

CSS

Cascading StyleSheets

Agenda

- Intro till css
 - vad är det?
 - infoga css
- grundläggande syntax
 - selectors, grouping, etc
- div-taggen
- properties, padding, margin, m.m
- övning

men först

1. börja med CSS2 och lär dig grundläggande funktioner, syntax etc.
2. fortsätt med CSS3 för att kunna använda mer avancerad funktionalitet.
3. lär dig använda minst ett s.k ramverk för css; Twitter Bootstrap, Foundation etc
<http://foundation.zurb.com/>
<http://getbootstrap.com/>
4. undersök extensioner (språk) till css såsom exempelvis sass <http://sass-lang.com/>

vad är css till för?

- Separerar layout från struktur;
 - HTML för struktur, CSS för utseende/layout/positionering.
- Används för att formatera html-elementen, genom att ange specifika format-regler.
- Med dessa regler kan vi formatera elementens olika attribut; t ex: typsnitt, ramar, färger, bakgrunder, position, m.m.

css3

css är idag i version 3.

Infoga css

dvs, hur man lägger till css till vår html-fil.

4 olika sätt att infoga css

- **Inline** style sheet
För ett specifikt avsnitt i hemsidan - direkt i taggarna med style-attributet.
- **Internal** style sheet
Inom head-sektionen - reglerna definierade inom `<head> </head>`
- **Extern** style sheet
Från en fil med bara CSS-kod (.css) med `<link>`-taggen.
Görs i headsektionen
- **Span**
över ett specifikt avsnitt i html-filen

Inline stylesheet

Med inline infogar vi css direkt i taggarna med style-attributet.

```
<html>
<head>
  <title>Om hemsidan</title>
</head>
<body style="background-color:black;">
  <a href="index.html">Förstasidan</a>
  <a href="about.html">Om hemsidan</a>
  <a href="http://www.google.se">Google</a>

  <h1>Om hemsidan</h1>

  <a href="http://google.se">
    
  </a>

  <p style="position:absolute; left:100px; top:300px; color:white;">
    Info om hemsidan lalalalalalalalal och la.
  </p>
</body>
</html>
```


Inline - pros & cons

Med style-attributet i taggarna kan vi påverka utseende för *en* specifik tagg/element.

- **Fördelar:**

- Samma (typer av) taggar kan ha olika utseende.

- **Nackdelar:**

- Blandar HTML med CSS -> filerna blir snabbt stora och oöverskådliga.
- Uppdateringsproblem -> måste ändra på flera platser i flera filer.

Internal stylesheet

Inom <head> skapar vi en intern stylesheet som kan användas i html-dokumentet.

```
1 <html>
2 <head>
3   <title>Om hemsidan</title>
4
5   <style type="text/css">
6     body { background-color: green; }
7     p { position: absolute; left: 100px; top: 300px; }
8   </style>
9
10 </head>
11
12 <body>
13   <a href="index.html">Förstasidan</a>
14   <a href="about.html">Om hemsidan</a>
15   <a href="http://www.google.se">Google</a>
16
17   <h1>Om hemsidan</h1>
18
19
20   <a href="http://google.se">
21     
22   </a>
23
24
25   <p>
26     Info om hemsidan lalalalalalalalal och la.
27   </p>
28 </body>
29
30 </html>
```

Internal - pros & cons

Med internal infogar vi ett stylesheet som ett internt element med style-taggar i head-sektionen.

- **Fördelar:**

- Återanvändning i dokumentet
- Något separerad kod, HTML/CSS

- **Nackdelar:**

- Fortfarande blandad kod
- Uppdateringsproblem

span

Med span kan vi tillämpa en stylesheet över ett eller flera avsnitt.

```
<html>
<head>
  <title>Om hemsidan</title>

  <style type="text/css">
    .blue { color:blue; }
    .green { color:green; }
    .red { color:red; }
    .yellow { color:yellow; }
  </style>
</head>
<body>
  <a href="index.html">Förstasidan</a>
  <a href="about.html">Om hemsidan</a>
  <a href="http://www.google.se">Google</a>

  <h1>Om hemsidan</h1>

  <a href="http://google.se">
    
  </a>

  <p>
    Info om <span class="red">hemsidan</span> lalalalalalalalal och la.
  </p>
</body>
</html>
```

Span - pros & cons

Fördelar:

vi kan tillämpa stylesheets över vissa avsnitt.

Nackdelar:

allt vi tidigare sagt * 2

External

Vi placerar all css-kod i en separat fil (.css). Denna fil länkar vi sedan in i head.

```
<html>
<head>
  <title>Om hemsidan</title>
  <link rel="stylesheet" href="default.css" type="text/css" />
</head>
<body>
  <a href="index.html">Förstasidan</a>
  <a href="about.html">Om hemsidan</a>
  <a href="http://www.google.se">Google</a>

  <h1>Om hemsidan</h1>

  <a href="http://google.se">
    
  </a>

  <p>
    Info om hemsidan lalalalalalalalal och la.
  </p>
</body>
</html>
```

External - pros & cons

Fördelar:

- Återanvändning
- Uppdateringar
- Separerar kod
- Flera .css-filer

Nackdelar:

- Fler filer på disken

Summa summarum

Vilket sätt att infoga *bör* man använda?

ALLTID* external!!!1!!1

HELST INTE !!!1!!!!!!1 1!

**) förutom vid väldigt små produktioner*

grundläggande css-syntax.

vi ska titta på några grundläggande koncept i css.

syntax == “uppsättningen tillåtna **reserverade ord** och möjlig tokenordning i ett **program**“

<http://sv.wikipedia.org/wiki/Syntax>

Hur man skriver css

- CSS syntax utgörs av tre delar:
 - Selector, Property och Value
 - det skrivs: `selector { property: value; }`
- **Selector** är ett html-element (tagg).
- **Property** är ett attribut man vill ändra (färg, storlek, typsnitt, position o.s.v.)
- **Value** är värdet man ger en property ("green", "20px" o.s.v.)
- Exempel: `h3 { width: 43px; height: 120px; }`

lära på egen hand

<http://www.w3schools.com/css/>

tutorials för både css2 och css3.

<http://www.codecademy.com/en/tracks/web>

html & css

<http://www.codecademy.com/courses/css-coding-with-style/0/1>

bara css

Grouping

- Grouping innebär att man samtidigt kan definiera regler för mer än en selector.
- Exempel:

```
h1, h2, h5, p {  
    color: pink;  
    background-color: yellow;  
}
```

- Varje selector separeras med , (kommatecken)

class selector (.)

Med class selector kan man definiera olika regler för samma html-element.

Syntax: `selector.className { ... }`

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

```
p.hej {color: green; }
```

- **html:**

```
<p class="minKlass">text text text</p>
```

```
<p class="hej">Grön</p>
```

```
<h1 class="hej">pålstek</h1> <!-- NEJ -->
```

class och grouping kombinerat

Om vi tillämpar både selectors och grouping kan vi uppnå:

```
p.hej h1, p.hej h2, p.hej h3 {  
  color: red;  
  font-size: 14px;  
}
```

html:

```
<p class="hej">  
  <h1>detta</h1><h2>kommer bli</h2><h3>samma</h3>  
</p>
```

till sist om class

Vi kan skapa 'fristående' class och tillämpa dessa på valfritt element.

```
.myClass { color: red; font-size: 14px; }  
.heading { color:black; }  
.links { text-decoration: none; color: pink; }
```

html:

```
<p class="myClass">  
  text text text.<br>text text. Lorem ipsum?  
</p>  
  
<h1 class=""heading>detta</h1>  
<a class="links" href="google.se">googla lite?</a>
```

id selector (#)

I stort sett samma funktionalitet som class.

Med id selector tillämpas reglerna för taggarna som har attributet id satt till samma namn i stilmallen.

```
p#minId { background-color: #FFFFFF; }  
#myId { color: yellow; font-family: Verdana;}  
#someText { width: 45px; height: 180px;}
```

- **html:**

```
<p id="myId">Vad fint det kan bli</p>  
<h4 id="myId">Rubrik</h4>  
<div id="someText"><p>Lite text här</p></div>
```


class eller id?

När ska man välja class och/eller id?

- id är tänkt för **ett** unikt element.
- class är tänkt för **flera** element

http://www.w3schools.com/css/css_id_class.asp

units

För de properties vars värden kan anges som numeriska värden (läng, höjd, bredd, avstånd etc) är följande enheter giltiga:

- % - procent
- in - inch(2,5cm)
- cm - centimeter
- mm - millimeter
- em - 1 em är samma som font-size i elementet.
- ex - 1 ex halva font-size
- pt - point (1/72 inch)
- pc - pica (12 points)
- px - pixel

div-taggen

div

- `<div>` - ett element/tagg som kan betraktas som en box i vilken man kan placera andra element..
- Div-taggen är alltså mycket användbar tack vare att man kan lägga andra element/taggar inuti en div och därigenom formatera allt med css.

exempel

```
#myDiv {  
    color: red; border-width: 2px; border-color: black;  
    position: absolute;  
    top: 250px; left: 450px;  
}
```

html:

```
<div id="myDiv">  
    <p>Här är lite text</p>  
     <br />  
    <h2>HEjsaN</h2>  
    <a href="bth.se">BeTeHå</a>  
</div>
```

kategorier

Allt man kan göra i CSS kan delas in i flera kategorier;

- styling
 - bakgrund, text, fonts, länkar listor, ...
- box model
 - kantlinjer, margin, padding
- dimension
 - storlekar
- positionering
 - placering
- pseudo-klasser och pseudo-element

Vi ska titta kort på några av dessa.

styling - background

Allt vad gäller bakgrund kan vi ställa in med css. För ett enstaka element eller för en hel sida (body).

- `background-color`
- `background-image`
- `background-repeat`
- `background-attachment`
- `background-position`

```
div {background-color:white;}
```

```
body {  
    background-image:url("bgpic.png");  
    background-repeat: repeat-x;  
}
```


Positioning

För att få våra element, t ex en div, på exakt det ställe vi vill använder vi positionering

```
div.test {  
    position: absolute; top: 25px; left: 400px;  
}
```

html:

```
<div class="test">  
    Lite text<br> lalalalala  
</div>
```

- Testa själva!

För att se en tydlig förändring kan ni lägga till bakgrundsfärg

forts positionering

`position: [relative, static, absolute]`

- `relative` - relativ position beroende på elementets position i förhållande till andra.
 - `static` - läggs bara till sidan i dess flöde
 - `absolute` - vi kan styra elementets absoluta position med `left`, `top` och `right`.
-
- `left` - avstånd från vänsterkant i webbläsaren
 - `top` - avstånd från överkant i webbläsaren
 - `right` - avstånd från från högerkant i webbläsaren

exempel positioning

```
#cls {  
    position: absolute;  
    left: 45px;  
    top: 145px;  
}
```

html:

```
<div id="cls"><p>Hey baberiba</p></div>
```

text och font

för text kan vi ställa in bl. a färg, stil och centrerering;

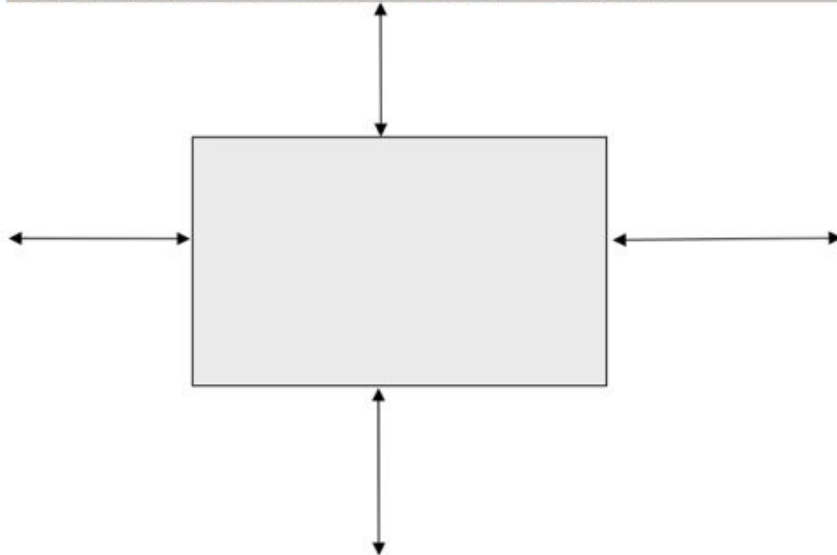
```
#txt {  
    text-decoration: underline; text-align: left;  
    text-color: blue;  
}
```

• Vad gäller typsnitt så kan vi bestämma bl.a vilket som ska användas samt storlek;

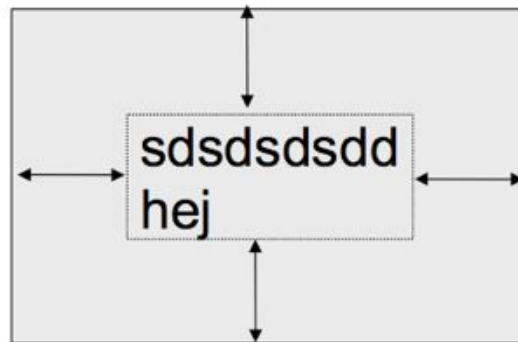
```
#txt {  
    font-family: Verdana, Helvetica, "Times new roman", serif;  
    font-size: 24px;  
}
```

Margin och padding

Med *margin* anges marginalerna runt elementet



Med *padding* anges marginalerna för innehållet i elementet



margin, padding

- Margin ställer in avstånd från elementets yttre kanter;

```
margin-left: 2px;
```

```
margin-right: 2px;
```

```
margin-top: 2px;
```

```
margin-bottom: 2px;
```

- samt att det finns s.k shorthands:

```
margin: 2px; 20px; /*top, bottom är 2px, left, right är 20px*/
```

- Padding ställer in marginalerna för innehållet i elementet från elementets innerkanter. Ser ut exakt som margin - byt ut margin mot padding.

- Finns också shorthands:

```
padding: 2px 20px;
```

Dimension

Properties som talar om elementens storlek.

Några exempel:

`width, height`

`max-height,`

`min-height, max-height, min-width, max-width`

- Möjliga värde för dessa properties;
 - `auto`
 - ***length*** /*använd en giltig unit*/
 - `%`
 - `inherit`

pseudo klasser

- Pseudo-klasser kan användas för att förändra enskilda properties för vissa element.
- Den vanligaste användningen av pseudo klasser är för att förändra 'default'-utseendet för länkar, <a>.

Syntax:

```
selector:pseudo-class {property:value;}
```

och med en egen klass:

```
selector.class:pseudo-class {property:value;}
```


exempel pseudo-klass

- För <a> gäller följande pseudo klasser:
a:link - obesökt länk
a:visited - besökt länk
a:hover - länk som 'hovras' (musmarkören är över)
a:active - aktiv/vald länk
- Exempel:
a { text-decoration: none; }
a:link {color:red;}
a:visited {color:red;}
a:hover {color:red;}
a:active {color:red;}

Liten övning

Gör en enkel html-sida med två stycken div-boxar.

I div-boxarna skriver du valfri text och infogar bilder.

Formatera div-boxarna och deras innehåll så att färger, typsnitt, ramar, position och storlek förändras.

Liten övning 2

Gör en enkel html-sida med 2st div-boxar.

- Placera samma text i 'divvarna'. Texten ska vara riktigt stor - minst 60px.
- Den ena textfärgen ska vara svart, den andra valfri färg.
- Placera diven med svart text under, fast något förskjuten så att den simulerar en skugga på texten.