Cascading Style Sheets (CSS)
An Introduction

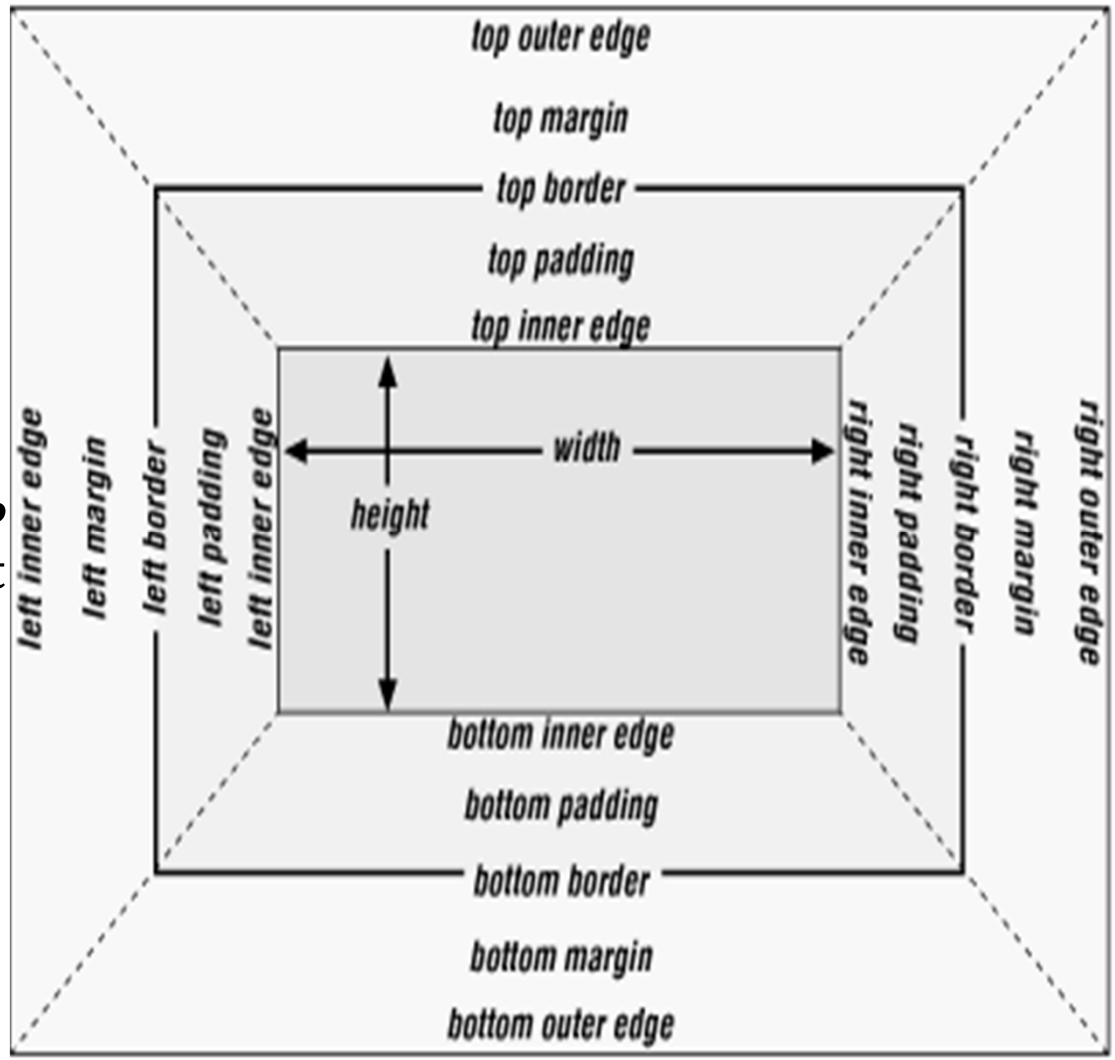
Nayeb Maleki

CSS

- CSS utvecklades under mitten av 90-talet för att kunna separera stilformatering och formateringsinformation från innehåll i HTML sidor.
- Fungerar väl med XML eftersom XML uteslutande har bättre och tydligare definierad struktur än HTML.
- HTML var inte menat för att användas för presentation.
- Taggar lades till för att ge möjlighet till presentation tillsammans med struktur:
  - <font>
  - <b>
  - <i>
- CSS förenklar globala och momentana ändringar.

The Box Model

- Varje element i DOM (Document Object Model) har en konceptuell "box" för presentation.
- Box:en består av margin, padding, border, content (width, height), och offset (top, left)



CSS Syntax

```
selector/element {
  property: value;
}
```

Selector kan vara en grupp av element, en identifierare, class eller viss kobimation av dessa.

Selector Förklaring

En selector är ett syntax som avgör vilka element som skall påverkas av vilka regler.

namn En visst typ av element, t ex alla namn element

.kurskod Alla element med attributet ”class” satt till något värde, t ex. ”class=kurskod”

#betyg Elementet med ett visst id, t ex id=”betyg”

div.kursnamn Alla div element med class=”kursnamn”

div kursbeskrivning Nestlade selectorer, t ex alla kursbeskrivningar som ligger innuti div

div.kursnamn teacher Alla kombinationer fungerar

Selector Class

```
<p class="intro">This is my introduction text</p>

.intro {
  font: 12px verdana, sans-serif;
  margin: 10px;
}
```

The Identifier Selector

```
<p id="intro"> This is my introduction text</p>

#intro {
  border-bottom: 2px dashed #fff;
}
```

CSS Selectors

Kom ihåg:

Id används för att kunna skilja ut element och taggar unikt. Class används om du vill använda samma formatering för flera områden på en webbsida.

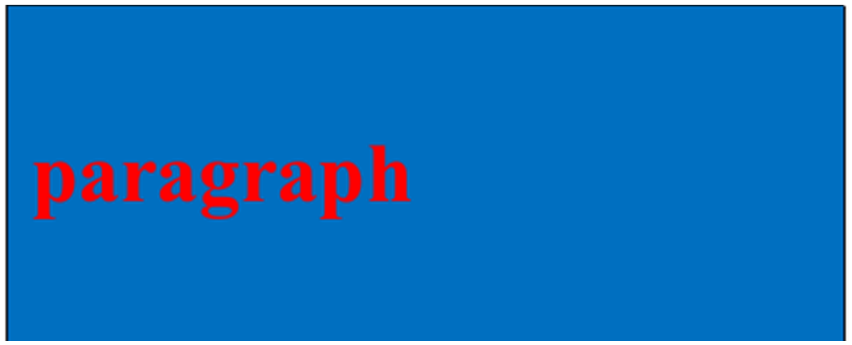
Läs mer:
http://reference.sitepoint.com/css/selectorref

Stil arv

- Om en still inte är specificerad för ett element, så ärver den stilen från sitt föräldrar -element.
- Detta kallas stil-arv

body {color: blue}

p {color: red}



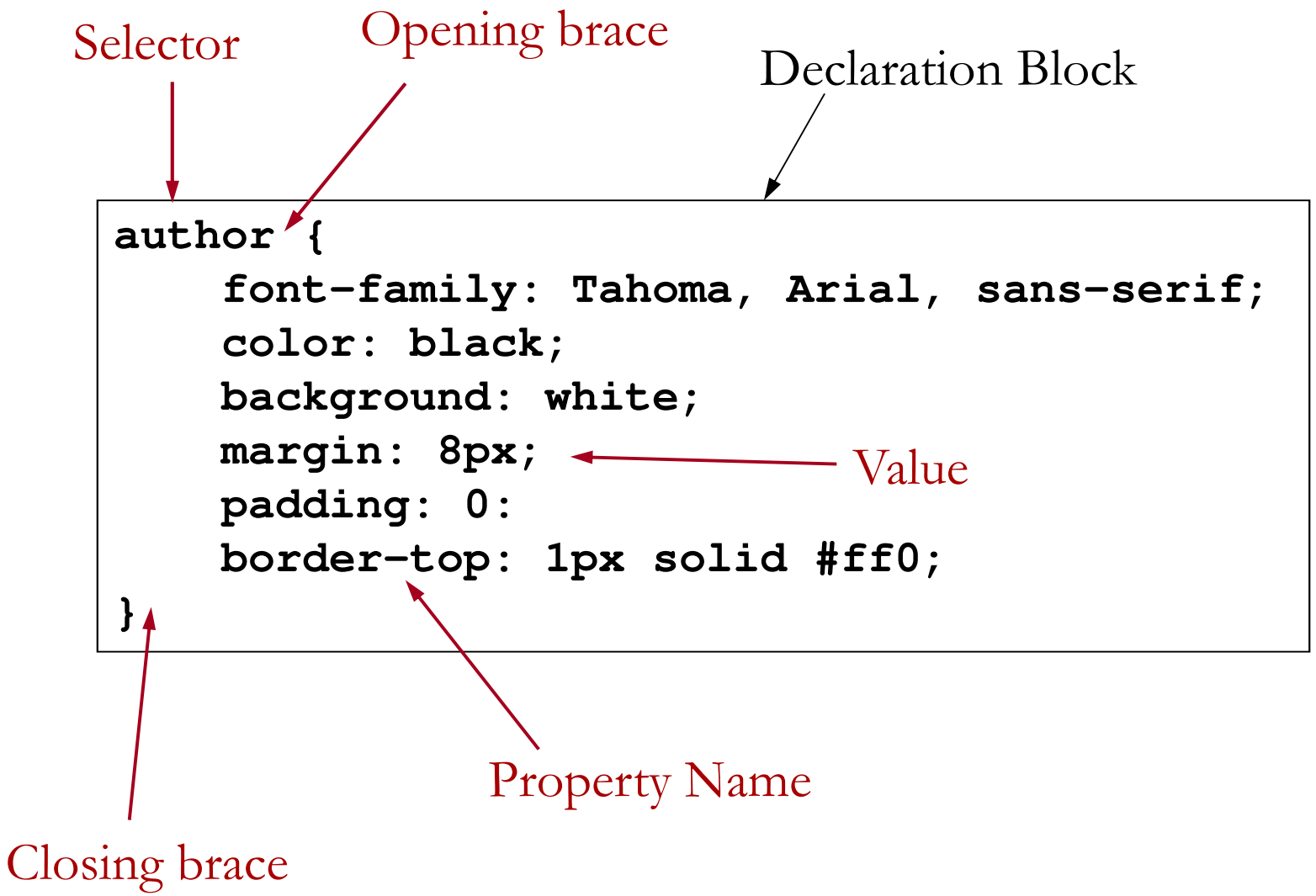
- I ovanstående exempel, så är 'body' elementet föräldra-element
- Genom stil-arv, så kan alla ändringar som man gör i ett 'style sheet' automatiskt skickas vidare till objekt och element nivåerna på websiten
- (Detta är anledningen till namnet "cascade style sheet").

Gruppering av Element

Ger möjlighet till att en stil kan specificieras till flera element på samma gång.

```
h1, h2, h3, h4, h5, h6 {
  font-family: “Trebuchet MS”, sans-serif;
}
```

CSS Regel


Selector
Opening brace
Declaration Block
author {
font-family: Tahoma, Arial, sans-serif;
color: black;
background: white;
margin: 8px;
padding: 0:
border-top: 1px solid #ff0;
}
Value
Property Name
Closing brace

Using Styles Sheets

Separate/external Stylesheet

```
<head>
  ...
  <link rel="stylesheet" type="text/css" href="myStyles.css" />
  <style type="text/css">
    body {
        font-family: Tahoma, Arial, sans-serif;
        ...
    }
  </style>
</head>
<body>
  ...
  <div style="padding:2px; ... ">
  ...
</body>
```

Page-Specific/embedded Styles

Element-Specific/ inline Styles

Ordning

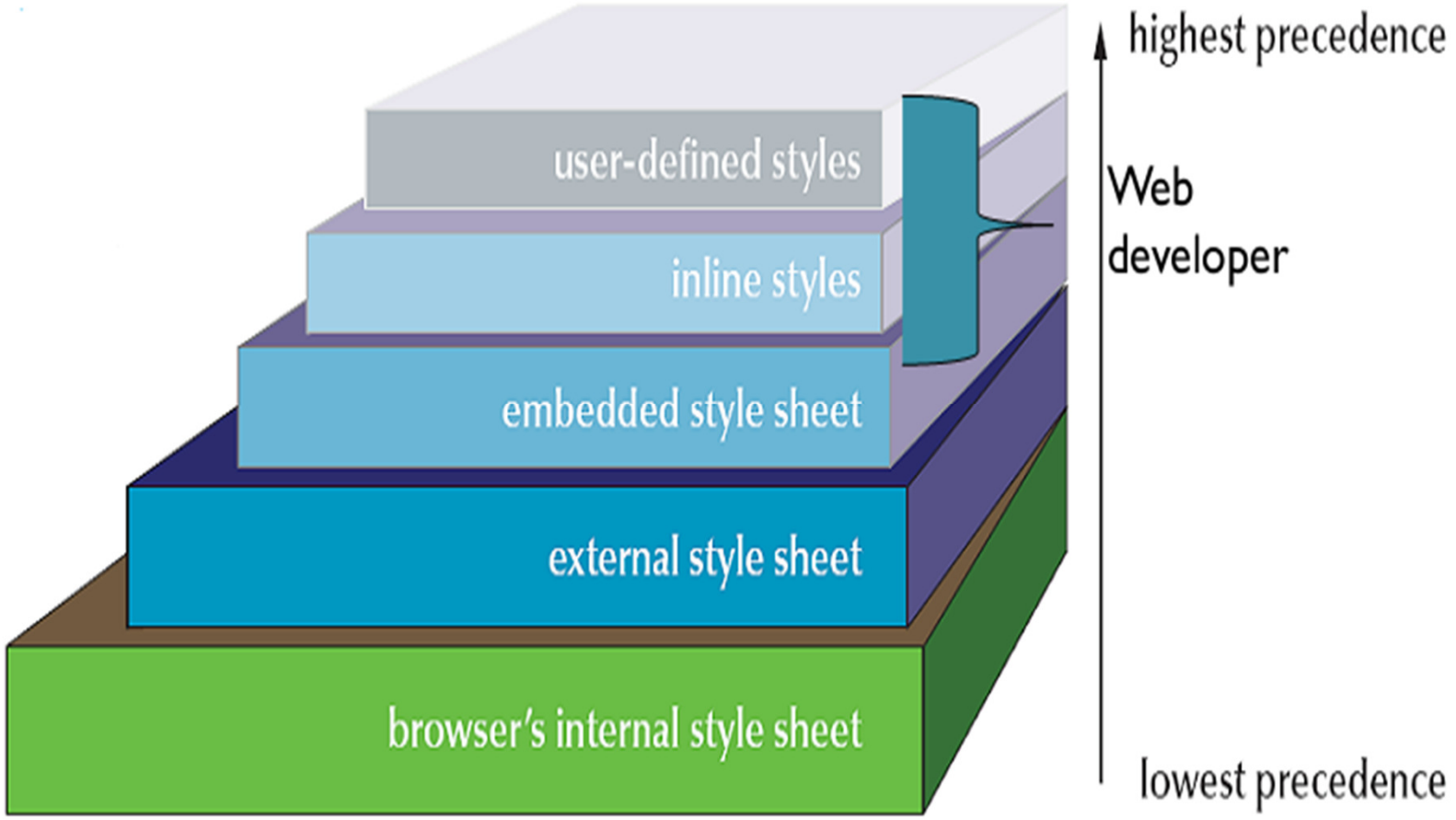
- När det uppstår stilkonflikter mellan stilar på samma nivå den senast deklarerade har prioritet

```
<style type="text/css">
        h1 {color: orange; text-align: center;
        h1 {color: blue}
</style>
```

- Skriv över prioriteten genom att lägga till !important egenskapen

```
<style type="text/css">
        h1 {color: orange !important;
        h1 {color: blue; text-align: center;}
</style>
```

Ordning


user-defined styles
inline styles
embedded style sheet
external style sheet
browser's internal style sheet
highest precedence
Web
developer
lowest precedence

CSS: Sample.css

```
p
{
   font-family: Tahoma;
   font-size:  1em     /*12pt  */
}

h1
{
   font-family: Arial;
   font-size:  2em;    /*  24pt */
   font-style: italic;
   color: red;
   text-align: center
}

li
{
   font-size: 0.75em;   /*9pt;*/
   font-weight: bold
}
```

HTML: Sample.html

```
<html>
<head>
<title>CSS Example</title>
</head>

   <body>
      <h1>Internet Technologies</h1>
      <p>Topics You Should Know:
      <ul>
        <li>HTML
        <li>XML
        <li>VBScript
        <li>Java
        <li>TCP/IP
      </ul>
   </body>
</html>
```

CSS: paragraph.css

```
body {
    font-family: Tahoma, Arial, sans-serif;
    font-size: 13px;
    color: black;
    background: white;
    margin: 8px;
}

h1 {
    font-size: 19px;
    margin-top: 15px;
    margin-bottom: 5px;
    border-bottom: 1px solid black
}

.shaded {
    background: #d0d0ff;
}
```

HTML: paragraph.html

```
<head>
<title>CSS Example</title>
<link rel=stylesheet type="text/css"
href="paragraph.css">
</head>
<body>
    <h1>First Section Heading</h1>
    <p>
        Here is the first paragraph, containing
        text that really doesn't have any use
        or meaning; it just prattles on and on,
        with no end whatsoever, no point to
        make, really no purpose for existence
        at all.
    </p>
    <div class="shaded">
        <h1>Another Section Heading</h1>
        <p>
          Another paragraph.
        </p>
    </div>
</body>
```

Färger i CSS

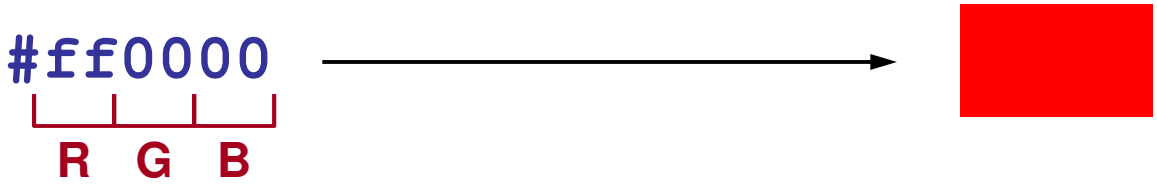
- Varje färg representeras av en triplett av tal som kallas RGB-tripletten baserat på styrkan av dess Röda, Gröna och Blå komponenter: rgb(red, green, blue)
- Intensiteten av var och en av de tre färgerna tilldelas ett tal från 0 (frånvaro av färg) till 255 (högsta intensiteten)
- rgb (0,0,0) vs. rgb (255,255,255)
- På det här sättet så får man 255 ^3 dvs. cirka 16.7 millioner olika färger

Färger i CSS

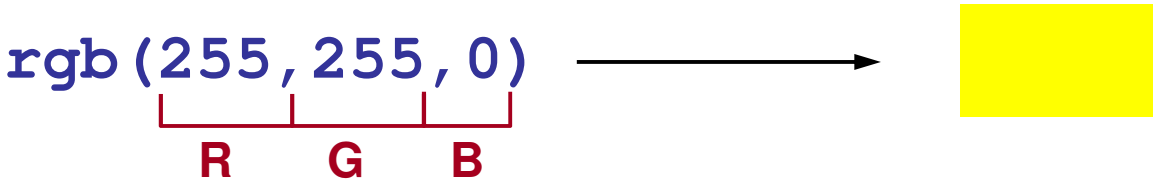
- Fördefinierade namn:

  white black red …

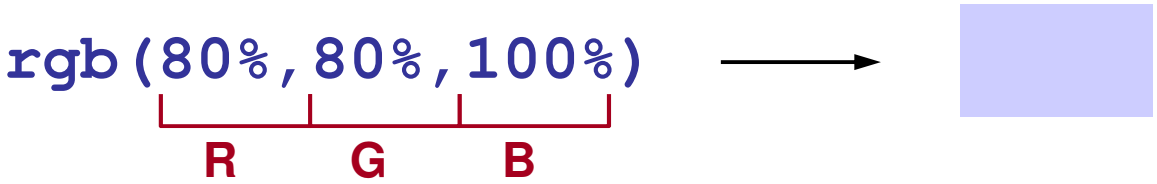
- 8-bit hexadecimal intensitet för röd, grön, blå:



- 0-255 decimal intensitet :



- Procentuella intensitet:



CSS Distances

2px	pixels
1mm	millimeters
2cm	centimeters
0.2in	inches
3pt	printer's points
2em, 4ex	other printer's units

Sätta upp alternativa 'Style Sheets'

<link href="large.css" rel="alternate stylesheet" type="text/css"

title="Large Text" />

<link href="regular.css" rel="alternate stylesheet" type="text/css"

title="Regular Text"/>

Se: multiple_css.xml

Återanvändning av Style Sheets

```
/* Start with a baseline style */

@import: url(HamletAct.css);

/* Don't display titles and speaker lines */

line {display: none}

title {display: none}
```

Ordning

Man kan länka ett 'style sheet' till flera dokument på web-siten genom att använda länkelement eller @import element

<style type="text/css">

@import url(company.css)

@import url(support.css)

</style>

Displaying elements (hamlet3.css)

```
speaker, title {
        display:block;
        font-size: 14 pt;
        font-weight: bold
   }
line, scndescr {
        display:inline;
        font-size: 8 pt
}
persona   {
        display:block;
        font-size: 10 pt;
        font-style: italic
}
```

Se: hamlet3.xml

The block parameter indicates that the element should start a new line, while inline(the default) says to run text together.
For other values on display see:
http://www.w3schools.com/cssref/pr_class_display.asp

The font-weight property sets how thick or thin characters in text should be displayed.
For other values on font-weight see:
http://www.w3schools.com/cssref/pr_font_weight.asp

The font-style property specifies the font style for a text.
For other values on font-weight see:
http://www.w3schools.com/cssref/pr_font_font-style.asp

Displaying elements (hamlet3.css)

```
personae title  {display: block;
                font-size: 2em}
 speaker, title {
                display:block;
                font-size: 1.5em;
                font-weight: bold
   }
line, scndescr {
                display:inline;
                font-size: 0.75em
}
persona        {
                display:block;
                font-size: 1em;
                font-style: italic
}
speaker >speech{ display: block;
                font-size: 2em }
```

Se: hamlet3.xml

Contextual Selectors

the title element is selected only when it appears as the descendant of a personae element.

Child Selector

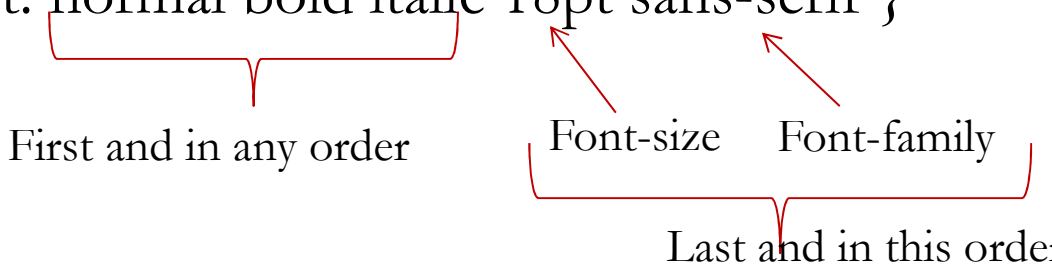
the **speaker** element is selected only when it appears as the child of a **speech** element.

Font attribut

- font-family == Serif, Sans-Serif, Monospace, Cursive, Fantasy, …
- font-style== normal, italic, oblique,
- font-variant == normal, small-caps
- font-weight == normal (400), bold (700), bolder, lighter
- font-size == absolut keyword (xx-small, x-small, small, medium, large, x-large, xx-large) , relative to parent (smaller, lager), percentage of parent (50%), absolut measure (18pt, 0.5in, 3cm, 30px)

Font shorthand

- focus { font: normal bold italic 18pt sans-serif }



Pseudo Elements and Classes

- CSS 1: first-letter, first-line : används för formatering av det första tecknet eller den första raden av ett element.
- CSS 2: lägger till before and after som gör det möjligt att skapa ett helt nytt element som inte existerade i käll dokumentet.
- CSS Level 2 introducerar pseudo classes som söker inte ett specifikt element utan vilket element som helst som faller inom ramen för sök-kriterierna. Det finns 7 sådana klasser som mestadels används inom publicering (till exempel: first, left, och right pages). Mest intressant är first-child som söker det första barnet till ett element.
- Exempel: document:first-child info{ color: red; }
- Läs mer: http://www.w3schools.com/css/css_pseudo_classes.asp

Adding Content

- I CSS Level 2 **content** egenskapen kan kombineras med before och after *pseudo element* för att lägga till text före och/eller efter ett element. Exemplet nedan lägger till innehåll i speaker och line elementen:

```
speaker {display: block; font-weight: bold}

speaker:before {content: "Character: "}

speaker:after {content: ":"}

line {display: block}

line:before {content: "Dialog: "}
```

Attribut Selektorer

- Course[Segments] { color: green }
- Element med attributet Segments oavsett värde.
- Course[Code="ADTC-510"] { color: purple }
- Element med attributet Code med specifikt värde.
- Sample[WeekDays~="Thursday"] { color: yellow }
- Element med ett attribut som har ett värde innehållande det givna värdet (innehållet kan vara en lista eller en mening).

Automatiska Räknare

- CSS Level 2 erbjuder enkla men eleganta sätt att öka, nollställa, visa räknarvärden genom att använda counter-increment, counter-reset, och counter( ) funktionen. Man kan ha flera oberoende räknare som måste tilldelas ett namn. Räknarvärdet som skall visas kan vara ett tal, bokstäverna A till Z eller a till z, romerska siffror eller många andra alternativ.
- /* Reset the counter whenever a new scene starts */

  scene {counter-reset: lineCount 0 }
- /* Increment the counter whenever we encounter a line element */

  line {counter-increment lineCount 1 }
- /* Display the counter value before the line */

  line:before {content: counter(lineCount)}

Automatiska Räknare, se: list_counter.xml

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="list_counter.css" type="text/css"?>
<root>
    <document>
        <info>Some info. Here </info>
        <list level="first">
            <item>Item</item>
            <item>Item</item>
            <item class="none">
                <list level="second">
                    <item>Item</item>
```

Automatiska Räknare, se: list_counter.css

```
list[level="first"] { counter-reset: first 0;  }

list[level="first"] > item:before { counter-increment: first 1;

        content: counter(first)". ";    }

list[level="second"] {  counter-reset: second; }

list[level="second"] > item:before { counter-increment: second;

content: counter(first)". "counter(second)" ";  }

list[level="third"] {  counter-reset: third; }

list[level="third"] > item:before { counter-increment: third;

        content: counter(first)"."counter(second)"."counter(third)" ";

}
```

Namnrymder i CSS

- **Två delar:**
  - Deklaration av namnrymden (länka till ett namn och URI) genom en @regel
  - Selektorer som refererar till namnrymden. namespaces
- **Syntax: @namespace "URI"**
  - Deklarerar default namespace
- **Syntax: @namespace prefix "URI"**
  - Deklarerar en prefix till en specifik namespace.
- **Exempel:**
- **@namespace "http://www.w3.org/1999/xhtml";**
  @namespace svg "http://www.w3.org/2000/svg";
  - Deklarerar xhtml namespace som default namespace samt skapar prefixen svg som skall gälla för svg namespace.

Namnrymder i CSS

namespace1.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet href="namespace1.css" type="text/css"?>
<p:Product xmlns:p="http://www.ns.com/ns/Product">
        <p:Code>This should be green.</p:Code>
        <p:Price>This should be purple.</p:Price>
</p:Product>
```

namespace1.css

```
@namespace p  "http://www.ns.com/ns/Product";
p|Product * { display: block; }
p|Code { color: green; font-weight: bold; }
p|Price { color: purple; font-weight: bold; }
```