

OIM CSS-klassemodel

Den digitale Taskforce

11. juni 2008

Version 1.0

Versionsstyring

Versio n	Dato	Ansvarli g	Ændringsbeskrivelse
0.1	22-1-08	CAF	Første version
0.2	23-1-08	CAF	Tilføjet udkast til regelsæt for brug af CSS-modellen
0.3	26-1-08	CAF	Revideret efter kommentering
0.4	5-3-08	CAF	Udvidet klassemodel, beskrivelse og påsat status efter styregruppemøde IV. Adskilt fra regelsættet.
0.5	7-3-08	CAF	Klargjort til review i Portalintegrationsprojektet
0.9	26-3-08	CAF	Revideret efter reviewkommentarer og gennemgang. Indstilling til OIM.
0.9.1	9-5-08	CAF	Tilføjet klasser identificeret af borger.dk
0.9.2	16-5-08	CAF	Tilrettet efter review
0.9.3	4-6-08	CAF	Tilrettet med BN's kommentarer omkring ikoner
1.0	11-6-08	CLAA	Godkendt som version 1.0

1. Om OIMCSS-klassemodel-arbejdet

OIMCSS-klassemodellen er udarbejdet af OIMCSS-styregruppen, bestående af repræsentanter fra borger.dk, Den Digitale Taskforce, SKAT og virk.dk.

Arbejdet blev igangsat i januar 2008. Processen tog udgangspunkt i virk.dk's klassebegreber og -model, og DBC, EDB-gruppen, KMD og Kommuneinformation, foruden OIMCSS-styregruppens medlemmer og disses leverandører Capgemini og NNIT, bidrog med kommentarer og forslag, hvor det er forsøgt at ramme ca. 80 % af de klasser, som benyttes til opbygning af portalservice-applikationer.

Alle deltagere i Portalintegrationsprojektet har haft adgang til at kommentere og foreslå rettelser og tilføjelser, hvilket resulterede i version 0.9 af klassemodellen. Denne er nu godkendt af Portalintegrationsprojektets styregruppe og dermed opnået status af version 1.0.

OIMCSS-Styregruppen har arbejdet ud fra følgende overordnede principper:

- a. Det skal være simpelt at anvende. Udviklerne skal hellere holde fokus på forretningslogikken/sideopbygningen, end på den aktuelle formattering og styling. Det tilstræbes derfor at have så logiske CSS-klasser som muligt og at de ikke skal være svære at huske eller anvende.
- b. Understøttelse af de fællesoffentlige standarder. Da den praktiske verden for browsere desværre er mere broget er der taget praktisk stilling til om konkrete designs kan realiseres i IE6, IE7 og Firefox 2.0 – dog har selve arbejdet med OIMCSS ikke været at realisere CSS'en, men at udpege de klasser, der kan være fælles.

2. Politik for navngivning i OIMCSS-klassemodellen

Det har været diskuteret, hvorvidt den fælles OIM-CSS-klassemodel skulle udvikles efter en 'attribut-baseret' eller 'hierarkisk-opbygget' model.

Med 'attribut-baseret' menes, at hvert relevant HTML-element kan påføres en eller flere CSS-klasser, der kan arbejde sammen i kombination. Dermed vil hver klassedefinition ikke være andet end en angivelse af den/de helt specifikke egenskaber, som klassen repræsenterer. Eksempelvis vil klassen 'text' justere et input-felts tekst, så det kan stå pænt på linje med omgivende ledetekster og radiobuttons, men skal feltet have en bestemt bredde skal man kombinere med f.eks. 'short'-klassen.

Fordelen ved den attribut-baserede model er, at hver klasse ikke nødvendigvis skal defineres fuldt ud (således at den kan stå alene). Til gengæld kræves det, at der i beskrivelsen fremgår tydeligt i hvilke sammenhænge en bestemt klasse kan anvendes og evt. i kombination med hvilke andre og hvornår de ikke kan kombineres.

Den 'hierarkisk-opbyggede' model opdeler CSS-klasserne i de varianter der kan benyttes, og beskriver dem via navngivning. Eksempelvis vil 'form_text_short' være en nøjagtig beskrivelse af et tekst-inputfelt, der skal justeres til radiobuttons, og have en bestemt bredde. Ulempen er, at man også er nødt til at definere en 'form_text_wide' osv. Med andre ord er der en vis redundans i den hierarkiske model.

OIMCSS-Styregruppen lægger op til en attribut-baseret model ud fra følgende argumentation:

OIMCSS-klassemodellen må ikke bidrage til en højere fejlrate alene på grund af det valgte princip, og den hierarkiske model *kunne* foranledige udviklere til at gætte på en given kombination, der ikke findes, eller bytte rundt på rækkefølger.

Det har ikke været muligt internt i styregruppen at kunne fremfinde konkrete eksempler på, at den attributbaserede model ikke kan bruges (læs: mappes fra eksisterende klasser), ligesom der ikke er kommet indvendinger hertil i review-processen.

Den hierarkiske model vil resultere i flere gentagelser i CSS filerne, idet en del regler skal gentages for fx "form_text_short", "form_text_wide", "form_text" og lign.

Der har ikke kunnet frembringes væsentlige argumenter for at bruge en hierarkisk opbygget model.

3. Klassemodellen

Inddeling af skærmens bredde

Den aktuelle inddeling af kolonner afgøres af den portal/website, som portalservicen skal præsenteres på. OIMCSS-klassemodellen søger at foretage en logisk opdeling af kolonner, således at en udvikler kan henholde sig til disse, og overlade den eksakte implementation til det CSS der kommer fra den enkelte portal.

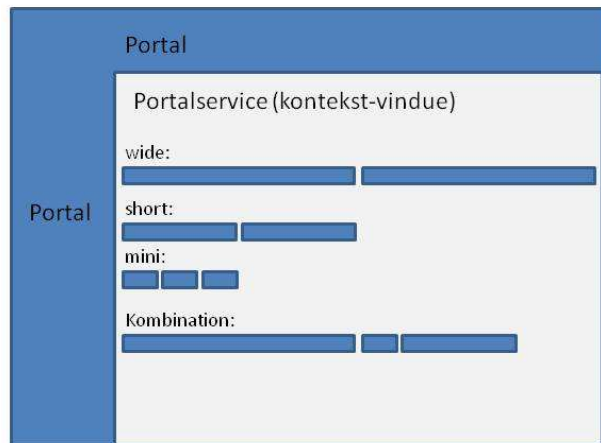
Der gives bredder i flg. størrelser, der angives relativt til den bredde, som kontekst-billedet har (dvs. fuld skærmstørrelse fraregnet portalens evt. venstremenu¹):

'form'	svarende til fuld bredde
'wide'	svarende til halv bredde
'short'	svarende til kvart bredde. Der går to shorts på en wide.
'mini'	svarende til 1/12 bredde. Der går 3 mini på en short og 6 mini på en wide.

Bredden skal implementeres i den enkelte portals CSS således, at den inkluderer margin til næste felt – flere felter skal kunne 'floate' ved siden af hinanden.

I eksemplet nedenfor vises ledeteksterne over input-feltet, hvor nogle portaler vil ønske at ledetekster og inputfelt står på samme linje. For at sikre at begge opsætninger er mulige skal bredden påsættes den container, der 'floates'. For at sikre at håndtering af ledetekster kan gøres korrekt gives klasserne: 'label.form', 'label.wide', 'label.short', 'label.mini'. Klasserne bør kunne benyttes på både input- og div-elementer.

¹ Det er op til den enkelte portal, om portalservicen får tilbudt hele browservinduets bredde, eller om portalen f.eks. påklitrer sin egen venstremenu. OIMCSS-klasserne refererer kun til det kontekst-vindue som den enkelte portal tilbyder.



Knapper

Knapper kan implementeres som input type, eller som et HTML-button-element. Input-type med klassen 'button' vil give den nødvendige fleksibilitet til at kunne håndtere browserkompatibilitetsproblemer ('button'-tagget sikrer submit-funktionaliteten i IE6).

Eksempel 1:

```
<p><input type="submit" value="Gem" class="button short" /></p>
```

Eksempel 2:

```
<button class="short" type="submit">
  Søg <strong>kun</strong> i denne mappe
</button>
```

Indkapslinger, afsnit og linjebud

Der gives klassen 'newline', der indføres aht. browserkompatibilitet. Klassen giver mulighed for at lave linjeskift efter en række af felter på samme linje. Den er kun nødvendig HTML-mæssigt ved flere felter på samme linje.

```
<br class="newline" />
```

CSS-klassen 'fieldset' grupperer fields. Skal benyttes sammen med en div-konstruktion for at kunne style grupperingsboksen særskilt, f.eks. til legend.

```
<div class="fieldset">
  <fieldset>
    <legend><span>Personoplysninger</span></legend>
    <p>Indtast personlige oplysninger</p>
    <label>Navn<span class="required">*</span><br/>
      <input type="text" id="navn" name="navn" class="text" value="" />
    </label><br/>
    <label>email<br/>
      <input type="text" id="email" name="email" class="text" value="" />
    </label><br/>
  </fieldset>
</div>
```

Alignment

Der gives to klasser, 'alignleft' og 'alignright', der justerer et element imod hhv. venstre og højre margin på siden. (de justerer ikke indenfor elementet).

```
<label class="alignleft">Vejnavn
  <input type="text" class="short text" value="" />
</label>
```

Pil/Markering frem/tilbage

Der gives to klasser, 'next' og 'previous', der formatterer en pil/billede i retning hhv. højre og venstre².

De to klasser benyttes med den fælles klasse, 'nextpreviousbasis', der formatterer farve, font-weight og kant, og derudover skaber afstand til elementerne ovenfor. 'nextpreviousbasis' sikrer også at de to knapper vil være horisonalt alignet.

```
<p class="nextpreviousbasis">
  <input type="button" value="Forrige" class="alignleft button previous" />
  <input type="submit" value="Næste" class="alignright button next" />
</p>
```

Klassen "disabled" kan tilføjes på inaktive knapper.

```
<p class="nextpreviousbasis">
  <input type="button" value="Forrige"
    class="alignleft button previous disabled" disabled="disabled" />
  <input type="submit" value="Næste" class="alignright button next" />
</p>
```

² Det ligger ikke indenfor OIM CSS-klassemodellen, at tage stilling til højre-venstre-vendte sider. Dette må om nødvendigt tilføjes ved en senere lejlighed.

Særlige markeringer

Der angives klassen 'required', der angiver at feltet er obligatorisk og skal udfyldes. Denne klasse kan kun bruges på ledeteksten (helt nøjagtigt på span-tagget der omkranser labelen), ikke på selve input-feltet.

```
<label class="alignleft">Vejnavn  
  <span class="required">*</span><br/>  
  <input type="text" class="short text" value="" />  
</label>
```

Der gives klassen 'readonly', hvor tekst ikke kan ændres. Denne virker ikke på input-felter og knapper, der istedet skal benytte attributten 'disabled'.

```
<label class="readonly">Readonly felt:  
  <span>Label vises med fed, værdi på samme linie</span>  
</label><br/><br class="newline" />
```

Der gives klassen 'totop', der formatterer et link, der går til sidens top.

```
<div><a href="#top" class="totop">Til toppen</a></div>
```

Der gives klassen 'information', der formatterer et spørgsmålstegn (?) eller tilsvarende markering, der benyttes til at give brugeren adgang til hjælp og vejledning.

```
<div><a href="helppage.htm?id=555" class="information">?</a></div>
```

Til brug på popup-vinduer med hjælpetekst findes klassen 'helpwindow', der giver en genkendelig baggrundsfarve. 'helpwindow' sættes på body elementet i det nye vindue.

```
<body class="helpwindow">Dette er et hjælpevindue - hvor du kan få hjælp til  
selvhjælp.</body>
```

Der gives klassen 'alternate', der kan give en anden baggrundsfarve, f.eks. til hver anden linje i en tabel.

```
<table border="0">
<tr>
  <td id="row1-1-1">kol 1</td>
  <td id="row1-1-2">kol 2</td></tr>
<tr class="alternate">
  <td id="row1-2-1">kol 1</td>
  <td id="row1-2-2">kol 2</td></tr>
<tr>
  <td id="row1-3-1">kol 1</td>
  <td id="row1-3-2">kol 2</td></tr>
<tr class="alternate">
  <td id="row1-4-1">kol 1</td>
  <td id="row1-4-2">kol 2</td></tr>
</table>
```

Tekstfelter

Der gives klassen 'text', der anvendes på input-felter. Klassen justerer indholdet i selve feltet, således at det kan benyttes på samme linje som radio-buttons (teksten bliver aligned vertikalt – hvilket sikrer browserkompatibilitet med IE6). Uden brug af denne klasse vil en radiobutton på samme linje stå skævt i forhold til tekstindholdet.

```
<input type="text" class="short text" value="" />
```

Hvert inputfelt-type har en tilsvarende klasse, der justerer disse ind:

'radio' – 'combobox' – 'checkbox' – 'text' – 'textarea' – 'submit' – 'reset'

Menuer

Der gives klassen 'leftmenu', der anvendes på en venstremenu, hørende til portalservicen. Det antages at portalens venstremenu ikke vises samtidigt. Venstremenuen giver mulighed for bedre navigation i meget komplekse/informationstunge portalservices. Afh. af den enkelte portal vil venstremenuen kunne implementeres vertikalt eller horisontalt (og dermed ligne faneblade).

```
<ul id="leftmenu">
  <li><a href="/">Forside</a></li>
  <li><a href="/menu1/">Menu1</a></li>
  <li><a href="/menu2/">Menu2</a></li>
  <li><a href="/menu3/">Menu3</a></li>
  <li><a href="/menu4/">Menu4</a></li>
</ul>
```

Lister

For nummererede lister gives klassen 'enumitem' til de elementer, der indgår i listen.

For ikke-nummererede lister gives klassen 'bulletitem' til de elementer, der indgår i listen.

Til selve listen gives klassen 'list'.

Det element der er valgt kan fremhæves visuelt ved at påsætte klassen 'selected'.

```
<ol class="list">
  <li class="bulletitem">Første i listen</li>
  <li class="bulletitem selected">Aktuelt valgte element i listen
</li>
  <li class="bulletitem">Sidste i listen</li>
</ol>
```

Lister der kan pakkes ud eller ind

Klassen 'foldedlist' påsættes listeelementet, der indeholder elementer, der kan pakkes ud/ind.

Der gives klassen 'folded', der giver et plus (+) ud for et element, der kan pakkes ud/ekspanderes.

Klassen 'unfolded' giver tilsvarende et minus (-) og markerer at elementet er udpakket.

Det element der er valgt kan fremhæves visuelt ved at påsætte klassen 'selected'.

```
<ol class="foldedlist">
  <li class="folded">Første i listen</li>
  <li class="unfolded selected">Aktuelle valgte<p>tekst der vises når den er
udpakket</p>
</li>
  <li class="folded">Sidste i listen</li>
</ol>
```

Forløbsindikator

Der gives klassen 'stepindicator' til at angive et forløb igennem flere (ordnede) trin. I tilgift kan man markere udførte og ikke-udførte, samt aktuelle trin med klasserne 'done', 'undone' hhv. 'current'.

```
<ol class="stepindicator">
  <li class="done">Første step <b class="tip"></b></li>
  <li class="done">Andet step<b class="tip"></b></li>
  <li class="current">Tredie step
    <p>Denne tekst vises nu</p>
    <p>Også denne linje</p>
    <b class="tip"></b>
  </li>
  <li class="undone">Fjerde step<b class="tip"></b></li>
  <li class="undone">Femte step<b class="tip"></b></li>
  <li class="undone final">Sidste step<b class="tip"></b></li>
</ol>
```

Fremhævelse af systembeskeder

Der gives klassen 'notificationbox', der kan suppleres med klasserne 'error', 'info', 'alert' og 'warning', hvilket typisk vil give en farvemæssig forskel på fejl/beskeder samt tilhørende ikoner.

Her gives tekstuelle eksempler på beskedtyper, der vil passe i hver variant:

- info: "Dine oplysninger er nu gemt"
- warning: "hvis du trykker ja vil alle dine indstillinger blive slettet"
- error: "Udfyld venligst postnummer korrekt 'DK-4000' er ikke et gyldigt postnummer"
- alert: "Denne er service er i øjeblikket ikke tilgængelig, der arbejdes på at udbedre fejlen."

```
<div class="notificationbox error">
  <h2>Der er sket en fejl</h2>
  <ul>
    <li>Udfyld venligst postnummer korrekt 'DK-4000' er ikke et gyldigt
    postnummer </li>
  </ul>
</div>
```

Der vil i visse portalservices være valgt en anden måde at vise valideringsfejl på, ved at markere det felt der ikke kan validere korrekt på en særlig måde. For at kunne imødekomme dette gives klassen 'fielderror', der angives på det felt, hvor der er en valideringsfejl.

```
<div class="fieldset">
  <fieldset>
    <legend><span>Firmaoplysninger</span></legend>
    <p>Indtast oplysninger om ansættelse</p>
    <label><span class="fielderror">Firmanavn<br/>
      <input type="text" id="firma" name="firma" class="text" value=""/>
    </span></label><br/>
    <label>CVR-nummer<br/>
      <input type="text" id="cvr" name="cvr" class="text" value=""/>
    </label><br/>
  </fieldset>
</div>
```

Ikoner

For at imødekomme standardisering af ikoner på portalerne gives et antal ikoner, der kan benyttes. Det tilrådes at benytte de fælles standard-ikoner, men dette er ikke et krav i denne version af OIMCSS.

Der gives flg. klasser, der vil indsætte ikoner (forklaring er eksemplificerende):

- | | |
|--------------------|---|
| 'icon totop' | et ikon for funktionen at gå til sidens top. |
| 'icon print' | et ikon for at åbne en printvenlig side/print-dialog |
| 'icon calendar' | et ikon der vil aktivere en kalender-kontrol, f.eks. til hurtigvalg af dato. |
| 'icon key' | et ikon der viser en nøgle – angivelse af f.eks. låste eller beskyttede oplysninger /sider. |
| 'icon clause' | et ikon der viser et paragraf-tegn (§) – angivelse af regler, love mv. |
| 'icon help' | et ikon der kan benyttes til at vise hjælp – se også tilsvarende link-formattering under særlige markeringer. |
| 'icon information' | et ikon der kan benyttes til at vise markere særlige informationer eller guide brugeren til yderligere information (som ikke er hjælp). |

Ikonerne vil vises efter teksten/elementet de tilknyttes, feks.:

```
<div class="icon key">CPR: 123456-7890</div>
```