COMP30680

Web Application Development

CSS part 1

David Coyle
d.coyle@ucd.ie

Cascading Style Sheets

HTML is primarily concerned with content, rather than style.

CSS provide Web authors with a powerful and flexible way to control the **presentation details** of documents.

Learning Expectations:

- What are CSS?
- Why are they Useful?
- Three Levels of Style Sheets
- Style Specification Formats
- Style Classes
- Properties and Property Values

Next class

What are CSS?

CSS stands for **C**ascading **S**tyle **S**heets

CSS describes **how HTML elements are to be displayed.**

They provide a method of imposing consistency on the style of Web pages

- E.g., allow authors to specify that all occurances of a particular tag use the same presentation style.
- Not a new idea – word processors and desktop publishing systems have used style sheets for a long time.

HTML style sheets are called cascading style sheets because they can be defined at three different levels to specify the style of a document.

Lower-level style sheets can override higher-level style sheets, so the style of the content of a HTML element is determined through a cascade of style sheet applications.

CSS **saves a lot of work**. It can control the layout of multiple web pages all at once.

CSS Solved a Big Problem

HTML was NEVER intended to contain tags for formatting a web page!

HTML was created to **describe the content** of a web page, like:

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

CSS removed the style formatting from the HTML page!

Demo: http://www.w3schools.com/css/css_intro.asp

3 Levels of Style Sheets

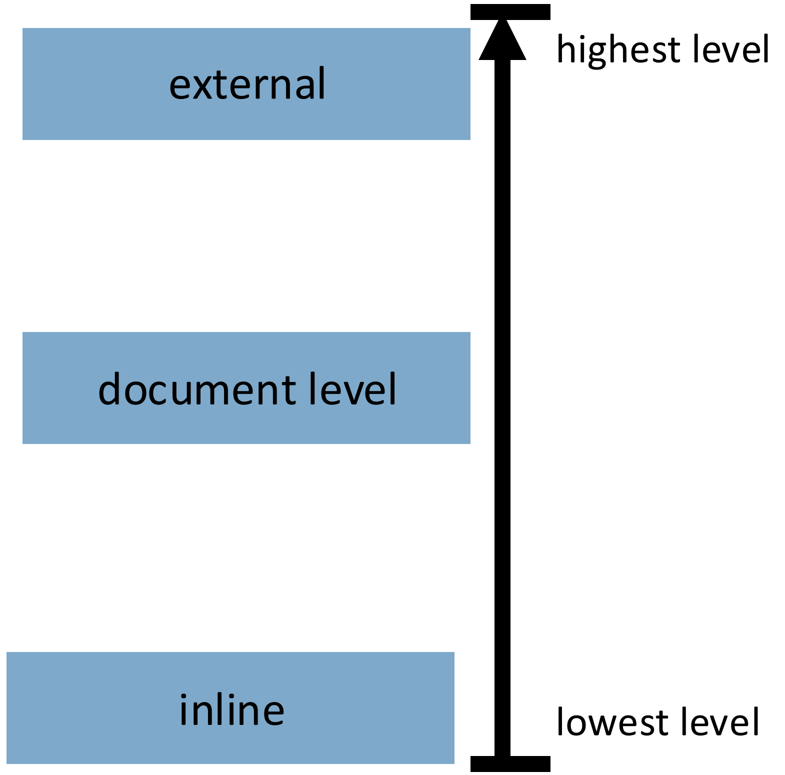
1. **External-level style sheets** can apply to the bodies of any number of documents.

2. **Document-level style sheets** apply to the whole body of a document.

3. **Inline-level style sheets** apply to the content of a single HTML element.

The format of a style specification depends on its level.

External and document level syntax is broadly similar.



NOTE:
The properties of an HTML element are those that result from a merge of all applicable style sheets, with lower-level style sheets having precedance in cases of conflicting instructions.

External style sheets

External Level

- External style sheets are not part of the documents to which they apply. They are stored separately and are specified by the documents that use them.
- The browser fetches external style sheets just as it fetches other Web pages. The **link** element is used to specify external style sheets. This appears in the head of the document.

```
<head>
  <link rel  = stylesheet type = “text/css”
  href = “mystyle.css” >
  </link>
</head>
```

Here the browser will read the style definitions from the file **mystyle.css**, and format the document according to it. An external style sheet can be written in any text editor and saved with a **.css** extension. The file should not contain any html tags/elements.

With an external style sheet, you can change the look of an entire Web site by changing just one file!

Internal / Document-Level style sheet

Document style specifications appear as the content of a <style>…</style> element within the **header** of a document.

<style type = “text/css”>

<!–- include list of rules here -->

</style>

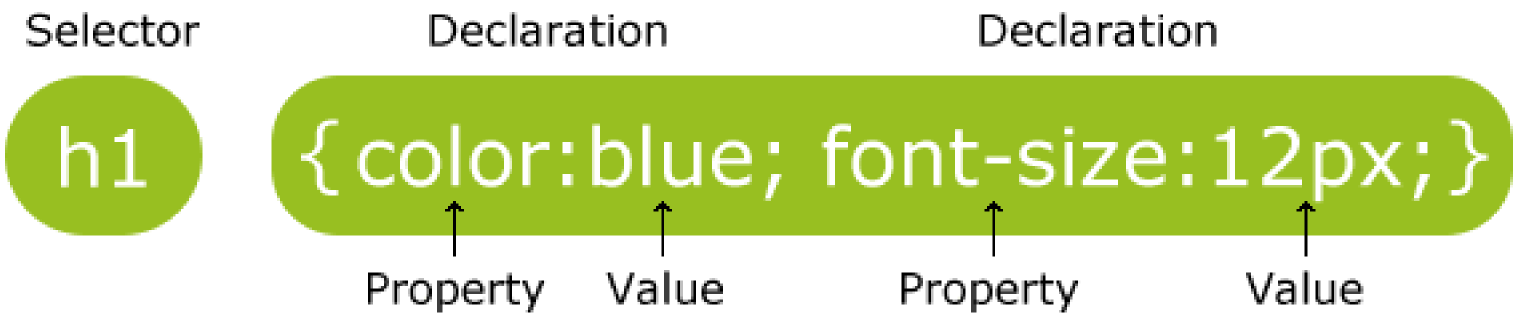
```
<head>
<style type = “text/css”>
    h1,h2,h3,h4,h5,h6{font-family: "sans serif“}
    h1{font-weight: bold; font-size: 14pt}
    h2{font-weight: lighter; font-size: 12pt}
    h1, h2{color: red}
</style>
</head>
```

The browser will read the style definitions, and format the current Web document according to it.

A document level or internal style sheet should be used when a single document requires a unique style.

CSS Syntax

A CSS rule-set consists of a selector and a declaration block:



The selector points to the HTML element you want to style.

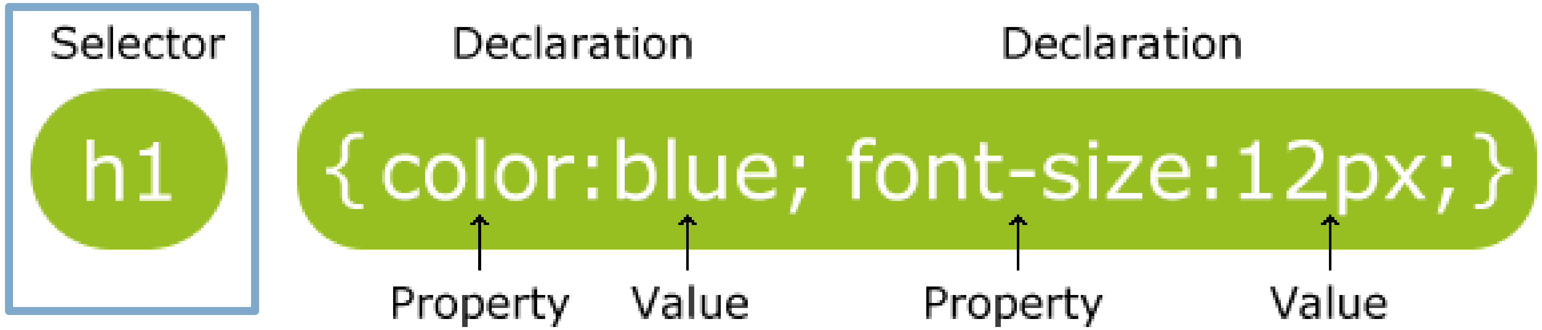
The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

Note The selector is normally the HTML element/tag you wish to define, the property is the attribute you wish to change. If the value is multiple words, put single-quotes around the value, e.g. `‘sans serif’`

CSS Selectors

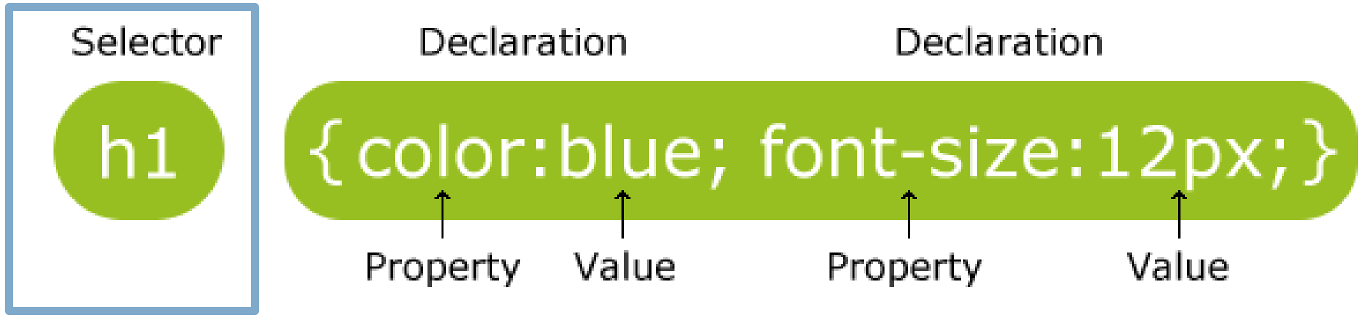


CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

Three approaches to creating selectors:

- element selector
- id selector
- class selector

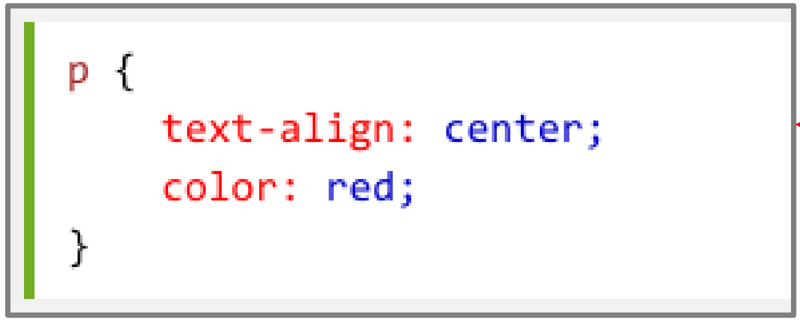
Element Selectors



The element selector selects elements based on the element/tag name.

tag_name{ property1:v1; property2:v2; …; propertyN:vN}

You can select all <p> elements on a page like this (in this case, all <p> elements will be center-aligned, with a red text color):

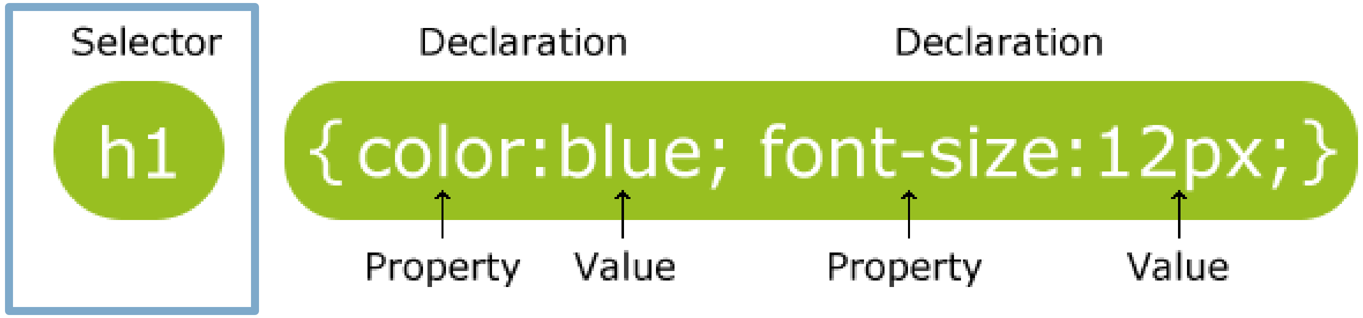


```
p {
    text-align: center;
    color: red;
}
```

**This style will be applied to all <p> tags,
unless the tag is overwritten
by another style at a lower level in the cascade.**

See element_selector.html

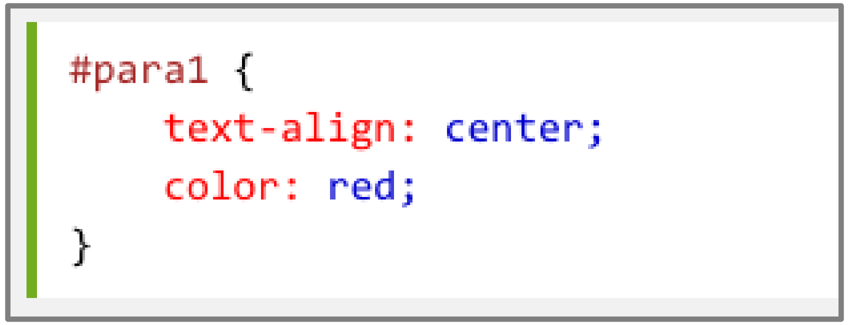
ID Selectors



The id selector allows you to apply styles more selectively.

It uses the **id attribute** of an HTML element to select a specific element.

To create a style with a specific id, write a hash (#) character, followed by the id.



```
#para1 {
    text-align: center;
    color: red;
}
```

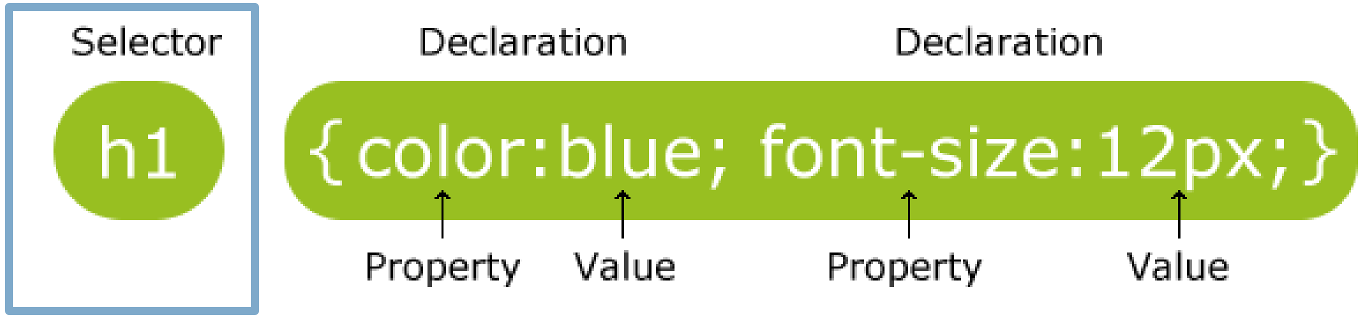
This style rule will be applied to the HTML element with id="para1”, e.g.

<p id="para1">Hello World!</p>

It is recommended that the id of an element should be unique within a page, so the id selector is used to select one unique element!

See id_selector.html

Class Selectors



Selector

h1

Declaration Declaration

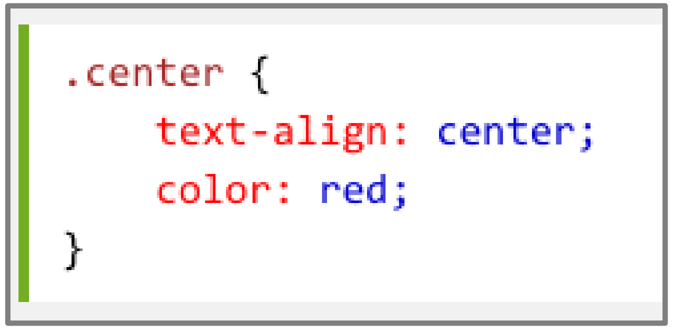
{ color:blue; font-size:12px; }

Property Value Property Value

The class selector applies styles to elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the name of the class.

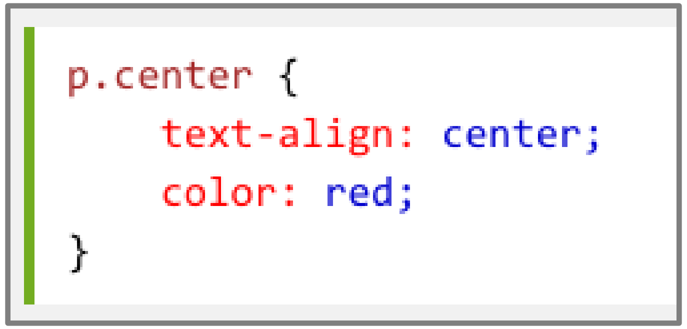
In the example below, all HTML elements with class="center" will be red and center-aligned:



```
.center {
    text-align: center;
    color: red;
}
```

This is called a generic class.

You can also specify that only specific HTML elements should be affected by a class. In the example below, only <p> elements with class="center" will be center-aligned:



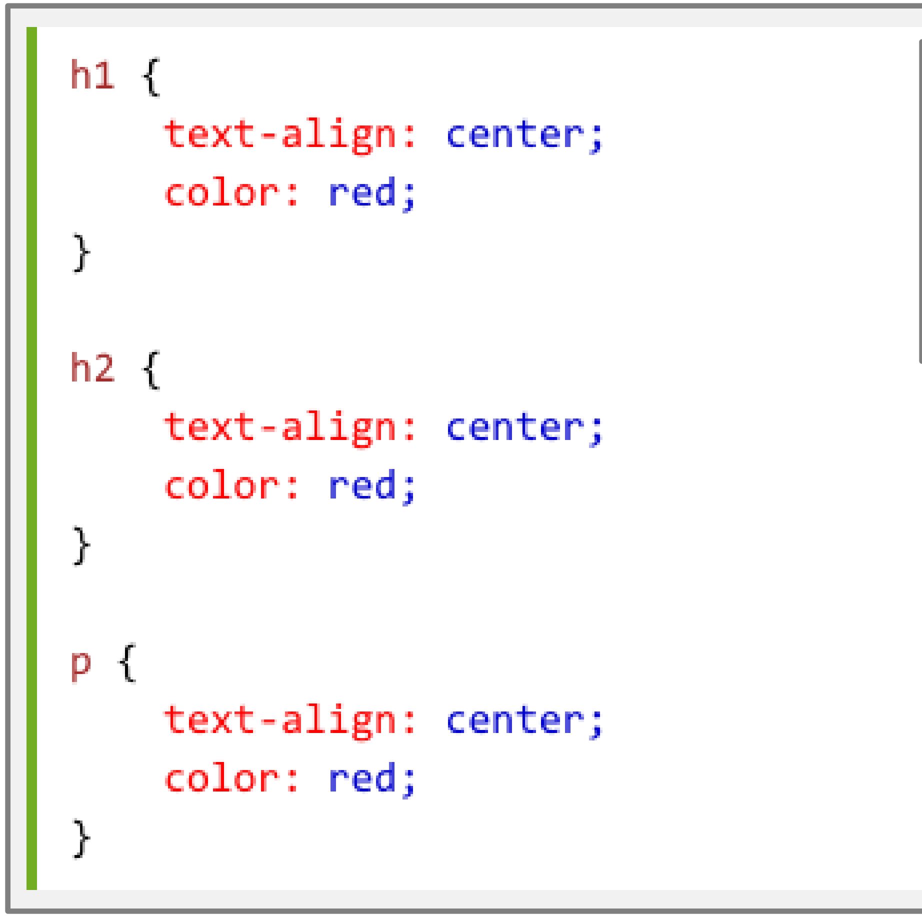
```
p.center {
    text-align: center;
    color: red;
}
```

See class_selector.html

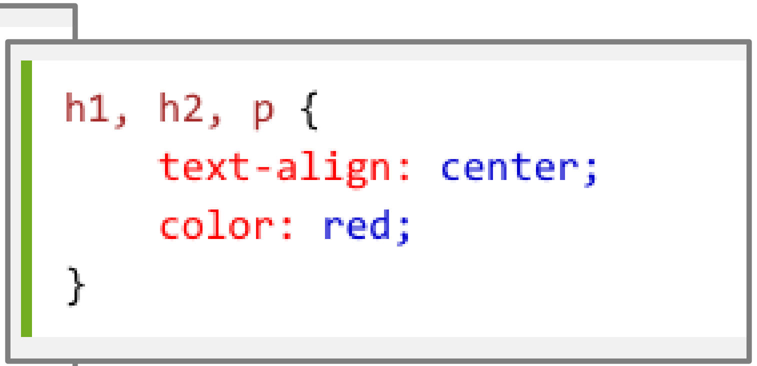
Grouping Selectors

If you have elements with the same style definitions, it is often better to group the selectors, to minimize the code.

To group selectors, separate each selector with a comma.



```
h1 {
    text-align: center;
    color: red;
}

h2 {
    text-align: center;
    color: red;
}

p {
    text-align: center;
    color: red;
}
```



```
h1, h2, p {
    text-align: center;
    color: red;
}
```

Another example:
p.first-line,h1 { font-style: italic }

Here the first line of each paragraph will appear in italics, as will the h1 heading.

Inline level styles

Inline styles are created by providing values for the style attribute of a HTML element:

style = “property1: v1; property2: v2; …; propertyN: vN”

To use inline styles you use the style attribute in the relevant tag. The example shows how to change the colour and the left margin of a paragraph:

```
<p style="color: red; margin-left: 20px">
     This is a paragraph
</p>
```

Inline styles should be used sparingly. They lose many of the advantages of style sheets by mixing content with presentation. Use this method sparingly, such as when a style is to be **applied to a single occurrence of an element**.

Cascading Order

What style will be used when there is more than one style specified for an HTML element?

Generally speaking we can say that all the styles will "cascade" by the following rules, where number one (i.e. the most specific level) has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

So, an inline style (inside a specific HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or a browser default value.

Cascading Order

What style will be used when there is more than one style specified for an HTML element?

Generally speaking we can say that all the styles will "cascade" by the following rules, where number one (i.e. the most specific level) has the highest priority:

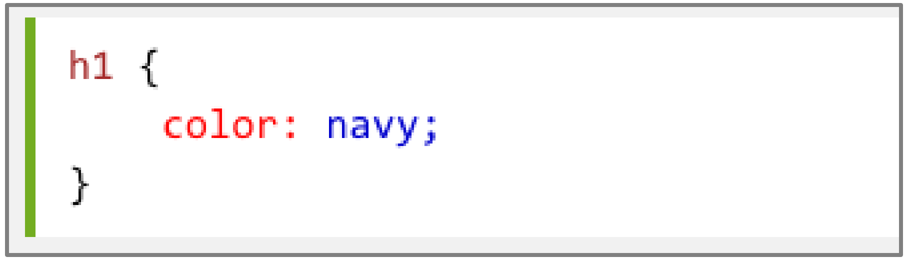
1. Inline style (inside an HTML element)
2. **External and internal style sheets (in the head section)**
3. Browser default

So, an inline style (inside a specific HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or a browser default value.

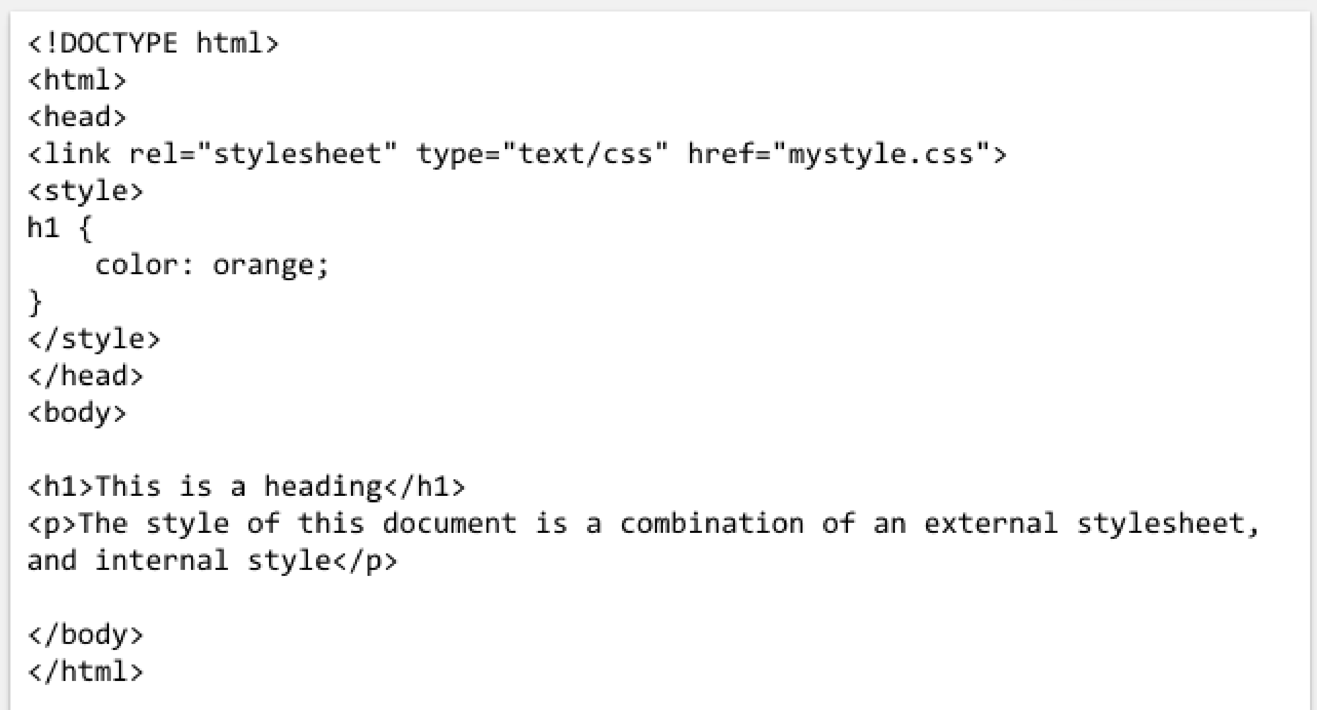
Multiple Style Sheets

If some properties have been defined for the same selector (element) in different style sheets (e.g. internal and external), the value from the last read style sheet will be used.

Assume that an external style sheet mystyle.css has the following style for the <h1> element



```
h1 {
     color: navy;
}
```



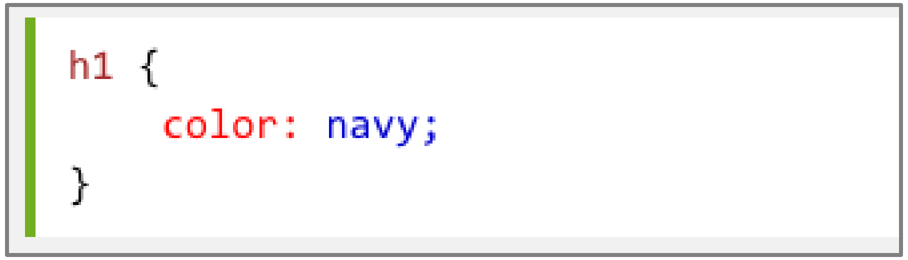
```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
    color: orange;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>The style of this document is a combination of an external stylesheet,
and internal style</p>

</body>
</html>
```

What colour is the h1 element here?

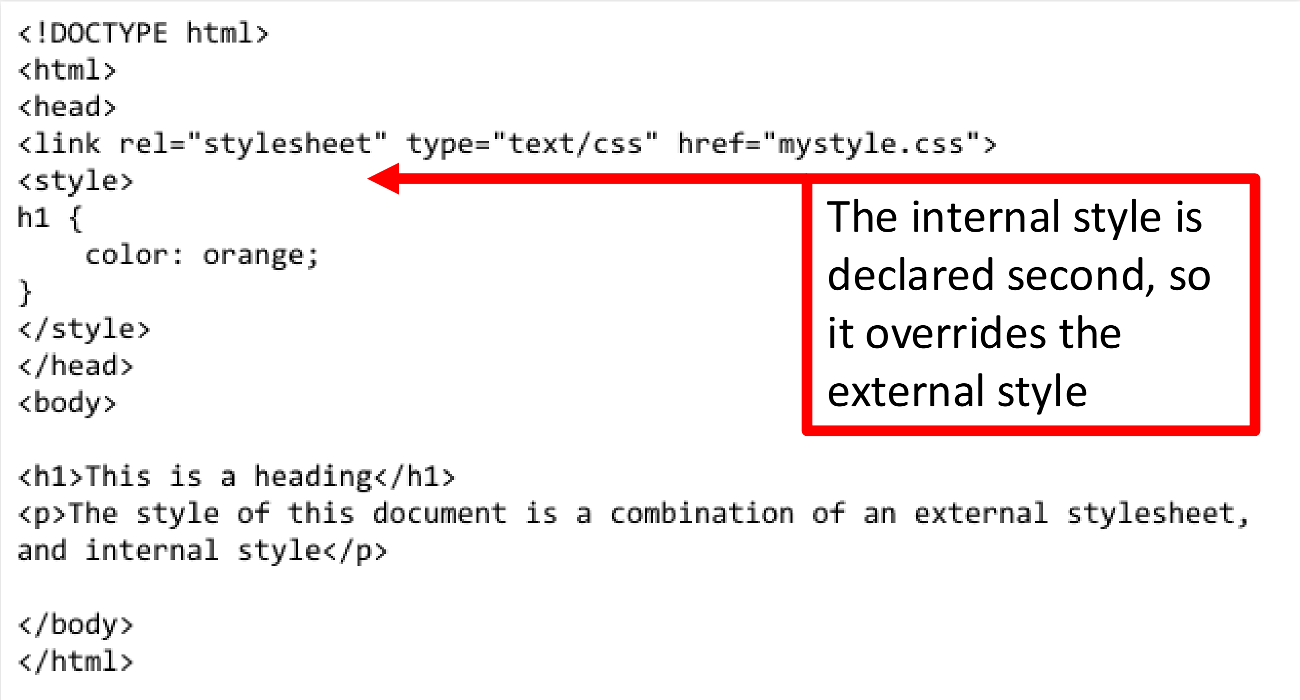
Multiple Style Sheets

If some properties have been defined for the same selector (element) in different style sheets (e.g. internal and external), the value from the last read style sheet will be used.

Assume that an external style sheet mystyle.css has the following style for the <h1> element

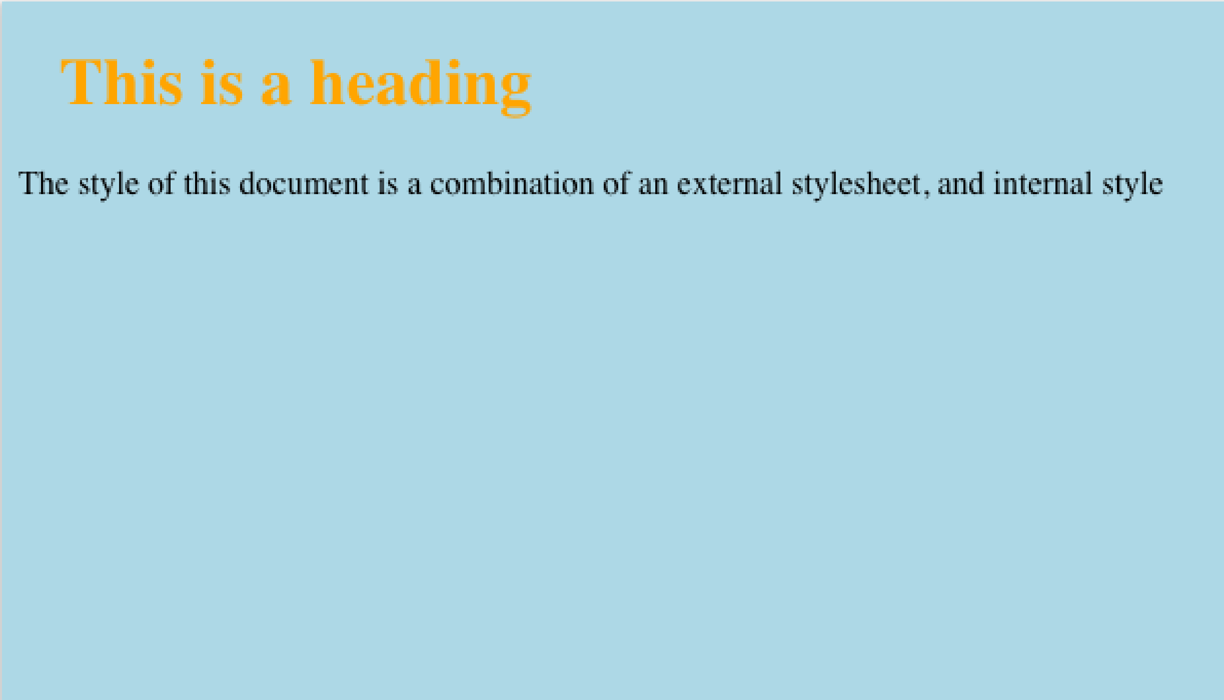


```
h1 {
      color: navy;
}
```



```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
    color: orange;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>The style of this document is a combination of an external stylesheet,
and internal style</p>

</body>
</html>
```

The internal style is declared second, so it overrides the external style



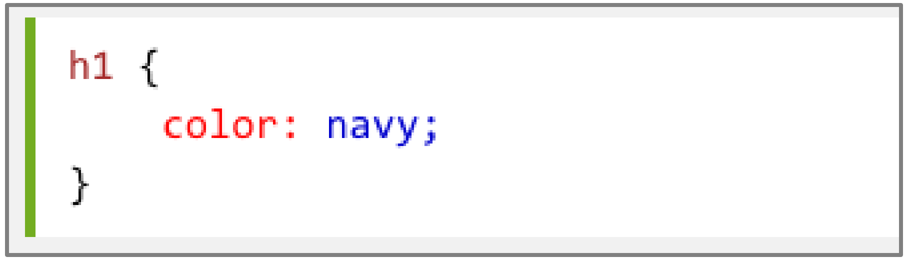
This is a heading

The style of this document is a combination of an external stylesheet, and internal style

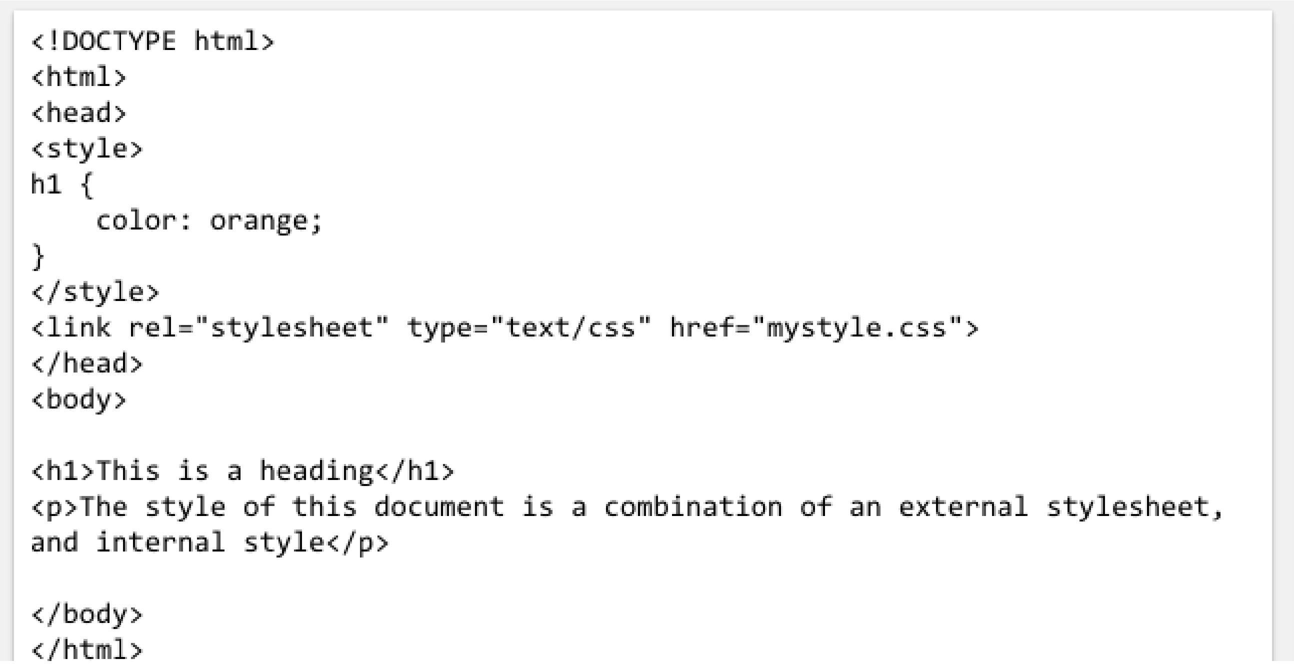
Multiple Style Sheets

If some properties have been defined for the same selector (element) in different style sheets (e.g. internal and external), the value from the last read style sheet will be used.

Assume that an external style sheet mystyle.css has the following style for the <h1> element



```css
h1 {
      color: navy;
}
```



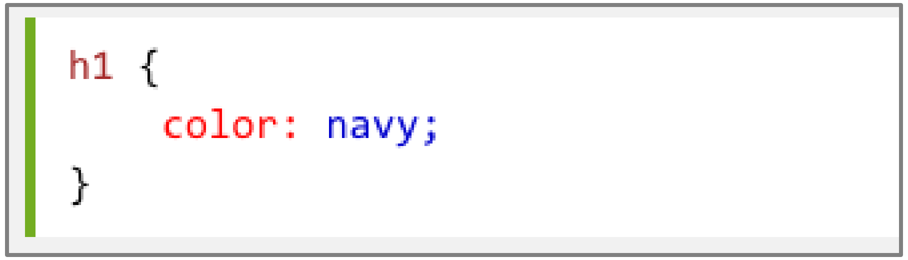
```html
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
    color: orange;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>The style of this document is a combination of an external stylesheet,
and internal style</p>

</body>
</html>
```

And here. What colour is the H1 element?

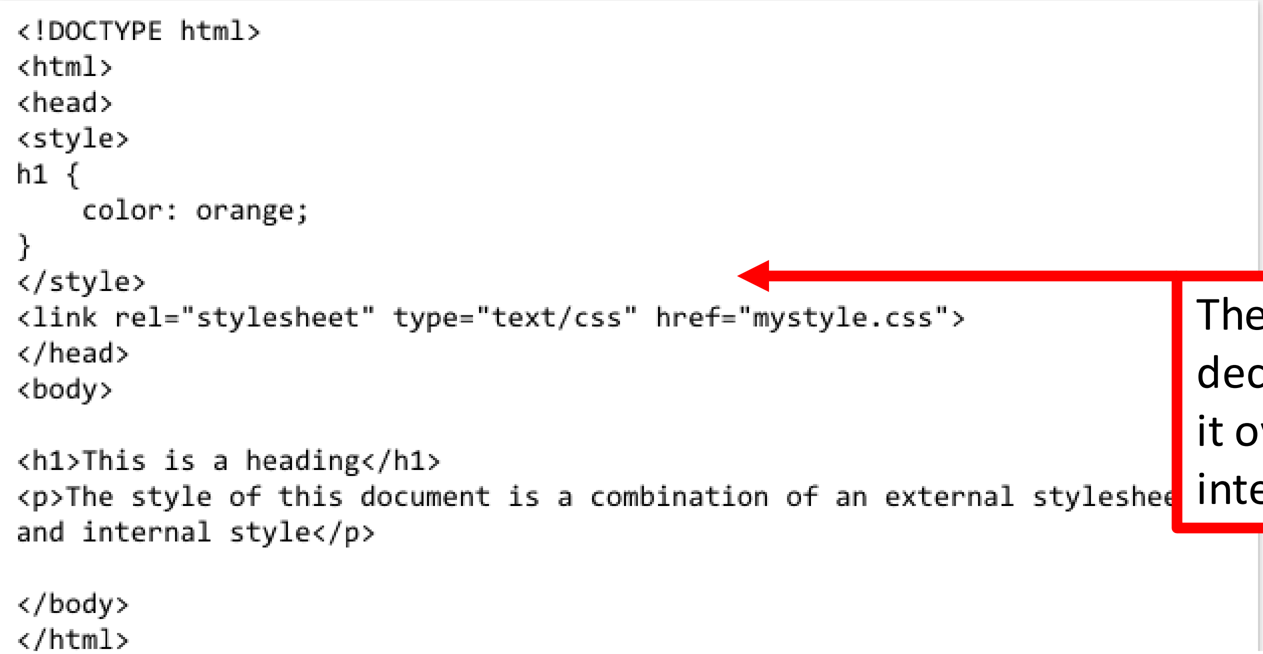
Multiple Style Sheets

If some properties have been defined for the same selector (element) in different style sheets (e.g. internal and external), the value from the last read style sheet will be used.

Assume that an external style sheet mystyle.css has the following style for the <h1> element

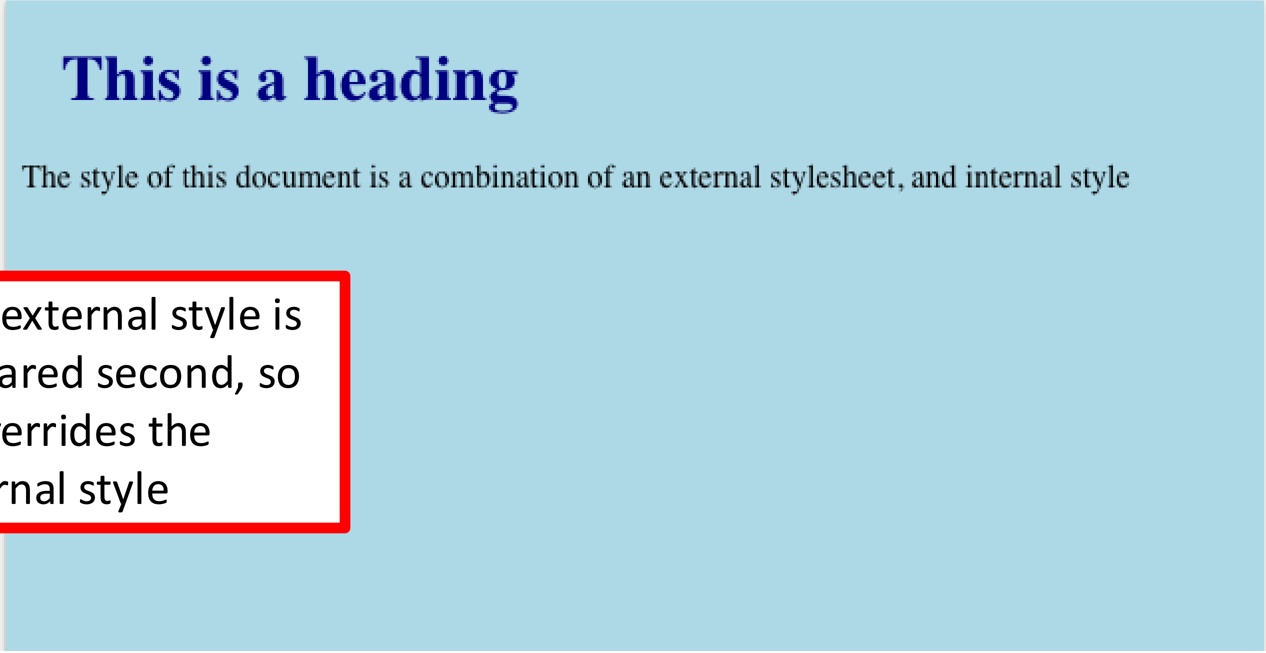


```
h1 {
      color: navy;
}
```



```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
    color: orange;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>The style of this document is a combination of an external styleshee
and internal style</p>

</body>
</html>
```

The external style is declared second, so it overrides the internal style

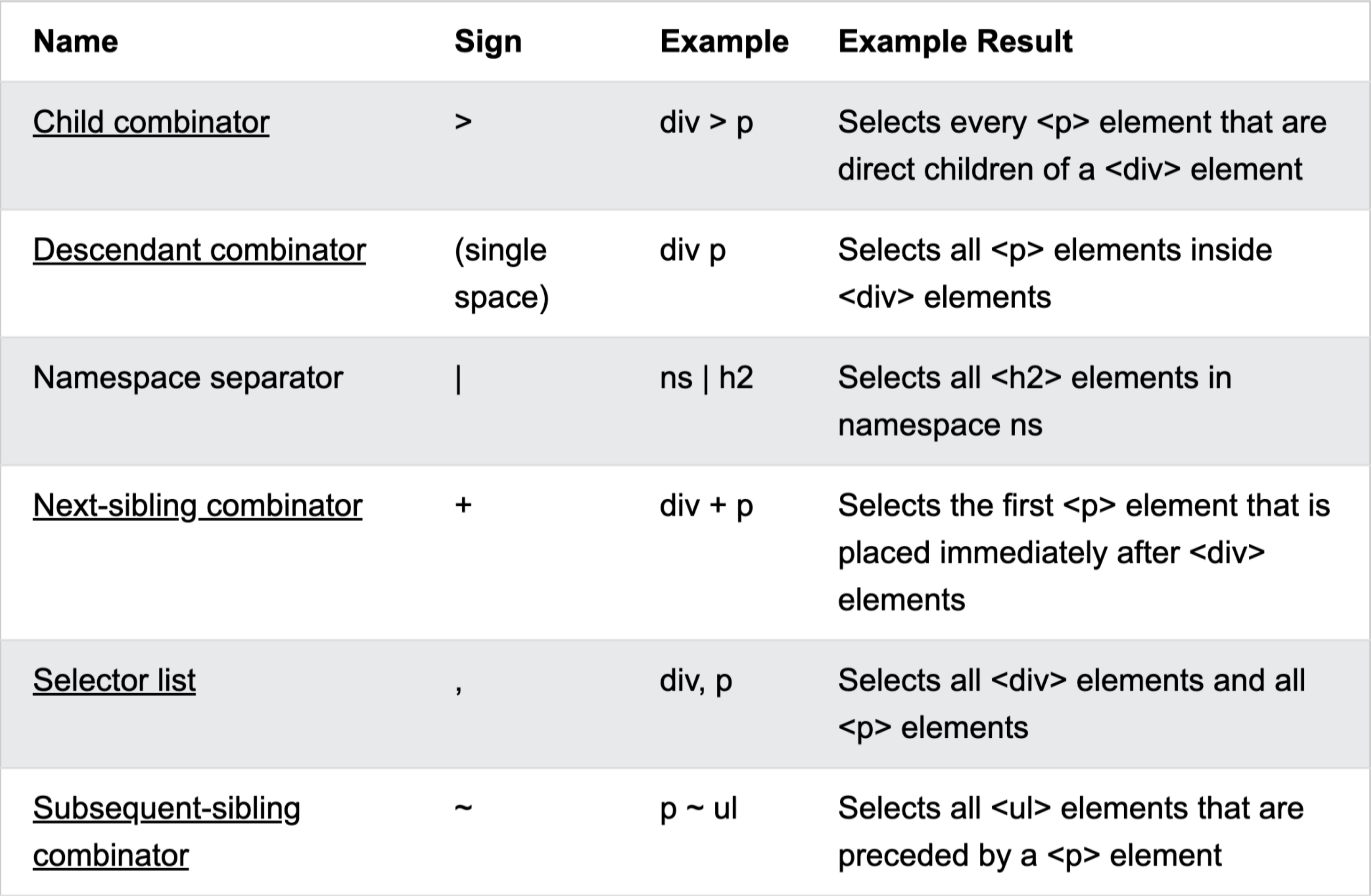


This is a heading

The style of this document is a combination of an external stylesheet, and internal style

CSS Combinators

Allows you to create more specific selector rules based on the relationships between elements



Name	Sign	Example	Example Result
Child combinator	>	div > p	Selects every <p> element that are direct children of a <div> element
Descendant combinator	(single space)	div p	Selects all <p> elements inside <div> elements
Namespace separator	|	ns | h2	Selects all <h2> elements in namespace ns
Next-sibling combinator	+	div + p	Selects the first <p> element that is placed immediately after <div> elements
Selector list	,	div, p	Selects all <div> elements and all <p> elements
Subsequent-sibling combinator	~	p ~ ul	Selects all <ul> elements that are preceded by a <p> element

http://www.w3schools.com/cssref/css_ref_combinators.php

CSS Pseudo-classes

These are built-in classes in the HTML and CSS specifications. They allow you to apply styles based on the status of elements and interactions with the webpage.

E.g. change the style of a link. See “pseudo_hover.html”

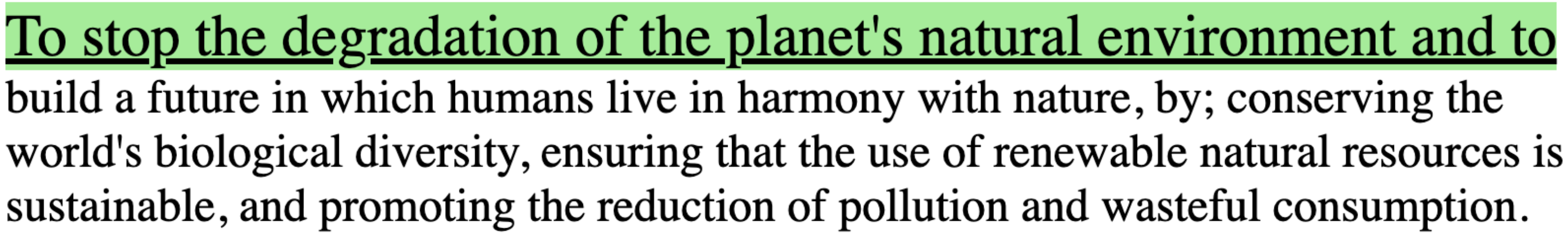
And the full reference here:

http://www.w3schools.com/cssref/css_ref_pseudo_classes.php

CSS Pseudo-elements

A CSS pseudo-element is used to style specific parts of an element.

```
p::first-line {
  font-size: 20px;
  text-decoration: underline;
  background-color: lightgreen;
}
```



To stop the degradation of the planet's natural environment and to build a future in which humans live in harmony with nature, by; conserving the world's biological diversity, ensuring that the use of renewable natural resources is sustainable, and promoting the reduction of pollution and wasteful consumption.

See pseudo_element.html

And the full reference here:

http://www.w3schools.com/cssref/css_ref_pseudo_elements.php

CSS Selector Reference

For more details on CSS selector patterns see:

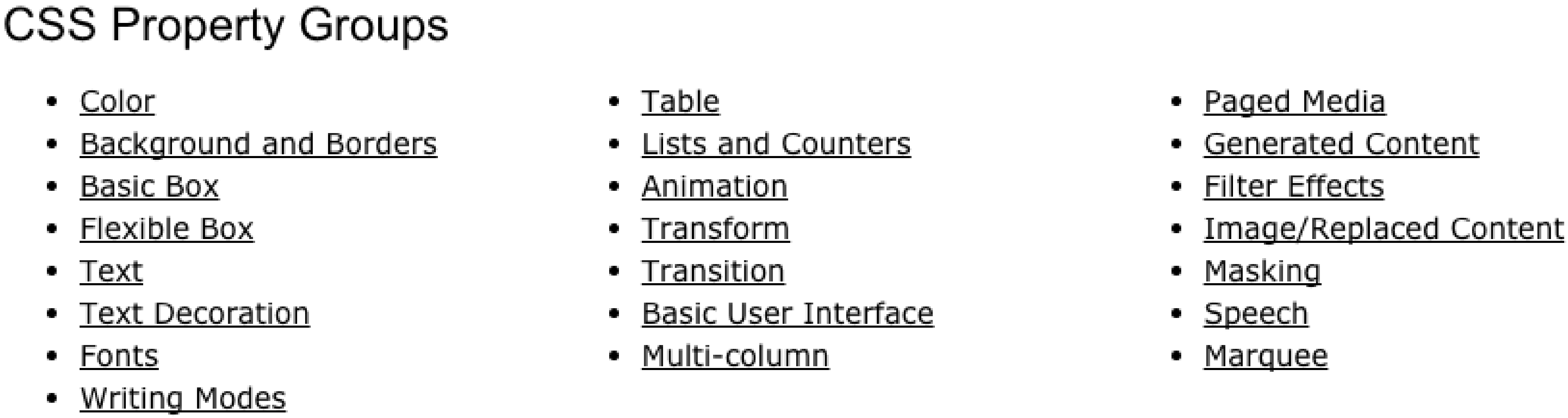
http://www.w3schools.com/cssref/css_selectors.asp

Also try out this CSS selector tester:

http://www.w3schools.com/cssref/trysel.asp

CSS Properties

CSS provides a large number of properties for you to use.



CSS Property Groups

- Color
- Background and Borders
- Basic Box
- Flexible Box
- Text
- Text Decoration
- Fonts
- Writing Modes
- Table
- Lists and Counters
- Animation
- Transform
- Transition
- Basic User Interface
- Multi-column
- Paged Media
- Generated Content
- Filter Effects
- Image/Replaced Content
- Masking
- Speech
- Marquee

Questions, Suggestions?

Next class:

CSS part 2: Properties.