



## Selectors i CSS

Artiklen giver en grundig gennemgang af selectors i CSS, og gennemgår samtidig, i hvilke browsere de kan bruges.

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

### Update

**13. marts 2004:** Som flere brugere har efterlyst, bliver det nu kommeteret hvilke browsere de forskellige selectors virker i!

**13. marts 2004:** Der var en mindre fejl omkring funktionaliteten bag `:first-child`. Det er nu rettet.

### Hvad er en selector?

En selector kan på dansk fortolkes som en vælger. Noget der vælger hvilket element vi har med at gøre, og under hvilken betingelse vi gør noget med det.

### Kompabilitet

Alt CSS i denne artikel er testet i følgende browsere på en dansk version af Windows XP Professional SP1. For at gøre det hurtigt overskueligt, har jeg udformet nogle forkortende navne for de forskellige test-browsere.

#### Browser Fuldt navn

IE 6	<code>&lt;a href="http://www.microsoft.com/windows/ie/default.asp"&gt;Microsoft Internet Explorer 6 SP1&lt;/a&gt;</code>
IE 5.5	<code>&lt;a href="http://www.mijnkopthee.nl/standalone_IE/"&gt;Microsoft Internet Explorer 5.5 SP2, NT&lt;/a&gt;</code>
IE 5.01	<code>&lt;a href="http://www.mijnkopthee.nl/standalone_IE/"&gt;Microsoft Internet Explorer 5.01 SP2, NT&lt;/a&gt;</code>
IE 4	Microsoft Internet Explorer 4
IE 3	Microsoft Internet Explorer 3, Windows 3.1
Mozilla	<code>&lt;a href="http://mozilla.org/"&gt;Mozilla 1.6&lt;/a&gt;</code>
Firefox	<code>&lt;a href="http://mozilla.org/products/firefox/"&gt;Mozilla Firefox 0.8&lt;/a&gt;</code>
Opera	<code>&lt;a href="http://www.opera.com/"&gt;Opera 7.23&lt;/a&gt;</code>

### Simple selectors

Den simpleste selector du kan lave er `*` (stjerne, eller asterisk på engelsk). Den sætter dine CSS-egenskaber på alle elementer.

#### Eksempel:

```
* { font-size: 13px }
```

### Kompabilitet:

Browser	Virker?
IE 6	Ja!
IE 5.5	Ja!
IE 5.01	Ja!
IE 4	Ja!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Ja!

Den næste er specifikt på elementer af samme navn. F.eks. et <div>-element. Her skriver man først elementets navn, efterfulgt af CSS-egenskaberne:

### Eksempel:

```
div { border: black 1px solid }
```

### Kompabilitet:

Browser	Virker?
IE 6	Ja!
IE 5.5	Ja!
IE 5.01	Ja!
IE 4	Ja!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Ja!

### Klasser og id'er

Før satte vi egenskaber for *alle* div-elementer, men hvad nu hvis vi kun ønsker at ramme ét element? Til det formål har vi HTML-attributten "id".

### Eksempel:

```
<table id="foo">  
  <tr>  
    <td>bar</td>  
    <td>bar</td>  
  </tr>  
</table>
```

Et element med id-attributten må, i følge W3C's standarder, kun optræde en gang i et HTML-dokument. Havde vi derfor i stedet skrevet `<td id="foo">bar</td>`, ville det generere en fejl. For at angive at vi gør brug af et id i CSS, bruger man et #-tegn (Pound Sign på engelsk), direkte efterfulgt af navnet på id'et.

#### Eksempel:

```
#foo { border-collapse: collapse }
table#foo { border: black 1px solid }
```

Sidste linie er også helt lovlig, men egentlig unødvendig, da id'er jo kun må optræde en gang.

#### Kompabilitet:

##### Browser Virker?

IE 6	Ja!
IE 5.5	Ja!
IE 5.01	Ja!
IE 4	Ja!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Ja!

Hvad gør man så hvis der er mange elementer som skal have de samme egenskaber? Til dem har vi HTML-attributten, "class".

#### Eksempel:

```
<div class="foo">
  <div class="bar">foobar</div>
  [div]foobar</div>
  <div class="bar">foobar</div>
</div>
```

For at forklare browseren at vi har med en klasse at gøre, angives et . (punktum) direkte efterfulgt af navnet på klassen.

#### Eksempel:

```
.bar { width: 30px }
```

Det kan også gøres så man kun berører klasser fra de rigtige elementer.

**Eksempel:**

```
span.foo { height: 10px }  
div.foo { height: 20px }
```

Den første sker der intet med, eftersom intet span-element er tilstede.  
Den sidste bliver til gengæld fortolket i alle div-elementer med klassen "foo".

**Kompabilitet:****Browser Virker?**

IE 6	Ja!
IE 5.5	Ja!
IE 5.01	Ja!
IE 4	Ja!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Ja!

**Underliggende elementer**

For at slippe for at angive klasser til f.eks. en masse <li>-elementer, kan det gøres lettere. Her er det nemlig muligt at gribe fat i alle <li>-elementer under et andet element, f.eks. meget passende et <ul>-element.

**Eksempel:**

```
<ul id="foo">  
  <li><a href="/">bar</a></li>  
  <li><a href="/">bar</a></li>  
  <li><a href="/">bar</a></li>  
</ul>
```

Her fortæller vi at ul'en har id'et "foo". For at få alle elementer inde i ul-elementet sætter man mellemrum mellem de to elementers navne.

**Eksempel:**

```
#foo li { list-style-type: none }
```

Vi kunne også skrive "ul li" i stedet for "#foo li", men brugen af id'er og klasser er en god måde at overskue tingene på.

Vi kunne også nemt gribe fat i alle a-elementerne under ul'en.

**Eksempel:**

```
#foo a { text-decoration: none }
```

Elementet inde i ul'en behøver altså ikke ligge lige inden for ul'en, men kan ligge langt inde i stakke af elementer.

**Kompabilitet:****Browser Virker?**

IE 6	Ja!
IE 5.5	Ja!
IE 5.01	Ja!
IE 4	Ja!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Ja!

Men hvad gør vi hvis vi kun vil have at den skal ramme de elementer der lige ligger inden for elementet? Det findes der selvfølgelig også en løsning på!

**Eksempel:**

```
<ul id="foo">  
  <li>foobar</li>  
  <li>foobar</li>  
  <li>  
    <ul>  
      <li>bar</li>  
      <li>bar</li>  
      <li>bar</li>  
    </ul>  
  </li>  
  <li>foobar</li>  
</ul>
```

Nu vil vi gribe fat i alle li-elementerne med teksten "foobar". Til det har vi tegnet > (større end). Først skriver vi moder-elements navn, derefter et >-tegn og til sidst navnet på det element lige under moderen, som skal findes.

**Eksempel:**

```
#foo>li { padding: 2px 5px 2px 0px }
```

Ønsker vi også at gribe fat i den underliggende ul kan vi bare gå længere ned i herakierne.

#### Eksempel:

```
#foo>li>ul>li { background-color: #AC0 }
```

I slutningen af artiklen vil jeg give eksempler på nogle længere typer af selectors.

#### Kompabilitet:

Browser	Virker?
IE 6	Nej!
IE 5.5	Nej!
IE 5.01	Nej!
IE 4	Nej!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Ja!

Den sidste inden for underliggende elementer vi skal hive fat i er `:first-child` pseudo-klassen. Som navnet så enkelt siger, så hiver den fat i det første underliggende element.

Tager vi udgangspunkt i HTML-eksemplet fra før kan vi se noget i denne retning.

#### Eksempel:

```
#foo :first-child { font-weight: bold }
```

Derved bliver det første element (i dette tilfælde `<li>`-elementet) vist med fed skrift.

Læg mærke til mellemrummet mellem `#foo` og `:first-child`. Denne selector finder altså først alle underliggende elementer (vha. mellemrummet), og vælger derefter `:first-child`. Ønsker vi at angive en speciel `:first-child`, kan det gøres således.

#### Eksempel:

```
li:first-child { list-style-image: url("foo/bar.png") }
```

Eksemplet for oven finder alle de steder hvor et `li`-tag er det første element indenfor et andet.

#### Kompabilitet:

Browser	Virker?
IE 6	Nej!
IE 5.5	Nej!

IE 5.01	Nej!
IE 4	Nej!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Ja!

### Flere pseudo-klasser

Der findes nogle egenskaber for elementer som gør vi kan give dem et andet udseende, ved forskellige handlinger.

Disse handlinger er *:hover*, *:active* og *:focus*.

**:hover** udføres når brugeren fører musen over et element.

**:active** udføres når brugeren klikker på elementet og gør det aktivt.

**:focus** udføres når brugeren har gjort elementet aktivt.

#### Eksempel:

```
input          { background-color: green }
input:active   { background-color: white }
input:hover    { background-color: gray }
input:focus    { background-color: lime }
input:focus:hover { background-color: red }
```

Dette giver et input-element forskellig baggrundsfarve, ved forskellige handlinger. Læg mærke til at handlingerne godt kan lægges oven på hinanden. *input:focus:hover* bliver på den måde udført, når elementet er aktivt, og brugeren har musen over det.

#### Kompabilitet:

##### Browser Virker?

IE 6	Kun ved links!
IE 5.5	Kun ved links!
IE 5.01	Kun ved links!
IE 4	Kun ved links!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Ja!

Folk tror ofte at disse egenskaber kun gælder for links, hvilket absolut ikke er tilfældet. Links har dog alligevel reserveret nogle pseudo-klasser til sig selv. Nemlig *:link* og *:visited*.

**:link** udføres når et link bliver vist første gang.

**:visited** udføres når brugeren har været på det pågældende link.

Følgende eksempler giver på den måde samme resultat.

**Eksempel:**

```
a { color: red }
:link { color: red }
a:link { color: red }
```

Dog ikke hvis linket har været besøgt før.

**Eksempel:**

```
a { text-decoration: none }
:visited { visibility: hidden }
```

**Kompabilitet:**

**Browser Virker?**

IE 6	Ja!
IE 5.5	Ja!
IE 5.01	Ja!
IE 4	Ja!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Ja!

Den næste vi skal kigge på er *:lang*.

Med den kan du give forskellige egenskaber, afhængigt af hvilket sprog dokumentet er på.

I HTML bliver sprog-egenskaben blandt andet valgt ud fra *lang*-attributten, *<meta>*-elementet og information fra protokollen, f.eks. HTTP headers.

I XML bliver sprog-egenskaben valgt ud fra *xml:lang*-attributten i *html*-elementet.

**Eksempel:**

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="da">
```

I CSS-eksemplet for neden, angiver vi hvordan quotes skal se ud ved de forskellige sprog-egenskaber.

**Eksempel:**

```
:lang(fr) { quotes: "«" "»" ' ' ' ' ' ' ' ' }
```



```
:lang(da) { quotes: '"" '"" '"" '""' }
```

#### Kompabilitet:

##### Browser Virker?

IE 6	Nej!
IE 5.5	Nej!
IE 5.01	Nej!
IE 4	Nej!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Nej!

Den næste pseudo-klasse hedder *:first-line*, hvilket hentyder meget til dens virkning.

#### Eksempel:

```
p:first-line { font-weight: 900 }
```

Ovenstående eksempel gør den første linie i en paragraf fed.

#### Kompabilitet:

##### Browser Virker?

IE 6	Ja!
IE 5.5	Ja!
IE 5.01	Nej!
IE 4	Nej!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Ja!

Ligesom man kan gøre noget ved den første linie, kan man også gøre noget ved det første bogstav, med *:first-letter*.

#### Eksempel:

```
p:first-letter { font-size: xx-large }
```

### Kompabilitet:

Browser	Virker?
IE 6	Ja!
IE 5.5	Ja!
IE 5.01	Nej!
IE 4	Nej!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Ja!

De sidste pseudo-klasser vi skal have fat i er *:before* og *:after*. Disse er mest effektive med CSS-egenskaben "content". En egenskab som f.eks. kan indsætte tekst før, eller efter, indholdet af et element. Her er det naturligvis *:before* og *:after* der bestemmer hvor teksten skal indsættes.

### Eksempel:

```
<p>En mindre tekst.</p>
```

```
p:before { content: "Note: " }
```

Vil i browseren fremstå som:

Note: En mindre tekst.

*:after* fungerer på samme måde, bortset fra at den indsætter teksten efter indholdet af elementet.

### Eksempel:

```
<p>To be, or not to be</p>
```

```
p:before { content: open-quote }  
p:after { content: close-quote }
```

Hvis vi havde valgt fr (fransk) som sprog for vores dokument, som i eksemplet med *:lang*, ville det i browseren fremstå som:

«**To be, or not to be**»

*:before* og *:after* skal altid angives som den sidste del af selector'en!

### Kompabilitet:

Browser	Virker?
IE 6	Nej!
IE 5.5	Nej!
IE 5.01	Nej!

IE 4	Nej!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Ja!

### Efterfølgende elementer

Som tidligere beskrevet kan man sætte egenskaberne for underliggende elementer i et andet element, men hvad nu hvis man ønsker at ramme det efterfølgende element?

#### Eksempel:

```
<body>
  <div id="foobar">
    <span>foo[/span]
  </div>
  <span>bar[/span]
</body>
```

For at sætte CSS-egenskaber for et efterfølgende element, bruges tegnet + (plus).

Denne regel kan kun bruges hvis elementerne har den samme moder. F.eks. kunne vi ikke ramme `<span>bar[/span]` med reference fra `<span>foo[/span]`.

#### Eksempel:

```
#foobar + span { background-color: red }
#foobar+span { color: white }
```

Om man sætter mellemrum før/efter '+' er lige meget. I eksemplet for oven rammer vi `<span>bar[/span]`-elementet.

#### Kompabilitet:

Browser	Virker?
IE 6	Nej!
IE 5.5	Nej!
IE 5.01	Nej!
IE 4	Nej!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Ja!

### Attribut selectors

En attribut selector, finder elementer ud fra hvilke attributter de har, og hvad der står i disse attributter. En attribut i HTML/XML er de egenskaber som findes inde i et tag.

#### Eksempel:

```

```

I ovenstående tag ser vi attributterne *src*, *alt* og *id*. Disse har samtidig en værdi, som vi kan gøre brug af senere.

Når man bruger attribut selectors, angives det med firkantede parenteser **[]**. Den simpleste attribut selector er at søge om attributten overhovedet er til stede.

#### Eksempel:

```
img[src] { border: red 1em solid }  
img[title] { float: right }
```

I ovenstående eksempel vil kun den første CSS-regel blive udført på vores *img*-tag, da tag'et ikke indeholder en *title*-attribut.

#### Kompabilitet:

##### Browser Virker?

IE 6	Nej!
IE 5.5	Nej!
IE 5.01	Nej!
IE 4	Nej!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Ja!

Den næste attribut-selector bruges ved en præcis sammenligning. Den angives stort set ligesom før, bortset fra at der efter attributtens navn indsættes et = (ligmed), og så indholdet i "" (gåseøjne, eller på engelsk: quotes).

#### Eksempel:

```
img[id="bar"] { clear: both }
```

#### Kompabilitet:

##### Browser Virker?

IE 6	Nej!
------	------

IE 5.5	Nej!
IE 5.01	Nej!
IE 4	Nej!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Ja!

Den næste attribut-selector kan bruges til at finde en værdi i en mellemrums-separeret liste. For at angive den, bruges tegnet ~ før ligmed. Altså ~=.

#### Eksempel:

```
img[alt~="bar"] { margin: 2px 0 1px 0 }  
img[alt~="oo"] { clip: rect(5px, 0px, 10px, 2px); }
```

Her vil kun den første CSS-regel blive udført, eftersom "bar" ligger blandt "foo" og "bar". ~= leder efter en præcis streng i den mellemrums-separeret liste. Derfor bliver "oo" ikke fundet, eftersom dets fulde værdi vil være "foo", altså bliver anden linie ikke udført.

#### Kompabilitet:

##### Browser Virker?

IE 6	Nej!
IE 5.5	Nej!
IE 5.01	Nej!
IE 4	Nej!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Ja!

En anden attribut-selector som minder meget om den foregående, finder det første ord i en bindestreks-separeret liste (-, eller på engelsk: hyphen).

Denne angives med et | (pipe) i stedet for ~.

#### Eksempel:

```
<p lang="en-us">Some kind of foobar-text.</p>
```

```
p[lang|="en"] { text-transform: capitalize }  
p[lang|="us"] { text-decoration: blink }
```

Her bliver kun den øverste CSS-regel udført, da værdien skal ligge i starten af listen.

### Kompabilitet:

Browser	Virker?
IE 6	Nej!
IE 5.5	Nej!
IE 5.01	Nej!
IE 4	Nej!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Ja!

Noget du sikkert sidder og savner lige nu, er muligheden for at finde tekst-værdier hvor som helst i attributterne. Med CSS3 bliver det muligt!

### CSS3's attribut selectors

#### ADVARSEL!!!

Følgende selectors er en del af CSS3, som der for øjeblikket arbejdes på. Dokumentationen er godt nok "Candidate Recommendation", hvilket vil sige at browser-udviklerne så småt godt kan begynde at implementere det i deres browser, men der tages stadig forbehold for små ændringer!

For at finde en streng i starten af en værdi angives det med ^=.

#### Eksempel:

```
<a href="http://www.google.com/">The one!</a>
```

```
a[href^="http://"] { text-decoration: underline overline }
```

### Kompabilitet:

Browser	Virker?
IE 6	Nej!
IE 5.5	Nej!
IE 5.01	Nej!
IE 4	Nej!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Nej!

Den næste er muligheden for at søge i slutningen af en attribut-værdi. Det kan vi med \$=.

#### Eksempel:

```


img[src$=".gif"] { display: none }
```

### Kompabilitet:

Browser	Virker?
IE 6	Nej!
IE 5.5	Nej!
IE 5.01	Nej!
IE 4	Nej!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Nej!

Sidst men ikke mindst, skal vi selvfølgelig have mulighed for at kunne søge hvor som helst i værdierne! Dertil har vi \*.=.

### Eksempel:

```
a[href*="google"] { text-transform: uppercase }
```

### Kompabilitet:

Browser	Virker?
IE 6	Nej!
IE 5.5	Nej!
IE 5.01	Nej!
IE 4	Nej!
IE 3	Nej!
Mozilla	Ja!
Firefox	Ja!
Opera	Nej!

### Flere selectors

Som tidligere lovet, vil jeg nu vise nogle eksempler på selectors der gør at vi kan ramme præcist det element vi vil!

### Eksempel:

```
<div id="foo">
  <div class="bar">
    <div></div>
```

```

<p>
  <div>
    <span><a href="http://www.google.com/">[/span]
  </div>
</p>
</div>
</div>

#foo>div.bar:hover>div+p span:first-child[href^="http://"]:after {
  content: " <-eksternt link";
  background-color: #cfc;
  color: black;
}

```

Ovenstående eksempel finder først elementet med id'et "foo", finder alle div-elementer som er direkte underliggende, og har klassen "bar". Når der er :hover på dem, findes alle div-elementer direkte under den. Det næste p-element vælges, og tager alle span-elementer under sig. Fra den findes det først underliggende element. Hvis dette element har attributten "href" og denne starter med "http://", indsættes der teksten " <-eksternt link" efter indholdet af elementet.

Dette eksempel er naturligvis helt ekstremt, og overdrevet. Alligevel viser det meget godt hvordan CSS' selectors fungerer. Selv har jeg en gang haft brug for en denne selector:

```
#menu li:hover a[href^="http://"]+span:after[/pre][[/div]
```

### Afsluttende bemærkninger

Jeg håber denne artikel har givet dig en udvidet forståelse for hvordan selectors i CSS fungerer, og kan hjælpe dig med at opbygge dine hjemmesider efter fremtidens standarder.

//hermandsen

#### Kommentar af limemedia d. 13. Mar 2004 | 1

Fin gennemgang af muligheder - jeg savner meget en "dette virker / dette virker ikke" i de forskellige browsere og operativsystemer. Blot fordi vi "seriøse" webudviklere anvender firefox og vores sider virkeligt spiller kanon i den, er der jo en broget skare kaldet brugere ;) og den sidste NN4 er ikke set endnu !

#### Kommentar af olebole d. 15. May 2004 | 2

Bedste og mest saglige (læs: mindst 'religiøse') artikel om CSS fra denne forfatter. Dog kunne man have skrevet, hvad IE3.X kan bruge af CSS, på én linie i begyndelsen af artiklen. Derimod savner jeg - som andre læsere - info om, hvad NS4.X forstår. Godt nok var CSS ikke implementeret i NS4-serien, men det er faktisk forbløffende, hvormeget CSS den via sit JSSS kunne oversætte og bruge :)



#### **Kommentar af minau d. 16. Mar 2004 | 3**

SUPER CSS - avanceret - men lige i øjet :-)

#### **Kommentar af jensgram d. 10. May 2004 | 4**

Fantastisk artikel! Gennemarbejdet, informativ og med gode eksempler. Thumbs up...

#### **Kommentar af repsac d. 18. Jun 2004 | 5**

Hvorfor ikke bare læse W3's udmærkede "documentation" og så kigge på et par af de allerede eksisterende tabeller? Hvad med folk der skriver CSS for Apple-folket?

[http://www.dithered.com/css\\_filters/css\\_only/index.php](http://www.dithered.com/css_filters/css_only/index.php)

<http://devedge.netscape.com/library/xref/2003/css-support/css2/selectors.html>

#### **Kommentar af mik789 d. 18. Mar 2004 | 6**

Super nyttig oversigt!

#### **Kommentar af ellert d. 14. Mar 2004 | 7**

Virkelig god artikel! CSS bliver vigtigere og vigtigere og denne artikel er god til at få sat det meste på plads!

#### **Kommentar af 7th d. 23. Apr 2004 | 8**

Virkelig god og brugbar artikel. Godt stykke arbejde ;o)

#### **Kommentar af roenving d. 27. Jul 2004 | 9**

Kanon !-)

Det eneste jeg for alvor savner er alle de minusser, hvor IE ikke er med ... -- og fedt med en opdatering !-)

#### **Kommentar af skovenborg d. 17. Mar 2004 | 10**

Super artikel - en god gennemgang af de forskellige selectore der findes i CSS. Bare ærgeligt IE ødelægger brugen af de fleste, men det er rart at få overblik over understøttelsen alligevel. Måske lidt synd du ikke har :target med under CSS3 - den er ellers meget sjov at bruge ;-)

#### **Kommentar af htm d. 13. Mar 2004 | 11**

Fremragende arktikel. Jeg anede virkelig ikke der var så store muligheder med css. Dejlig arktikel.

#### **Kommentar af athlon-pascal d. 15. Mar 2004 | 12**

God, lang, grundig artikel, der giver et godt indblik i mere avanceret anvendelse af CSS. Når der står, hvor vidt de enkelte eksempler virker i IE 3 (der slet ikke understøtter CSS), bør der så ikke også stå, om de virker i

Netscape 4? \_

Du bør også evt. få fat i en nyere Knoppix Linux-cd (helst med nyeste KDE), og teste eksemplerne i Konqueror - hvis de virker der, virker de nok også i Safari til MacOS X (alt andet ville være mærkeligt) :o) \_  
Desuden kan du godt skrive ved alle eksemplerne, at de ikke virker i Lynx ;o)

#### **Kommentar af thesurfer d. 31. Mar 2004 | 13**

100%!

-- Det fede er at man kan bruge CSS3 til at blokkere reklamer, med FireFox -- Se <http://texturizer.net/firefox/adblock.html>

#### **Kommentar af michael-k d. 14. Mar 2004 | 14**

Kan kun være enig med de andre. Dette er bestemt noget man kan bruge til noget. Tak for det.

#### **Kommentar af mm12010 d. 12. Dec 2004 | 15**

#### **Kommentar af jensendk d. 29. Jun 2004 | 16**

Grundig og velskrevet

#### **Kommentar af stich d. 24. Jan 2006 | 17**

#### **Kommentar af dripz d. 18. Mar 2005 | 18**

God artikel. Det er bare synd at IE ikke forstår en pølse af det meste I denne artikel :(

#### **Kommentar af viciodk d. 14. Mar 2004 | 19**

Thumbs up :)

#### **Kommentar af lordhead d. 13. Mar 2004 | 20**

En artikel der absolut er værd at gemme i sine bookmarks, og tage frem hvis man bliver i tvivl om noget ;) Jeg kan personligt ikke huske alt hvad der er blevet skrevet da artiklen i sig selv er meget lang. Ellers godt stykke arbejde. Dog skal du stadig huske den store mængde af brugere der bruger IE, netscape og andre browsere!

#### **Kommentar af dustie d. 09. Sep 2005 | 21**

"God artikel. Det er bare synd at IE ikke forstår en pølse af det meste I denne artikel :("

Hvad har det med artiklen at gøre?

Fordi IE sutter betyder det ikke artiklen er dårlig.

#### **Kommentar af dxs d. 08. Dec 2004 | 22**

Utrolig flot og velformuleret artikel om CSS!

**Kommentar af drip d. 21. Mar 2004 | 23**

Super Duper oversigt over funktioner i css!

**Kommentar af slawomir d. 13. Mar 2004 | 24**

**Kommentar af jokkejensen d. 16. Apr 2007 | 25**

En skam ie7 ikke er med i compabilitets skemaerne... hvorfor ik ?