

# Föreläsning 4

CSS – Stilmallar för webben



```
body {  
  font: x-small;  
  background: #  
  color: black;  
  margin: 0;  
  padding: 0;
```

# Innehåll

1. CSS, vad är det?
2. Repetition av HTML-element
3. CSS, hur, var, när?
4. CSS & arv?



Agenda!

Frågor så långt?

# CSS – Cascading Style Sheet

Same, same, but different



```
body {  
    font: x-small;  
    background: #  
    color: black;  
    margin: 0;  
    padding: 0;  
}
```



Samma innehåll  
– fast olika utseende



IA-dag vt2016





















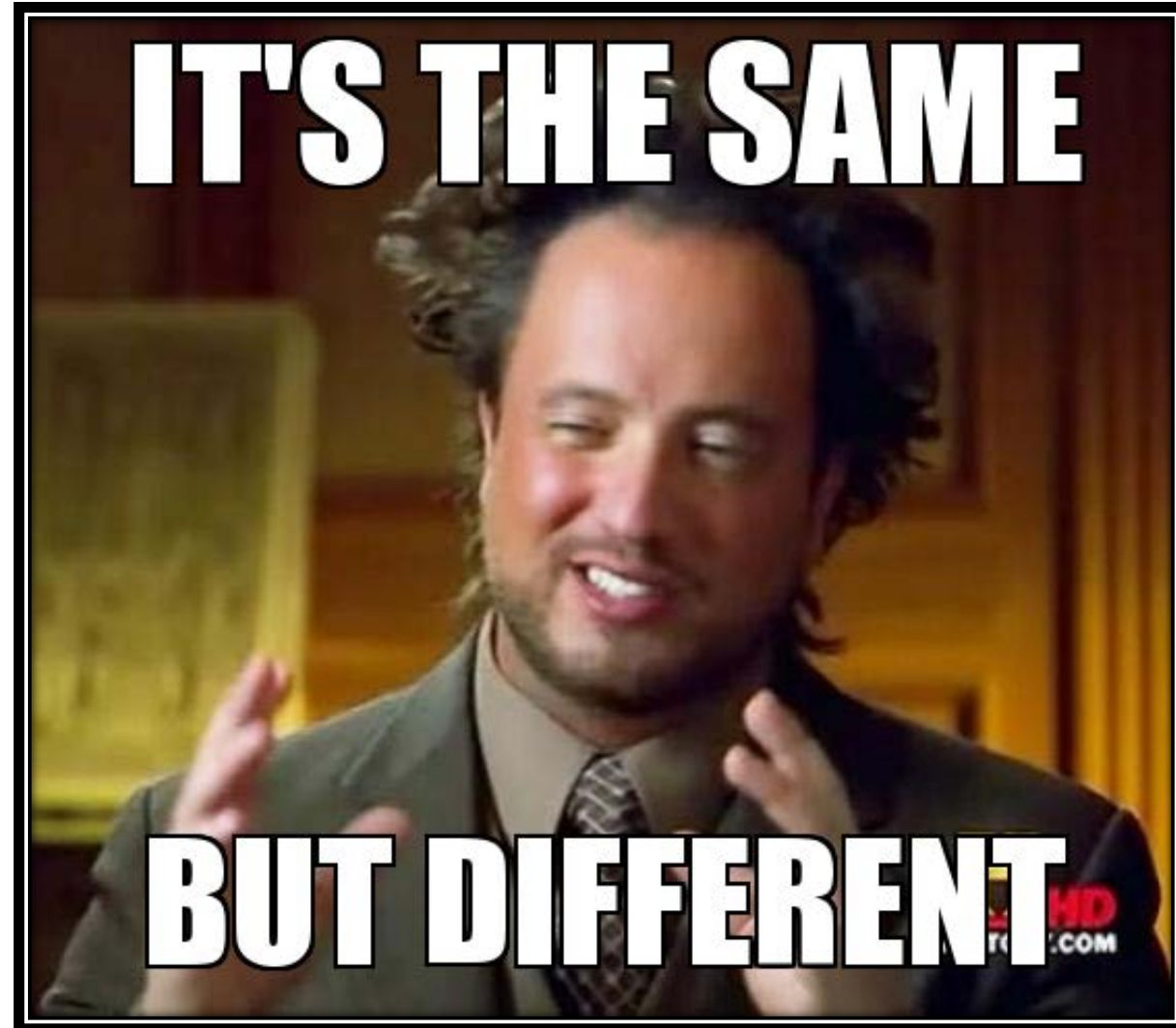


Samma innehåll  
– fast olika utseende

# Samma innehåll – fast olika utseende

- Produkter – olika utssende, samma innehåll







# Samma innnehåll – fast olika utseende

<http://www.csszengarden.com>

Cascading Style Sheets är ett språk som beskriver presentationsstilen för ett strukturerat dokument som till exempel typsnitt, textstorlek och färg.

# CSS – Snabb historik

- Hette ursprungligen Cascading HTML Style Sheets (CHSS)
- Håkon Wium Lie
- Döptes sedan om för att inte vara specifikt till HTML







## Har tre nivåer

- Nivå 1: Ursprung rekommendation från 1996
- Nivå 2 (2.1): 2007, primär version med bra (dock inte fullständigt) webbläsarstöd
- Nivå 3: Under aktiv utveckling. Kan användas, men har bara stöd i de nyaste webbläsarna.

# Kort om CSS

- CSS sköter formgivning av innehåll, medan HTML sköter struktur av innehåll.
- CSS styr hur HTML-elementen ska visas
- Typisk användning av CSS är för typografi, layout, bildhantering, animationer m.m.
- CSS är ett eget språk, men går att använda tillsammans med HTML
- Genom att koppla olika CSS-mallar till samma HTML-dokument kan informationen presenteras på olika sätt



# Varför CSS?

- **Det gör webbplatser grafiskt tilltalande**
- Det är smidigt att bygga upp layouter
- Det gör webbplatser lättare att underhålla
- M.m.
- Anpassa webbplatsen efter målgruppen
- Öka användbarhet
- Öka attraktivitet
- Optimera för användaren



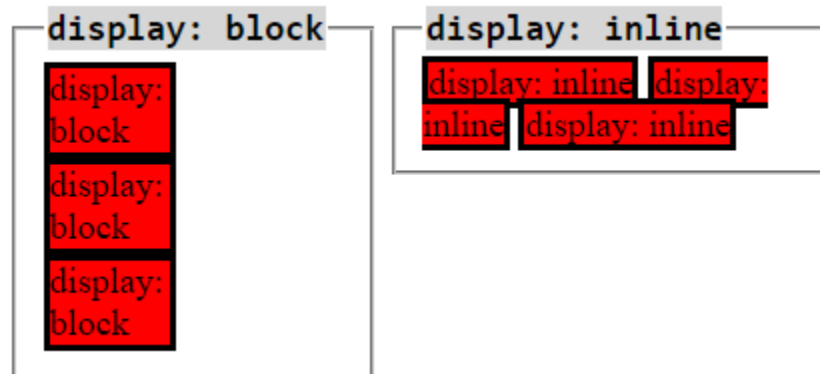
# HTML

Olika typer av element



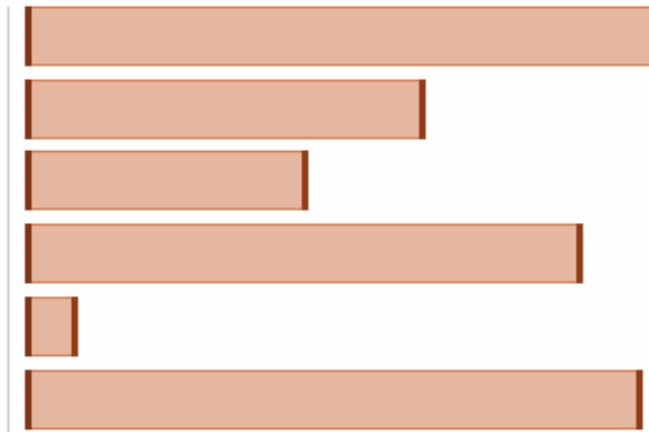
# Olika typer av element

- Det finns två huvudtyper av element (finns fler, mer om det sen).
  - Block
  - Inline
- Block-element visas generellt (utan styling från css) upp över hela skärmen horisontellt, som ett block
- Inline-element visas generellt (utan styling från css) endast upp där dess innehåll finns.

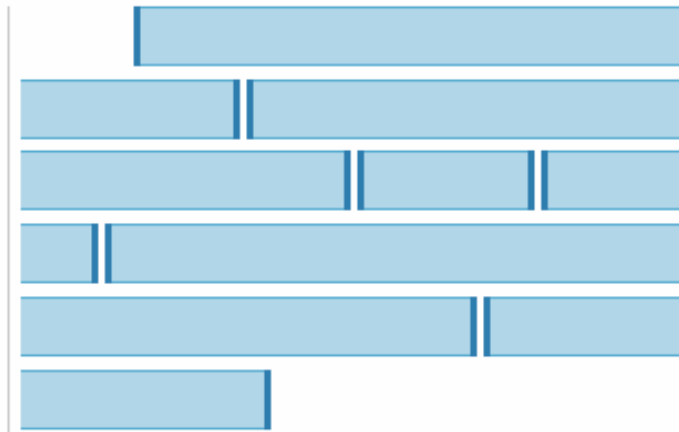


# Olika element

- Block-element för layout:
  - Section, article, div m.m.
- Block-element för innehåll
  - p, table, ul, h1
- Inline-element
  - i, span, strong, b, a, img



Block elements stack, regardless of their widths.



Inline elements flow from one line to the next.

# Introduktion till webbutveckling (DA156A)



# Introduktion till webbutveckling

Denna webbplats innehåller material från en grundkurs i webbutveckling. De som läser [kursen](#) är den primära målgruppen, men mycket kan förhoppningsvis vara användbart även för andra. Allt material är publicerat under en [fri licens](#), vilket gör att det kan återanvändas av andra, så länge upphovsman anges.

## Syfte

I den bakomliggande kursplanen beskrivs syfte och mål:

Kursen syftar till att introducera fundamentala koncept och tekniker som ingår i modern webbutveckling. Studenten ska ges praktiska färdigheter i utveckling av webbsidor, liksom en teoretisk bakgrund till de begränsningar och möjligheter som finns i olika plattformar. Studenten ska också utveckla en inledande förståelse av hur användargränssnittet kan styra utvecklingen av ett gränssnitt.

Bra länkar

Malmö Högskola

## It's Learning

## Kursplan

## Senaste händelser

### Föreläsning 3: Tabeller & formulär

## Föreläsning 2: Introduktion till HTML

## Föreläsning 1: Kursintroduktion

Memory Audits Elements

```
<!DOCTYPE html>
<html lang="en">
  <#shadow-root (open)>
    <head>...</head>
    <body style> == $0
      <div class="navbar navbar-inverse navbar-fixed-top">...</div>
      <div class="container">...</div>
      <!-- /.container -->
      <div id="lightboxOverlay" class="lightboxOverlay" style="display: none;"></div>
      <div id="lightbox" class="lightbox" style="display: none;">...</div>
    </body>
  </html>
```

Styles Computed Event Listeners >>

```
Filter :hov .cls -
element.style {
}
body {
  font-family: 'PT Sans', sans-serif;
  font-size: 150%;
}
body {
  font-family: "Helvetica Neue", Helvetica,
    Arial, sans-serif;
  font-size: 14px;
  line-height: 1.42857143;
  color: #333;
  background-color: #fff;
}
body {
}
```

Demo!





Tillbaka till CSS

# CSS – Same same but different

- Formgivning åt HTML-dokument
- Typografi
- Layout
- Färger
- Bildhantering m.m.



# CSS

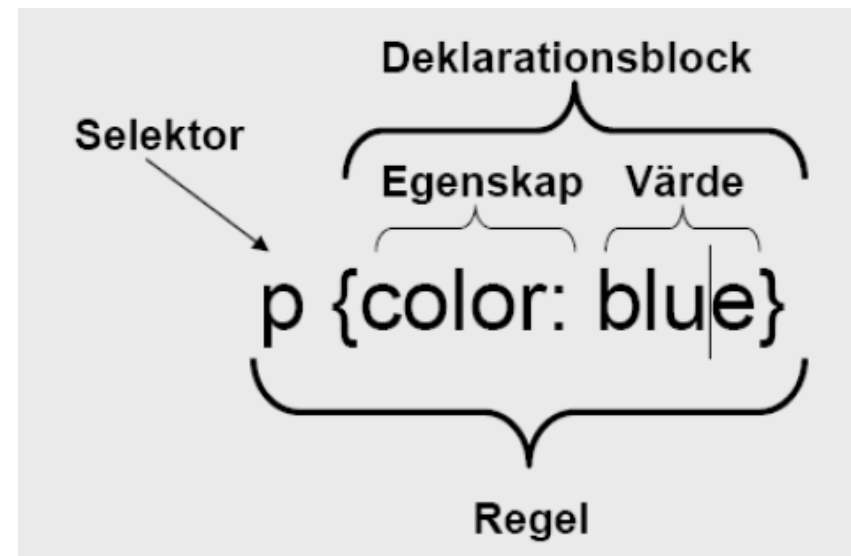
## SELECTORS

### TARGETING HTML LIKE A PRO

# Hur skriver man CSS?

- Först måste man ange vad som ska stylas (s.k. selektor). T.ex. om man vill styla alla h1-taggar skriver man "h1" följt av tecken {}.  
Mellan {} skrivs de egenskaper som <h1>-taggarna ska ha. T.ex.

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





# Hur fungerar det?

`<h2 style="color: red">`

BAD =(

- Man kan skriva CSS-kod på tre olika ställen:  
1. i html-taggar m.h.a. attributet style="":

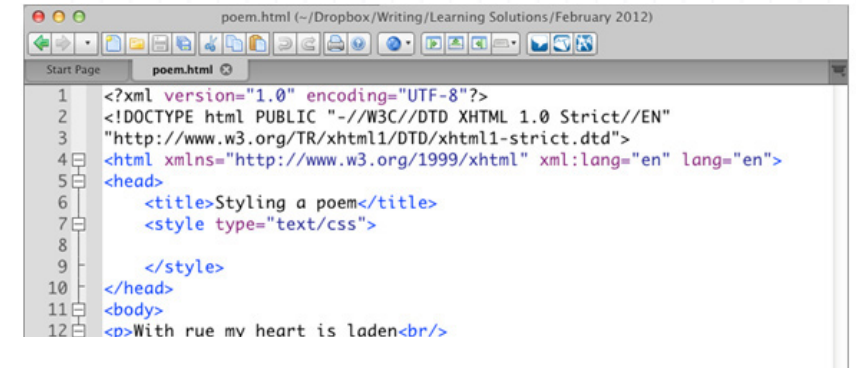
```
<h1 style="color:blue;">Rubrik</h1>
```

- 2. i html-dokumentet inom <style>-taggen:

```
<style type="text/css">
h1{color:blue}
</style>
```

- 3. i ett externa CSS-dokument:

```
h1 {
  color:blue
}
```



```
<html>
<head>
<title>My Web Page</ title>
<link rel="stylesheet" type="text/css" href="whatever.css">
</head>
<body>
</body>
</html>
```

---

```
h1 {color : green}
body {background-color : black ; color : purple}
```

# Exempelkod

- Göra alla paragrafer kursiva:

```
p{  
    font-style: italic;  
}
```

- Gör alla rubriker av typen h2 understrukna och gröna:

```
h2{  
    text-decoration: underline;  
    color: green;  
}
```

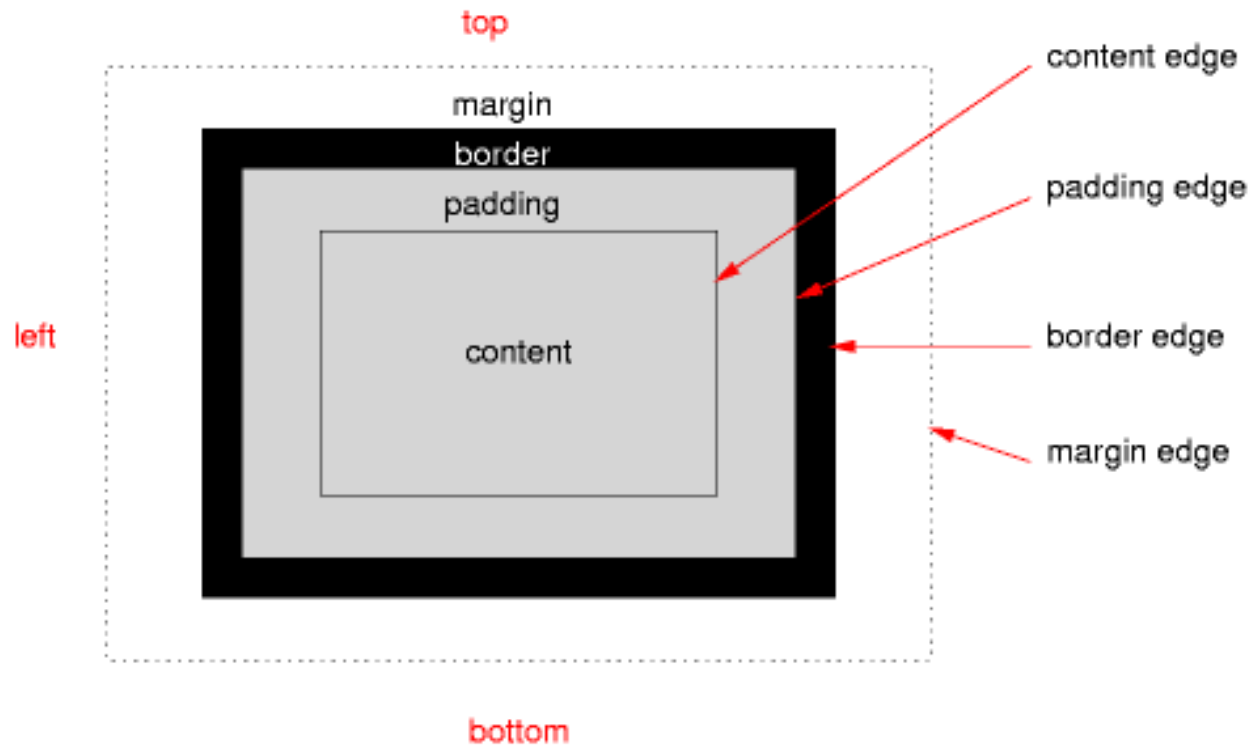
# Olika egenskaper för text i css

```
/*  
    Mall för CSS-egenskaper:  
    egenskap: värde;  
*/  
body{  
    font-family: arial, courir; /* Typsnittsfamilj */  
    font-size: 12px;           /* Textstorlek */  
    font-weight: bold;         /* Fetstil text */  
    font-style: italic;        /* Kursiv text */  
    text-decoration: underline; /* Understruken text */  
    text-align: center;        /* Centrerad text */  
    color: red;                /* Färg på text */  
}
```

# CSS - Boxmodellen

- Element i HTML kan beskrivas som boxar, och kan tilldelas olika egenskaper som t.ex.
  - Höjd
  - Bredd
  - Ram
  - Marginaler mellan varandra
  - Var de ska ligga
  - Färg
  - Osv.
- **OBS. block-element**

# Boxmodellen



Total bredd för en box är: **Content (width) + Padding + Border**.  
Margin är sedan marginalen från boxen



# <h1> som en box

- En vanlig <h1>-tagg:

**Min rubrik**

- En vanlig <h1>-tagg med CSS:

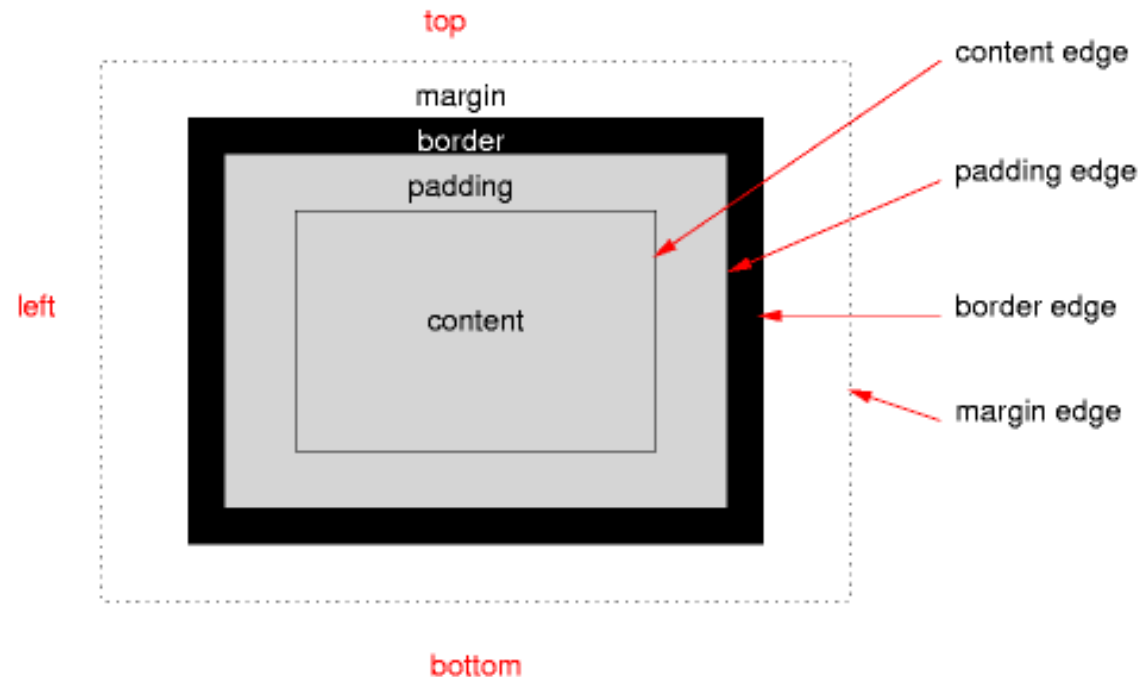


```
h1{  
    color: navy;  
    background-color: green;  
    border: 5px solid red;  
    padding: 10px;  
}
```

# Boxar

```
h1{  
  color: navy;  
  background-color: green;  
  border: 5px solid red;  
  padding: 10px;  
}
```

**Min rubrik**



Exempel

Om jag vill formge bara en  
paragraf?



# Identifiera element genom id

- I HTML kan man identifiera specifika element genom att ge dem ett unikt id
- Ett id ges genom attributet "id" samt namnet på id:t
- T.ex. för att ange ett id för en paragraf kan man skriva:

```
<p id="start">...</p>
```

- Eller för att ge en rubrik ett id:

```
<h1 id="first">...</h1>
```

- Eller för att ge en bild ett id:

```

```

# Att formge element genom att ange id i CSS

- För att referera från CSS till ett element med ett specifikt id i HTML använder man tecknet # (bräddgård)
- T.ex. för att referera till paragrafen:

```
<p id="start">...</p>
```

Så skriver man i CSS:

```
#start
```

- Och för att ge texten i paragrafen med id:t "start" blå färg skriver man i CSS:

```
#start{  
    color: blue;  
}
```

Om jag vill formge några  
paragrafer?

# Identifiera element genom klasser

- I HTML kan man identifiera element genom att ge dem en klass
- En klass ges genom attributet "class" samt namnet på klassen
- T.ex. för att ange en klass för en paragraf kan man skriva:

```
<p class="intro">...</p>
```

- Eller för att ge en rubrik ett id:

```
<h1 class="center">...</h1>
```

- Eller för att ge en bild ett id:

```

```



# Att formge element genom att ange klasser i CSS

- För att referera från CSS till de element som har tilldelat sig en viss klass skriver man . (punkt) och klassnamnet
- T.ex. för att referera till paragraferna:

```
<p class="blue">...</p>  
<p class="blue">...</p>  
<p class="blue">...</p>
```

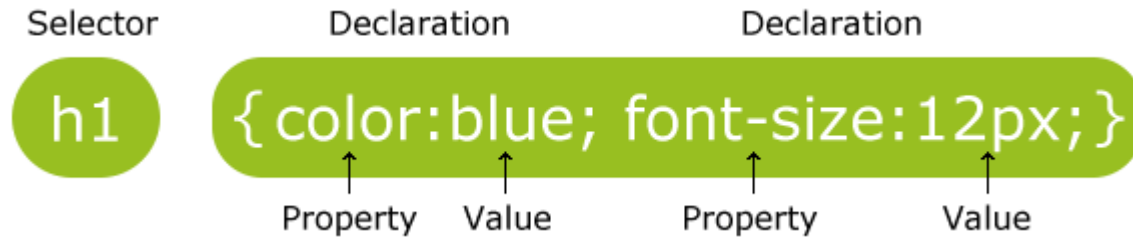
Så skriver man i CSS:

```
.blue
```

- Och för att ge texten i paragraferna med klassen "blue" blå färg skriver man i CSS:

```
.blue{  
    color: blue;  
}
```

# CSS Repetition

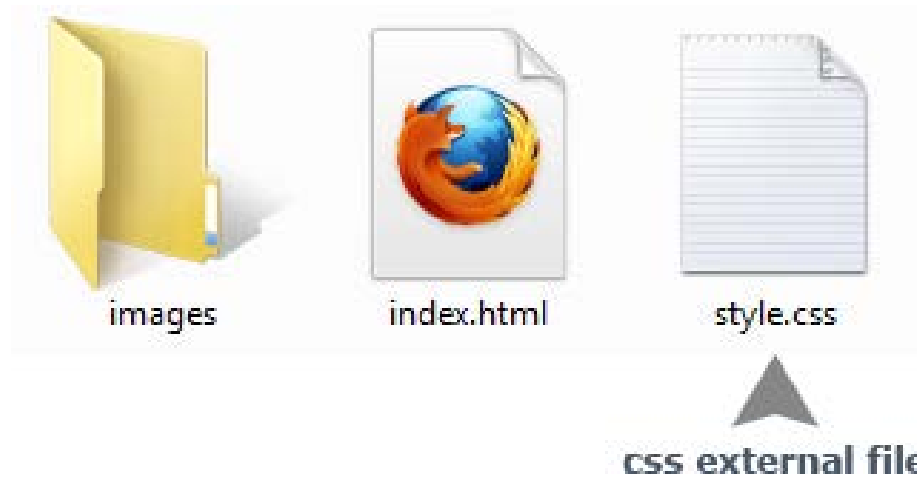


- För att referera till ett element skriver man:
  - `element{egenskap:värde;}`
- För att referera till element med en klass skriver man:
  - `.klassnamn{egenskap:värde;}`
- För att referera till ett element med ett id skriver man:
  - `#id{egenskap:värde;}`

**#id** vs. **.class**

Who wins?

DEMO – id och klasser

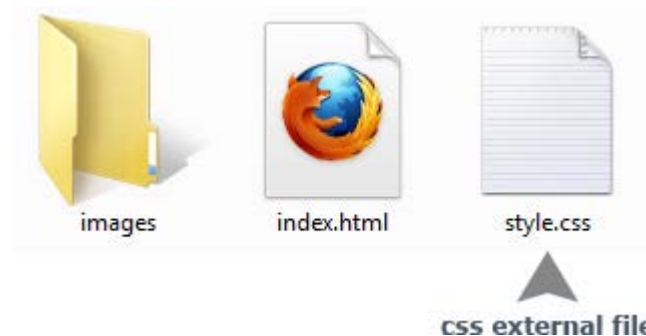


# CSS i ett externt dokument?



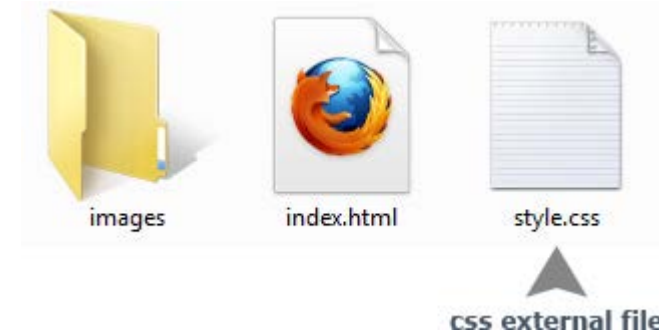
# CSS i ett externt dokument

- Ofta när man jobbat med CSS så separerar man på HTML och CSS-koden. De största fördelarna med detta är:
  1. Om en webbplats består av mer än en HTML-sida så återanvänds CSS-koden på alla sidor där den är inkluderad.
  2. Genom att separera HTML och CSS så får man en "renare" och mer lättläst källkod.



# Hur länkar man till ett externa CSS-dokument

- Det sker i taggen <head>
- Man skriver:  
`<link href="sökväg" rel="stylesheet" type="text/css">`
- T.ex.  
`<link href="style.css" rel="stylesheet" type="text/css">`
- OBS. Det externa dokument som innehåller all CSS-kod måste sluta på ".css" alltså t.ex. "style.css".





Demo – CSS i externt dokument

# Tips

- [http://www.w3schools.com/css/css\\_intro.asp](http://www.w3schools.com/css/css_intro.asp)

FRÅGOR?



Arv

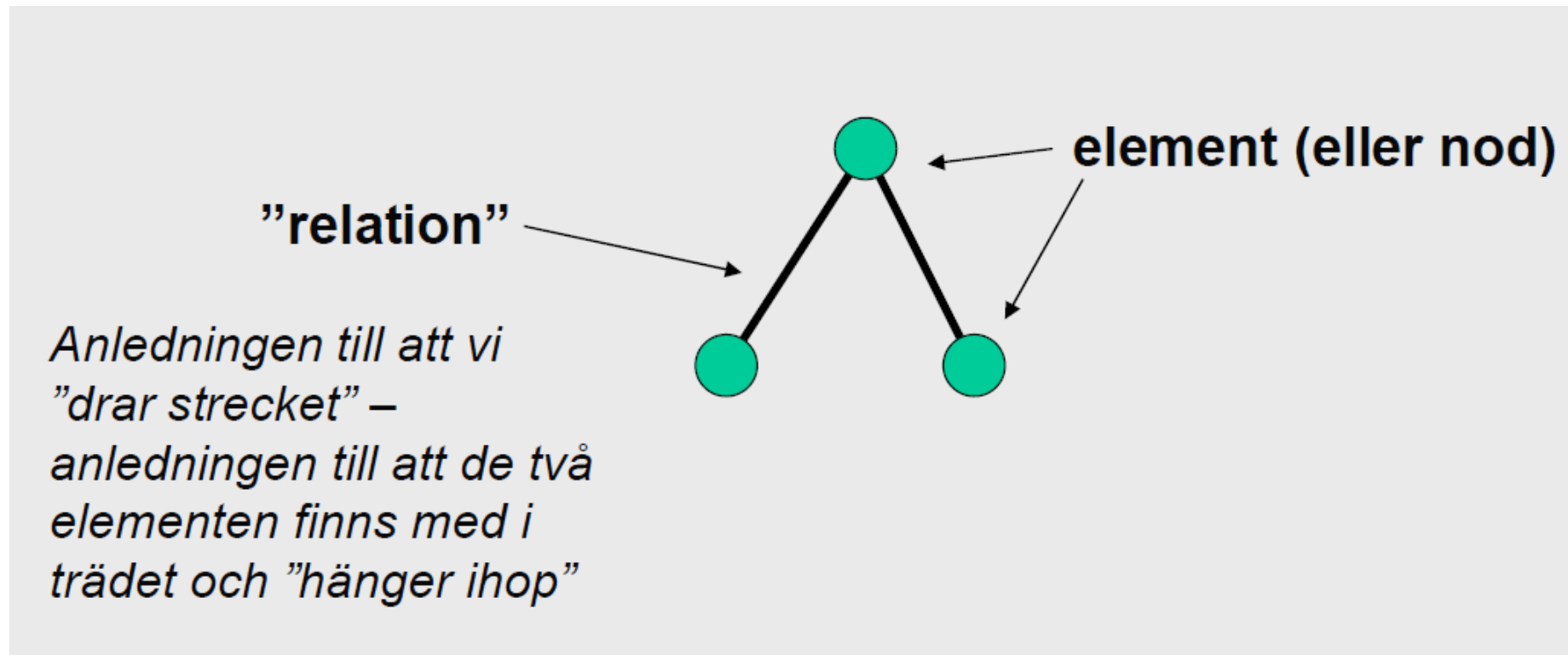
# Trädstruktur



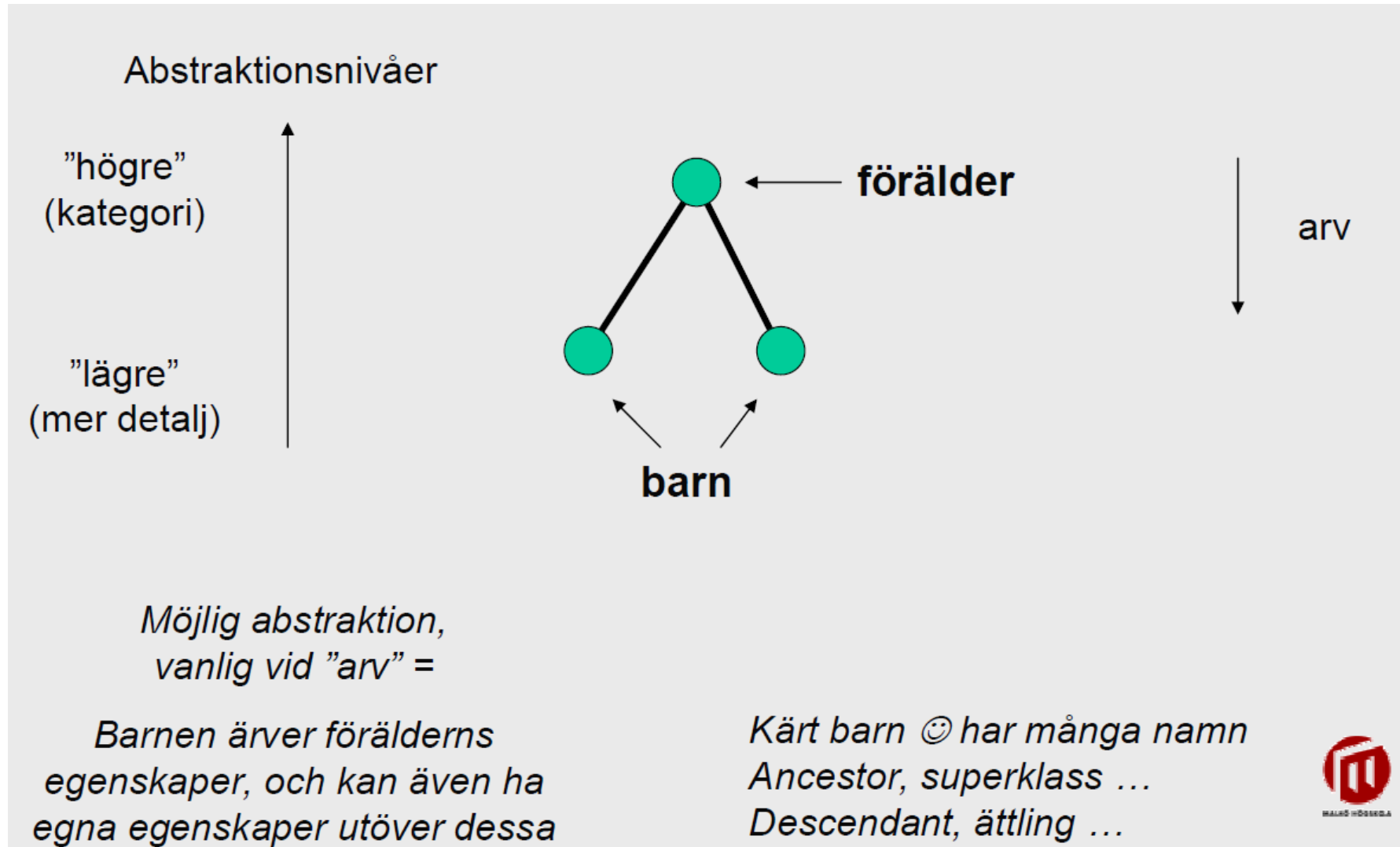
# Trädstruktur

- Trädstruktur används i datalogi för att beskriva en struktur
  - Ofta hierarkisk
- Är ett verktyg för abstraktion och därmed klassificering, arv, regler etc.
- Kan implementeras som en datastruktur för lagring/sökning av information
- Trädstruktur kan liknas med
  - Filstruktur med mappar och filer
  - Släktträd vid släktforskning
  - Biologin, klassificering av arter
    - Växter (flora), insekter osv.

# Träd – Centrala begrepp

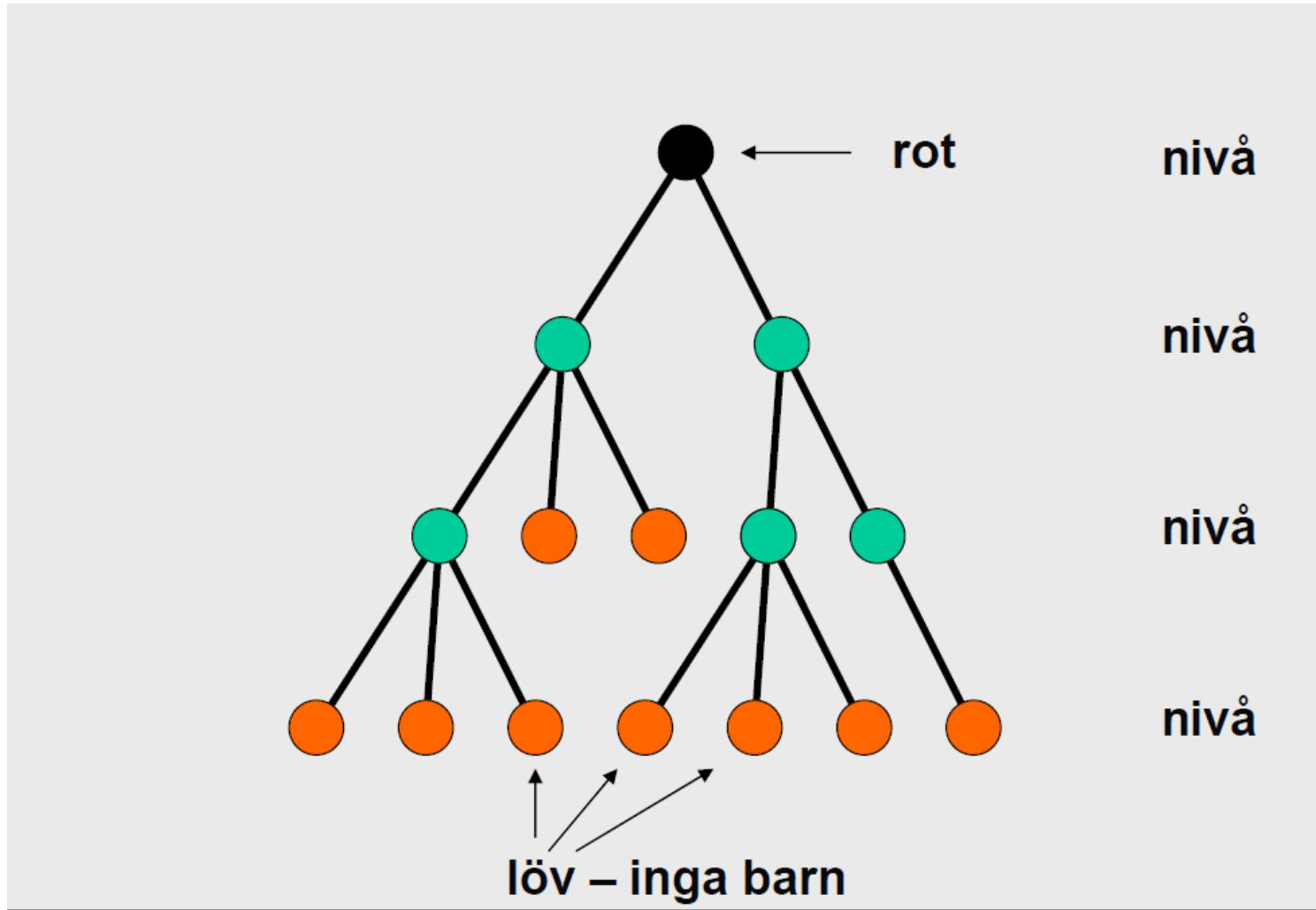


# Träd centrala begrepp

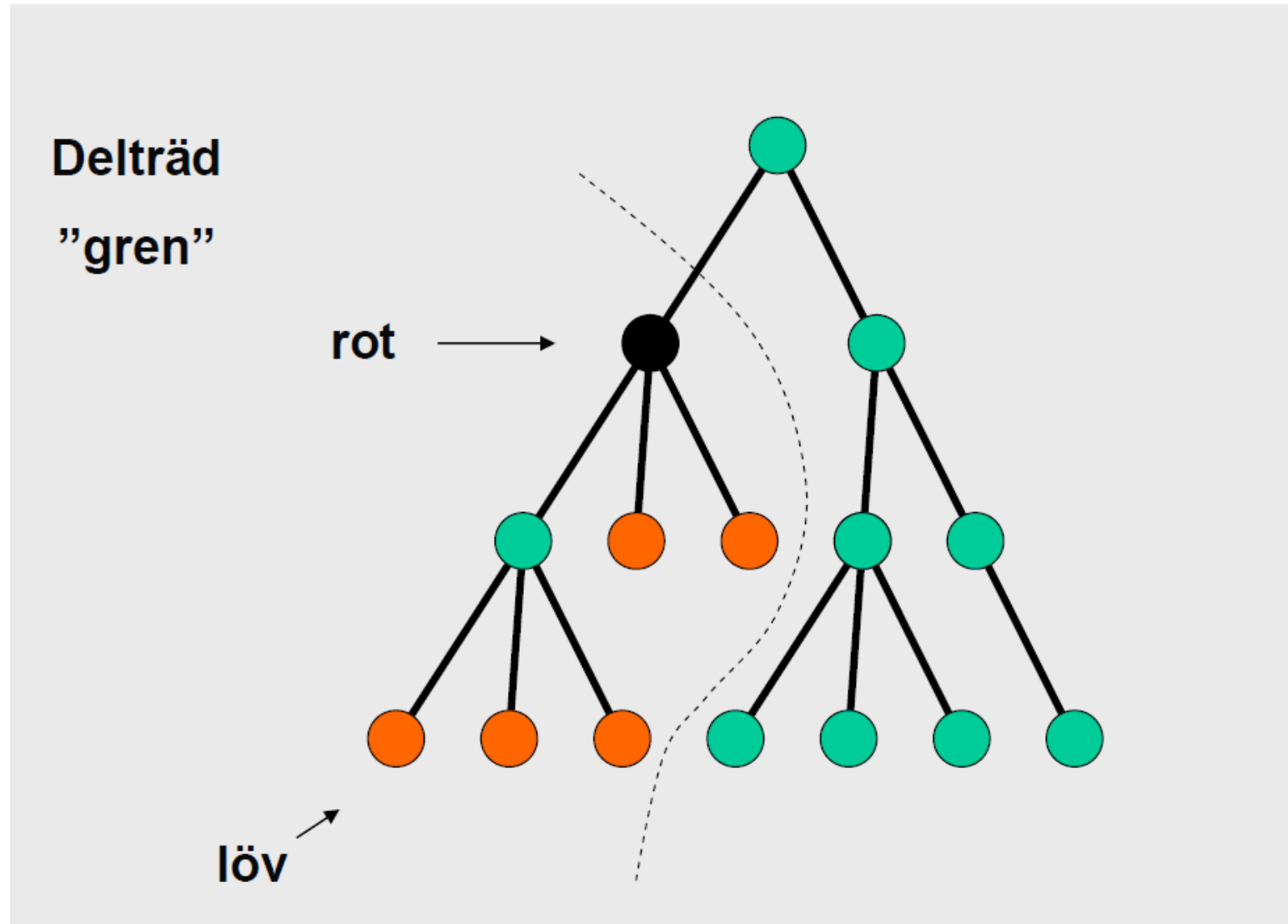




# Träd centrala begrepp



# Träd centrala begrepp

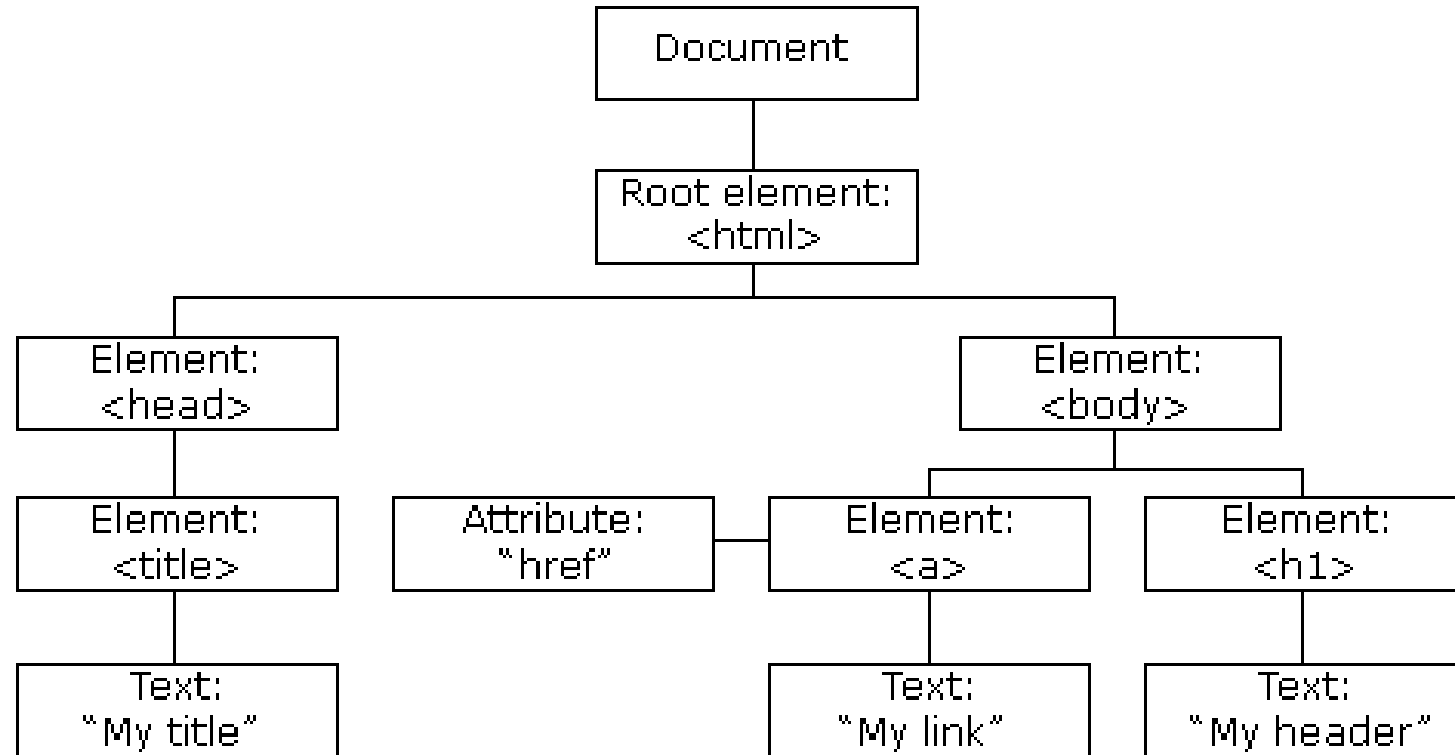


# Centralt i webbt teknik – DOM Modellen

- **Document Object Model**

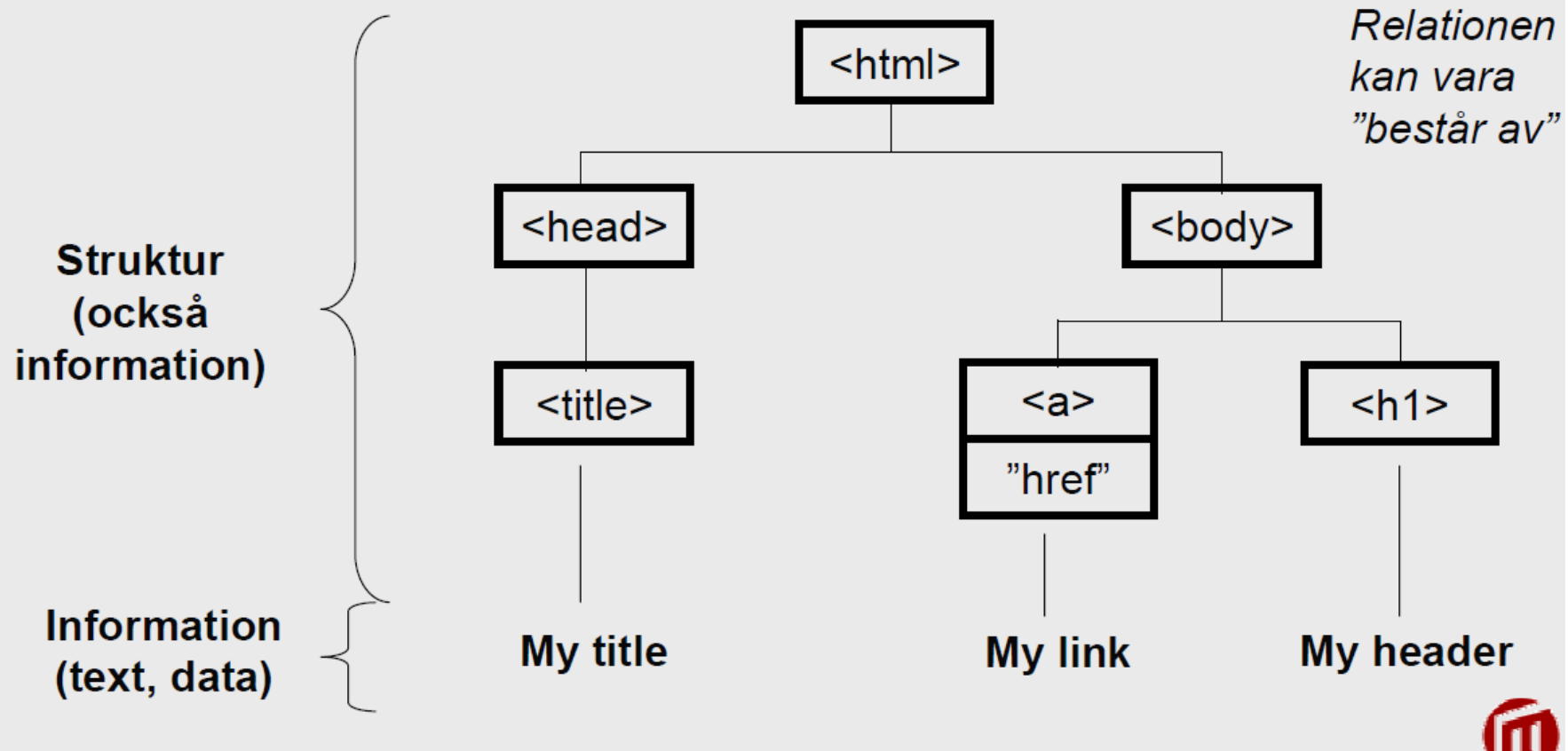
- Är en presentation av HTML-dokumentets struktur som en modell, där de olika elementen är kopplade till varandra genom olika relationer.
- DOM är standardiserat av W3C
- Trädliknande modell av sidan som den "finns inne i webbläsaren"
- Stödjer koppling mellan statisk information t.ex. HTML och dynamiskt beteende som t.ex. JavaScript.
- Exempel på användning är AJAX, vilket kort innebär att sidan modifieras (saker tas bort, läggs till, förändras) utan att man behöver ladda om sidan.
- T.ex. Facebook, Gmail, m.m.

# HTML - DOM



# HTML – DOM

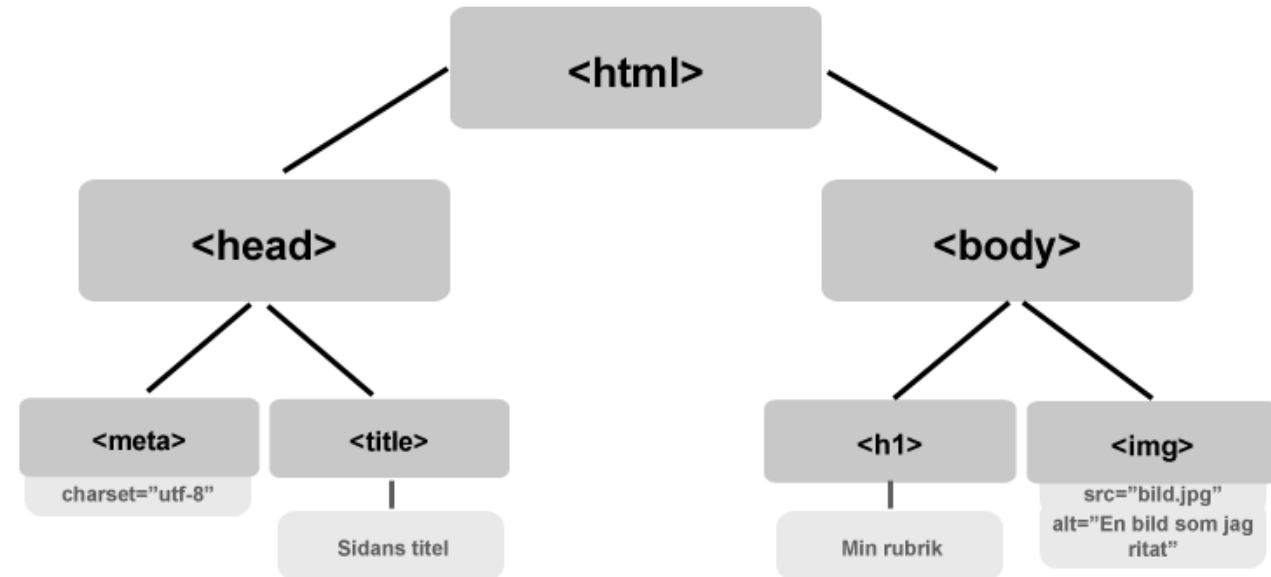
- Samma bild, ungefär ... men vi frilägger strukturen ...



# Källkod vs. noder

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Sidans titel</title>
  </head>

  <body>
    <h1>Min rubrik</h1>
    
  </body>
</html>
```



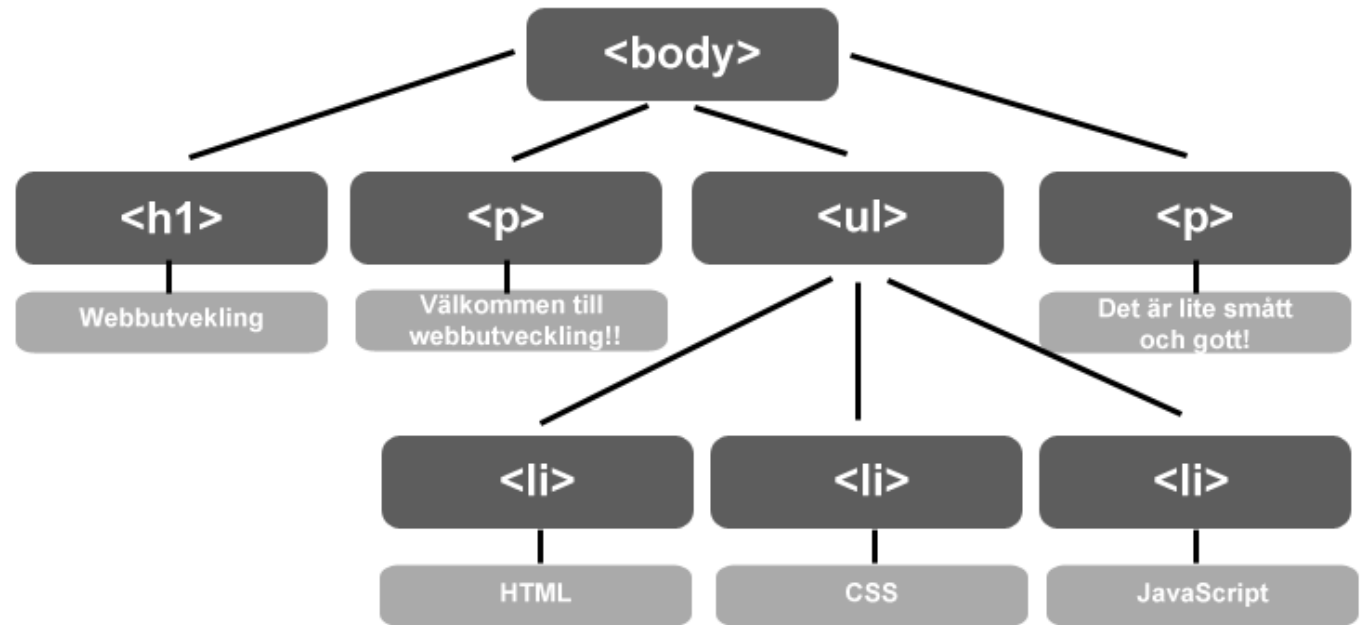


# Arv i CSS

- För att inte behöva skriva specifika egenskaper för alla element på hela webbsida så finns något som kallas för arv.
- Detta innebär att, i ett nodträd, så ärver barnet sin förälders egenskaper (i CSS).

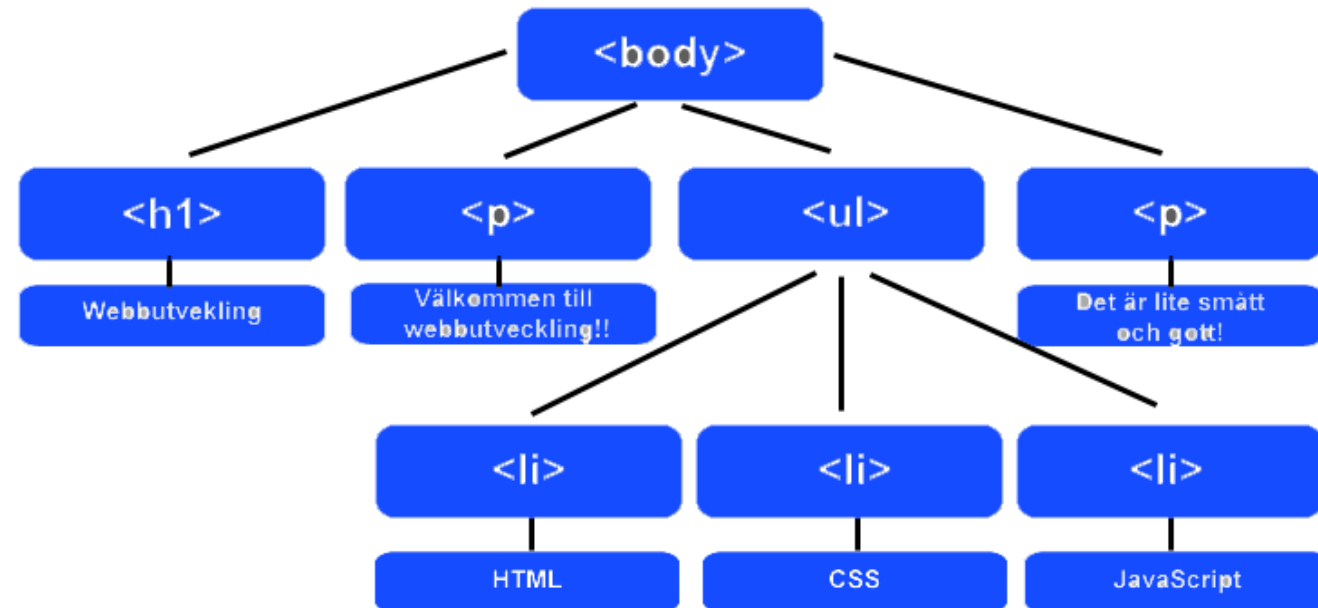
# Arv - exempel

```
<body>
  <h1>Webbutveckling</h1>
  <p>Välkommen till webbutveckling!</p>
  <ul>
    <li>HTML</li>
    <li>CSS</li>
    <li>JavaScript</li>
  </ul>
  <p>Det är lite smått och gott!</p>
</body>
```



# Arv - färgläggning

```
Body{  
    color: blue;  
}
```

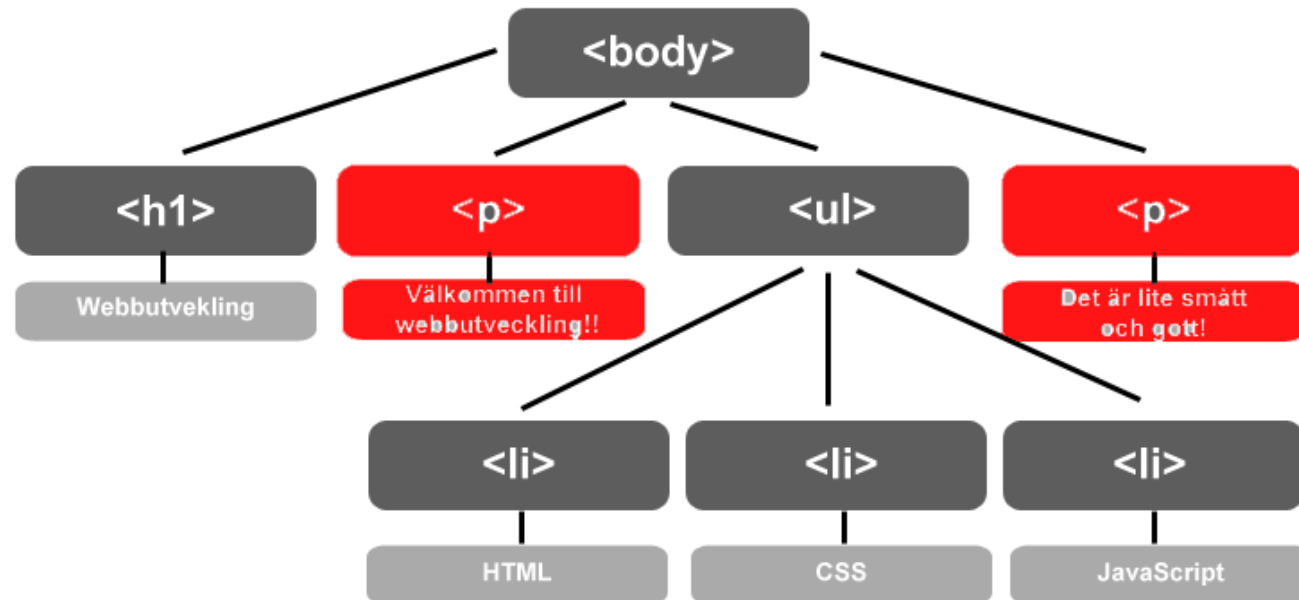


# Arv förläggning

p{

color: red;

}

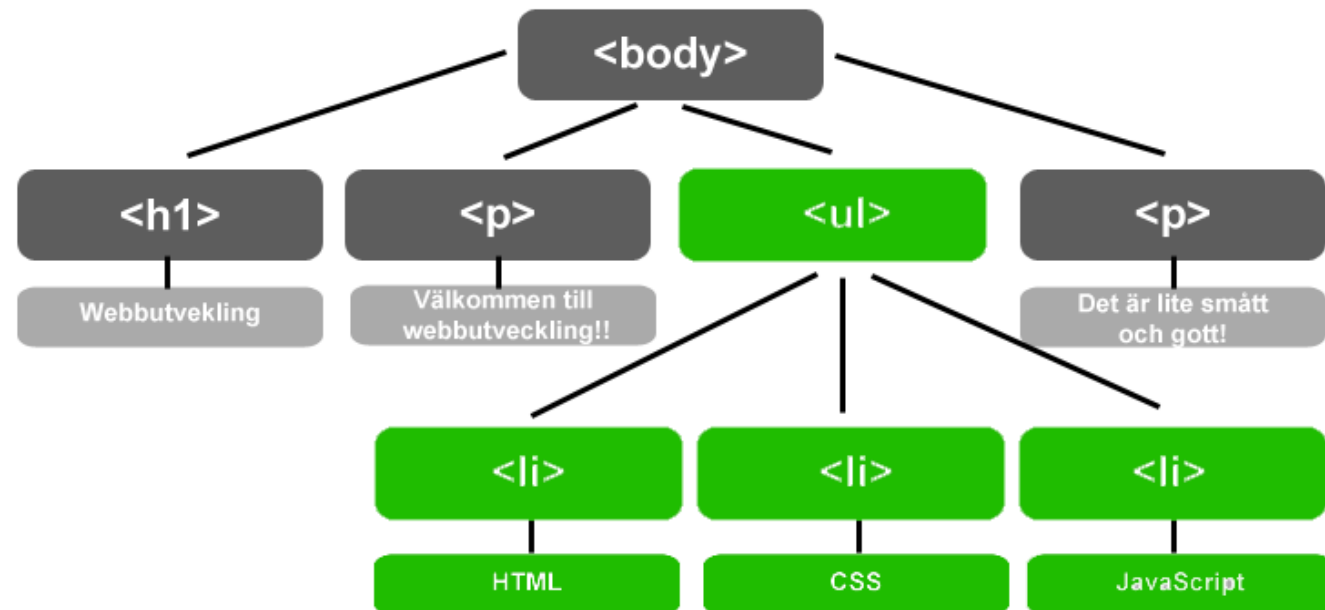


# Arv förläggning

```
ul{
```

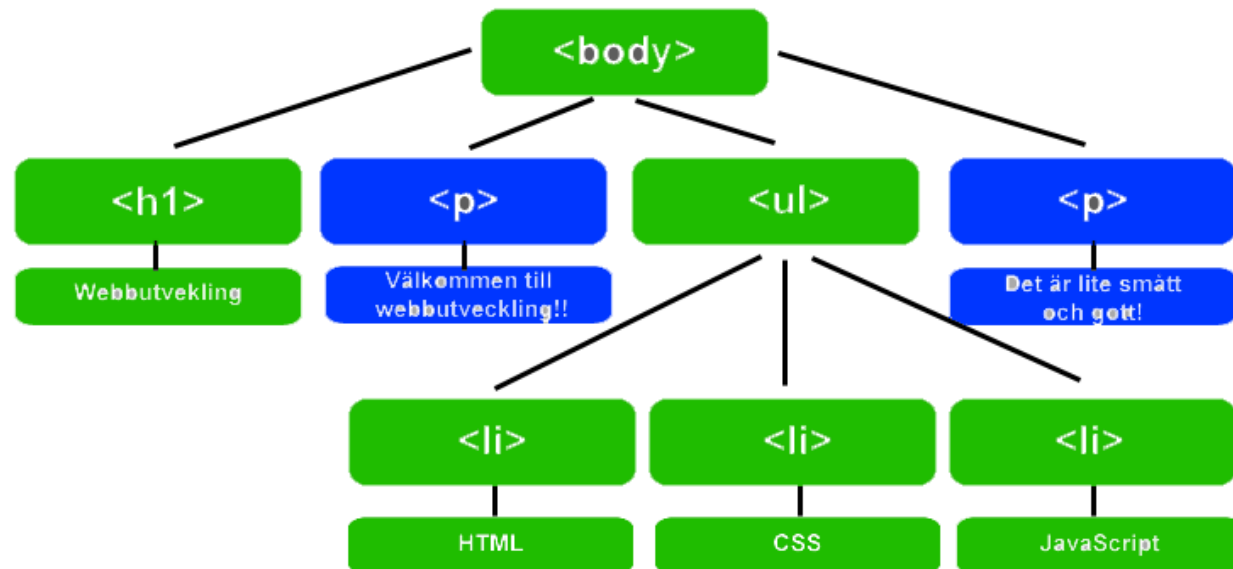
```
  color: green;
```

```
}
```



# Arv – flera färger

```
body{  
    color:green;  
}  
p{  
    color:blue;  
}
```





# Representation – att använda en kod (tal) för att presentera något

- Nödvanligt att använda då datorn bara förstår 1 och 0.
  - Det betyder att allt i datorn (färger, figurer, knapptryckningar, beräkningar, musrörelser, bokstäver, siffror, typsnitt, bilder) representeras binärt.
  - T.ex.  
bokstaven A = 65 (dec) = 01000001 (bin) = 41 (hex)
- I webbt teknik kommer vi i kontakt med flera sådana bestämda regelsystem & standarder för hur saker och ting är representerade
  - CharacterEncoding(charset), hur tecken ska beskrivas, exempel ISO 8859, UTF 8, Unicode, ...(äldre standard: ASCII)
  - Färgmodell, hur färger ska beskrivas, exempel RGB, CMY, CMYK

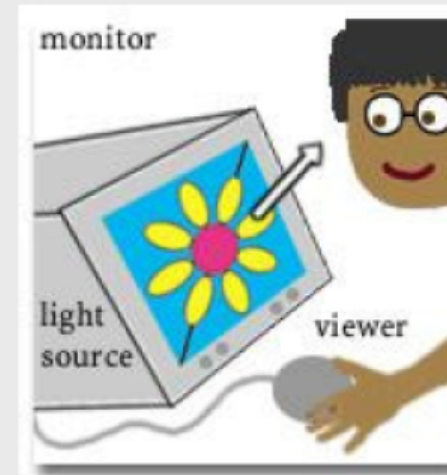
# Färger i HTML/CSS

- Man kan beskriva färger
- Istället för #08ht5

RGB - används för additiv

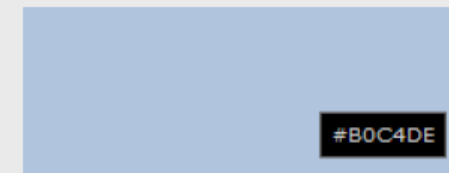
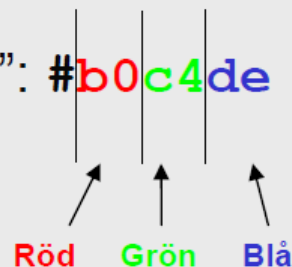
beskrivning

– Red	0-255	= 00 – FF (hex)
– Green	0-255	= 00 – FF (hex)
– Blue	0-255	= 00 – FF (hex)



En kombination av dessa 3 färger ger hela färgspektrumet på  $256 \times 256 \times 256 = 16777216 = 16,8$  Milj färgnyanser

En "webbfärg": #b0c4de anges hexadecimalt →



Modifierat material från kursen Data- och systemvetenskap I, 1-30



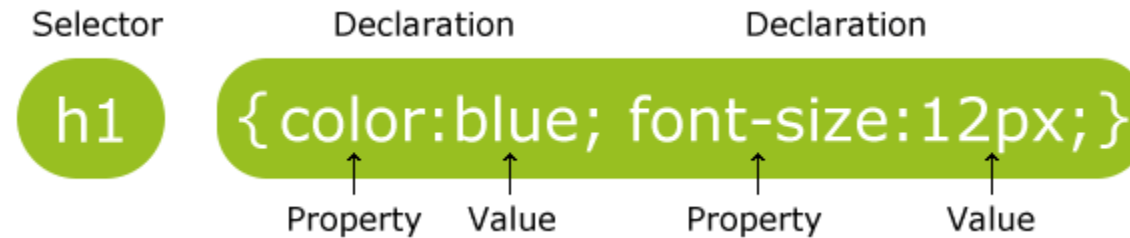
# Färgval

- <http://colorschemedesigner.com/>

# CSS – text-egenskaper

```
/*  
    Mall för CSS-egenskaper:  
    egenskap: värde;  
*/  
body{  
    font-family: arial, courir; /* Typsnittsfamilj */  
    font-size: 12px;           /* Textstorlek */  
    font-weight: bold;         /* Fetstil text */  
    font-style: italic;        /* Kursiv text */  
    text-decoration: underline; /* Understruken text */  
    text-align: center;        /* Centrerad text */  
    color: red;                /* Färg på text */  
}
```

# CSS Repetition



- För att referera till ett element skriver man:
  - element{egenskap:värde;}
- För att referera till element med en klass skriver man:
  - .klassnamn{egenskap:värde;}
- För att referera till ett element med ett id skriver man:
  - #id{egenskap:värde;}



**ONE DOES NOT SIMPLY...**

**KNOW HELL UNTIL THEY'VE TRIED  
WEB DEVELOPMENT WITH CSS**

ht

CSS

Global  
HD