

Slideserie 2: Introduksjon til CSS

**PRO100/Kreativt webprosjekt
Rolando Gonzalez
Høst 2018**

Innhold slideserie

- CSS definisjon
- CSS oppbygning
- Klasser
- ID

CSS-definisjon

- “Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g. fonts, colors, spacing) to Web documents*.”

Kilde: www.w3c.org

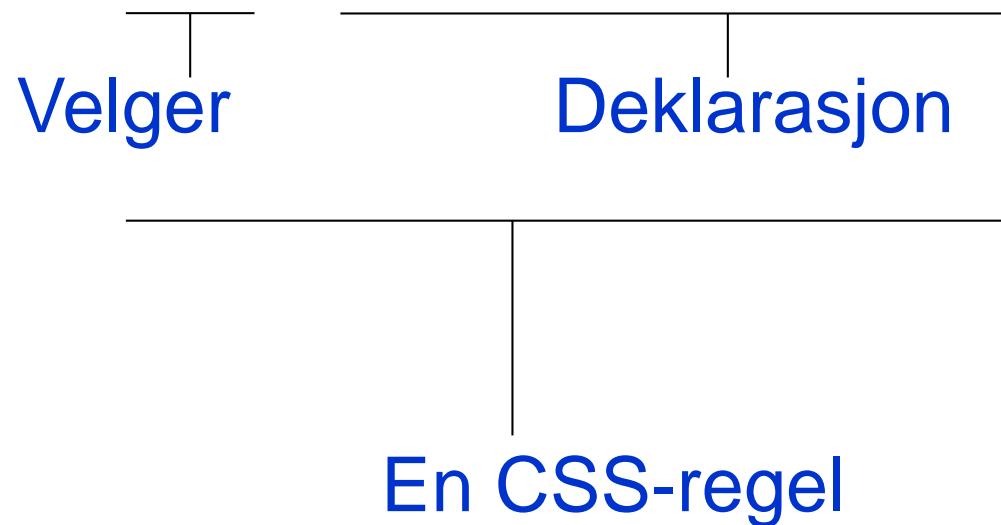
*Med 'Web documents' menes HTML-filer

CSS Oppbygging 1

Anatomien til en regel:

CSS-regel = Velger + Deklarasjon

h1 { color : red }



CSS – Oppbygging 2

Anatomien til en deklarasjon:

Deklarasjon = Egenskap + Verdi

h1 { color : red }

Egenskap

Verdi

Deklarasjon

CSS-kode kan hentes inn i HTML på forskjellige måter:

- Inline (den minst optimale måten*, **):

- `<h1 style = "color:#CC0000">Overskrift</h1>`

- Internal:

`<head>`

`<style type = "text/css">`

`h1{color: #CC0000;}`

`</style>`

`</head>`

- External (den optimale måten*):

- Linker til en fil lagret som .css. Plasserer denne koden i <head>:

- `<link rel="stylesheet" href="stil1.css" type="text/css"/>`

*Med utgangspunkt i kodeprinsipper om kodegjentakelse og kodeseparasjon.

**Brukes kun om nødvendig.

Inline style

- Eksempel:

```
<h1 style ="color:#CC0000; background-  
color:yellow;">
```

Hver regel skilles med semikolon!

Ekstern css-fil:

- Med ekstern CSS-fil menes en fil som bare inneholder CSS.
- CSS-filen skal ha .css-endelse

Innholdet i filen kan for eksempel se slik ut:

```
@charset "utf-8";  
/* CSS-dokument */  
h1{color: #CC0000;}  
p{ font-style: italic}
```


CSS-kommentar

- Kommentarer omslutes med /* og */

...

```
/*Dette er en kommentar*/
```

```
h1{color: rgb(20, 20, 20);}
```

...

Eksempel bruk av CSS-kode

...

Ekstern fil:

```
h1{
  color:blue;
  text-align:center;
}
p{
  color:red;
  text-align:center;
}
```

...

```
<h1>Dette er en overskrift</h1>
<p>Dette er et avsnitt (paragraph)</p>
```

....



Gjenbruk av kode

```
h1{  
color:blue;  
text-align:center;  
font-family: sans-serif;  
}
```

```
h2{  
color:blue;  
text-align:center;  
font-family: sans-serif;  
}
```

```
p{  
color:red;  
text-align:center;  
}
```

Kan endres til dette ->

```
h1, h2{  
color:blue;  
font-family: sans-serif;  
}  
p{  
color:red;  
}  
h1, h2, p{  
text-align:center;  
}
```

Gjenbruk av kode

- Merk at man ikke skal gjenbruke kode i alle tilfeller. Det kan være tilfeller hvor elementene som skal påvirkes er så annerledes at det ikke er formålstjenlig å samle kodeblokker for dem.

Noen CSS-egenskaper mht. tekst

- text-align <!--center, left, right-->
- font-family <!-- arial, times new roman-->
- font-weight <!-- bold -->
- font-style <!-- Tekst stil, for eksempel: italic -->
- background-color
- color <!-- Tekstfarge-->

Skape rom rundt element

"elementnavn">{margin: x x x x}

- Rekkefølge:

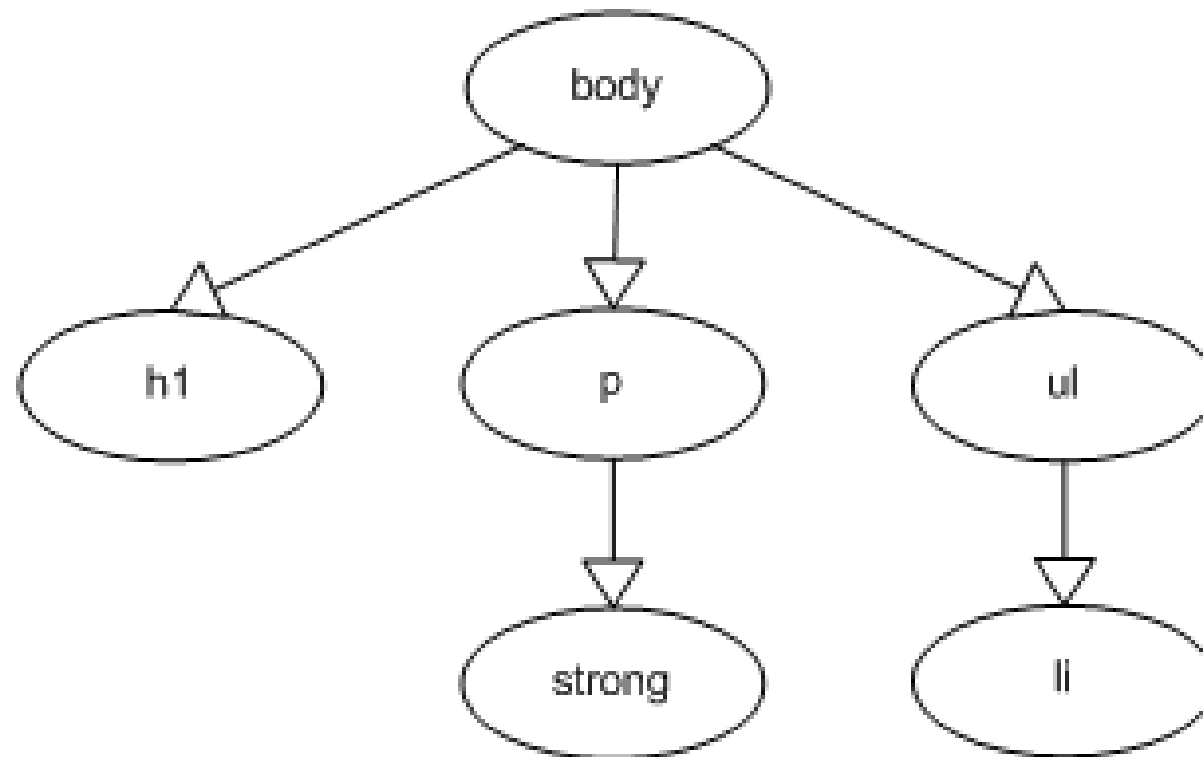
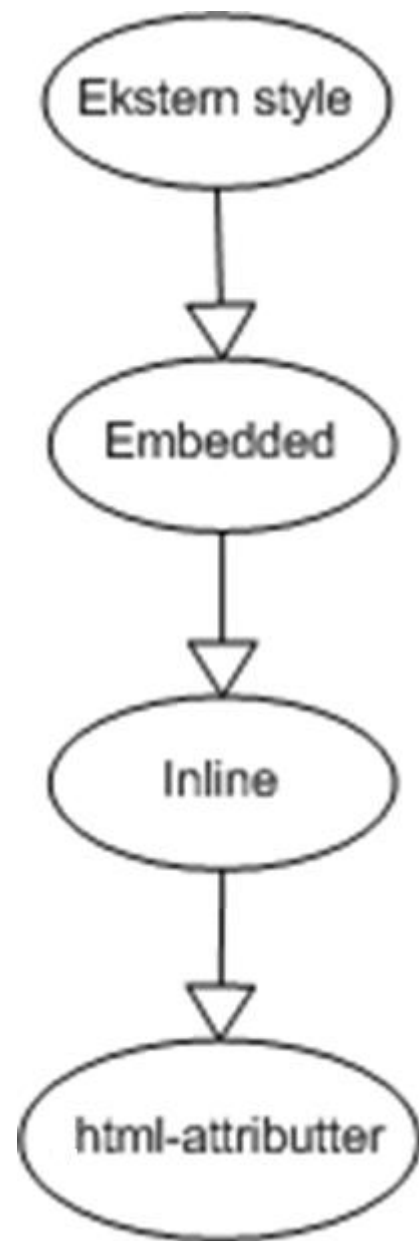
{margin: topp høyre bunn venstre}

- Hvis bare første verdi tilstede får alle marginene lik verdi
- Hvis et element er av typen "display: block" og har en bredde mindre enn viewport kan den midtstilles med *margin: auto;*



”Cascade” - CSS

Modellene viser i hvilken rekkefølge stilene applikeres. For hvert trinn overskrives forrige trinn.



Overstyre cascade

- Hvis en ikke ønsker at en stil skal overskrives nedover i cascade-hierarkiet kan en legge til !important etter en regel:

```
p{font-family: Times New Roman !important;}
```


Angående prioritet

- Både cascade og typen velger påvirker prioritet til en velger
- I følgende eksempel får eksternt stilark høyest prioritet og divisjonen med id "div1" blir 10em bred:

HTML-kode:

```
<div id="wrapper"><div id="div1"></div></div>
```

Eksternt stilark:

```
#wrapper div{width: 10em;}
```

Embedded stilark:

```
#div1{width: 20em;}
```

Hvordan blir h1 og p seendes ut?

HTML-filen:

```
<head>
  <title>ArvTest</title>
  <link rel = "stylesheet" type = "text/css" href = stil1.css" />
  <style type = "text/css">
    h1{font-style: normal;
      color:red;
    }
  </style>
</head>
<body>
  <h1>Dette er tittelen</h1>
  <p>Dette er et avsnitt</p>
</body>
</html>
```

stil1.css-filen:

```
body{
  font-style: italic;
  color:blue;
}
```

Fargesetting med css

- Det finnes flere måter å sette fargeverdi:
 - Fargenavn (engelsk)
 - RGB-kode (Red, Green, Blue)
 - Hex-kode

Les mer her:

http://www.w3schools.com/cssref/css_colors_legal.asp

Fargesetting eksempler

- Fargenavn
 - {color: red;}
- Hex
 - {color: #FF0000;}
 - {color: #F00;} <!-- - Forkortelse av hex
parene - ->
- RGB
 - {color: rgb(255,0,0);}

CSS-velgere: type-velger

- Typevelger er navnet på de velgerne som har samme navn som HTML-elementene de referer til:

h1{color:blue}

p{font-size:20pt}

li{font-style:italic}

CSS-velgere: klasse-velger

- I noen tilfeller er det flere samlinger (i eksempel under spørsmål og svar) av elementer som vi ønsker å gi forskjellige stiler til:

`<p style = "font-style:italic; color: red;"/>Hva er dette?</p>`

`<p style = "font-weight:bold"; color: blue;/>'Dette' er et ord.</p>`

`<p style = "font-style:italic"; color: red;/>Hva med dette?</p>`

`<p style = "font-weight:bold; color: blue;"/>Hva mener du egentlig med 'dette'?</p>`

- Kan ikke bruke type-velger her da alle p vil bli like. Vi ønsker å bruke klassevelger for å gi utvalg av elementer visse stiler -> se neste slides

Hvordan opprette en klasse?

- Mal:

```
.klassenavn{ /*Klassenavnet er valgfritt*/  
    ...  
}
```

Hvordan opprette en klasse?

- Eksempel:

```
.spoersmaal{  
    font-weight:bold;  
    color: red;  
}
```

```
.svar{  
    font-style:italic;  
    color: blue;  
}
```


Hvordan benytte klassen?

<h3 class = "spoersmaal">What is CSS?</h3>

<p class = "svar">CSS stands for Cascading Style Sheets and is a simple styling language which allows attaching style to HTML elements.</p>

<h3 class = "spoersmaal">What are Style Sheets?</h3>

<p class = "svar">Style Sheets are templates, very similar to templates in desktop publishing applications, containing a collection of rules declared to various selectors (elements).</p>

CSS-velgere: Id

- id attributtet fungerer som klasser bare at hver id må være unik og kun kan brukes én gang per side
- Id kan brukes til minst to ting:
 - hjelper oss å style ett element på en viss måte
 - med et logisk navn gjør det det lettere å forstå et html-elements rolle på siden:
`<section id=«bildegalleriSection">`

Hvordan gi id til element

```
<body>
```

```
<section id="omHTMLSection">
```

```
  <h1 id="omHTMLH1">Om HTML</h1>
```

```
  <p>Avsnitt</p>
```

```
</section>
```

```
<section id="omCSSSection">
```

```
  <h1 id="omCSSH1">Om CSS</h1>
```

```
  <p>Avsnitt</p>
```

```
</section>
```

```
</body>
```

Hvordan referere til en id

```
#omHTMLSection{  
    padding: 1em;  
}  
#omHTMLH1{  
    font-family: helvetica;  
    font-weight: bold;  
    text-decoration: underline;  
}
```

CSS: Kombinere velgere

- Velgere kan kombineres slik at man kan med mer nøyaktighet spesifisere hvilke elementer som skal bli påvirket av en regel.

Velger-kombinasjon på klasser

```
p.spoersmaal{  
  font-weight:bold;  
  text-transform: uppercase;  
}
```

Betyr at to krav må være innfridd: Må være av klassen "spoersmaal" OG være en p

Eksempel kombinasjon p.k1

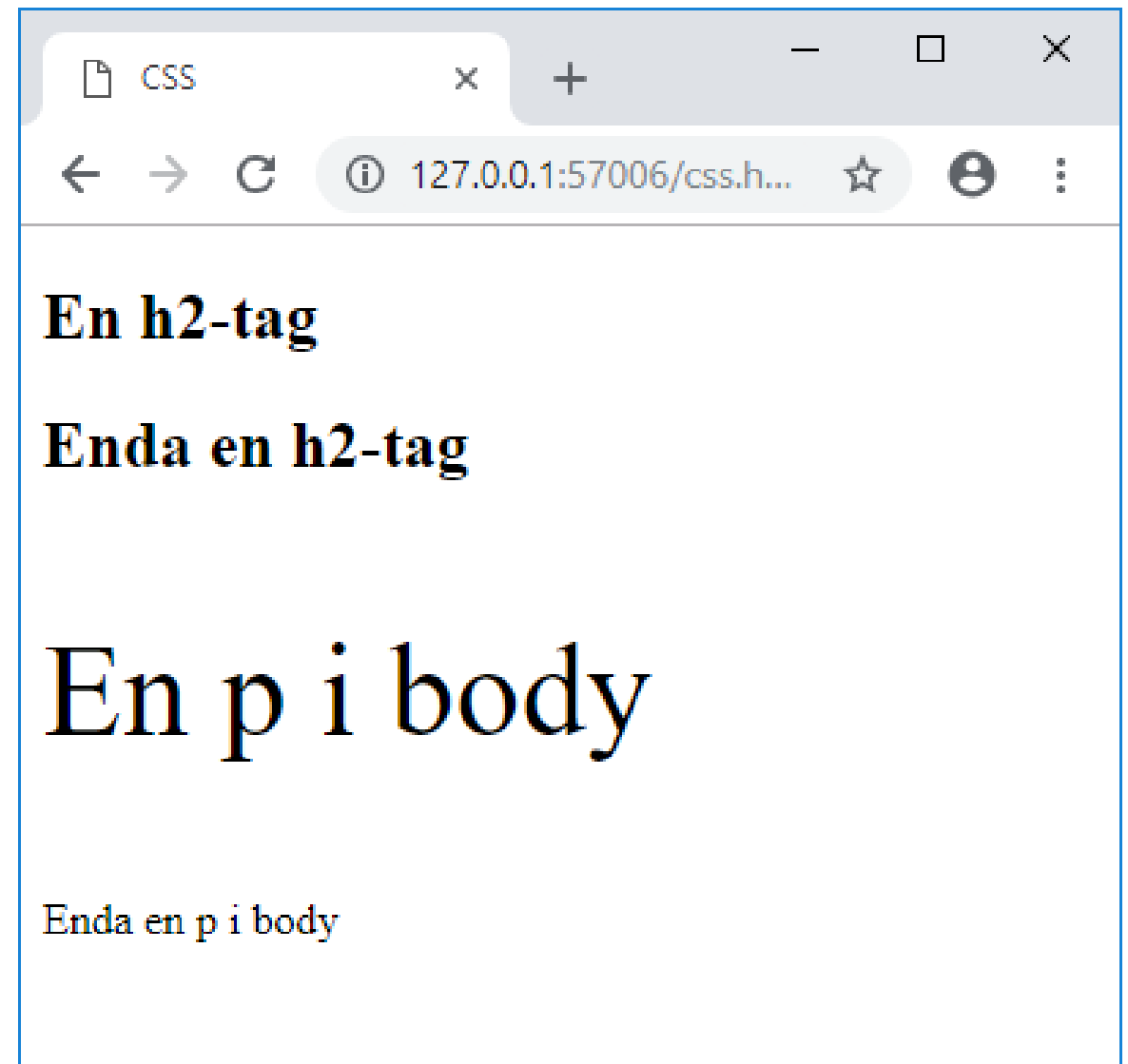
...

```
p.stor-tekst{  
  font-size: 3rem;  
}
```

...

```
<h2 class="stor-tekst">En tittel</h2>  
<h2>Enda en tittel</h2>  
<p class="stor-tekst">En p i body</p>  
<p>Enda en p i body</p>
```

...



Kun p med klasse stor-tekst påvirkes.

Kombinasjonsvelgere: Avkom

```
<head>
```

```
  <title>CSS</title>
```

```
  <style type = "text/css">
```

```
    h1 em{
```

```
      background: red;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
<h1>Dette er en setning med <em>ett</em> ord som er  
  emphasised</h1>
```

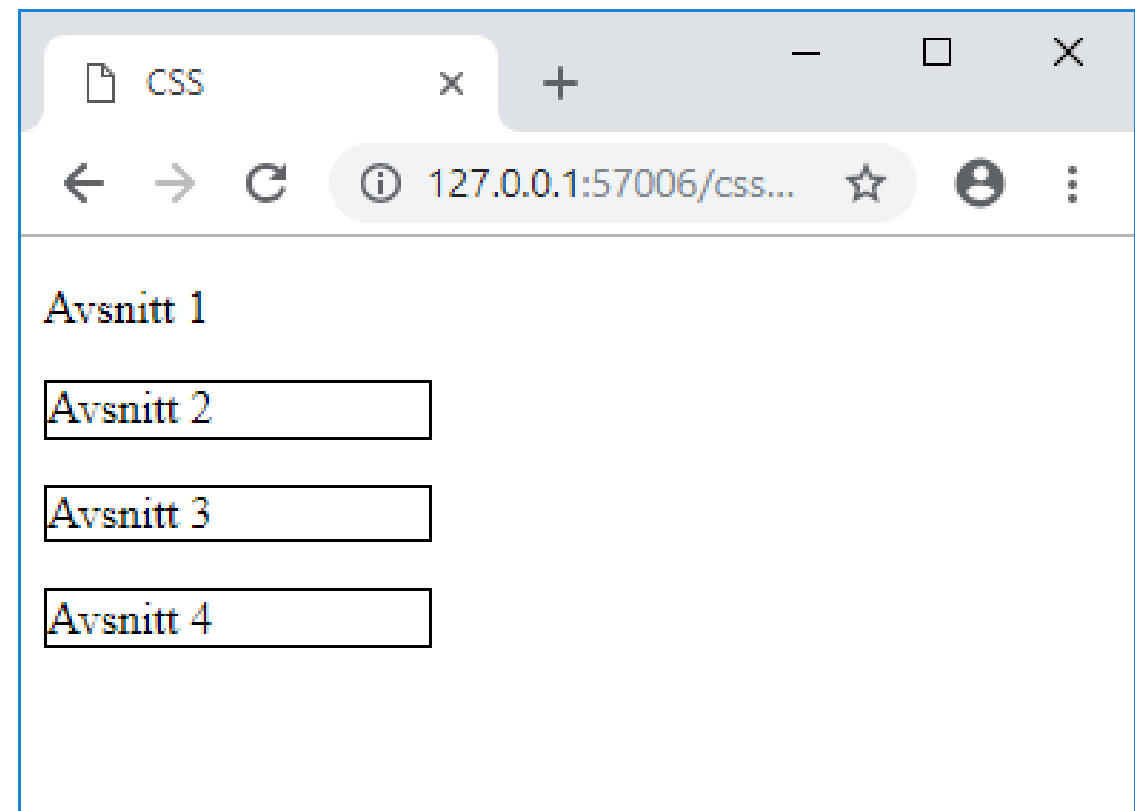
```
</body>
```



- Denne typen velger kalles også for kontekstuell velger.

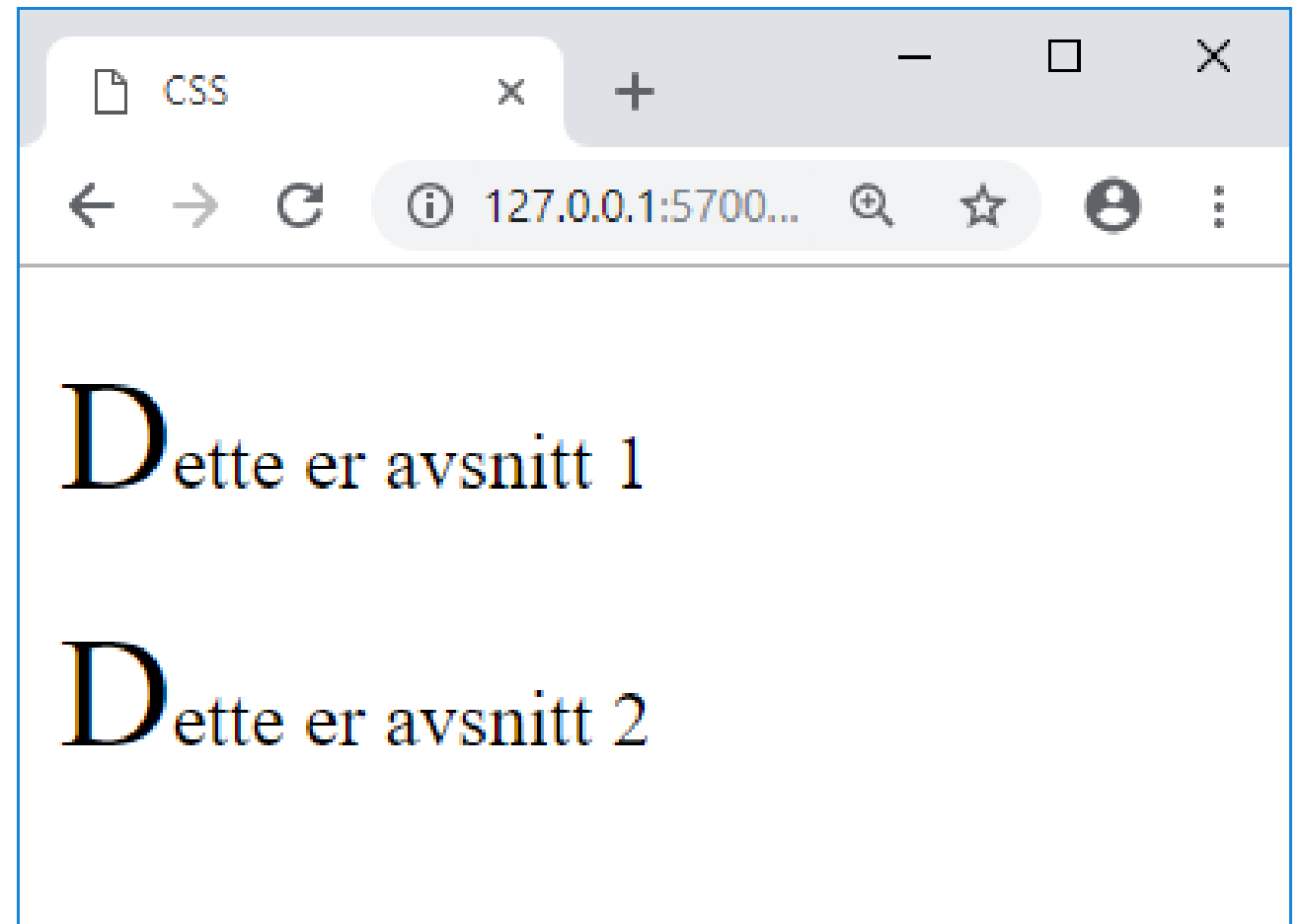
Kombinasjonsvelgere: søsken-velger

```
<style type = "text/css">
  p + p{
    width:100pt;
    border-style: solid;
    border-width: 1pt;
  }
</style>
</head>
<body>
<p>Avsnitt 1</p>
<p>Avsnitt 2</p>
<p>Avsnitt 3</p>
<p>Avsnitt 4</p>
```



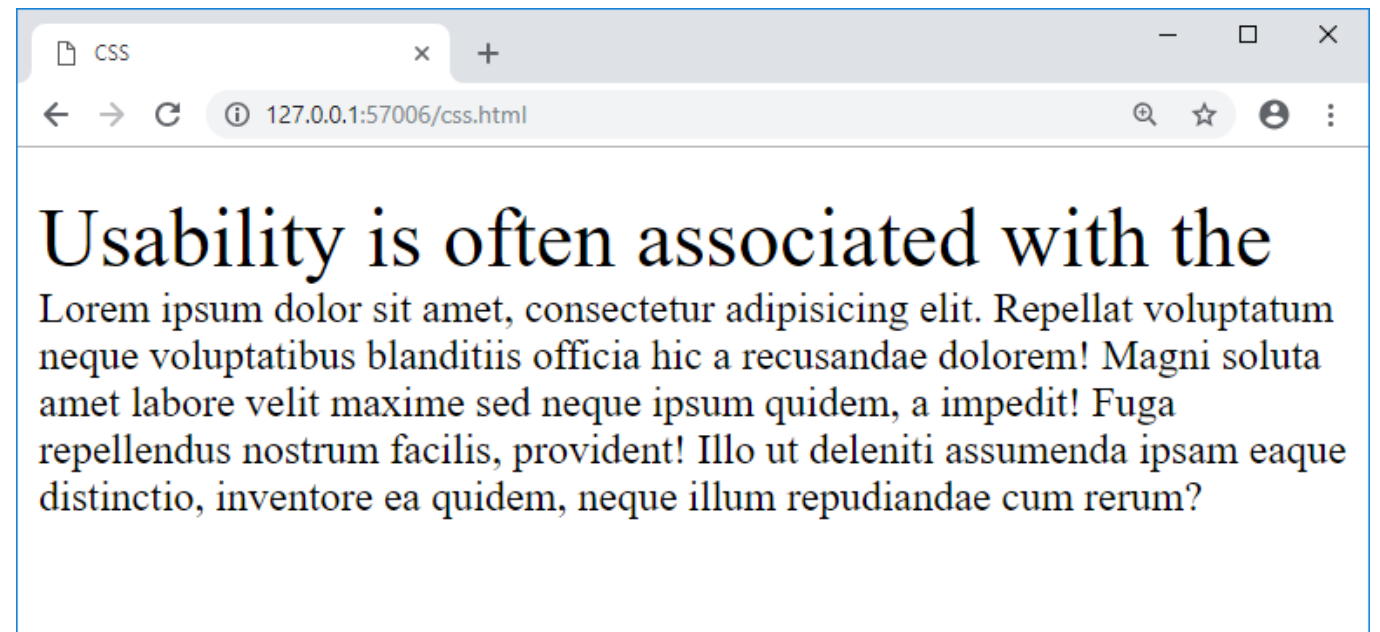
Kombinasjonsvelgere: Pseudo

```
<style type = "text/css">
  p:first-letter{
    font-size:xx-large;
  }
</style>
</head>
<body>
<p>Dette er avsnitt 1</p>
<p>Dette er avsnitt 2</p>
...
```



Kombinasjonsvelgere: Pseudo

```
<style type="text/css">
  p:first-line{
    font-size:xx-large;
  }
</style>
</head>
```



```
<body>
<p>Usability is often associated with
the.....
```

Kombinasjonsvelgere:

Pseudo

```
li:first-child{  
    font-size:x-large;  
}
```

```
<ol>
```

```
    <li>Velg det invitasjonsdesignet du ønsker.</li>
```

```
    <li>Fyll ut skjemaet med når arrangementet  
starter, hvor det er osv.</li>
```

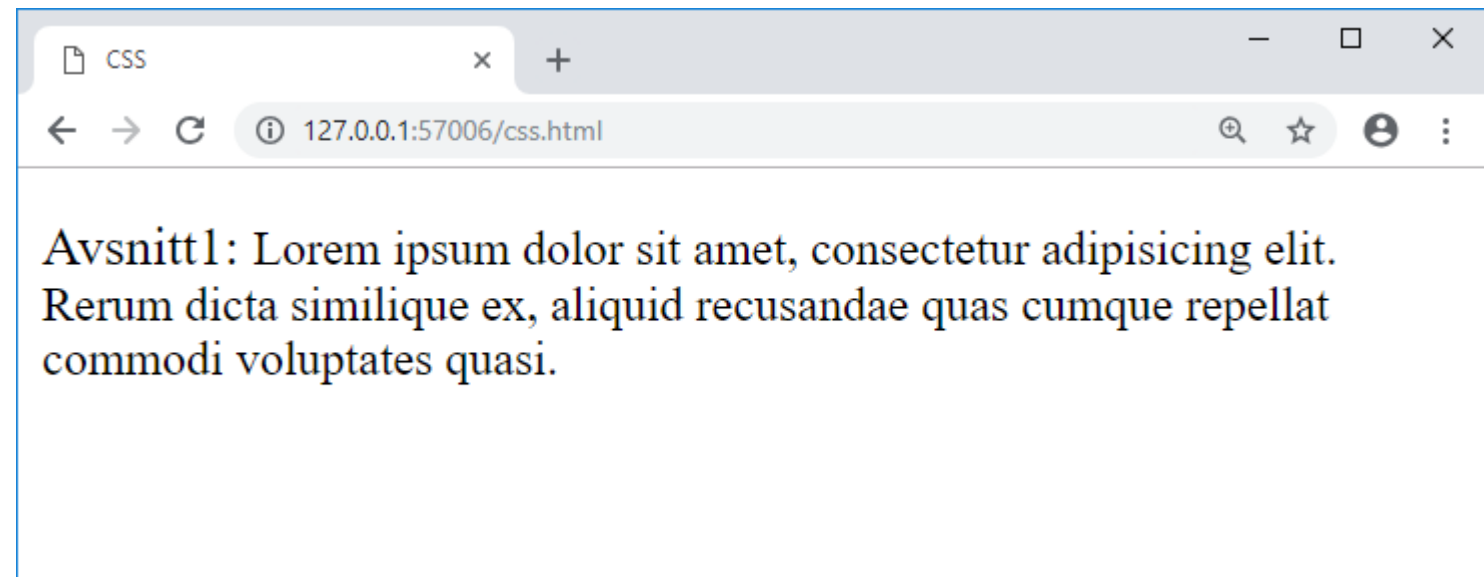
```
    <li>Lag gjestelisten og klikk send.</li>
```

```
</ol>
```



Kombinasjonvelgere: Pseudo

```
<style type = "text/css">  
  p:before{  
    content:"Avsnitt1: ";  
    font-size: large;  
  }  
</style>
```



Kombinasjonvelgere: Pseudo

```
<style>
```

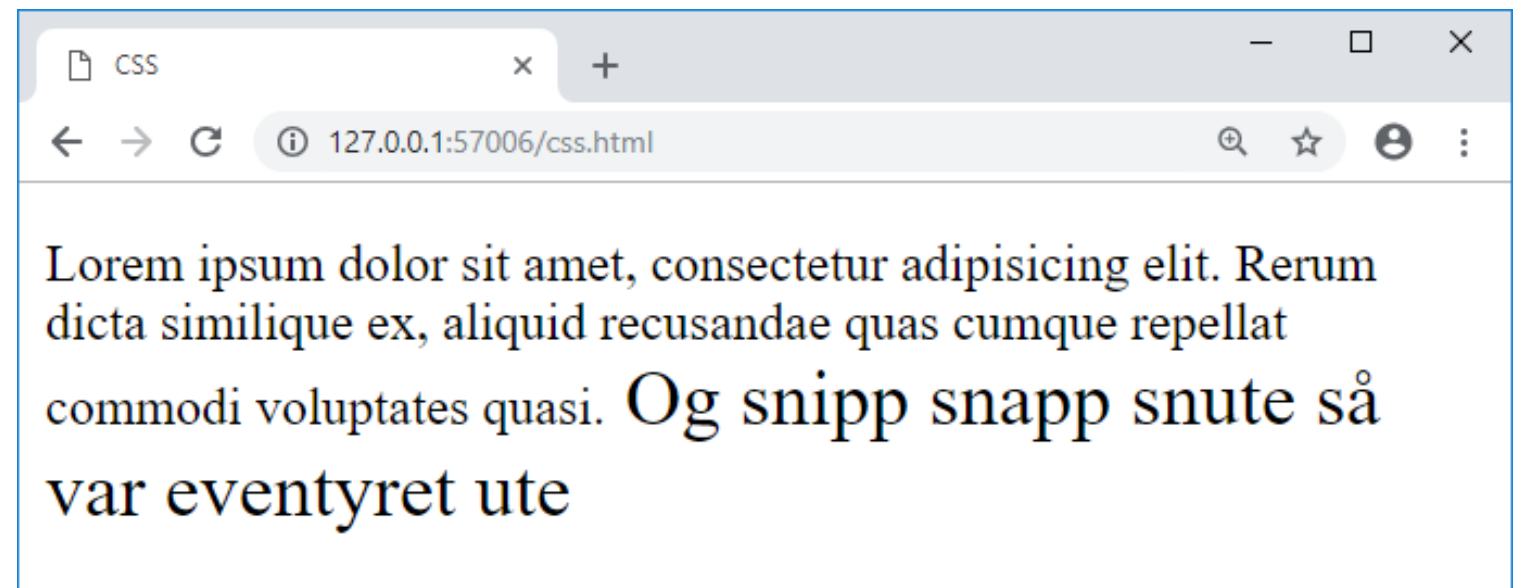
```
  .siste-avsnitt:after{
```

```
    content:" Og snipp snapp snute så  
var eventyret ute";
```

```
    font-size: x-large;
```

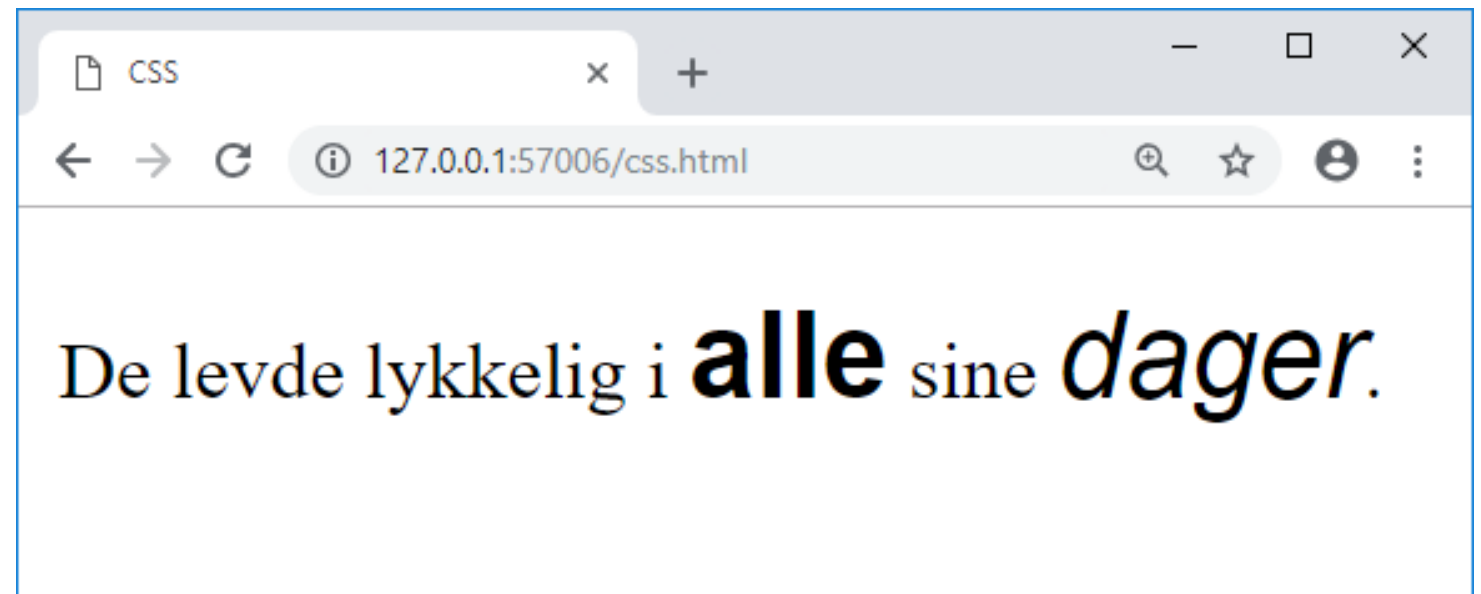
```
  }
```

```
</style>
```



Kombinasjonvelgere: Mor/barn

```
p > *{  
  font-size:x-large;  
  font-family: sans-serif;  
}
```



<p>De levde lykkelig i alle sine dager.</p>

Attributtvelgere

Eksempler:

HTML-elementer med attributtet border:

`[border]{stil}`

Bilder som har verdien "Katt" i attributtet alt:

`[alt="Katt"]{stil}`

Bilder som har et alt attributt hvor man finner ordet Katt:

`[alt~="Katt"]{stil}`

CSS 3

attributtvelgere

- Attribute

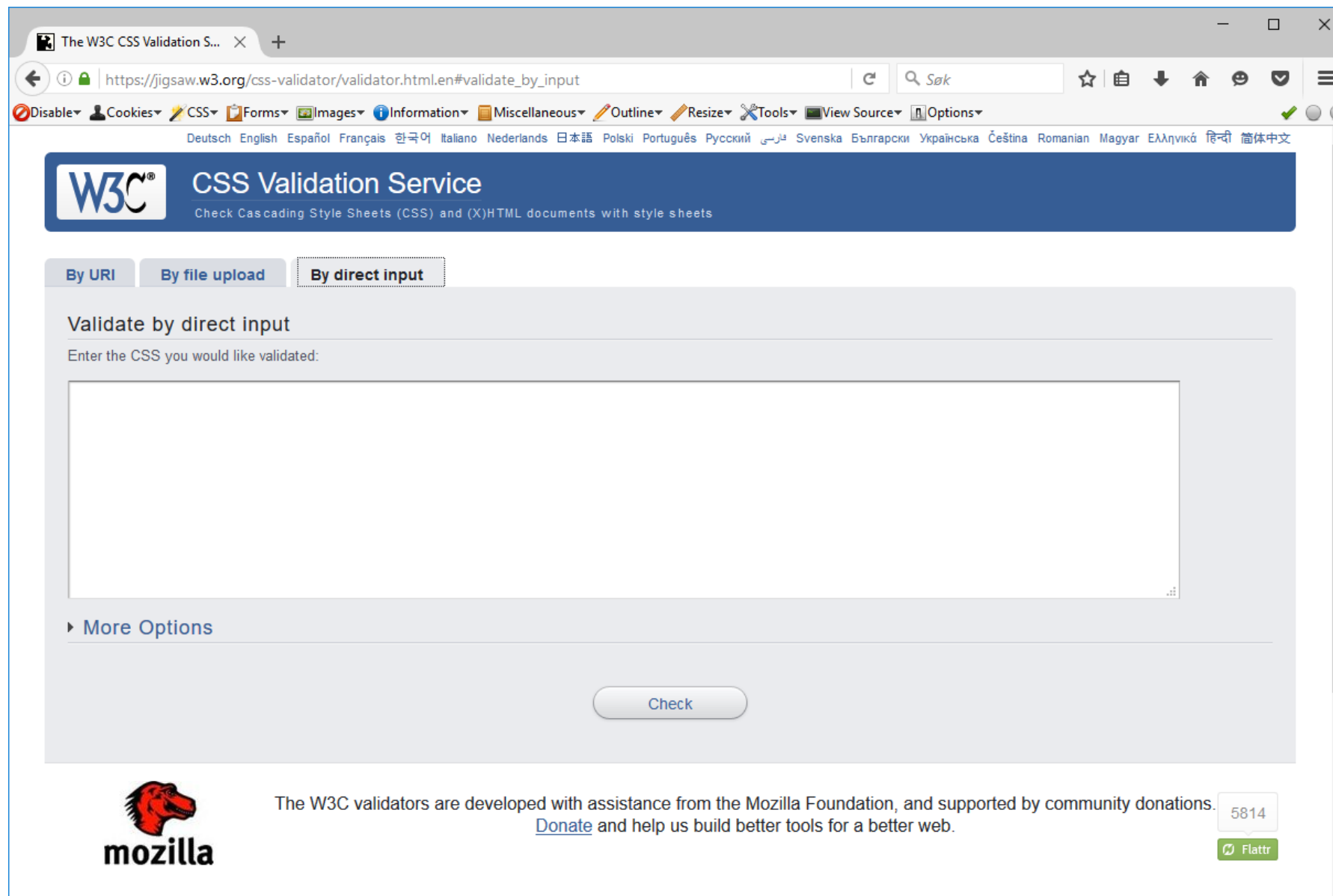
selector[attribute^=value] : starter med value

selector[attribute\$=value] : slutter med value

selector[attribute*=value] : inneholder substring
med value

CSS-Validator

- https://jigsaw.w3.org/css-validator/validator.html.en#validate_by_input



The screenshot shows the W3C CSS Validation Service interface in a web browser. The browser's address bar displays the URL https://jigsaw.w3.org/css-validator/validator.html.en#validate_by_input. The page features a blue header with the W3C logo and the text "CSS Validation Service" and "Check Cascading Style Sheets (CSS) and (X)HTML documents with style sheets". Below the header, there are three tabs: "By URI", "By file upload", and "By direct input", with "By direct input" being the active tab. The main content area is titled "Validate by direct input" and contains the instruction "Enter the CSS you would like validated:". Below this is a large, empty text input field. At the bottom of the input area, there is a link "More Options" and a "Check" button. The footer of the page includes the Mozilla logo, a statement about the W3C validators being developed with assistance from the Mozilla Foundation, a "Donate" link, and a Flattr button with the number 5814.

The W3C CSS Validation Service

Check Cascading Style Sheets (CSS) and (X)HTML documents with style sheets

By URI By file upload By direct input

Validate by direct input

Enter the CSS you would like validated:

More Options

Check

The W3C validators are developed with assistance from the Mozilla Foundation, and supported by community donations. [Donate](#) and help us build better tools for a better web.

5814

Flattr

span

- "Span of words".
- HTML-tag som brukes når du ønsker å gi deler av en tekst en stil.

<p> Dette er en
paragraf!</p>

(W3.org, 2008)

Posisjonering

Posisjonering: position

- Static (default)
 - Relative
 - Absolute
 - Fixed
-
- Kan brukes på alle elementer!

Static

- Default ved tilføyning av elementer på en side
- Elementer legges default fra øvre venstre hjørne og nedover (block) eller etter hverandre (inline) uten linjeskift.

Relative, absolute og fixed

- Bortsett fra static bruker de andre posisjoneringene egenskapene: top, right, bottom og left
- Verdiene som kan brukes til å angi plassering er blant annet em, % og px.

Posisjonering: Absolutt

- Absolute tar ikke hensyn til hvor andre elementer på skjermen ligger
- Absolute tillater overlapping av elementer. Eksempel: hvis 4 bilde elementer settes til `img{position:absolute}` vil disse 4 komme overlappende på skjermen relativt til rekkefølgen de legges til i body.
- Posisjonerte elementer vil ligge over elementer uten position på skjermen.

Posisjonering: Relativ

- Ved å posisjonere et element relativt, tar man utgangspunkt i hvor elementet opprinnelig ville dukket opp og flytte det derfra.

Fixed

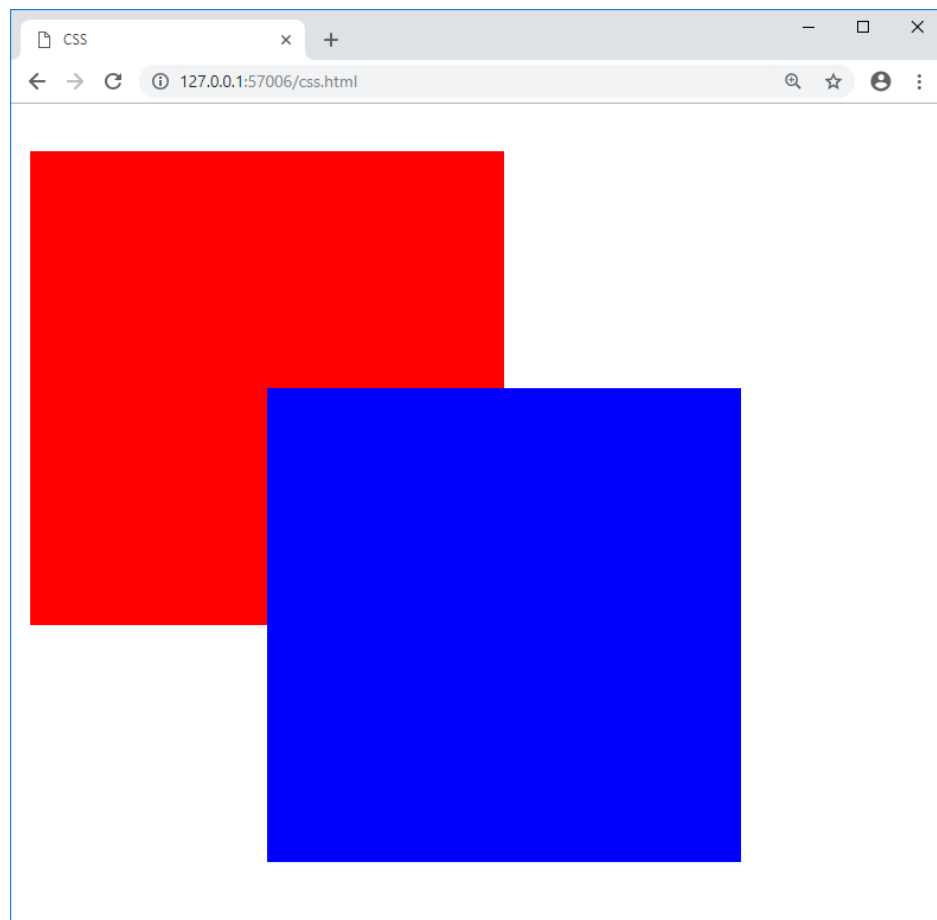
- Gjør at element befinner seg på samme sted på skjermen selv om man scroller ned/opp.
- Kan ha problemer i forhold til noen IE versjoner
- Ved utskrift av web dokumenter vil fixed elementer komme på alle sider på samme plass.

Posisjonering: Nøsting av elementer

- Man kan ha en div med en div i seg. Man kan da posisjonere den indre i forhold til den ytre.

HTML:

```
<div id="div-red"><div id="div-blue"></div></div>
```



CSS:

```
div{
  width: 200px;
  height: 200px;
  position: absolute;
}
#div-red{
  top: 100px;
  background-color:red;
}
#div-blue{
  left: 100px;
  background-color:blue;
}
```

Position og Z-index

- Kan brukes for å få et element til komme bak/foran et annet posisjonert element.
- Man bruker en tallverdi (negativ eller positiv) som settes for å angi z-index til elementer relativt til hverandre. Jo høyere verdien jo lengre frem plasseres elementet i forhold til de andre elementene.