Web Development

Lecture 9 – CSS Theory

CSS Theory

- Terminology
- Adding Styles
    - Inline
    - Embedded
    - External
- Alternate Style Sheets
- Importing Style Sheets
- Modular Style Sheets
- Document Structure
- Selectors
    - Type
    - Contextual
    - Class and ID
    - Attribute
- The Cascade

Terminology

```
h1
{
  font-size: 20pt;
  color: green;
}

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

Properties can take several types of values:

- predefined keywords
- percentage values
- length measurements
- colour values
- URLs

A specific property will only accept certain types of value.

W3Schools contains a reference section which lists all valid properties and their values.

Adding Styles to a Document

This can be done in three ways:

- Inline Styles
- Embedded Style Sheets
- External Style Sheets

Inline Styles

You can add style information within the HTML tag for that element by using the **`style`** attribute.

```
<h1 style="color: red">This heading will be red</h1>

<p style="font-size: 12px; font-family: Arial">This is
the content of the paragraph</p>
```

Although valid, this breaks the rule of keeping content and style separate.

Only used when you want to override higher-level rules.

The Style attribute has been deprecated in XHTML 1.1.

Embedded Style Sheets

A style block can be included at the top of the web page in the <head> tags. Again, this is not recommended.

```
<html>
  <head>
    <style type="text/css">
       h1 { color: red; }
        p {
            font-size: 90%;
            font-family: Verdana, sans-serif;
          }
    </style>
    <title>Example one</title>
  </head>
<body>
...
</html>
```

External Style Sheets

The styles should be stored in a separate document.

This is the most flexible way as an external style sheet can be used by all pages on the website.

This makes it easy to make stylistic changes to an entire website very easily.

You can put comments into a style sheet by using this syntax:

```
/* This is a comment */
```

The normal way to link to an external style sheet is to use the link tag at the top of the main document.

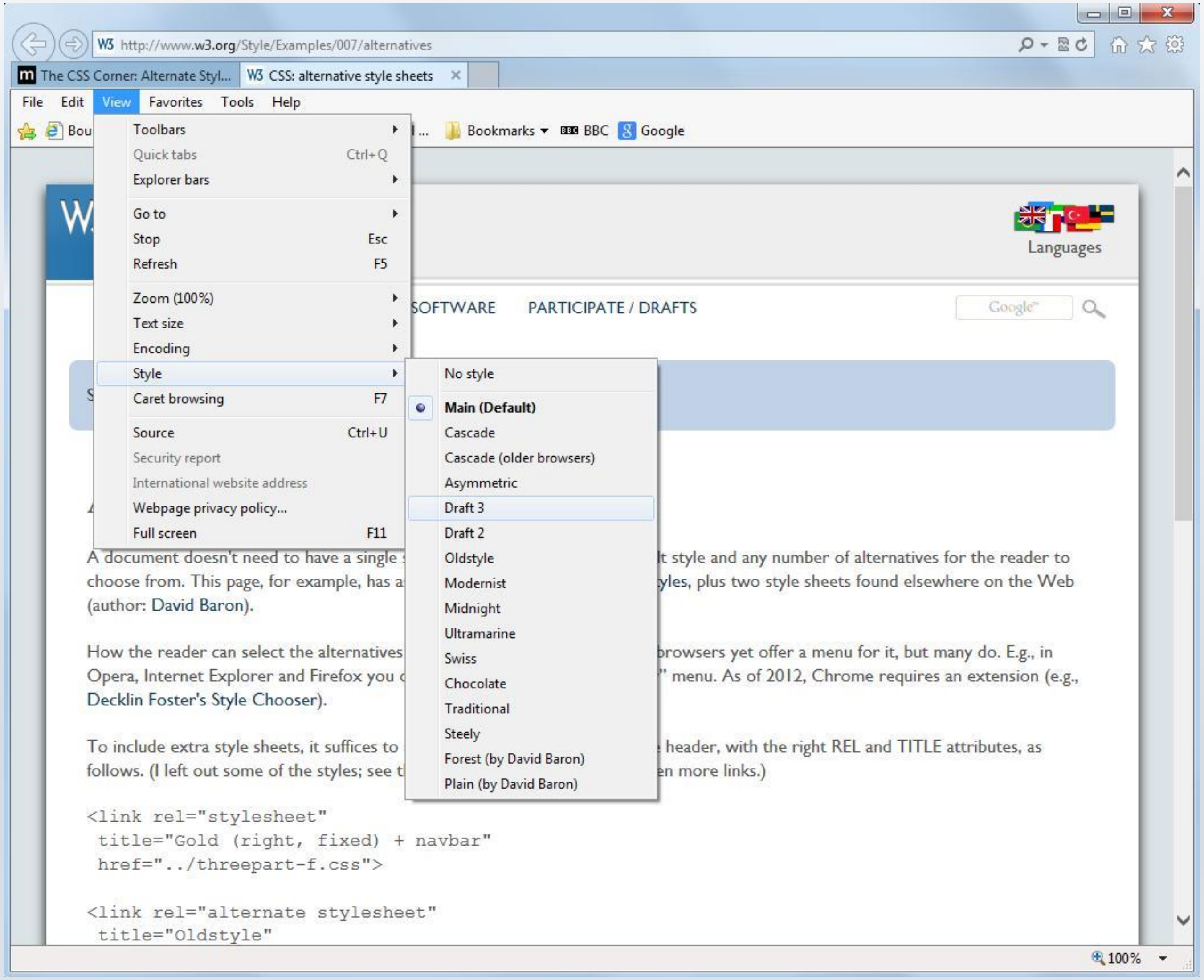
```
<head>
<link rel="stylesheet" type="text/css" href="otherdocs/style01.css" />
</head>
```

Alternate Style Sheets

CSS version 2 allows you to specify alternate stylesheets for the same document:

```
<head>
<link rel="stylesheet" type="text/css"
                href="basic.css" title="Basic Style" />
<link rel="alternate stylesheet" type="text/css"
                href="largetype.css" title="Larger Type" />
<link rel="alternate stylesheet" type="text/css"
                href="minimal.css" title="Minimal Design" />
</head>
```

http://www.w3.org/Style/Examples/007/alternatives



A document doesn't need to have a single … t style and any number of alternatives for the reader to choose from. This page, for example, has a … yles, plus two style sheets found elsewhere on the Web (author: David Baron).

How the reader can select the alternatives … browsers yet offer a menu for it, but many do. E.g., in Opera, Internet Explorer and Firefox you c … " menu. As of 2012, Chrome requires an extension (e.g., Decklin Foster's Style Chooser).

To include extra style sheets, it suffices to … header, with the right REL and TITLE attributes, as follows. (I left out some of the styles; see t … en more links.)

```
<link rel="stylesheet"
 title="Gold (right, fixed) + navbar"
 href="../threepart-f.css">

<link rel="alternate stylesheet"
 title="Oldstyle"
```

Importing

As well as linking to a style sheet, you can import a style sheet into a document:

```
<head>
    <style type="text/css">
       <!--
        @import url(http://pathname/stylesheet.css);
        -->
    </style>
    <title>Example one</title>
 </head>
```

Importing allows multiple style sheets to be applied to the same document.

The last one read (the one at the bottom of the list) takes precedence over the previous ones.

Modular Style Sheets

The @import command can be used within a style sheet, to pull styles in from other style sheets.

```
/* basic font styles */
@import url("type.css");

/* form inputs */
@import url("forms.css");

/* navigation */
@import url("list-nav.css");
```

Conditional Stylesheets

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
  <head>
    <title>Test</title>
    <link href="all_browsers.css" rel="stylesheet" type="text/css">
    <!--[if IE]> <link href="ie_only.css" rel="stylesheet" type="text/css"> <![endif]-->
    <!--[if lt IE 7]> <link href="ie_6_and_below.css" rel="stylesheet" type="text/css"> <![endif]-->
    <!--[if !lt IE 7]><![IGNORE[--><![IGNORE[]]>
                             <link href="recent.css" rel="stylesheet" type="text/css">
    <!--<![endif]-->
    <!--[if !IE]>--> <link href="not_ie.css" rel="stylesheet" type="text/css"> <!--<![endif]-->
  </head>
  <body>
    <p>Test</p>
  </body>
</html>
```

In the above example, all_browsers.css applies to all browsers, ie_only.css only applies to all versions of Internet Explorer, ie_6_and_below.css applies to all versions of Internet Explorer below IE 7, recent.css applies to all browsers except IE versions below 7, and not_ie.css applies to all non-IE browsers.

This approach seems to be falling out of use with newer browsers.

See: http://www.webdevout.net/css-hacks/ for more information.

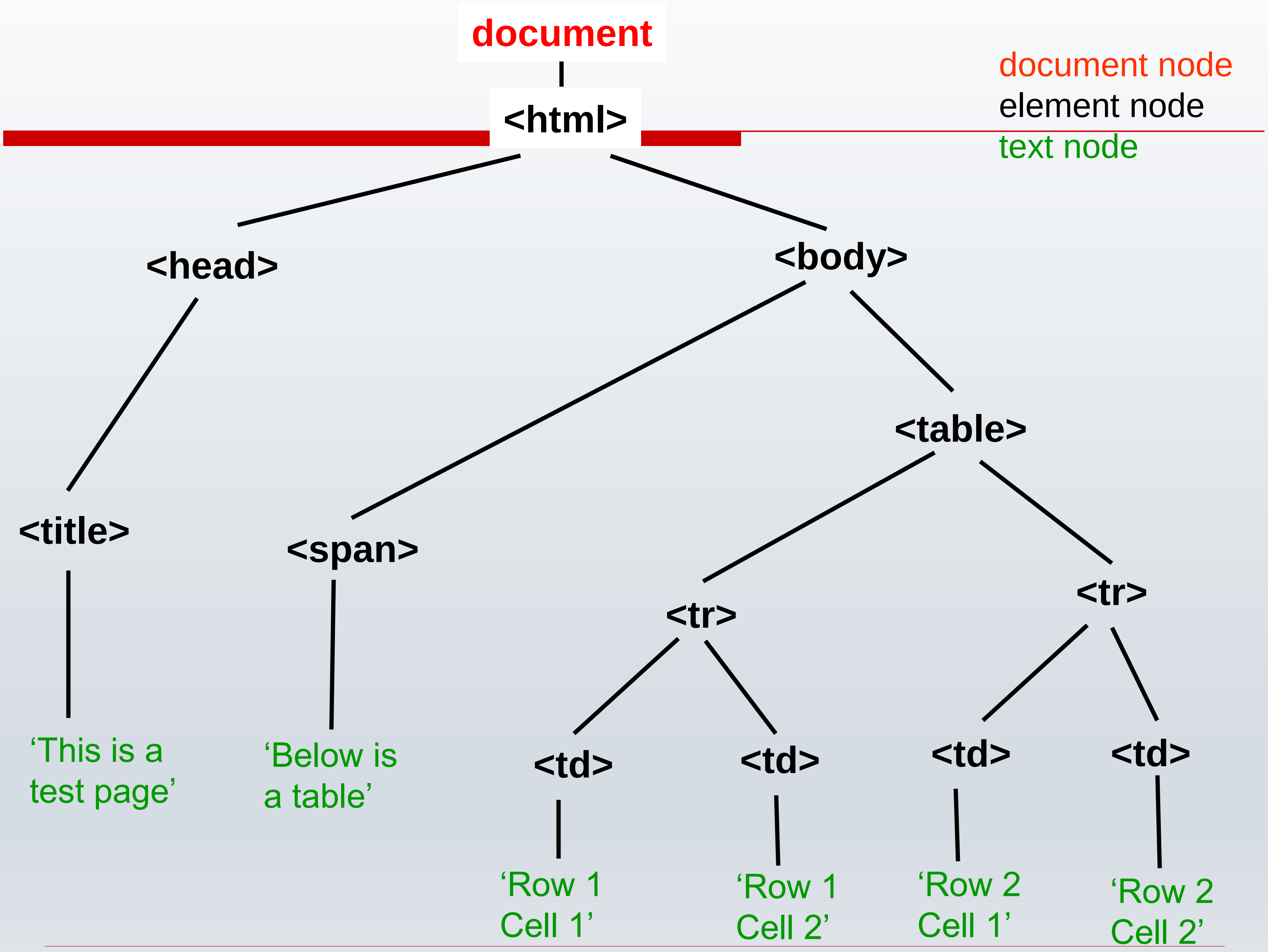
Document Structure

```
<html>
  <head>
    <title>
       This is a test page
    </title>
  </head>

<body>
  <span>Below is a table</span>
  <table border="1">
    <tr>
      <td>Row 1 Cell 1</td>
      <td>Row 1 Cell 1</td>
    </tr>
    <tr>
      <td>Row 1 Cell 1</td>
      <td>Row 1 Cell 1</td>
    </tr>
  </table>
</body>
</html>
```

All XML, HTML and XHTML documents have an implicit structure or hierarchy.

They can be thought of as an upside down tree structure.


document
<html>
<head>
<body>
<title>
<span>
<table>
<tr>
<tr>
<td>
<td>
<td>
<td>
'This is a test page'
'Below is a table'
'Row 1 Cell 1'
'Row 1 Cell 2'
'Row 2 Cell 1'
'Row 2 Cell 2'
document node
element node
text node

Selectors

There are several types of selector that you can use on your style sheets:

- Type (element) selectors
- Contextual selectors
    - descendant
    - child
    - adjacent sibling
- Class and ID selectors
- Attribute selectors

Type (element) selectors

These are the simplest ones, that just target an element by name:

```
h1 { color: blue; }
h2 { color: blue; }
p { color: blue; }
```

You can group them together:

```
h1, h2, p { color: blue; }
```

Contextual Selectors

You can apply style properties to select elements, depending upon their context.

Descendant

```
li p { color: green; }
```

Any paragraph elements which are descendants of list elements will be green – all other paragraph elements will be unaffected.

```
h1 span.phone, h2 p { color: red; }
```

```
h1 p li span#surname { font-size: 12pt; }
```

Child Selector (>)

A child selector is similar to a descendant selector, but it only targets direct children of a given element. (Not grandchildren, or great-grandchildren etc)

```
p > a    { background-color: gray; }
li > p > h1    { color: red; }
```

Adjacent Sibling Selector (+)

Targets an element that comes immediately after another element with the same parent element.

```
h1 + p    { font-family: Arial; }
```

will style the first paragraph immediately following an h1 heading.

Class and ID Selectors

To specify styles for elements of a particular class, use the dot:

```
h1.special { color: red; }
p.special  { color: red; }
```

To apply a style to all elements of the same class, leave out the tag name:

```
.special { color: red; }
```

To specify a style for elements with ID codes, use the hash symbol:

```
div#banner { height: 300px; width: 200px; }
```

You can also leave out the tag name if you wish:

```
#banner { height: 300px; width: 200px; }
```

Attribute Selectors

```
img[title] { border: 3px; }
```

All image tags with an attribute called 'title' will be given a border.

```
img[title="kitten"] { border: 3px; }
```

All image tags with an attribute called 'title' which has the value "kitten" will be given a border.

Inheritance

A child element can inherit style properties from its parent element.

Most properties are inherited, but some (margins, backgrounds) are not.

Styles applied to specific elements override settings higher in the hierarchy.

So, if you want most of the text on the page to be blue, set the color property on the <body> tag.

You can then apply a different rule to tag lower down in the hierarchy if you want it to be a different colour.

The idea that some rules can override others brings us to an important idea: The cascade.

Conflicting Style Rules: The Cascade

It is possible for elements in a document to get style instructions from several different sources.

They may conflict.

CSS rules define three orders of precedence to resolve these conflicts.

They are:

- Style Sheet Origin
- Selector Specificity
- Rule Order

Style Sheet Origin

If an element is getting four sets of style information from different stylesheets, then the one with the most weight will 'win'.

- Default Browser Styles (least weight)
- Reader Style Sheets
- Author Style Sheets
- Reader !important style declarations (most weight)

If the Reader Style Sheet contains a rule like this:

```
p {color: blue !important; }
```

then it cannot be overridden by a subsequent rule.

Selector Specificity

Once the browser has chosen which style sheet has precedence, there still may be conflicts within that style sheet.

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

All paragraph text will be red.
But, any paragraph text which is inside a table will be blue.

The more specific the selector, the more weight it is given.

■ Individual element selectors	p	(least weight)
■ Contextual selectors	table p	
■ Class selectors	div.special	
■ ID selectors	div#intro	(most weight)

Rule Order

If there are still several conflicting rules of equal weight, whichever one comes last, wins.

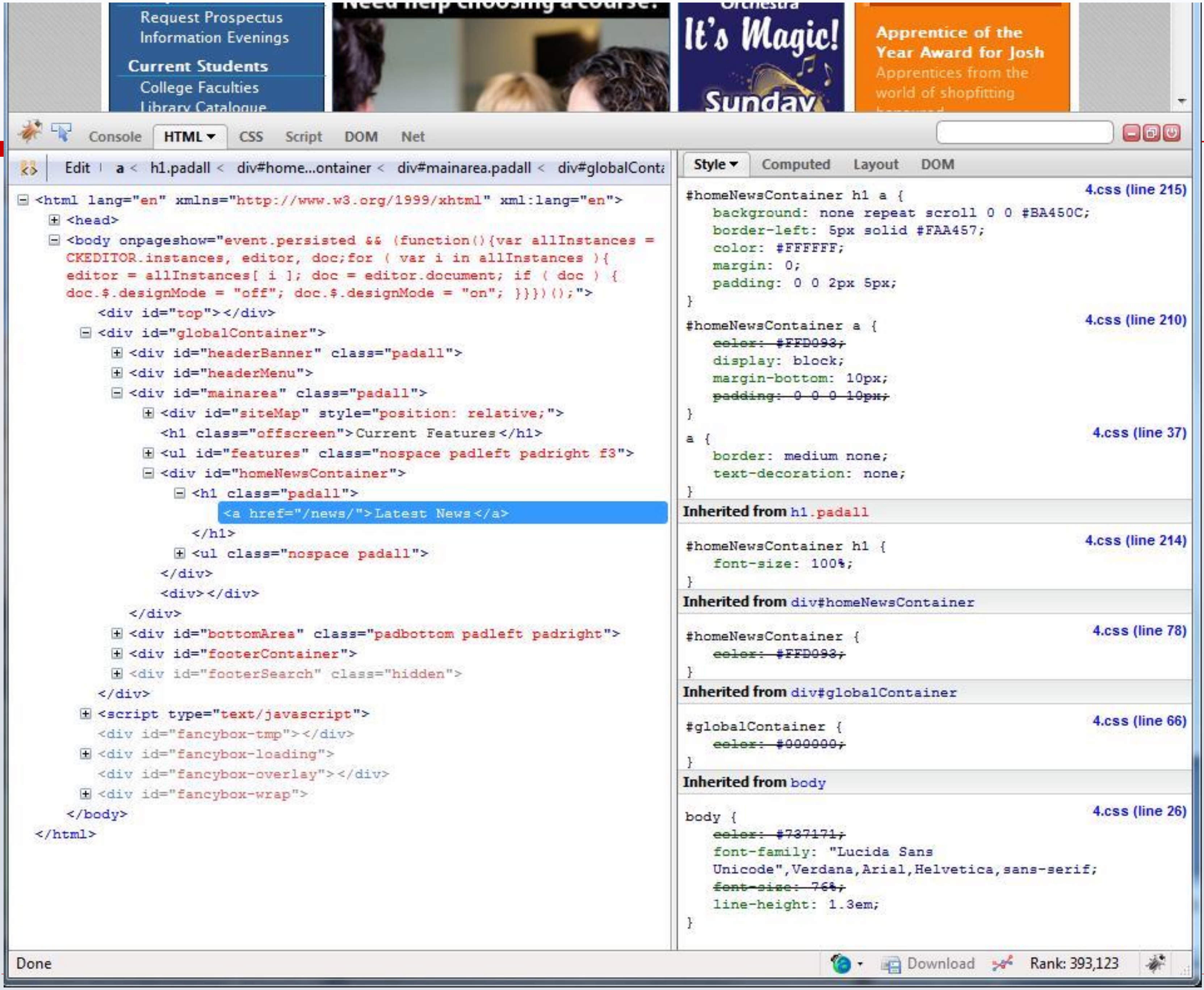
```
h1 {color: green; }
h1 {color: blue; }
h1 {color: red; }
```

So the h1 tags will be red.

This also works within a declaration block:

```
div#side  {  border-color: gray;
              border-color-top: black;   }
```

The second rule overrides the first.



```
<html lang="en" xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
  <body onpageshow="event.persisted && (function(){var allInstances =
  CKEDITOR.instances, editor, doc;for ( var i in allInstances ){
  editor = allInstances[ i ]; doc = editor.document; if ( doc ) {
  doc.$.designMode = "off"; doc.$.designMode = "on"; }}})();">
    <div id="top"></div>
    <div id="globalContainer">
      <div id="headerBanner" class="padall">
      <div id="headerMenu">
      <div id="mainarea" class="padall">
        <div id="siteMap" style="position: relative;">
        <h1 class="offscreen">Current Features</h1>
        <ul id="features" class="nospace padleft padright f3">
        <div id="homeNewsContainer">
          <h1 class="padall">
            <a href="/news/">Latest News</a>
          </h1>
          <ul class="nospace padall">
        </div>
        <div></div>
      </div>
      <div id="bottomArea" class="padbottom padleft padright">
      <div id="footerContainer">
      <div id="footerSearch" class="hidden">
    </div>
    <script type="text/javascript">
    <div id="fancybox-tmp"></div>
    <div id="fancybox-loading">
    <div id="fancybox-overlay"></div>
    <div id="fancybox-wrap">
  </body>
</html>
```

```
#homeNewsContainer h1 a {                                4.css (line 215)
   background: none repeat scroll 0 0 #BA450C;
   border-left: 5px solid #FAA457;
   color: #FFFFFF;
   margin: 0;
   padding: 0 0 2px 5px;
}
#homeNewsContainer a {                                   4.css (line 210)
   ~~color: #FFD093;~~
   display: block;
   margin-bottom: 10px;
   ~~padding: 0 0 0 10px;~~
}
a {                                                      4.css (line 37)
   border: medium none;
   text-decoration: none;
}
Inherited from h1.padall
#homeNewsContainer h1 {                                  4.css (line 214)
   font-size: 100%;
}
Inherited from div#homeNewsContainer
#homeNewsContainer {                                     4.css (line 78)
   ~~color: #FFD093;~~
}
Inherited from div#globalContainer
#globalContainer {                                       4.css (line 66)
   ~~color: #000000;~~
}
Inherited from body
body {                                                   4.css (line 26)
   ~~color: #737171;~~
   font-family: "Lucida Sans
   Unicode",Verdana,Arial,Helvetica,sans-serif;
   ~~font-size: 76%;~~
   line-height: 1.3em;
}
```