Cascading Style Sheets

- CSS - Cascading Style Sheets language
- Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g. fonts, colors, spacing) to Web documents
- designed for HTML in 1996
- Has its own syntax
- CSS1 (1996), CSS2 (1998), CSS Mobile Profile
- CSS3 (working drafts)
- browser support can be checked:
  - http://www.w3schools.com/css/css_reference.asp

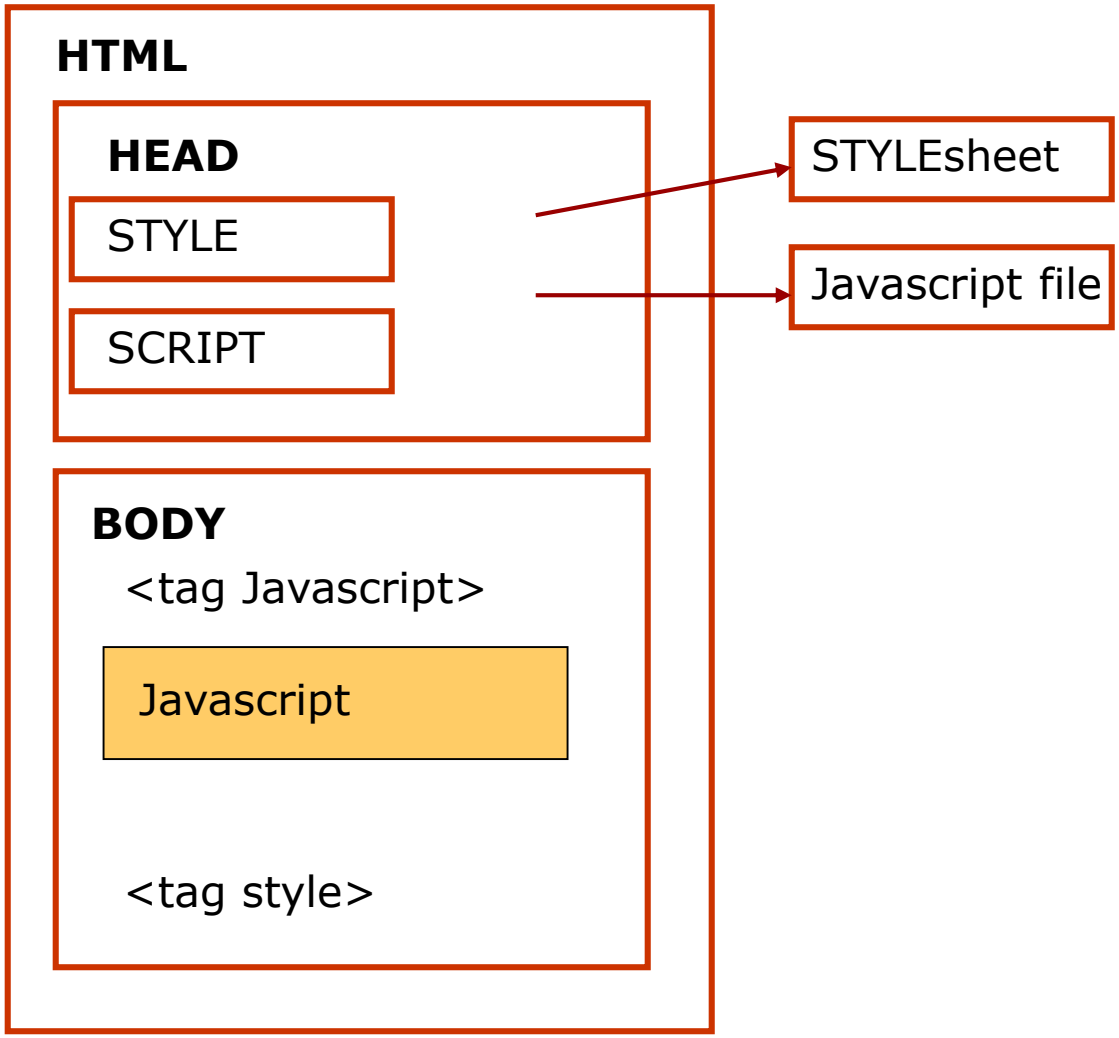
CSS rules

- CSS style rules:
  - Each rule starts with an element name or other selector
  - followed by a list of style properties bracketed by { and }
  - element name defines where it is applied
  - Each style property starts with the property's name, then a colon and lastly the value for this property. When there is more than one style property in the list, you need to use a semicolon between each of them
    - property : value
    - property – i.e. font-size
    - value – i.e. `16 pt'
  - h2 {font-size: 16pt; font-weight:bold}
  - latest rule is valid

Selectors

- Element name(s)
  - Nesting separated by space li b
  - Parallel separated by commas h1, h2
- Class .
- Pseudoclass :
- Id #
- Attribute a[href^="https"]

Styles and Javascript on HTML pages



Using inline styles

```
<html>
<!-- Exercise 1: inline style -->
   <head>
   <title>Inline Styles</title>
   </head>
   <body>
   <p>This text does not have any style applied to it.</p>
   <!-- The style attribute allows you to declare inline -->
   <!-- styles. Separate multiple styles with a semicolon. -->
   <p style = "font-size: 20pt">This text has the font-size
   style applied to it, making it 20pt.</p>

   <p style = "font-size: 20pt; color: #0000ff">This text has
   the font-size and color styles applied to it,
   making it 20pt and blue.</p>
   </body>
   </html>
```

Internal styles

```
<html>
   <!-- Exercise 2: internal style -->
   <head>
   <title>Introduction to CSS </title>
<!-- Declaring a style in the header section -->
   <style type = "text/css">
       i { background-color: #8000ff; color: white }
       h1 { font-family: arial, sans-serif }
       p { font-size: 14pt }
       .blue { color: blue }
   </style>
   </head>
<body>
   <!-- This class attribute applies the .blue style -->
   <h1 class = "blue">Main heading</h1>
   <p> Some content   etc.
```

External styles

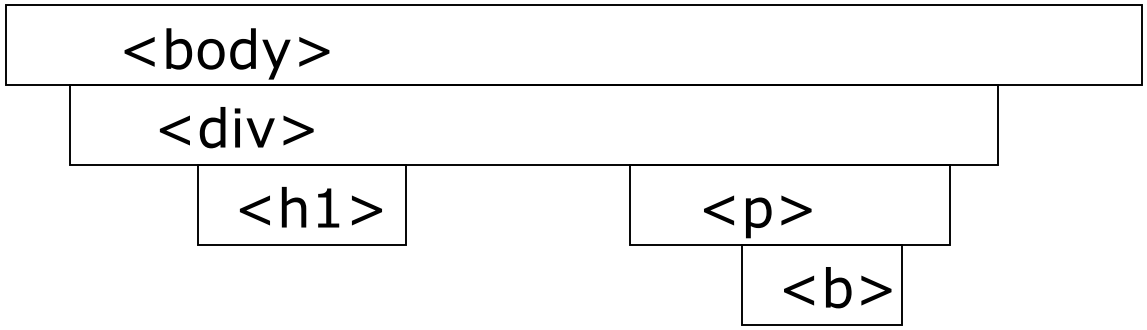
```
<html>
<head>
    <title> Linking an external style sheet </title>
    <link rel = "stylesheet" type = "text/css" href = "style4.css">
    </head>
<body>
    html document continues
```

File style4.css

```
/* Example: style4.css */
/* External style sheet */
a.nodec { text-decoration: none; } /* class
a:hover { text-decoration: underline; /* pseudoclass
    color: red;
    background-color: #ccffcc; }
li em { color: red; font-weight: bold; }
ul { margin-left: 2cm; }
```

Cascading Style Sheets

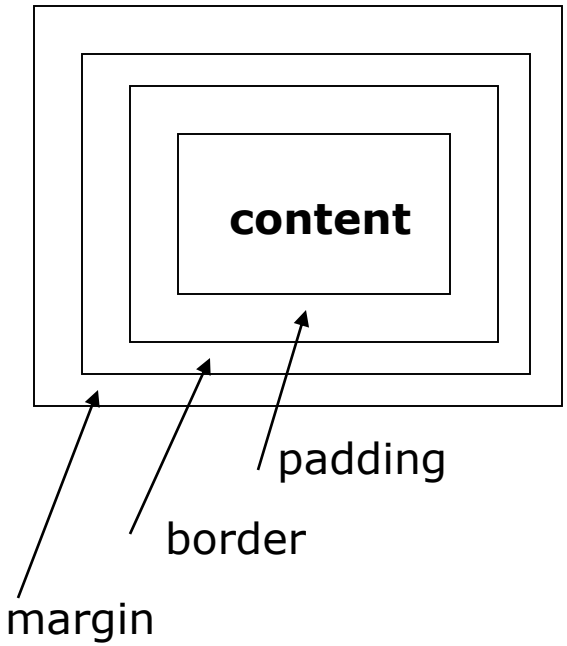
Inheritance

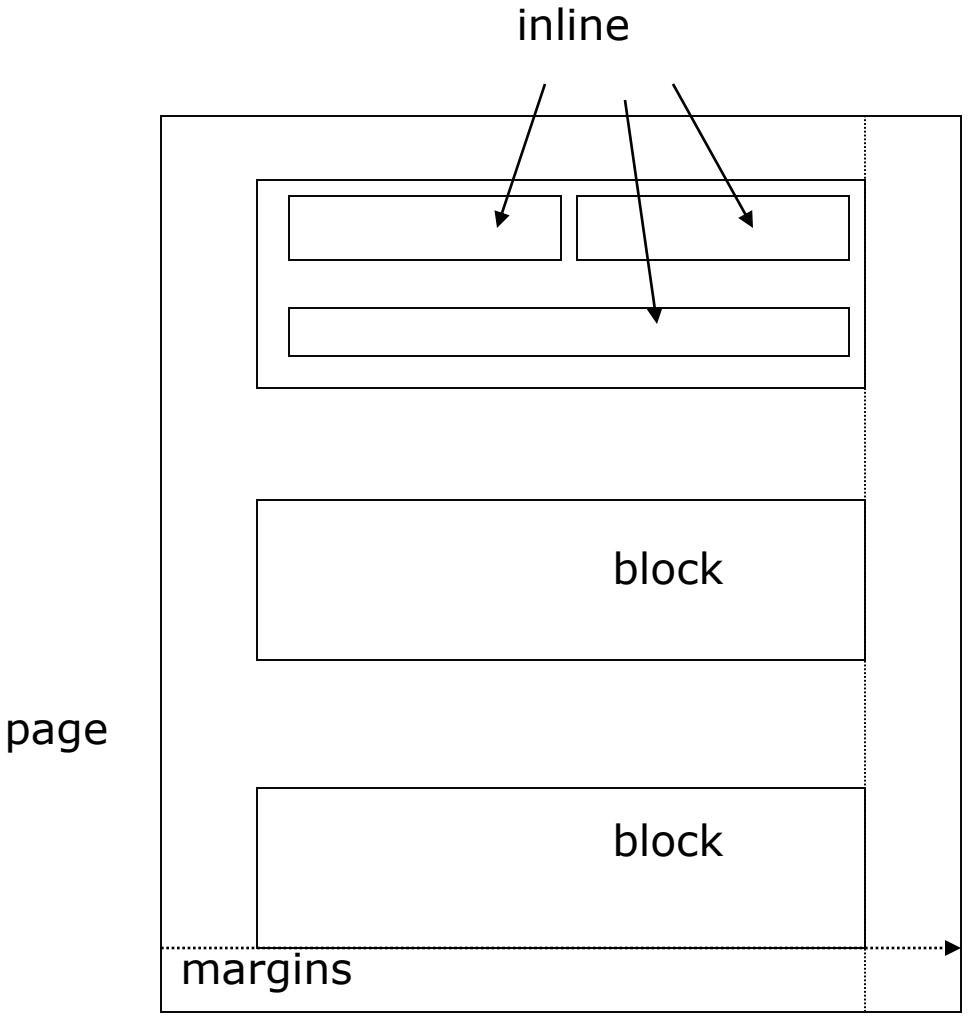


Inheritance

```
<style type="text/css">
  body {color : red}
  h1 {color : blue}
  p { font-family: arial, helvetica, sans-serif; padding:2px}
</style>
```

Text boxes


content
padding
border
margin


inline
block
block
margins
page

Text boxes

- padding: 2pt 3 pt 4pt 5pt
  - between frame and content
- margin: 2px, 3px, 4px, 5px
  - between boxes (order: top, right, bottom, left)
  - margin-left, margin-right, etc.
- width: 50%; height: auto
  - size of image
- float: left /* right */ (makes the box float)
- clear: both /* left, right*/ (no floating)

How the browser sets priority for style definitions

- Styles declared as important by the page creator
  {color : red !important }
- Styles declared as important by the user
- Normal styles by the page creator
- Normal styles by the user
- Default styles by the browser if there are no user styles

External information : Pseudo-classes

- normally CSS styles are based on HTML tags and attributes. Sometimes it is not sufficient.
- CSS introduces the concepts of pseudo-elements and pseudo-classes to permit formatting based on information that lies outside the document tree.
- Anchor pseudo-class:
  - Links (LINK) are VISITED or ACTIVE
  - Example:
    a:link {color : red}
    a:visited {color : blue}
    a:active {color : white}
  - The :hover pseudo-class applies while the user designates an element (with some pointing device), but does not activate it.

Using CSS on HTML pages

- Generalized use
  - Write style definitions to fit several pages, preferably the entire site. If you need page specific styles, use the STYLE attribute inside the document.
- Scalability
  - Avoid using absolute values for font size, margins, etc. Users can have very different browsers, screen resolutions, window sizes, etc. Use relative measures.
- You might need media queries to optimize for device type

CSS @media example

```
body {background: #f9f3e9; color: #594846; font: serif;}
.header, .footer {clear: both;}
.header {height:200px;}
.navigation{min-height: 25px;}
.header, .footer, .navigation {width: 100%;}

@media screen and (min-width:481px;)
        {#visit {width: 25%; float: left;}
        #points {width: 25%; float: right;}}

@media screen and (max-width:480px;)
        {#visit, #points {width: 100%;}}
```

Note: this uses div tag with id's (#) and classes

Using CSS on HTML pages

- Test the styles with different browser versions,
  - Design your pages to work also without style sheet support
  - Browsers behave and render pages in non-standard ways!
- Select fonts with care
  - people may not have all nice fonts on their PC
- Don’t push it too far!
  - Not every single element and attribute needs a pre-set value.
- Standardization: http://www.w3.org/Style/CSS/current-work

CSS2 extensions

- media
  - aural: sound
  - @media aural {
    document{play-during:url(backgroundmusic.mp3) repeat mix;}
    title {voice-family:'Elvis' male;volume:medium;}
    }
  - @media visual
  - @media braille
  - @media print {
    BODY { font-size: 10pt }
    }
  - @media screen {
    BODY { font-size: 12pt }
    }
  - @page

CSS3 modules

- Web fonts: @font-face downloads specific fonts
- Backgrounds: multiple backgrounds; background placement
- Borders
- Transformations: rotate; skew; scale; translate
- Transitions
- Animations

CSS3 and HTML5

- CSS 3: more decorative options and small animations
- HTML 5:
  - Structural approach
  - Added graphics canvases
  - Scalability possibilities

- Browser support for CCS2 and CSS3 features

  http://www.quirksmode.org/css/contents.html