



DA106A

# Introduktion till webbutveckling

HT21

7.5hp

Anton Tibblin

Föreläsningen kommer att spelas in, dock inte deltagarna

# Föreläsning 4

css



```
body {  
    font: x-small;  
    background: a  
    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 innan vi börjar?



# 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 utseende, samma innehåll



# Samma innehåll – fast olika utseende

Produkter – olika utseende, samma innehåll



# IT'S THE SAME



# BUT DIFFERENT

HD  
COM

Samma innehåll  
– 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
- Anpassa webbplatsen efter målgruppen
- Öka användbarhet
- Öka attraktivitet
- Optimera för användaren

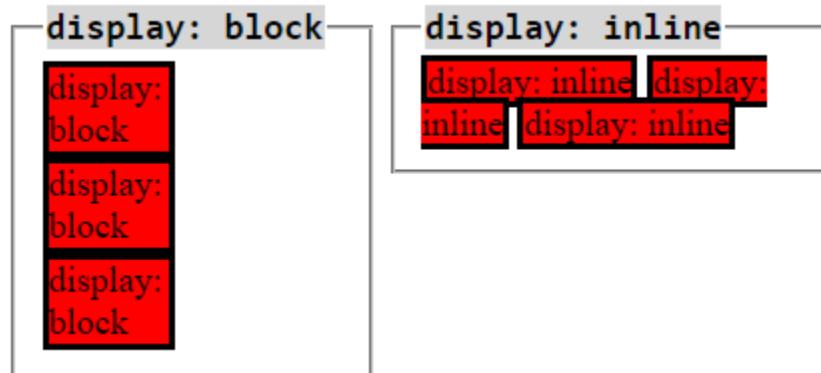


# HTML

## Olika typer av element

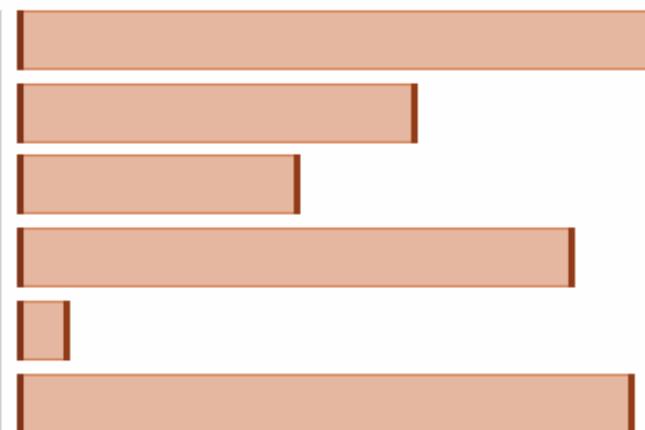
# Olika typer av element

- 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.

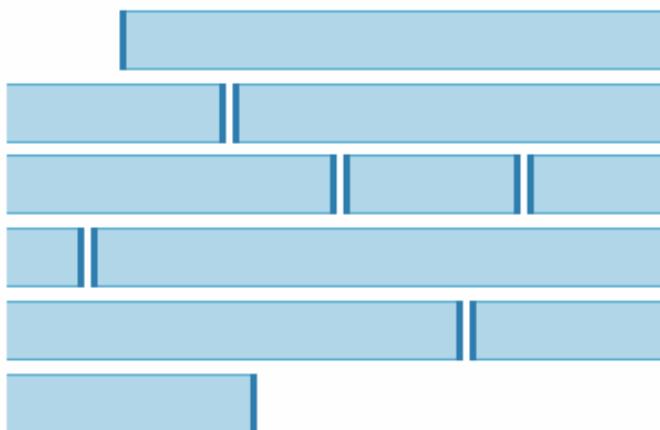


# Olika element

- 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 utvecklingstekniker, liksom en teoretisk bakgrund till de begränsningar och möjligheter som omges med att utveckla en plattformar. Studenten ska också utveckla en inledande förståelse av hur användargränssnittet kan styra utvecklingen av ett gränssnitt.

Den beskrivna syftet formulerad utan att nämna specifika tekniker. Istället ligger fokus på "modern

## Bra länkar

Malmö Högskola

It's Learning

Kursplan

## Senaste händelser

Föreläsning 3: Tabeller & formulär

Föreläsning 2: Introduktion till HTML

Föreläsning 1: Kursintroduktion

The screenshot shows a browser window with the developer tools open. The 'Elements' tab is selected in the toolbar. A large blue arrow points from the text in the 'Syfte' section towards the DOM tree in the elements panel. The DOM tree shows the structure of the page, including the header navigation bar and the main content area. The styles panel on the right shows the CSS rules applied to the selected element, with specific rules for the body element and its styling.

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body style="...">
    <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

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;

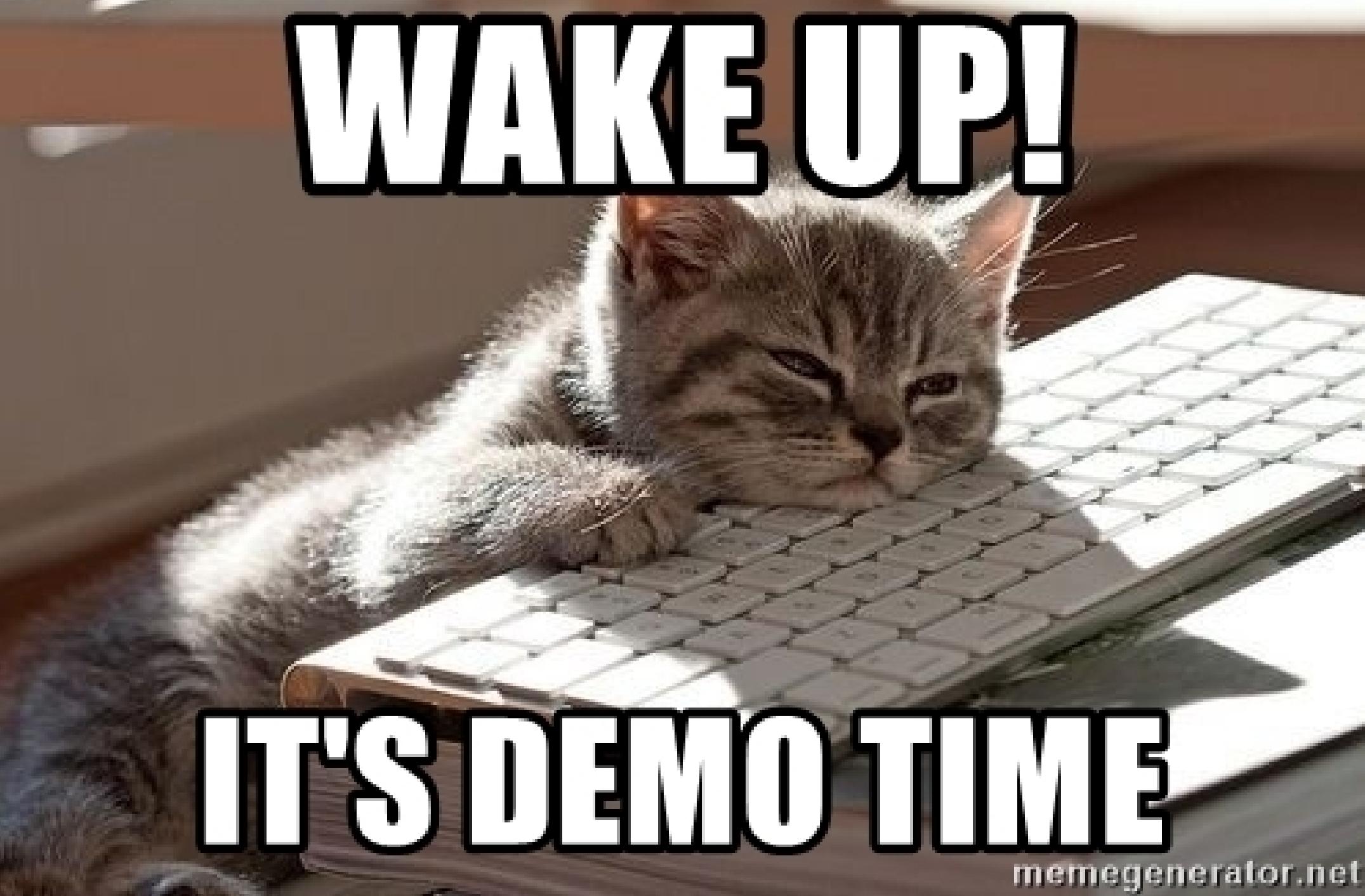
background-color: #fff;

}

body {

normalize.less:19

# WAKE UP!

A fluffy, light-colored kitten is sleeping peacefully on a white computer keyboard. The kitten's fur is a mix of light brown and white, and it has its eyes closed. The keyboard is a standard layout with white keys. The background is a warm, orange-toned wall.

# IT'S DEMO TIME

**CSS**



Tillbaka till CSS

# CSS – Same same but different

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



# CSS

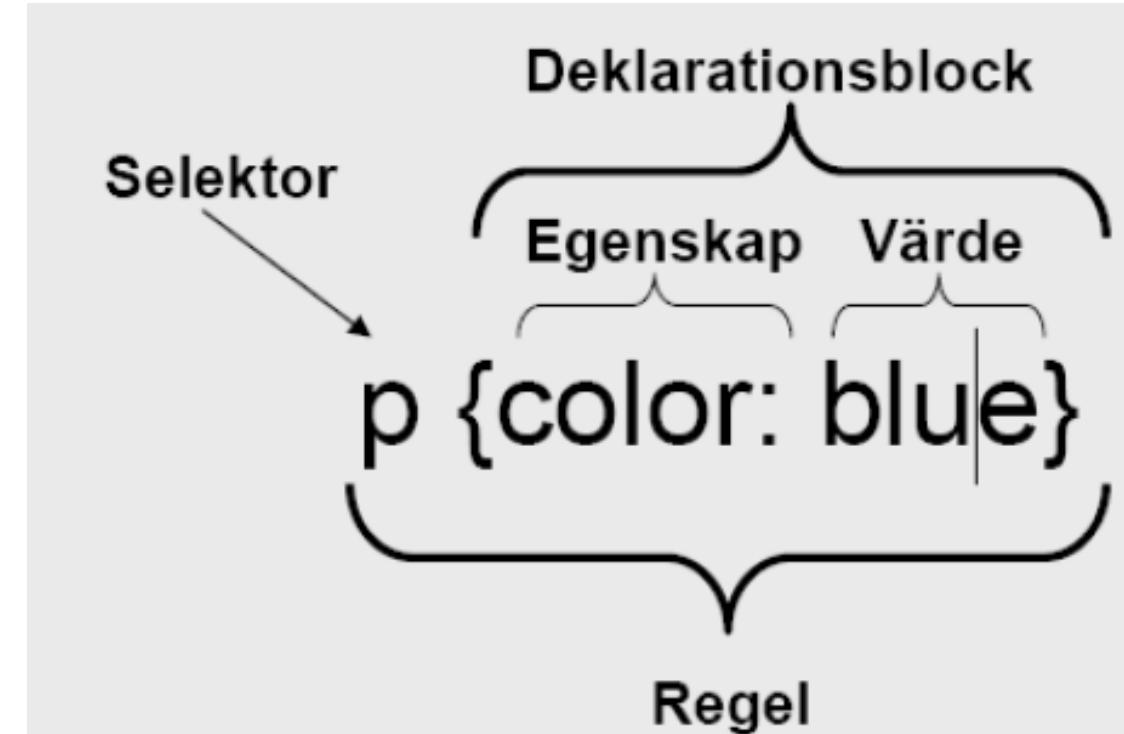
## SELECTORS

### 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 styra alla h1-taggar skriver man "h1" följt av tecknen {}. 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="":

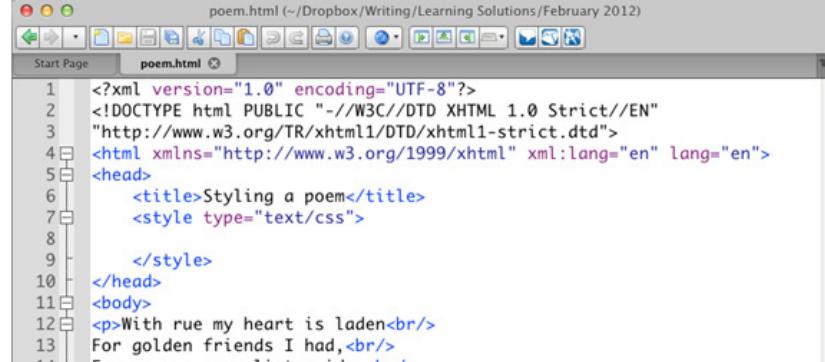
```
<h1 style="color:blue;">Rubrik</h1>
```

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

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

- 3. i ett externa CSS-dokument:

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



```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">  
<head>  
    <title>Styling a poem</title>  
    <style type="text/css">  
        </style>  
</head>  
<body>  
    <p>With rue my heart is laden<br/>  
    For golden friends I had,<br/>  
    For many a nose, list, maiden b...</p>
```

```
<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

- Gör alla paragrafer kursiva:

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

- Gör alla rubriker av typen h2 understrukna och gröna:

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

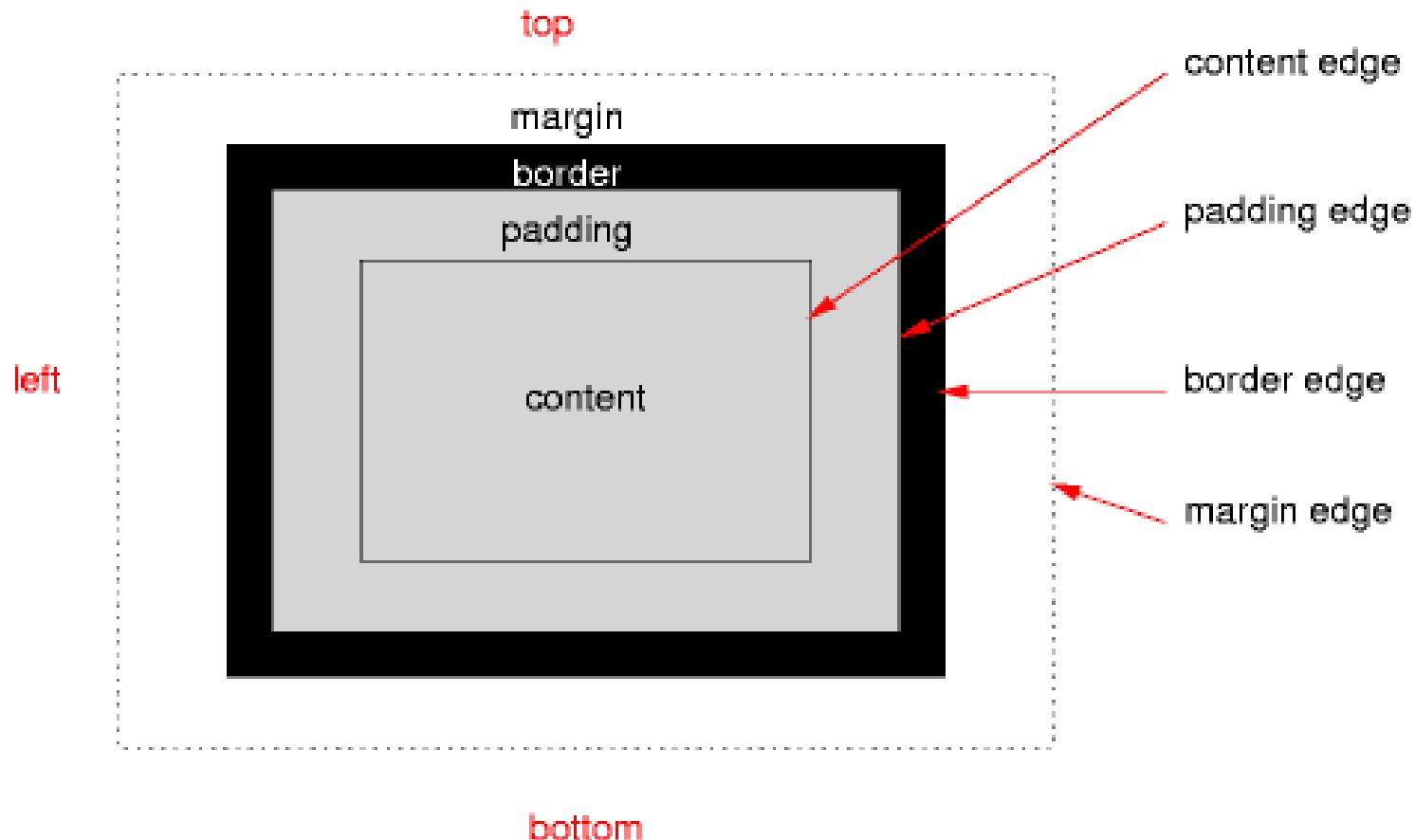
# Olika egenskaper för text i css

```
/*
    Mall för CSS-egenskaper:
    egenskap: värde;
*/
body{
    font-family: arial, courir; /* Tysnittsfamilj */
    font-size: 12px;           /* Textstorlek */
    font-weight: bold;         /* Fetstil text */
    font-style: italic;        /* Kursiv text */
    text-decoration: underline; /* Understrucken 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

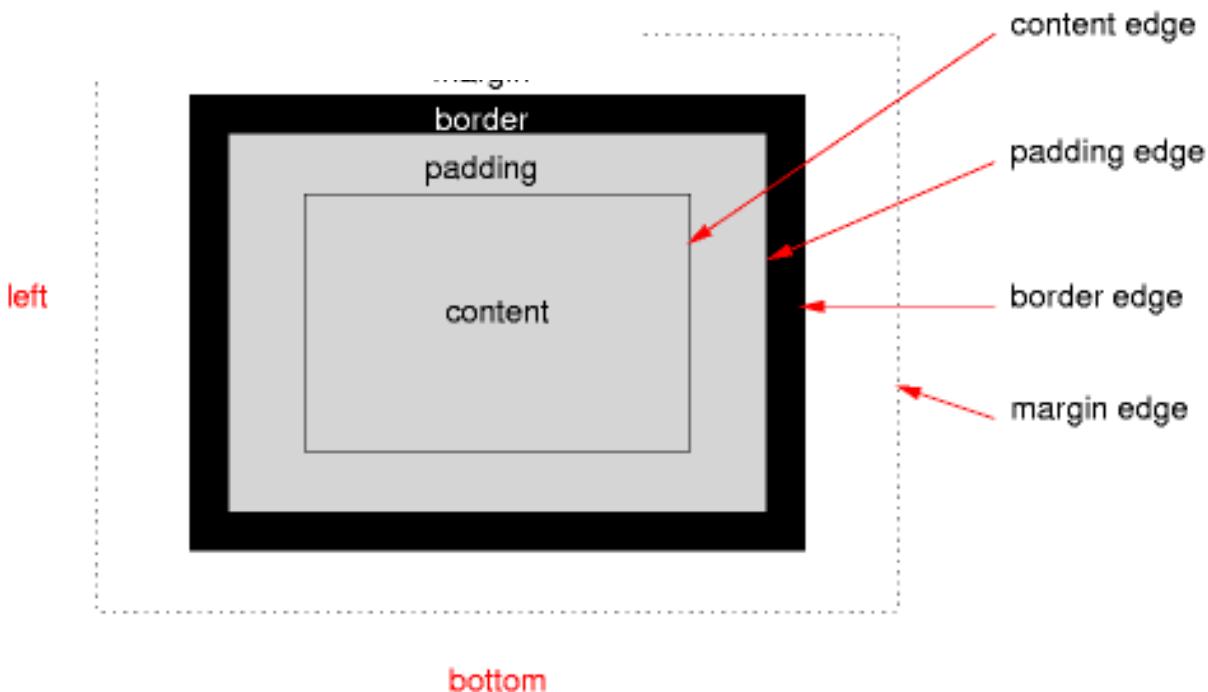
- En vanlig <h1>-tagg med CSS:

Min rubrik

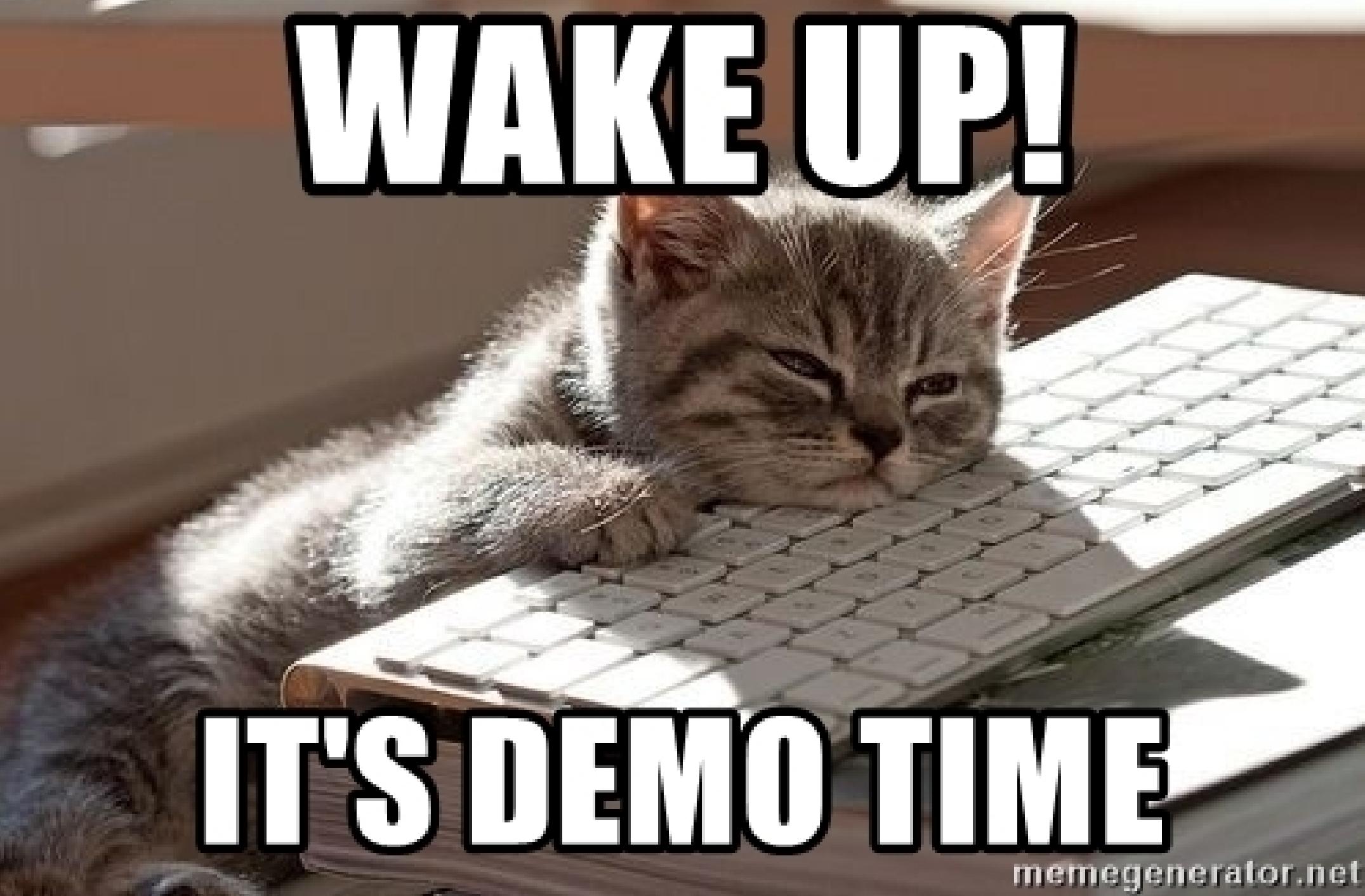
```
h1 {  
    color: navy;  
    background-color: green;  
    border: 5px solid red;  
    padding: 10px;  
}
```

# Boxar

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



# WAKE UP!

A fluffy, light-colored kitten is sleeping peacefully on a white computer keyboard. The kitten's fur is a mix of light brown and white, and it has its eyes closed. The keyboard is a standard layout with white keys. The background is a warm, orange-toned indoor setting.

# IT'S DEMO TIME

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 # (brädgård)
- 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 en klass :

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

- Eller för att ge en bild en klass :

```

```

# 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>
```

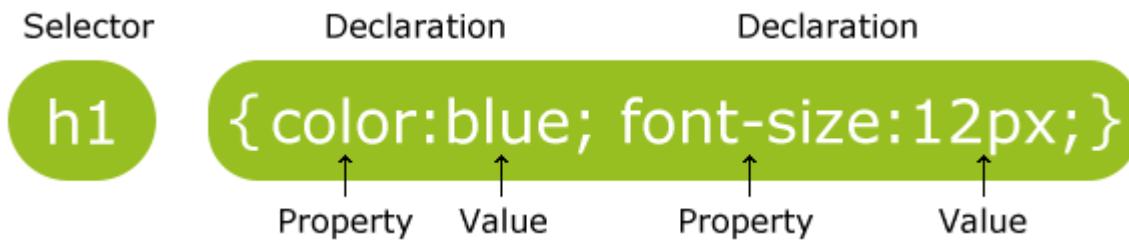
Så skriver man i CSS:

```
.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;}



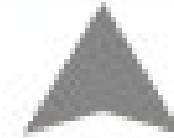
images



index.html



style.css

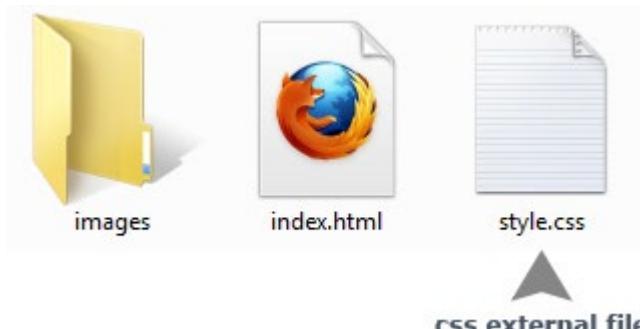


css external file

# CSS i ett externt dokument?

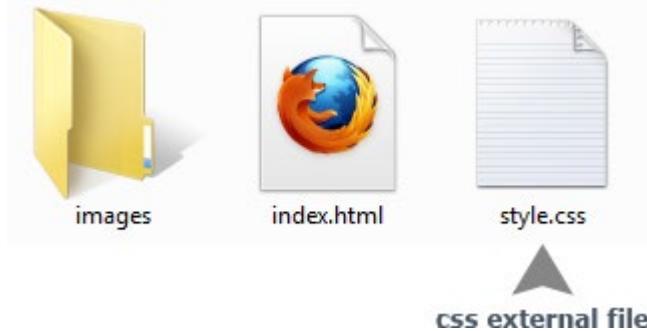
# CSS i ett externt dokument

- 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".



THE MAGIC OF  
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

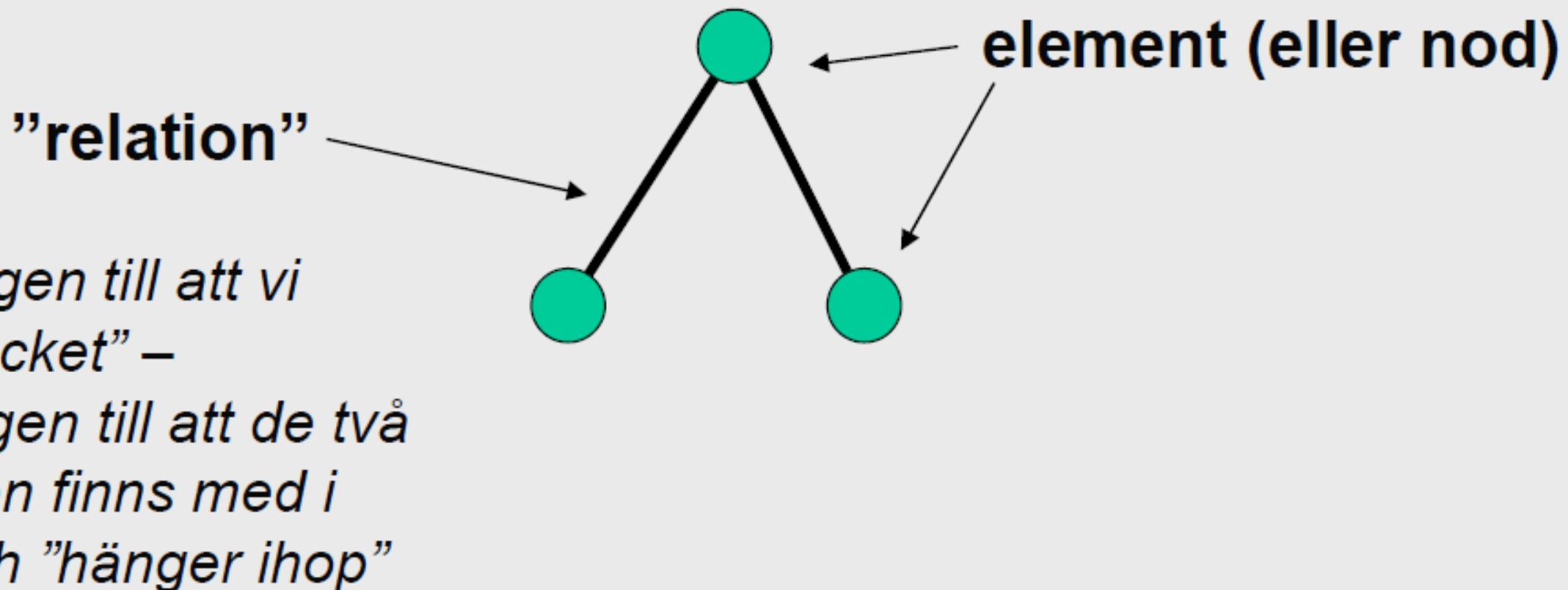
# Trädstruktur



# 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.

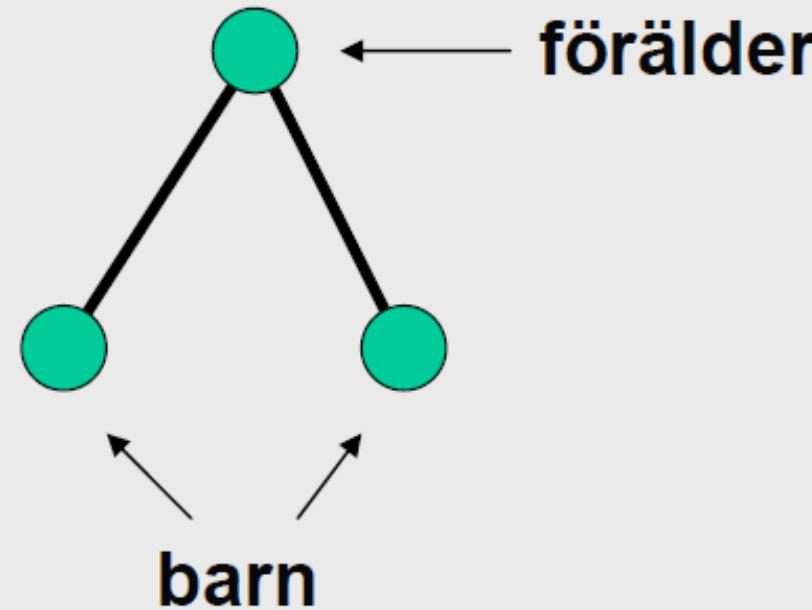
# Träd – Centrala begrepp



## Abstraktionsnivåer

"högre"  
(kategori)

"lägre"  
(mer detalj)

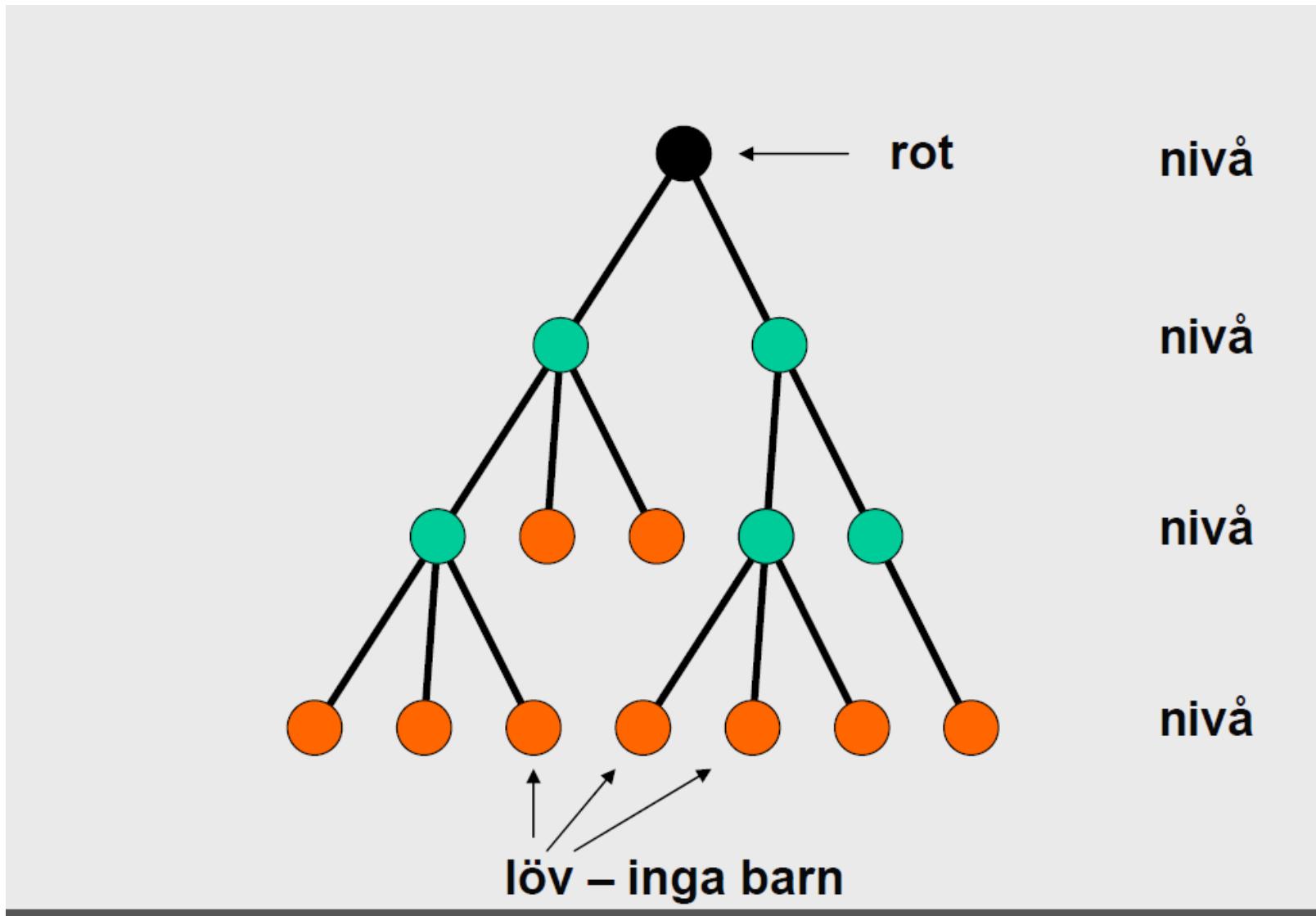


*Möjlig abstraktion,  
vanlig vid "arv" =*

*Barnen ärver förälderns  
egenskaper, och kan även ha  
egna egenskaper utöver dessa*

*Kärt barn ☺ har många namn  
Ancestor, superklass ...  
Descendant, ättling ...*

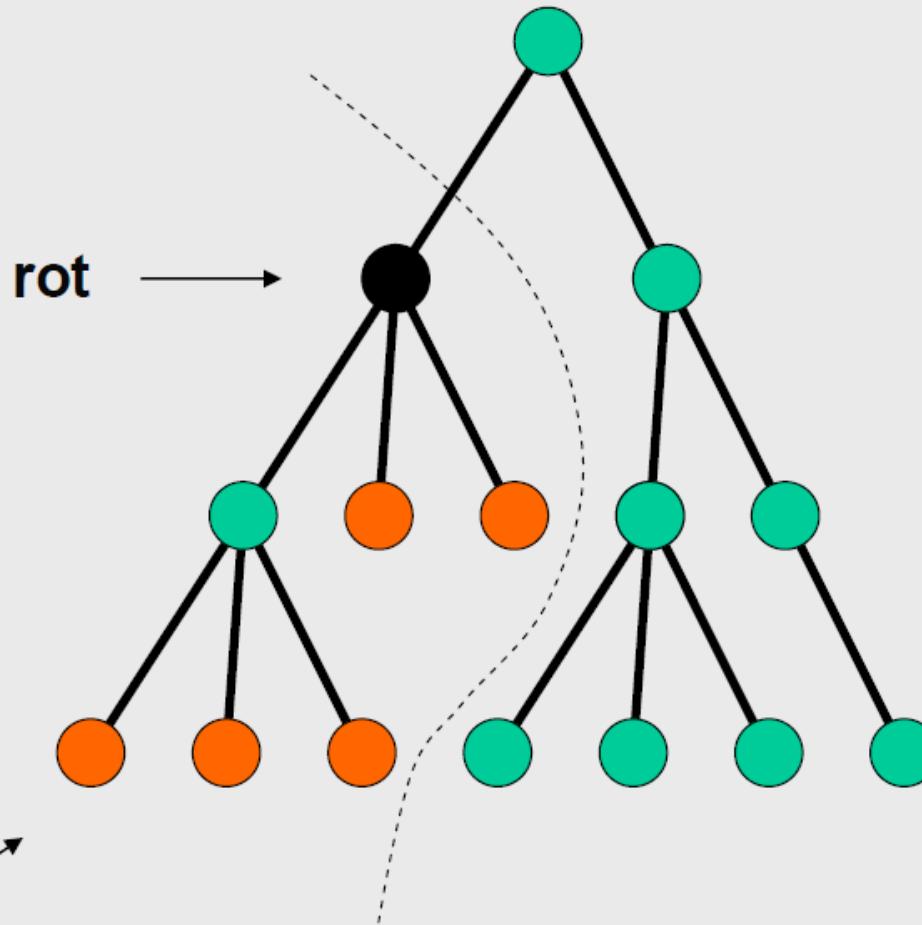
# Träd centrala begrepp



# Träd centrala begrepp

**Delträd**

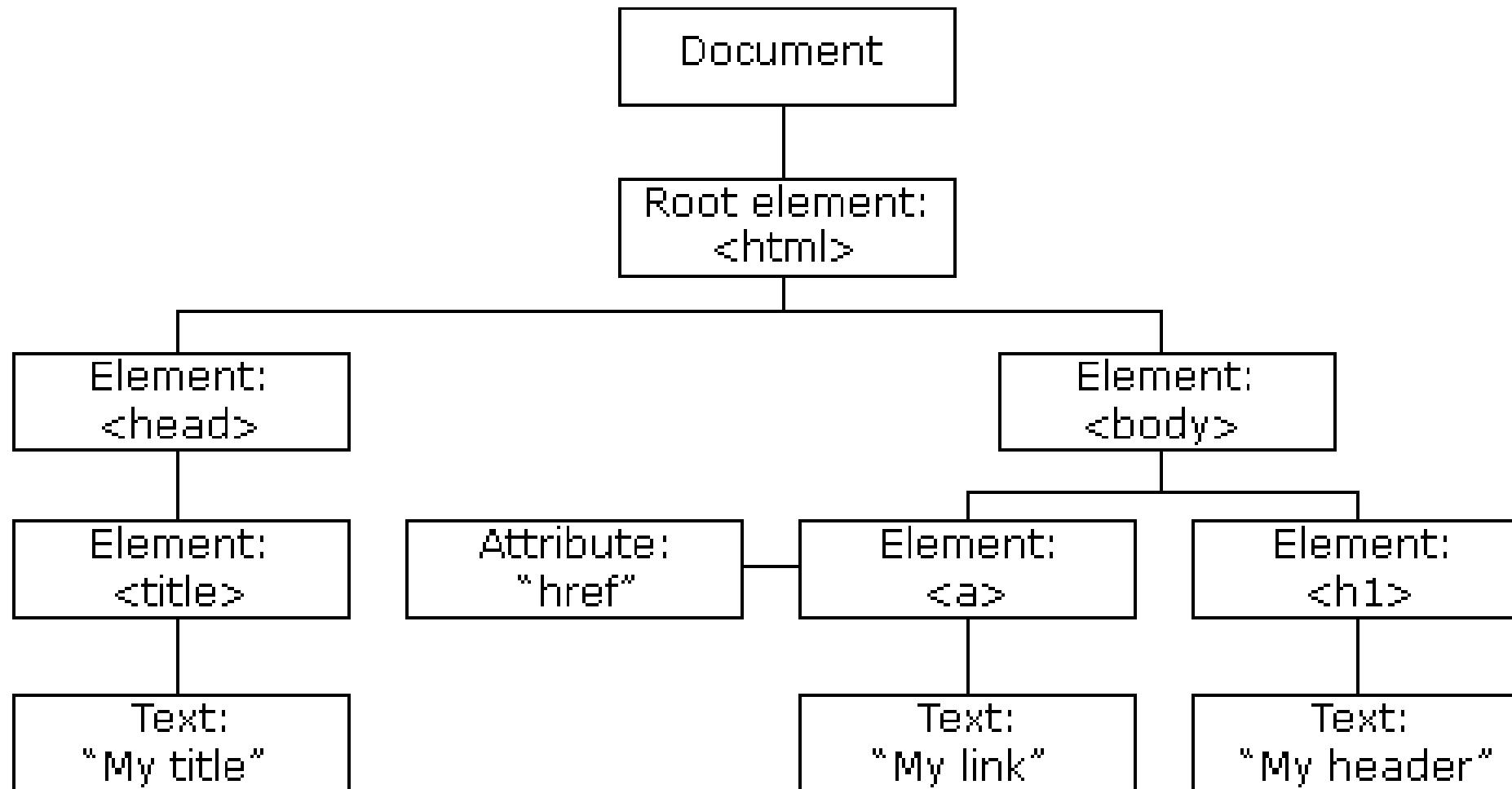
"gren"



# Centralt i webbteknik – DOM Modellen

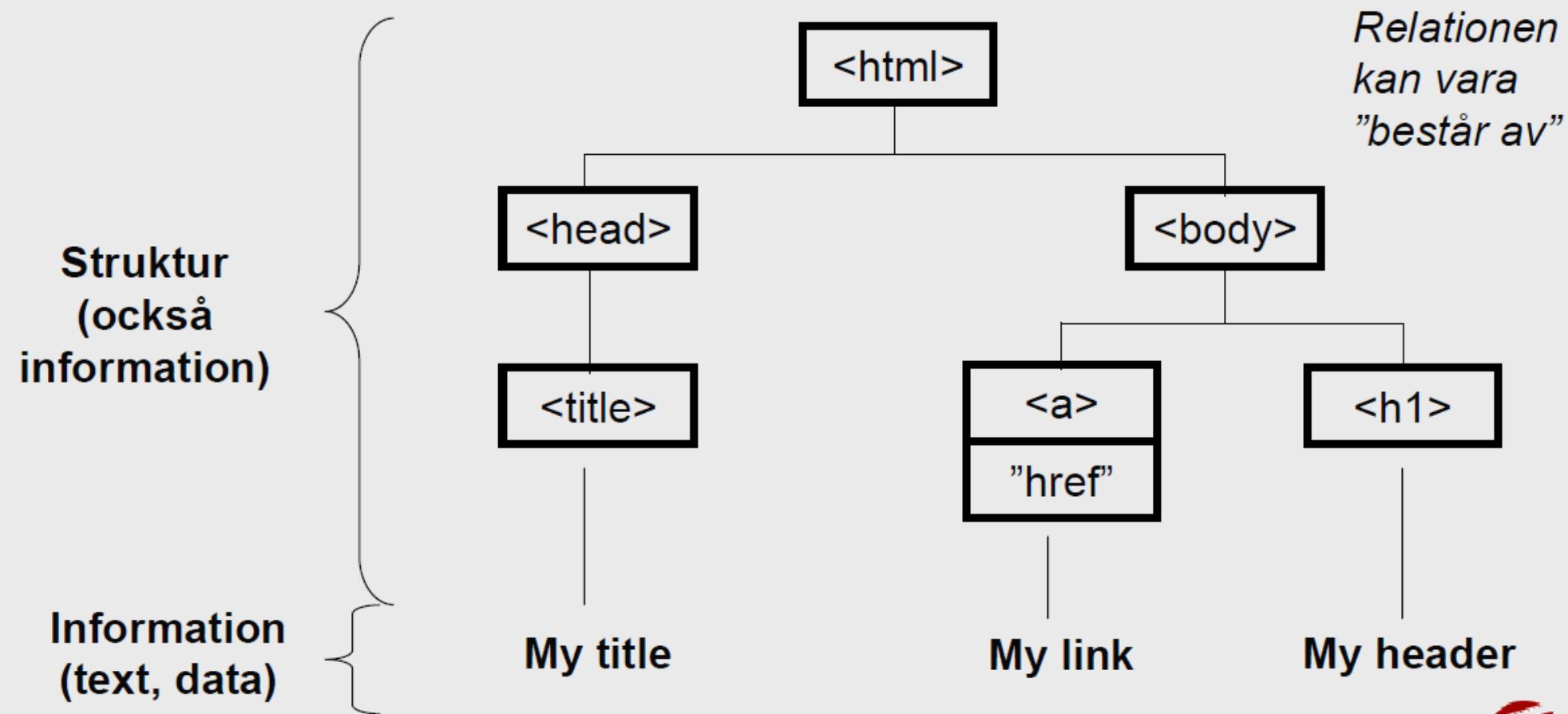
- **Document Object Model**
  - Är en presentation av HTML-dokumentets struktur som en modell, där de olika elementen är kopplade till varandra genom olika relationer.
  - DOM är standardiserat av W3C
  - Trädliknande modell av sidan som den ”finns inne i webbläsaren”
  - Stödjer koppling mellan statisk 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

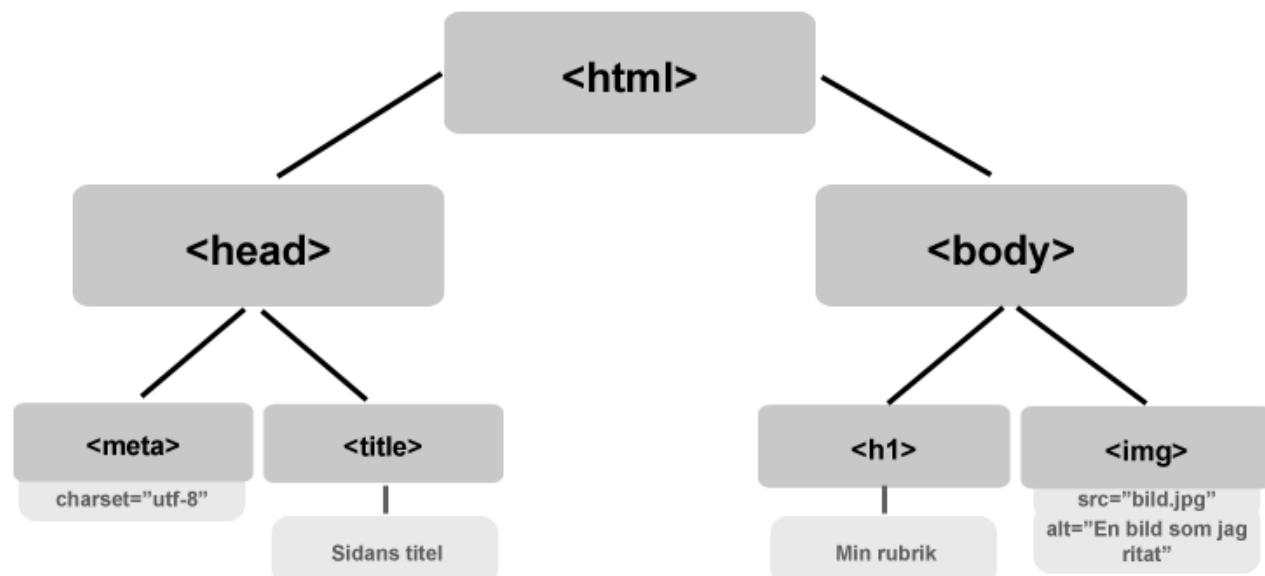
- Samma bild, ungefär ... men vi frilägger strukturen ...



# Källkod vs. noder

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

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

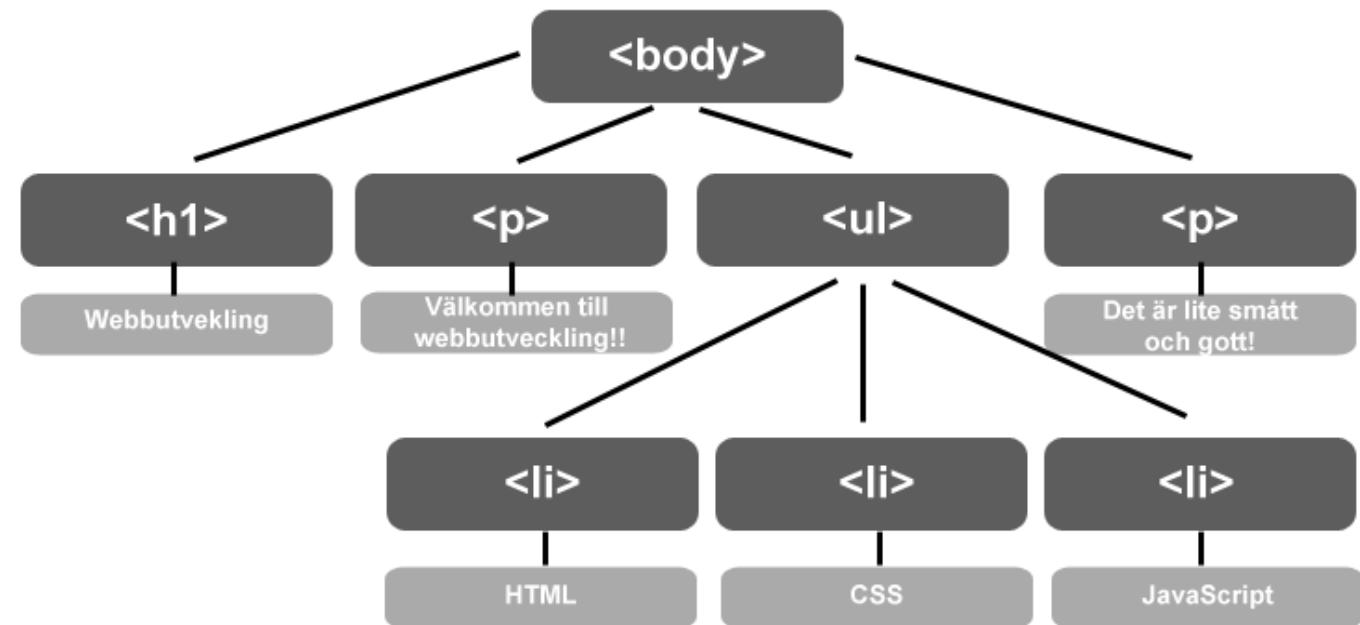


# Arv i CSS

- 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).

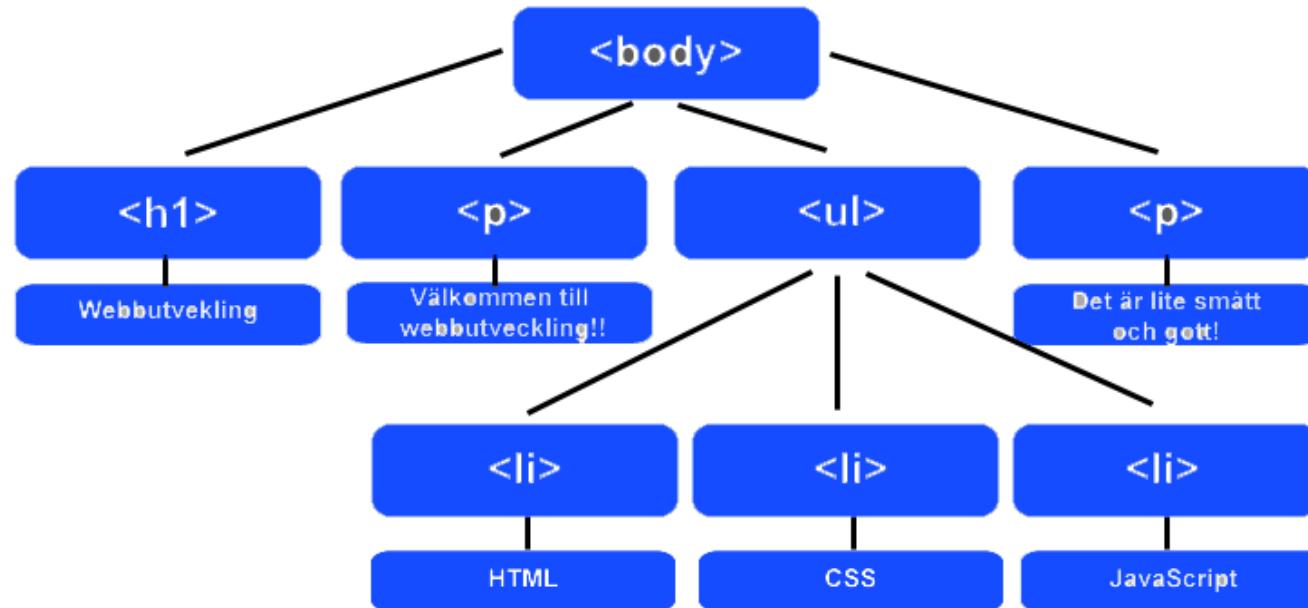
# Arv - exempel

```
<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>
```



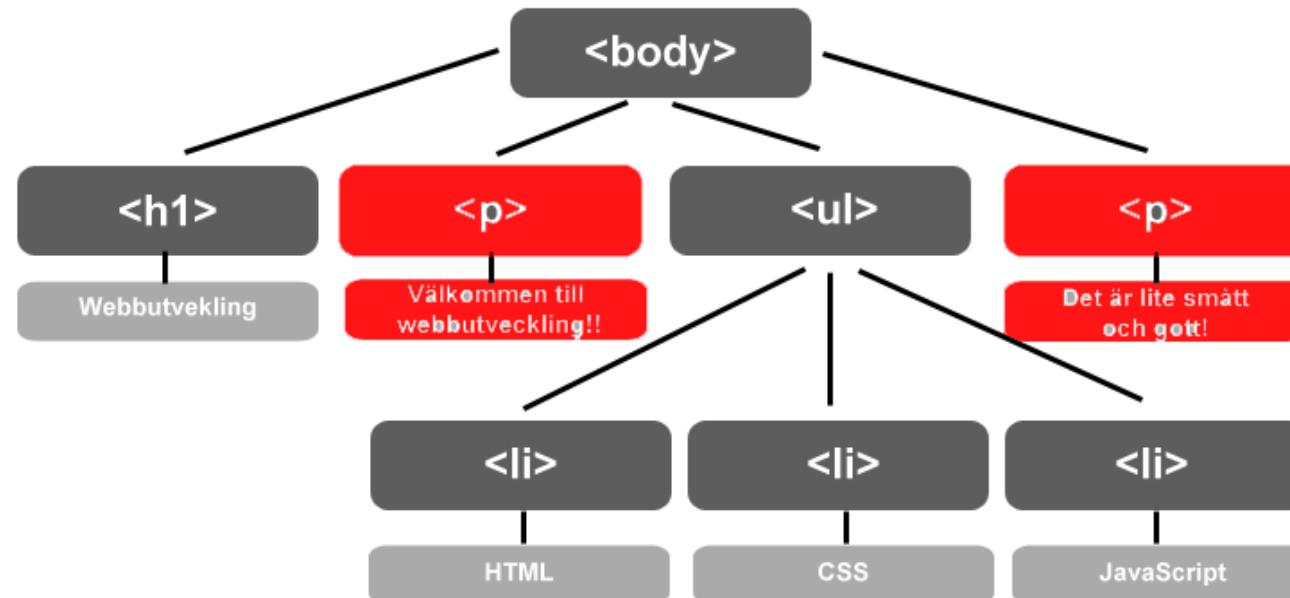
# Arv - färgläggning

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



# Arv färgläggning

```
p{  
    color: red;  
}
```

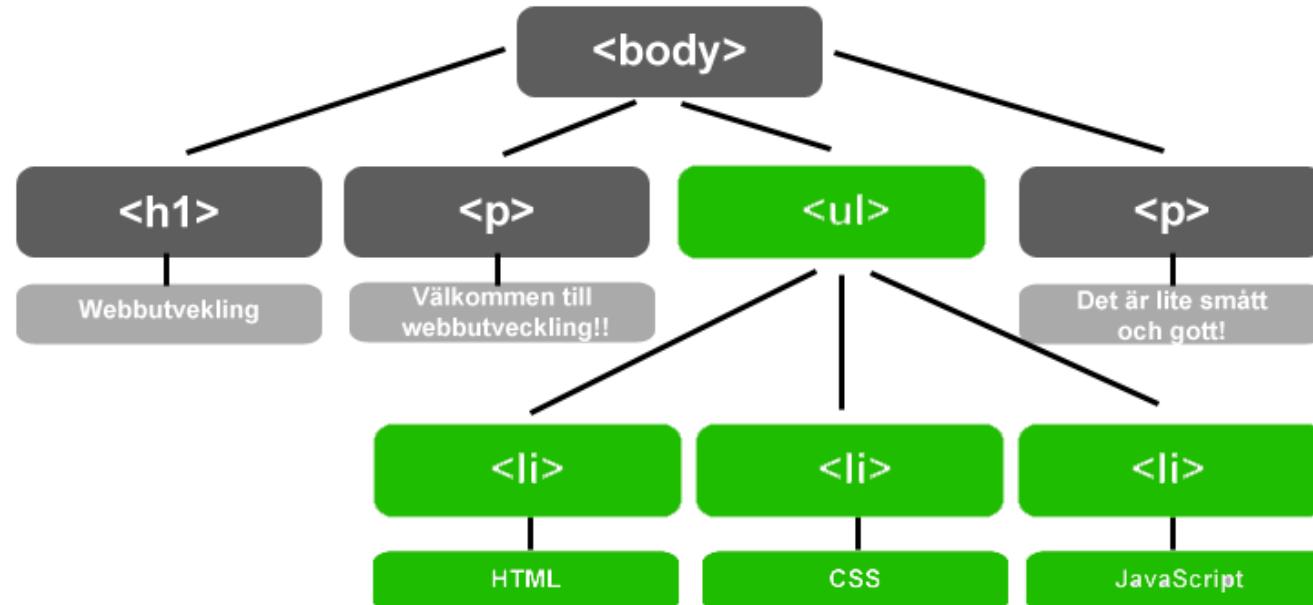


# Arv färgläggning

ul{

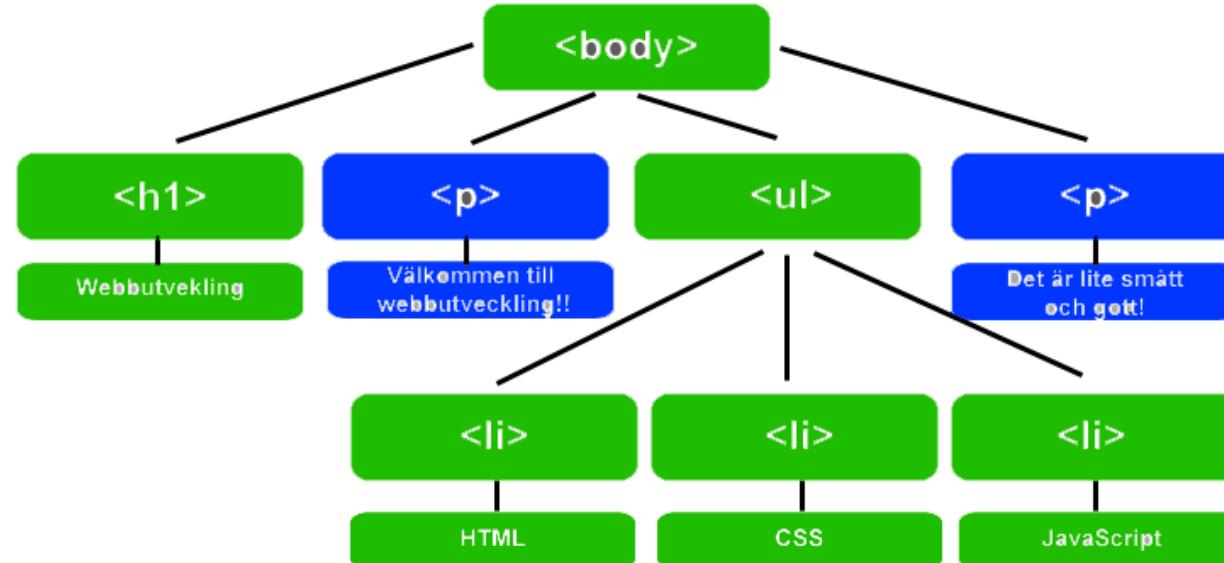
color: green;

}



# Arv – flera färger

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



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

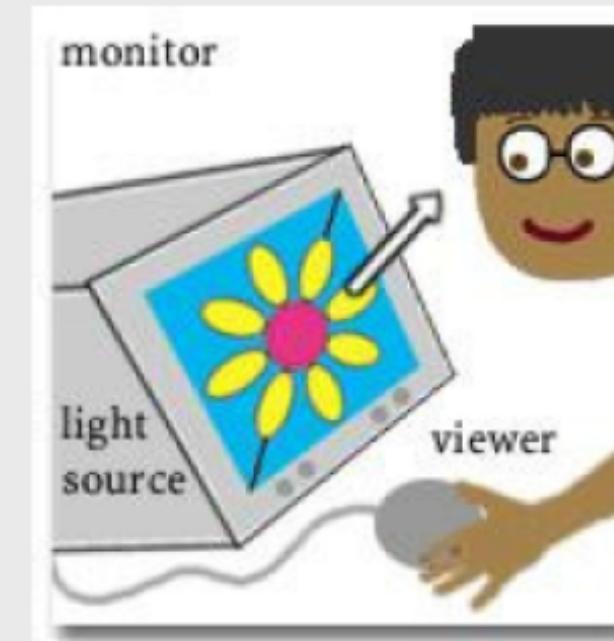
- Nödvändigt att använda då datorn bara förstår 1 och 0.
  - Det betyder att allt i datorn (färger, figurer, knapptryckningar, beräkningar, musrörelser, bokstäver, siffror, typsnitt, bilder) representeras binärt.
  - T.ex.  
bokstaven A = 65 (dec) = 01000001 (bin) = 41 (hex)
- I webbteknik kommer vi 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 använda namn ”red/green/blue” osv.  
Fast detta begränsar färgvalet något enormt!
- Istället använder man s.k. hexadecimal t.ex.  
#08ht5s för att ange en viss färg.

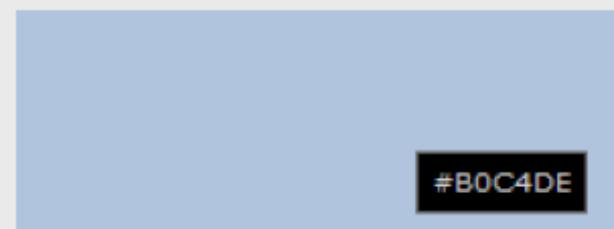
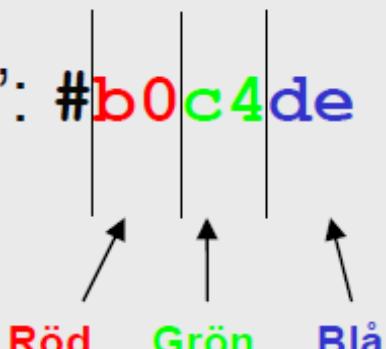
# RGB - används för additiv beskrivning

- 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  $256^3 = 16777216 = 16,8 \text{ Milj färgnyanser}$

En "webbfärg": #**b0c4de** anges hexadecimalt →



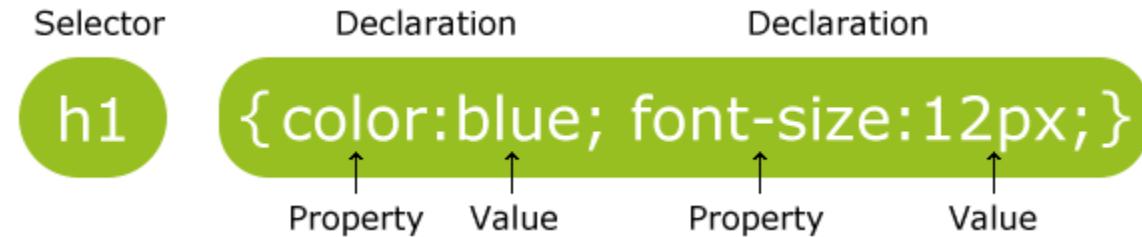
# Färgval

<http://colorschemedesigner.com/>

# CSS – text-egenskaper

```
/*
    Mall för CSS-egenskaper:
    egenskap: värde;
*/
body{
    font-family: arial, courier; /* Typsnittsfamilj */
    font-size: 12px;             /* Textstorlek */
    font-weight: bold;           /* Fetstil text */
    font-style: italic;          /* Kursiv text */
    text-decoration: underline;  /* Understrucken 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



**CSS**

Global  
HD