baggrundsfarverne så boksene kommer til at gå i ét med deres omgivende container bokse. Tilsammen får vi tre bokse der *ser ud som om* de flygter og som har hver sin farve:

CSS

```
.container {  
    position: absolute;  
    text-align: left;  
    left: 50%;  
    margin-left: -401px;  
    height: auto;  
    width: 800px;  
    background-color: yellow;  
    border: 1px solid black;  
}  
.top-box {  
    position: relative;  
    top: 0px;  
    left: 0px;  
    width: 800px;  
    height: 98px;  
    background-color: red;  
}
```

```

.fl-left {
  float: left;
}
.left {
  width: 200px;
  background-color: yellow;
}
#center-right-container {
  float: left;
  width: 600px;
  background-color: blue;
}
.center {
  width: 300px;
  background-color: white;
}
.right {
  width: 300px;
  background-color: blue;
  color: white;
}

```

HTML

```

<div id="tabel-3" class="container">
<div class="top-box">&nbsp;< /div> <!-- HUSK AT FJERNE MELLEMRUM -->
<div class="left fl-left">
Menu<br>
P u n k t 1<br>
P u n k t 2<br>
[...]
< /div>
<div id="center-right-container">
<div class="center fl-left">
Bla bla bla bla<br>
Bla bla bla bla<br>
Bla bla bla bla [...]
< /div>
<div class="right fl-left">
Mere tekst<br>
Mere tekst
< /div>
< /div>
< /div>

```

Der opstår dog endnu et problem hvis vi ønsker border mellem boksene. For så vil man se at de ikke er lige høje. Hvis vi på forhånd ved at midterste boks har mest indhold og altså bliver den højeste, løser vi det bare ved at sætte left og right border på den alene. Men hvis det er boksen til venstre eller boksen til højre der er den højeste har vi et problem, hvis der skal være border imellem alle tre. Disse begrænsninger betyder at dette design kun er anvendeligt hvis der ikke skal være border mellem boksene, eller hvis vi ved at det er den midterste boks der er den højeste.

Så alt i alt giver dette design os ikke samme fleksibilitet med hensyn til højde som en "rigtig" tabel. Men vi kan ikke komme længere med CSS end hertil. Og da Internet Explorer ikke forstår "display: table-cell"

må vi gribe til andre metoder.

Løsning 4

F.eks. kan vi bruge JavaScript. Her er et script der løser problemet. Det er lavet af "roenving" her på Eksperten, se: <http://www.eksperten.dk/spm/477594>

Script

```
<script type="text/javascript">
function setDivsHeight() {
  var divsH = ["leftBox","centerBox","rightBox"]; // indsæt div-ID'er
  var maxH = 0;
  for(i=0;i<divsH.length;i++){
    maxH = Math.max(maxH,document.getElementById(divsH[i]).offsetHeight);
  }
  for(i=0;i<divsH.length;i++){
    document.getElementById(divsH[i]).style.height = maxH + 'px';
  }
}
</script>
```

CSS

```
/* Den samme som i løsning 2 */
```

HTML

```
<body onload="setDivsHeight();">
<div id="tabel-4" class="container">
<div class="top-box">&nbsp;< /div> <!-- HUSK AT FJERNE MELLEMRUM -->
<div id="leftBox" class="left fl-left">
Menu<br>
P u n k t 1<br>
P u n k t 2<br>
[...]
< /div>
<div id="centerBox" class="center fl-left">
Bla bla bla bla [...]
< /div>
<div id="rightBox" class="right fl-left">
Mere tekst<br>
Mere tekst<br>
< /div>
< /div>
```

I forhold til løsning 2 har vi altså bare sat ID på de tre bokse så de kan styres af scriptet som looper igennem alle tre bokse og finder den højeste. Det sker ved hjælp af metoden max(), som sammenligner to tal og vælger det højeste. Ved igennem loopet at sætte variabelen maxH til højden af hver enkelt boks og så sammenligne med sig selv, ender den med at have værdien af den højeste boks. Dernæst sættes boksenes style.height til denne værdi.

Med denne metode har vi hverken brug for den ekstra center-right-container eller for at tænke på de omgivende baggrundsarver, for nu fylder alle tre bokse hele container boksen ud. Det er heller ikke noget problem at sætte border på dem. Det kan f.eks. bare gøres på den midterste boks.

Der er dog den avlorlige ulempe ved JavaScript-løsninger på vores problem (som det også bliver nævnt i kommentarerne til denne artikel) at folk jo kan have slået JavaScript fra i deres browser. Når løsningen alligevel er kommet med her, er det derfor ikke for at anbefale den - jeg ville ikke selv bruge den - men blot for at vise at det kan lade sig gøre. Hvis opgaven som her er at lave et trefløjet design hvor alle fløje automatisk alignes i højden med den fløj der har mest indhold - uden at der angives en fast højde - ja, så ville jeg selv vælge en tabel der jo netop forærer os denne egenskab. Men hvis man derimod kan leve med at der angives faste højder på fløjene, så ville jeg foretrække en CSS-løsning uden tabel.

Kendte problemer

Hvis man ønsker helt at erstatte de absolutte mål i px med relative mål, kan der opstå problemer med "float", specielt hvis der sættes border på boksene og denne angives i px. Til sammen kan det gøre at boksene i browserens fortolkning kommer til at fylde mere end 100%, og så bliver de stillet under hinanden i stedet.

Links

http://www.e-tidsskrift.dk/dev/non_table_design.php (demo af ovenstående)

<http://bluerobot.com/web/layouts/default.asp> (godt sted til andre tre- og tofløjs layouter)

Kommentar af olebole d. 29. Apr 2004 | 1

Fin lille artikel - med visse forbehold :)

1. White-space før DTD'en betyder ikke noget - heller ikke i IE.
2. Hvorfor linke til W3C-Schools og ikke til W3C? De to har intet med hinanden at gøre.
3. 'Trefløjet-layout' laves stadig bedst med en tabel (som man efterfølgende kan fylde med evt. 'container'-design ... altså div/span med CSS). At skulle rette op på 'container'-designs mangler med JavaScript er ikke så heldigt. Der er en hel del, der slår JS fra - og også en hel del, der surfer med tekstbaserede browsere.

I W3C's arbejdsrapporter anbefales tabeller til tabulerede data - og kolonne-opdeling.

Ellers en udmærket artikel :)

/mvh

Kommentar af basicq d. 17. Sep 2004 | 2

Ok artikel hvis man er lidt tør på CSS'en men jeg mener stadig at tabeller kommer i første række. Hvis alle browser producenter kunne blive enige om fast standard både hvad angår HTML og css, ville det være super at bruge CSS. Men forskellen fra browser til browser er stadig for stor. Selv dine flotte eksempler har problemer. Men ellers en udmærket artikel. (Man kunne måske argumenter for at overskriften og den sidste linie i artiklen kunne have et sammenspil.) BasicQ :-)

Kommentar af thorvall d. 02. Apr 2004 | 3

Kommentar af roenving d. 20. Dec 2005 | 4

Ganske udmærket gennemgang. Eneste alvorlige mangel er den fraværende præcisering af, at det absolut er tåbeligt at basere layout på javascript-funktioner, når der findes andre muligheder, altså tabeller !-)

Kommentar af doodoo d. 24. Mar 2004 | 5

Lækker artikel om CSS/HTML meget brugervenlig

Kommentar af jakobdo d. 26. Jul 2004 | 6

Fin artikel, havde netop oprettet et spørgsmål om samme emne! :o)

Kommentar af netnerd d. 19. Mar 2004 | 7

Kommentar af jensendk d. 29. Jun 2004 | 8

Najs, og bestemt brugbar hvis man ikke i forvejen roder med CSS og positioning

Kommentar af googolplex d. 19. Mar 2004 | 9

Rigtig god artikel - et område jeg selv har haft problemer med. Nu venter vi så bare på at IE bliver opgraderet til det 21. århundrede :(

Artiklens overskrift er dog temmelig uheldigt valgt!

Kommentar af stich d. 20. Dec 2005 | 10

Fin artikel, bortset fra du konkluderer noget forkert mht. hvad man kan opnå med CSS, hvilket ikke er så heldigt, når artiklen netop handler om det emne. For det første har man i lang tid kunnet opnå det man ville, med CSS, specielt mht. 2-kolonne-layouts. Nogle ting har dog krævet noget langhåret kode, når det fx drejede sig om 3-kolonne layouts- men det lod sig gøre.

Der er dog nu fundet en ny måde at klare tingene på, som gør det på en betydeligt bedre måde, end dem man hidtil har benyttet:

<http://positioniseverything.net/articles/onetruelayout/>

Desuden vil jeg lige nævne, hvad W3C skriver i introduktionen til tabeller:

"Tables should not be used purely as a means to layout document content as this may present problems when rendering to non-visual media. Additionally, when used with graphics, these tables may force users to scroll horizontally to view a table designed on a system with a larger display. To minimize these problems, authors should use style sheets to control layout rather than tables."

Kommentar af lordhead d. 21. Mar 2004 | 11

Absolut en artikel der er god at gemme, byder på mange gode eksempler!

Kommentar af uso d. 26. Mar 2004 | 12

Kommentar af jonasjersborg d. 12. Apr 2007 | 13

rigtig god artikel