

CASCADE STYLE SHEET - INTRODUCTION

2.5 Introduction

- The CSS1 specification was developed in 1996
- CSS2 was released in 1998
- CSS3 is on its way
- CSSs provide the means to control and change presentation of HTML documents
- CSS is not technically HTML, but can be embedded in HTML documents
- Style sheets allow you to impose a standard style on a whole document, or even a whole collection of documents
- Style is specified for a tag by the values of its properties

2.6 Levels of Style Sheets

- There are three levels of style sheets
 - Inline - specified for a specific occurrence of a tag and apply only to that tag
 - This is fine-grain style, which defeats the purpose of style sheets - uniform style
 - Document-level style sheets - apply to the whole document in which they appear
 - External style sheets - can be applied to any number of documents
- When more than one style sheet applies to a specific tag in a document, the lowest level style sheet has precedence
 - In a sense, the browser searches for a style property spec, starting with inline, until it finds one (or there isn't one)
- Inline style sheets appear in the tag itself
- Document-level style sheets appear in the head of the document
- External style sheets are in separate files, potentially on any server on the Internet
 - Written as text files with the MIME type text/css

2.7 Linking an External Stylesheet

- A <link> tag is used to specify that the browser is to fetch and use an external style sheet file

```
<link rel = "stylesheet" type = "text/css"  
href = "http://www.wherever.org/termpaper.css">  
</link>
```

- External style sheets can be validated

[http://jigsaw.w3.org/css-validator/
validator-upload.html](http://jigsaw.w3.org/css-validator/validator-upload.html)

2.8 Style Specification Formats

- Format depends on the level of the style sheet
- Inline:
 - Style sheet appears as the value of the style attribute
 - General form:

```
style = "property_1: value_1;  
property_2: value_2;  
...  
property_n: value_n"
```

2.9 Format for Document-level

- Style sheet appears as a list of rules that are the content of a <style> tag
- The <style> tag must include the type attribute, set to "text/css"
- The list of rules must be placed in an HTML comment, because it is not HTML
- Comments in the rule list must have a different form - use C comments (/*...*/)

2.10 General Form, Document Level

- General form:

```
<style type = "text/css">  
<!--  
rule list  
-->  
</style>
```

- Form of the rules:
selector {list of property/values}
 - Each property/value pair has the form:
property: value
 - Pairs are separated by semicolons, just as in the value of a <style> tag

General Form, External style sheets

- Form is a list of style rules, as in the content of a <style> tag for document-level style sheets

Selector Forms: Simple

- The selector is a tag name or a list of tag names, separated by commas
 - h1, h3
 - p
- *Contextual selectors*
 - ol ol li

Class Selectors

- Used to allow different occurrences of the same tag to use different style specifications
- A style class has a name, which is attached to a tag name
 - ❑ p.narrow {property/value list}
 - ❑ p.wide {property/value list}
- The class you want on a particular occurrence of a tag is specified with the class attribute of the tag
- For example,

```
<p class = "narrow">
```

```
...
```

```
</p>
```

```
...
```

```
<p class = "wide">
```

```
...
```

```
</p>
```

Generic Selectors

- A generic class can be defined if you want a style to apply to more than one kind of tag
- A generic class must be named, and the name must begin with a period
 - Example,

```
.really-big { ... }
```

- Use it as if it were a normal style class

```
<h1 class = "really-big"> ... </h1>
```

```
...
```

```
<p class = "really-big"> ... </p>
```

id Selectors

- An id selector allow the application of a style to one specific element
- General form:

```
#specific-id {property-value list}
```

- Example: