NTNU – Trondheim
Norwegian University of
Science and Technology

Cascading Style Sheets: Selectors and Properties

Michalis Giannakos
Norwegian University of Science & Technology (NTNU)
Department of Computer and Information Science

Cascading Style Sheets

- CSS1 – official W3C, Dec 1996
- Introduced styles for:
  - Fonts (eg typeface, emphasis)
  - Text (eg spacing)
  - Color (eg of text, backgrounds)
  - Alignment (eg of text, images, tables)
  - Block-level Elements

Cascading Style Sheets

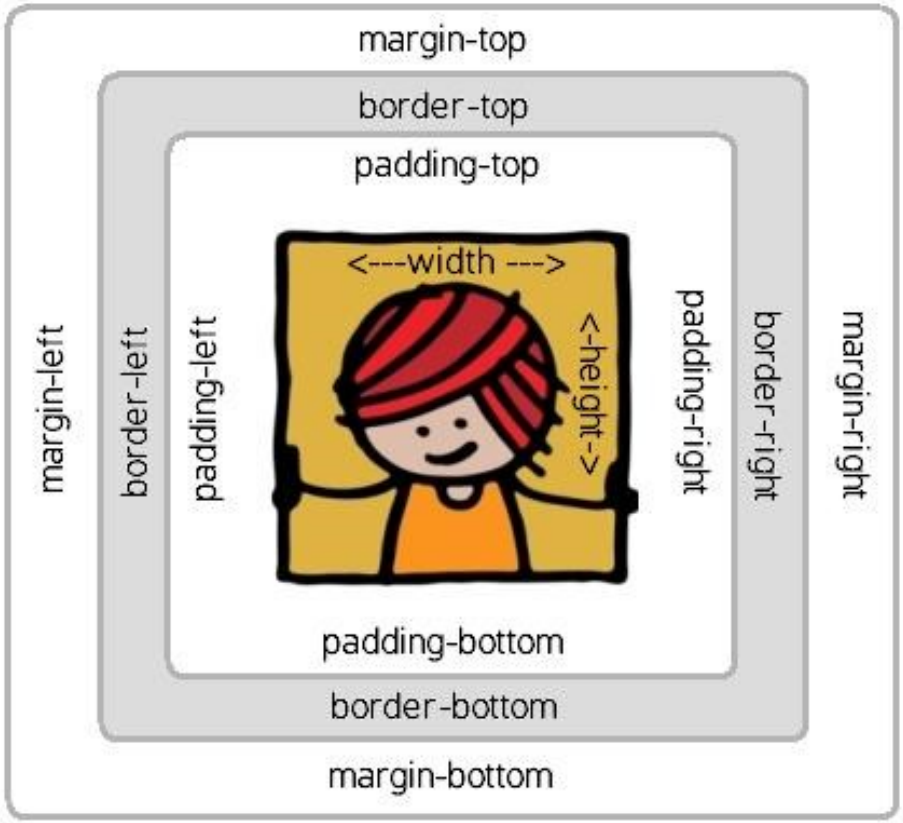
- CSS2 – official W3C, Dec 1998
- Introduced styles for:
  - Positioning (abs, rel, z-index)
  - Visual Formatting (Box Model)
  - Media Types (screen, paper, speech, braille)

Cascading Style Sheets

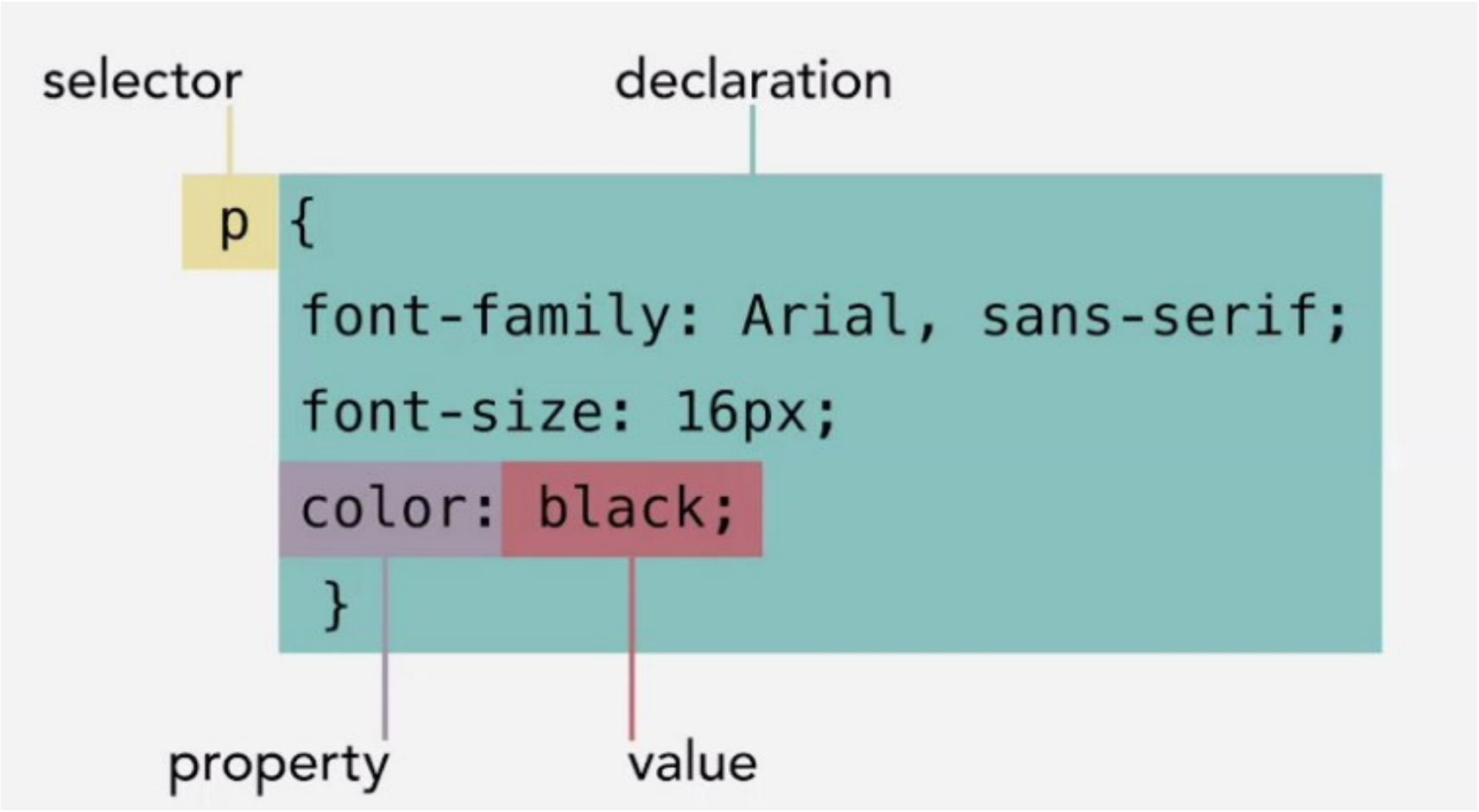
- CSS3 (which is still under development from W3C) introduces styles for the following document features:
  - User Interfaces
  - Accessibility
  - Columnar layout
  - Rounded borders,
  - Box and text shadows,
  - Gradient backgrounds
  - Scalable Vector Graphics

How CSS sees an HTML document?

- CSS treats HTML element as it appears inside its own box and uses rules to indicate how that element should look



CSS Syntax


selector
declaration
p {
font-family: Arial, sans-serif;
font-size: 16px;
color: black;
}
property
value

LEARNING CSS

- **Focus on selectors:** These allow you to efficiently target elements
- **Properties:** Learn which properties can be used for specific elements, and which values they'll accept
- **Implementing CSS:** Where you can add it to your HTML, how to externalize CSS into a single external file

Resolving Rule Conflict

- Use !important to override.
  - **p {color: blue !important}**
- All user rules and author rules have more weight than the default supplied by the web browser.
- **If two rules of the same origin conflict, the more specific rule applies.**
- **If the rules are equally specific, the last-specified rule is chosen.**

http://www.idi.ntnu.no/~michailg/IT2805/examples/lecture4/6.html

Later styles over-rule earlier styles

- If you define a style property, and later define an alternative style property for the same thing, the later definition over-rules the earlier one.

```
<style type=”text/css”>
        body {
                background-color:yellow;
                font-weight:bold;}
        div {
                background-color:#afa;
                font-weight:normal;}
</style>
```

```
<p>Some text here, inherits properties of the body.</p>
<div>

<p>However, the div’s rules over-ride the body’s rules, as the div’s rules apply later (i.e. nearer to this text in the document).</p>

</div>
```

Style specificity (prioritization) for one document

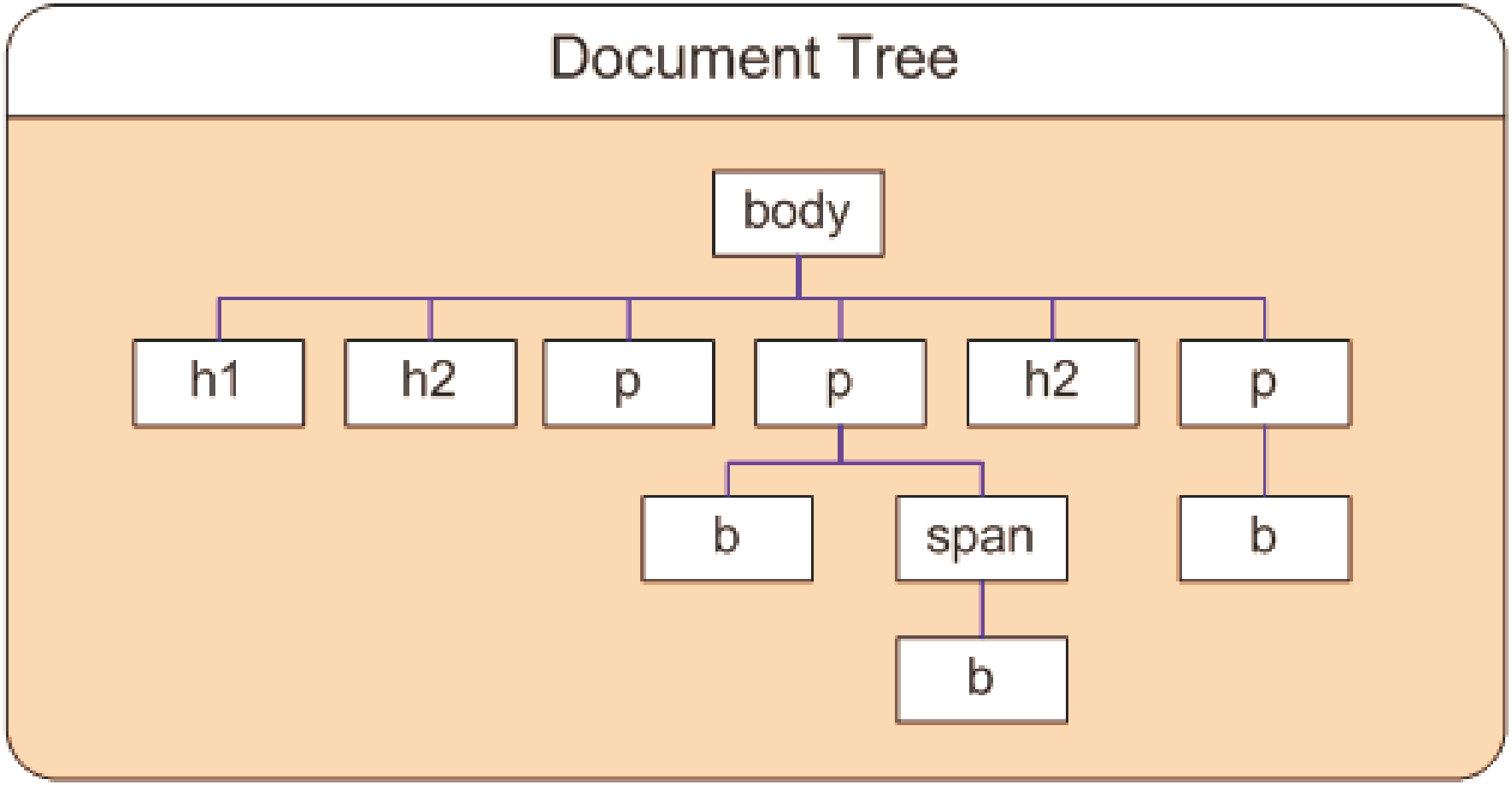
- Three ways to apply a style to an HTML document:
  - **Inline Styles**
  - **Embedded Styles**
  - **External Styles**
- Style prioritization
  - **Inline > embedded > external**

  The more local a style is defined, the higher priority has

Build the Document/HTML tree

```
<html>
    <head>
        <style type="text/css">
            </style>
    </head>
    <body>
        <h1> Heading 1</h1>
        <h2> Heading 2.a</h2>
        <p> First paragraph. </p>
        <p> Second paragraph has a <b>bold</b>
            and a <span>span with another <b>bold</b></span>. </p>
        <h2> Heading 2.b</h2>
        <p> Third paragraph has a <b>bold</b> also. </p>
    </body
</html>
```

the Document/HTML tree



Multiple declarations

- Multiple declarations for the same selector may be organized into semicolon separated groups. For example,

```
h1 {font-weight: bold}
h1 {font-size: 12pt}
h1 {font-family: Helvetica}
h1 {font-style: normal}
```

is equivalent to:

```
h1 {
    font-weight: bold;
    font-size:12pt;
    font-family: Helvetica;
    font-style: normal
 }
```

Grouping

- When several selectors share the same declarations, they may be grouped as a comma-separated list.

```
h1 {font-family: sans-serif}
h2 {font-family: sans-serif}
h3 {font-family: sans-serif}
```

is equivalent to:

```
h1,h2,h3 {font-family: sans-serif}
```

Selecting Elements

- There are numerous ways of specifying to which elements style rules apply. Here are examples of some of the more commonly used selectors:

p {color:red}	Every p element
h1,h2,h3 {…}	Group selector
strong em {…}	Contextual selector
div[secret="yes"] {…}	Attribute selector
span.important {…}	Class selector
p#1234 {…}	ID selector

Universal Selector

This selector consists of the asterisk character, like this:

```
* {
    background-color: red;
}
```

When used on its own like above, this selects every element on the page.

Element Type Selector

Also called just the “type selector”, this selector matches HTML elements by tag name. Two examples:

```
h2 {
    background-color: red;
}

div {
    background-color: red;
}
```

Class Selector

Selects an element that matches a class name defined in a class attribute in the HTML.

```
.element {
    background-color: red;
}
```

This is easily my favorite selector, and all good CSS developers should use it abundantly. You can put multiple classes separated by spaces on a single class attribute, which makes this selector quite powerful.

ID Selector

This selects an element that matches an id defined in an id attribute in the HTML.

```
#elementID{
    background-color: red;
}
```

Descendant Selector

This selector is defined with a space character separating two selectors, and represents a child element, but not just immediate children, further nested ones as well.

```
p a {
    background-color: red;
}
```

This example targets **any <a> elements that sit inside a <p>** element, even if there are no elements nested between them

Attribute Selector

This selector **targets an element based on an HTML attribute and/or attribute value**. Both examples below are valid attribute selectors:

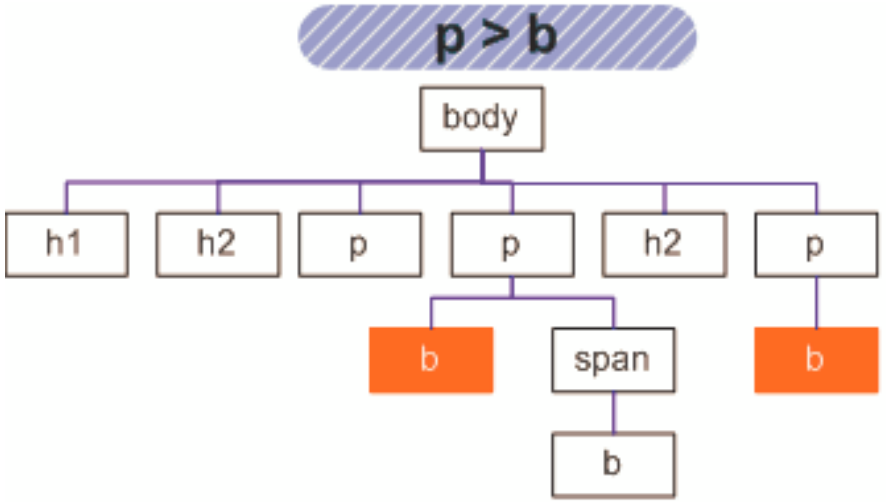
```
div[style] {
    background-color: red;
}

input[type="text"] {
    background-color: red;
}
```

The first example **targets any element that has a "style" attribute**. The second example **targets any <input> element that has a "type" attribute with a value of "text"**.

Child Selector

This selects **an element based on it being an immediate child of another element**:

```
one > two {
    background-color: red;
}
```

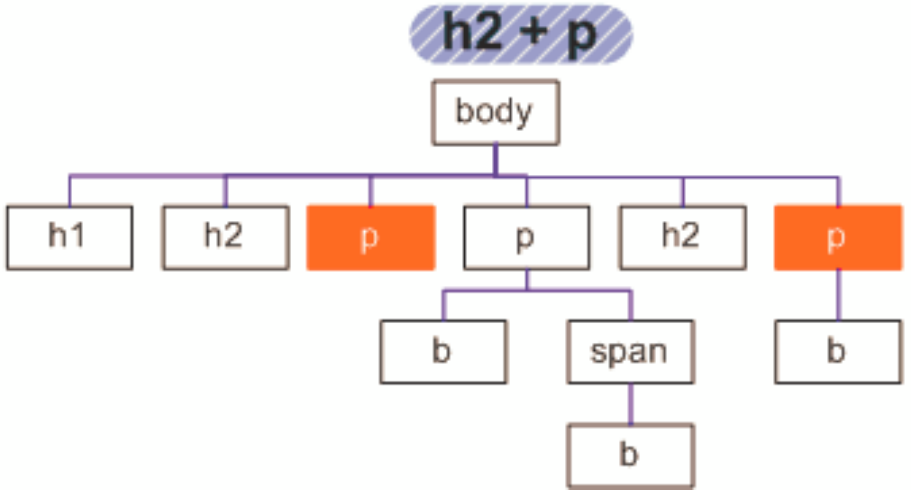


So this **will not style a "two" element unless it is an immediate child of a "one" element**. It can't be nested inside another element, it has to be an immediate child element. And only the child is styled, not the parent.

Adjacent Sibling Selector

This selector, which uses the plus sign, **targets elements that are "adjacent" to each other**, or immediate siblings, and they must have the same parent element.

```
h2+p {
    background-color: red;
}
```

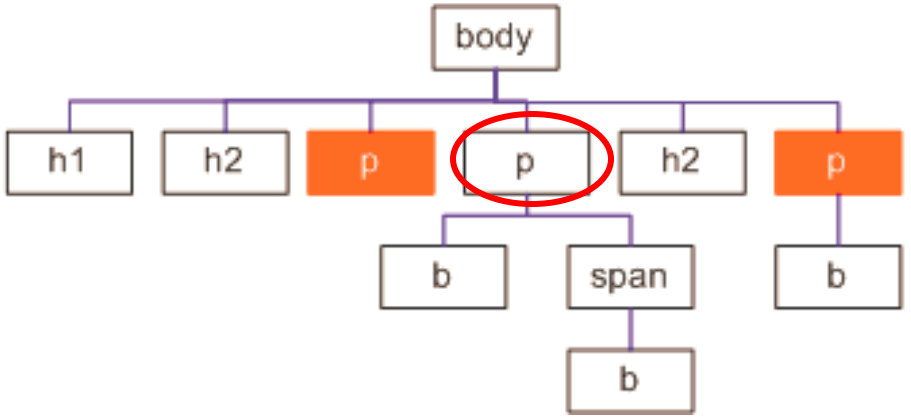


Targets **the first <p> element after any <h2> element**, but not other <p> elements

General Sibling Selector

This uses the **tilde character and is exactly the same as the adjacent sibling selector except the elements don't have to be immediate siblings**.

```
h2~p {
    background-color: red;
}
```



If you had two <p> elements that are sibling of an <h2> element, this rule would apply to both.

Pseudo-class

While technically falling under the category of “selectors”, these are not normally referred to as “selectors”, but just pseudo-classes. Pseudo-class **selects an element based on a state the element is in**. Here are a few examples:

```
a:visited {
    background-color: red;
}

a:hover {
    background-color: red;
}
```

Pseudo-element

CSS pseudo-elements are used to add special effects to some selectors.

Again, not normally referred to as a selector, these actually represent elements in the HTML page that are not really part of the rendered HTML:

```
p::first-letter {
    background-color: red;
}

p::before {
    content: "Read this -";
    background-color: yellow;
}
```

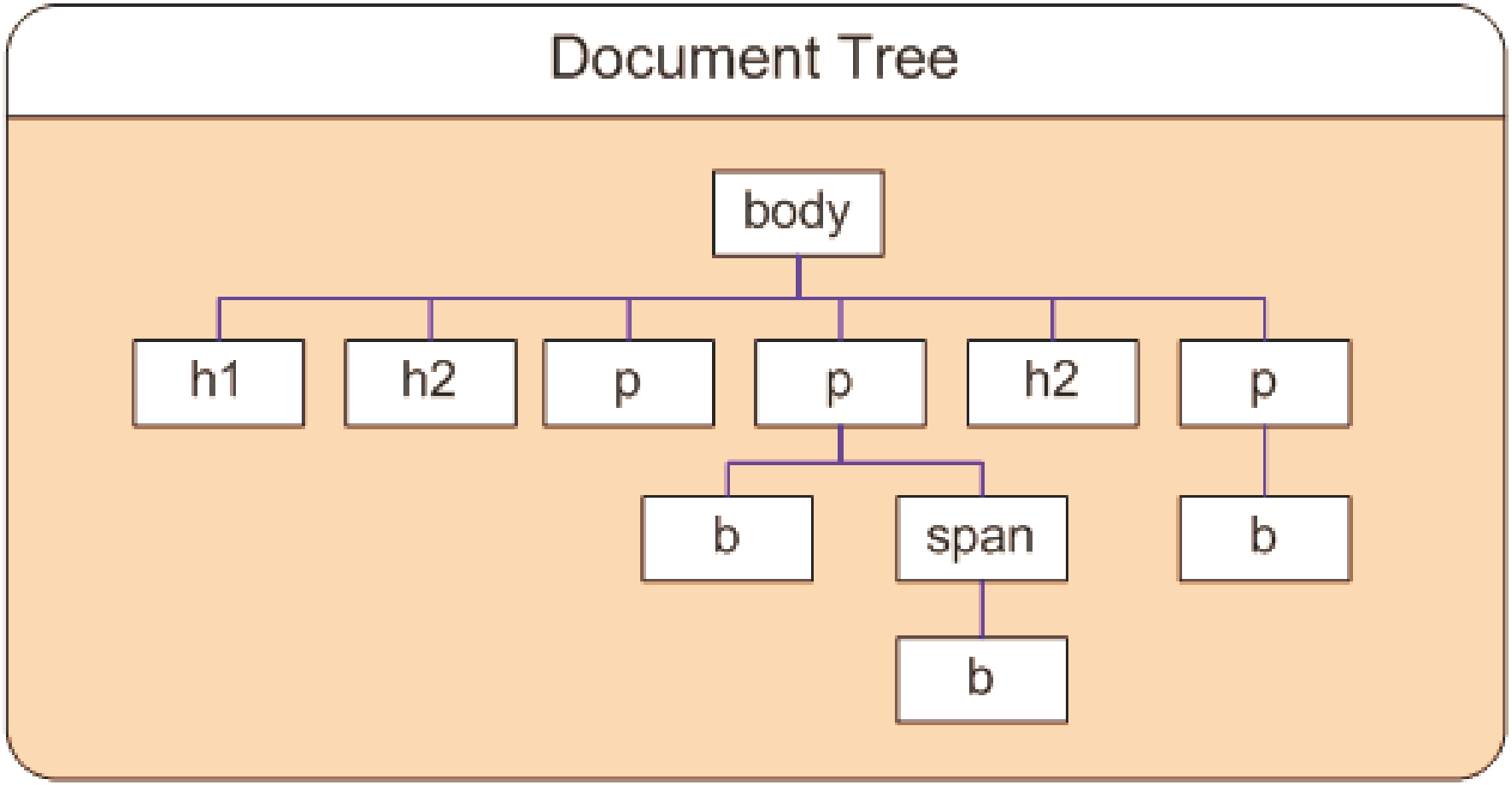
Why sometimes double colons (::)?

- Sometimes you will see double colons (::) instead of just one (:).
- This is part of CSS3 and an attempt to distinguish between pseudo-classes and pseudo-elements. Most browsers support both values.
- Pseudo-classes act as simple selectors in a sequence of selectors and thereby classify elements on non-presentational characteristics, **pseudo-elements create new virtual elements**.

All CSS Pseudo Classes/Elements

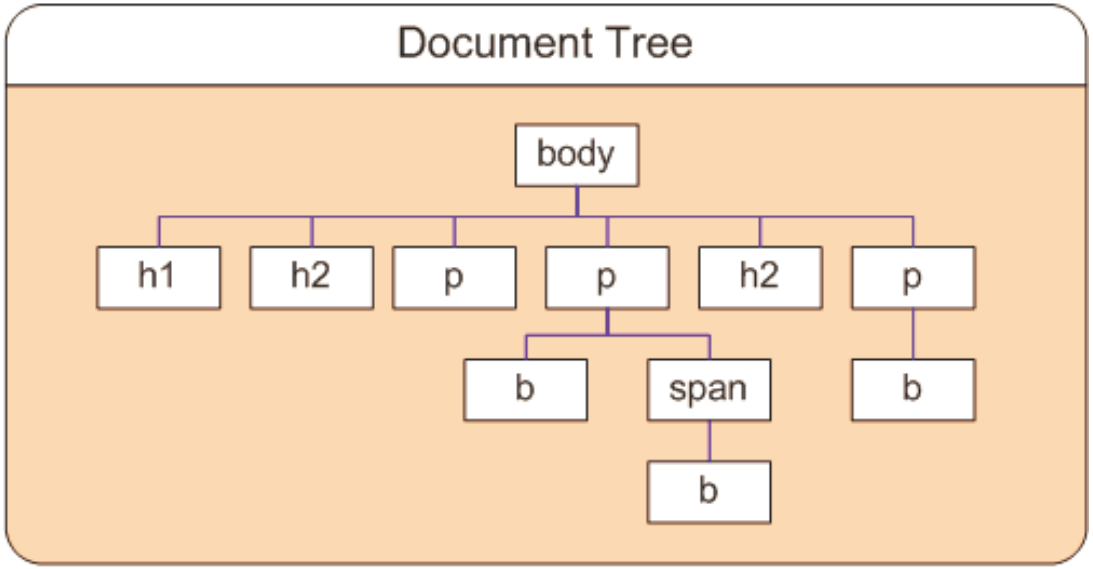
Selector	Example	Example description
:link	a:link	Selects all unvisited links
:visited	a:visited	Selects all visited links
:active	a:active	Selects the active link
:hover	a:hover	Selects links on mouse over
:focus	input:focus	Selects the input element which has focus
::first-letter	p::first-letter	Selects the first letter of every <p> element
::first-line	p::first-line	Selects the first line of every <p> element
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent
::before	p::before	Insert content before every <p> element
::after	p::after	Insert content after every <p> element
:lang(*language*)	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"

the Document/HTML tree

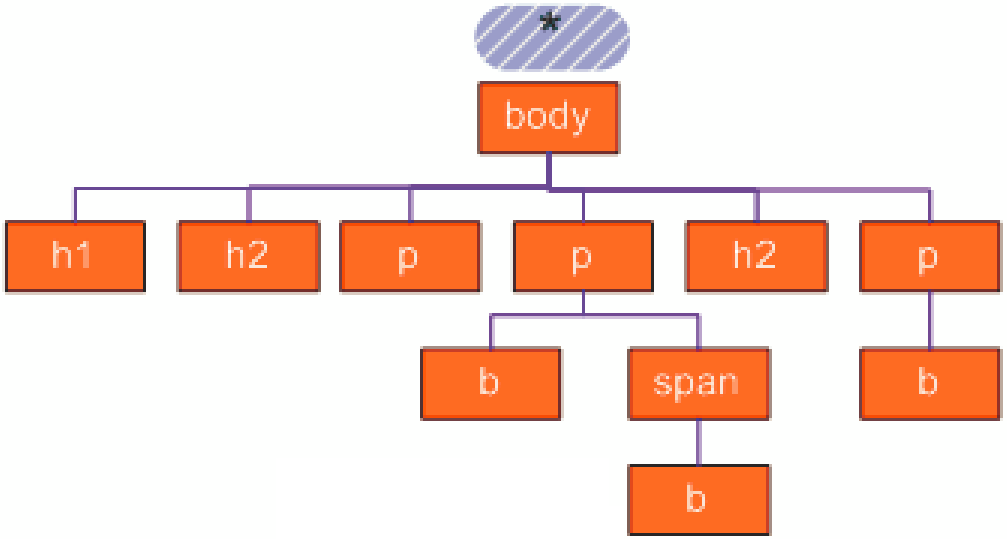


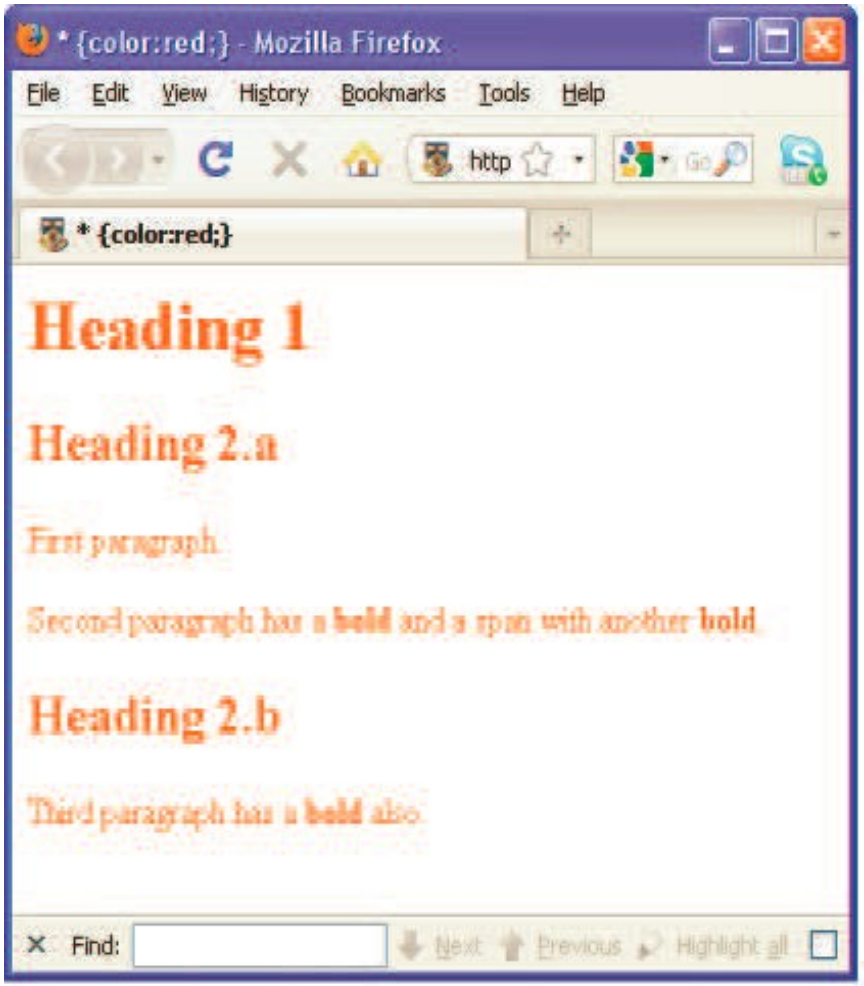
Selector: Any element (universal)

```
<html>
    <head>
        <style type="text/css">
            * {color:red;}
        </style>
    </head>
    <body>
        <h1> Heading 1</h1>
        <h2> Heading 2.a</h2>
        <p> First paragraph. </p>
        <p> Second paragraph has a <b>bold</b>
                and a <span>span with another <b>bold</b></span>. </p>
        <h2> Heading 2.b</h2>
        <p> Third paragraph has a <b>bold</b> also. </p>
    </body>
</html>
```



Selector: Any element

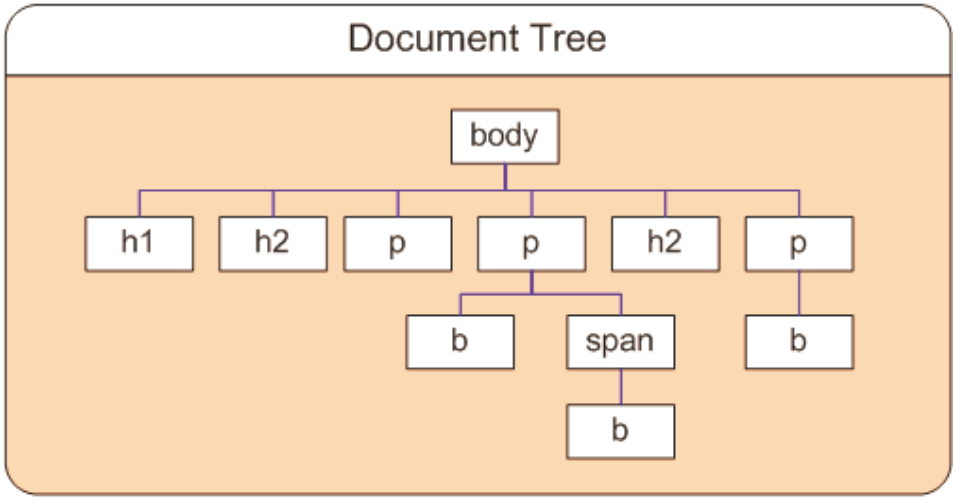




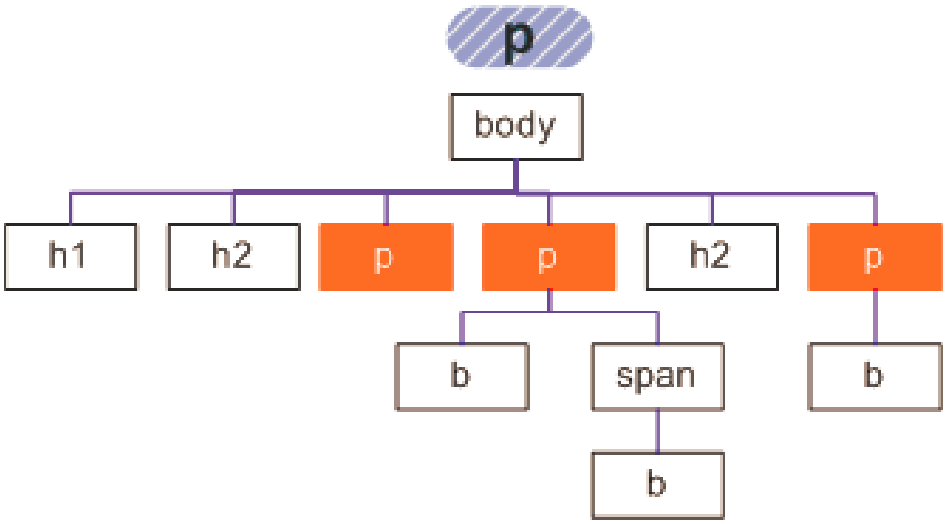
Any selector example

Example: Selector p

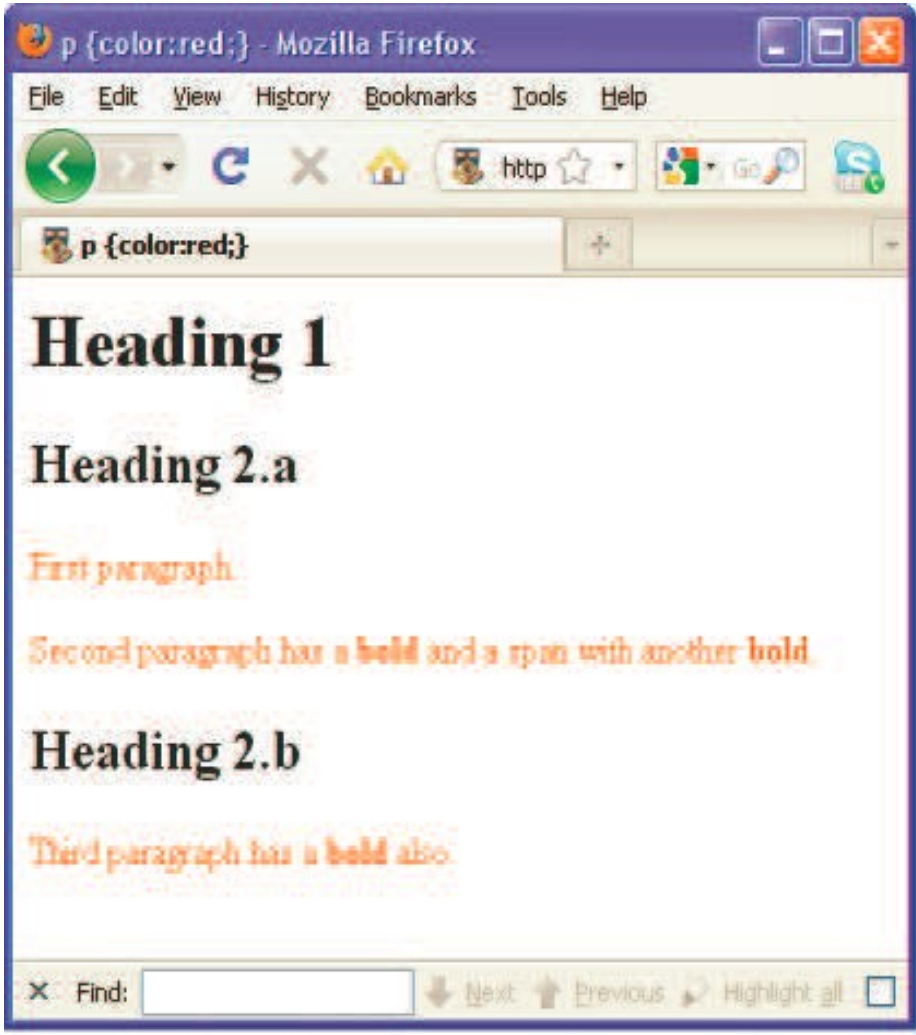
```
<html>
    <head>
        <style type="text/css">
        <!–
            p {color:red;}
        -->
        </style>
    </head>
    <body>
        <h1> Heading 1</h1>
        <h2> Heading 2.a</h2>
        <p> First paragraph. </p>
        <p> Second paragraph has a <b>bold</b>
            and a <span>span with another <b>bold</b></span>. </p>
        <h2> Heading 2.b</h2>
        <p> Third paragraph has a <b>bold</b> also. </p>
    </body>
</html>
```



Example: Selector p

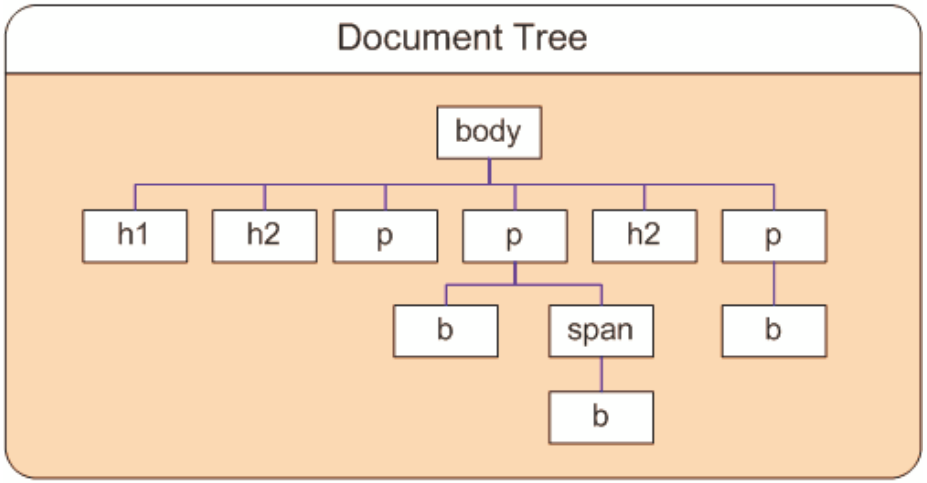


Selector p example

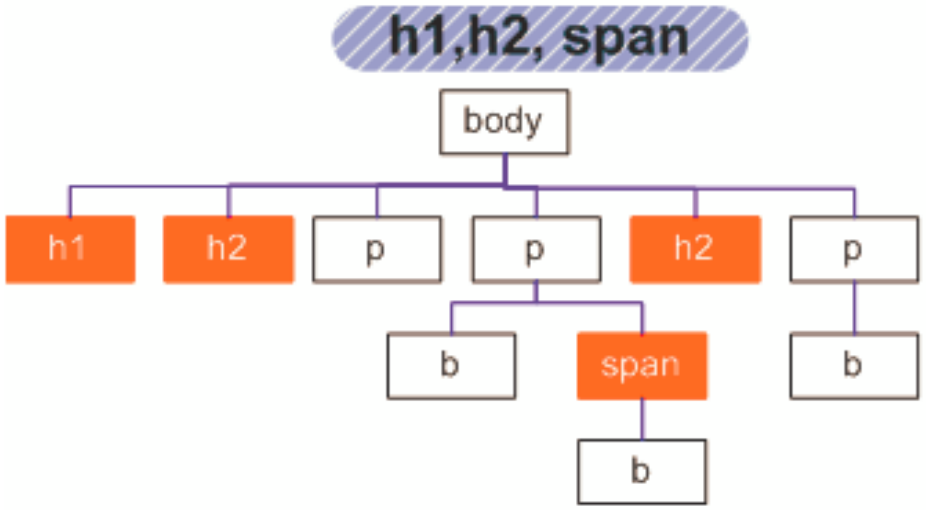


Example: Selector h1,h2,span

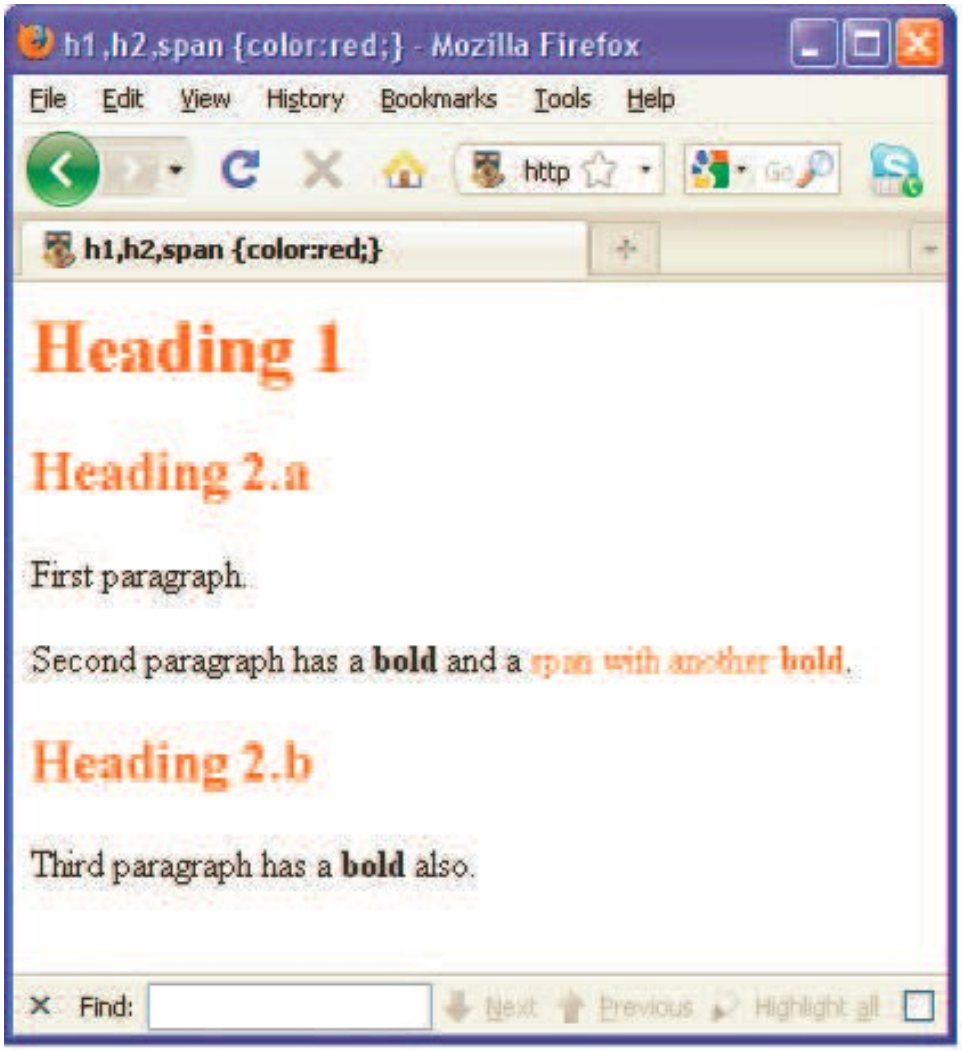
```
<html>
    <head>
        <style type="text/css">
        <!–
          h1,h2,span {color:red;}
        -->
        </style>
    </head>
    <body>
        <h1> Heading 1</h1>
        <h2> Heading 2.a</h2>
        <p> First paragraph. </p>
        <p> Second paragraph has a <b>bold</b>
          and a <span>span with another <b>bold</b></span>. </p>
        <h2> Heading 2.b</h2>
        <p> Third paragraph has a <b>bold</b> also. </p>
    </body>
</html>
```



Example: Selector h1,h2,span

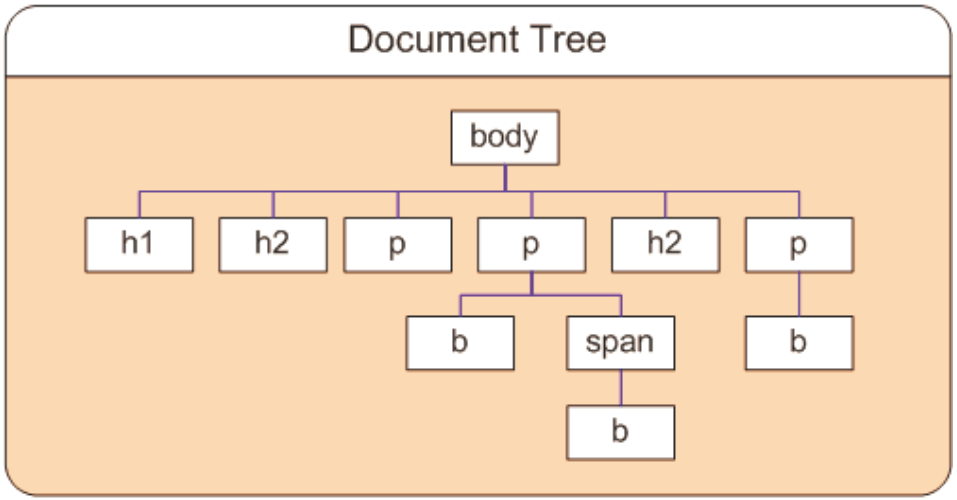


Selector h1,h2,span example

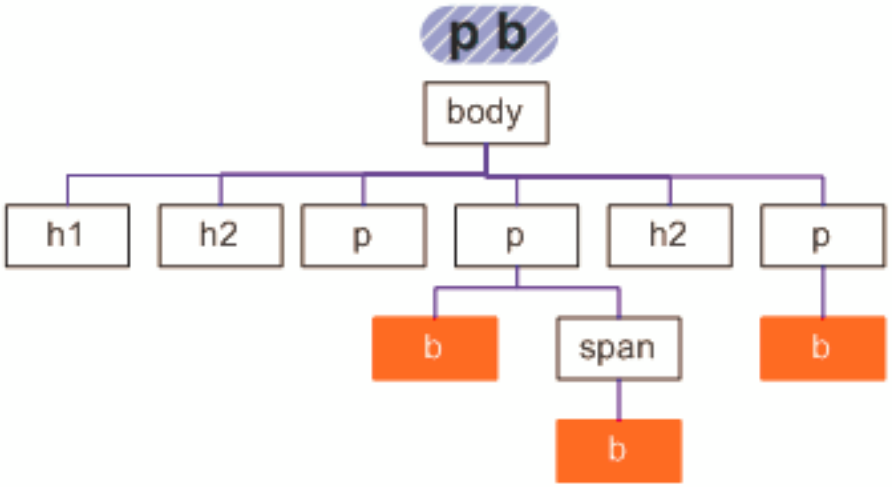


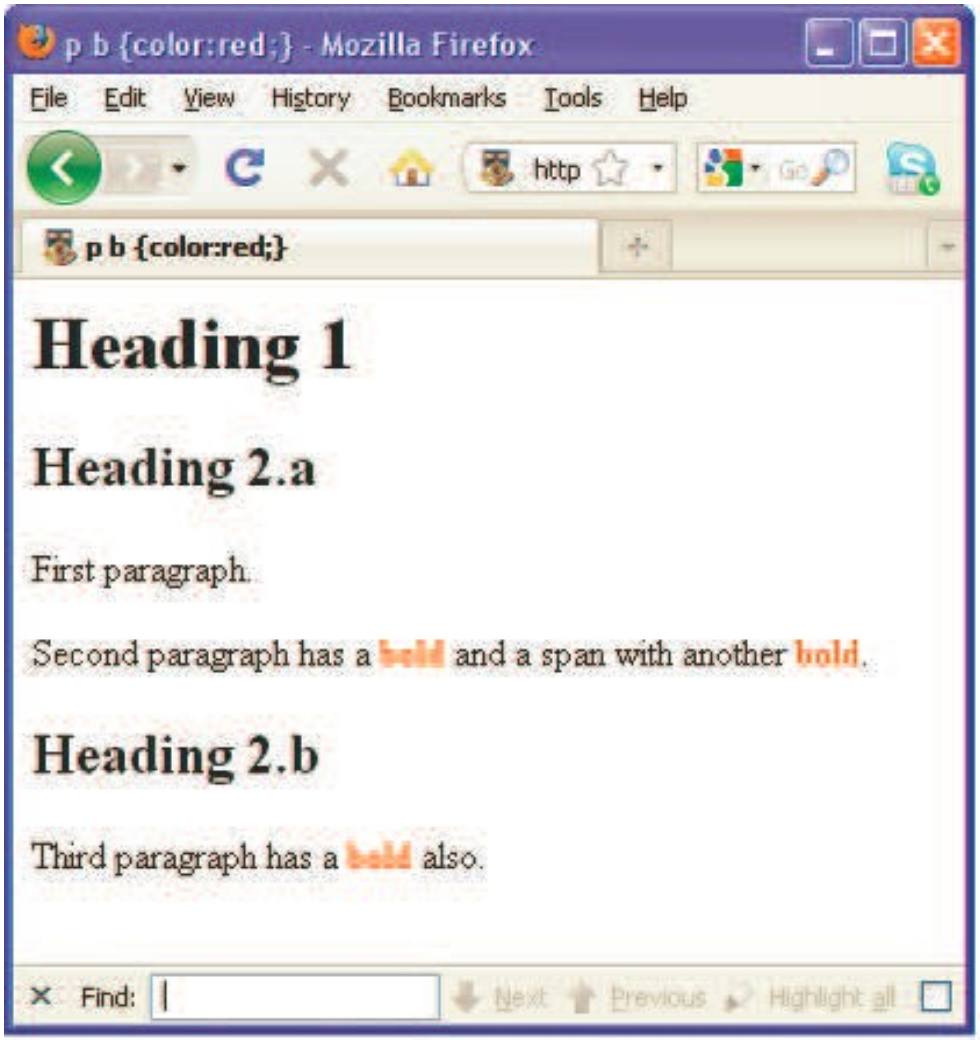
Example: Selector p b

```
<html>
    <head>
        <style type="text/css">
        <!–
          p b {color:red;}
        -->
        </style>
    </head>
    <body>
        <h1> Heading 1</h1>
        <h2> Heading 2.a</h2>
        <p> First paragraph. </p>
        <p> Second paragraph has a <b>bold</b>
          and a <span>span with another <b>bold</b></span>. </p>
        <h2> Heading 2.b</h2>
        <p> Third paragraph has a <b>bold</b> also. </p>
    </body>
</html>
```



Example: Selector p b

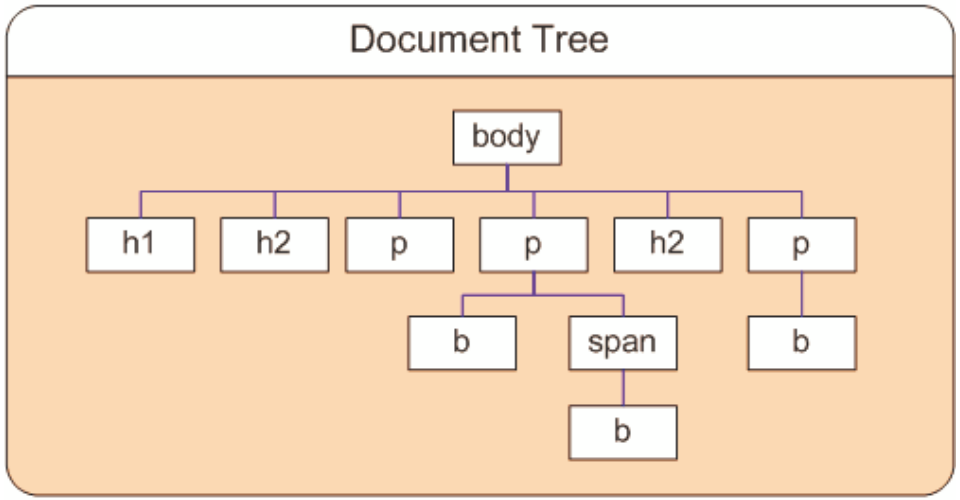




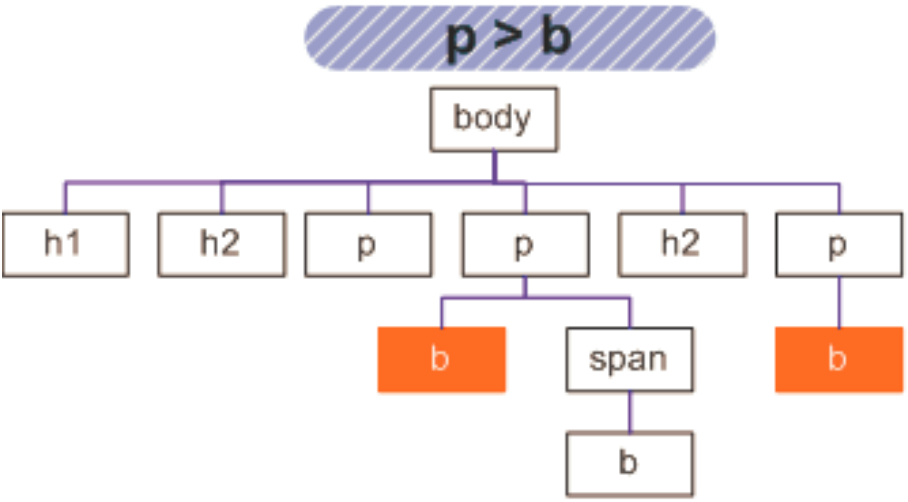
Selector p b example

Example: Selector p>b

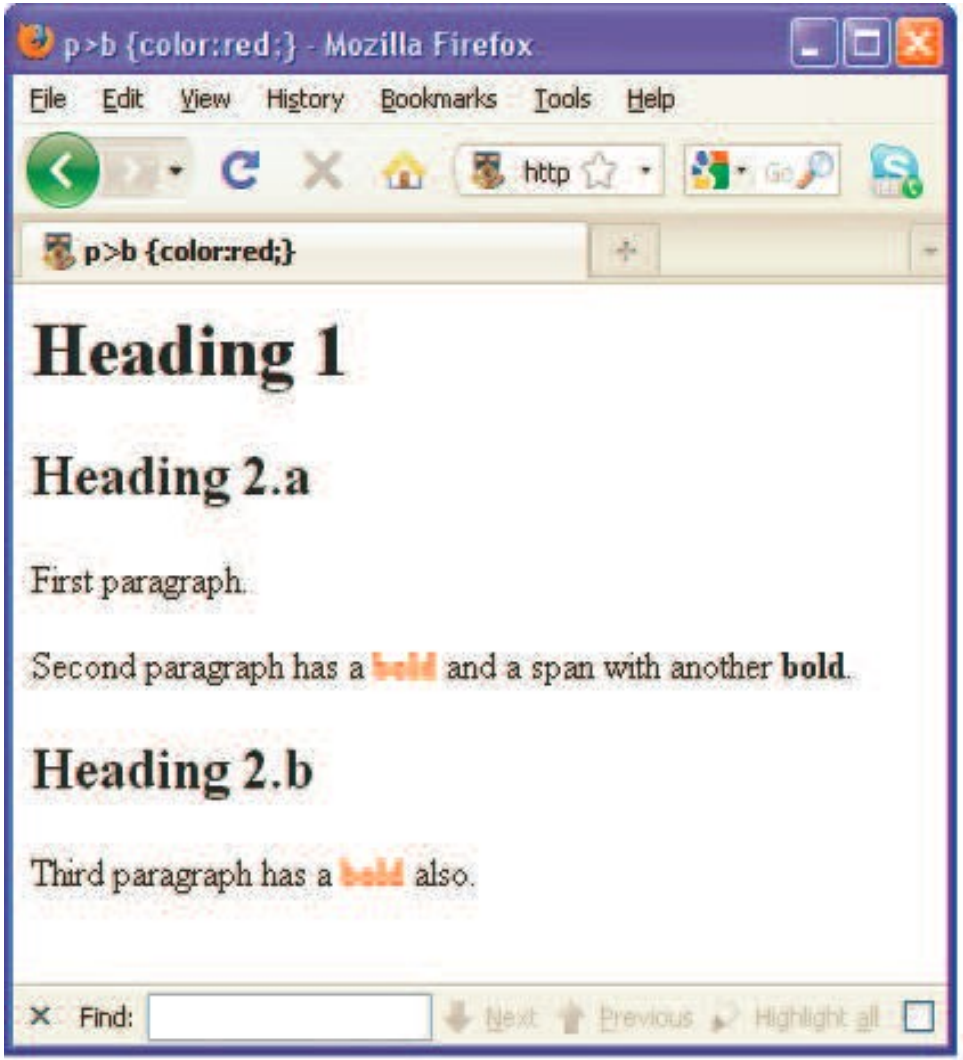
```
<html>
    <head>
        <style type="text/css">
        <!–
          p>b {color:red;}
        -->
        </style>
    </head>
    <body>
        <h1> Heading 1</h1>
        <h2> Heading 2.a</h2>
        <p> First paragraph. </p>
        <p> Second paragraph has a <b>bold</b>
          and a <span>span with another <b>bold</b></span>. </p>
        <h2> Heading 2.b</h2>
        <p> Third paragraph has a <b>bold</b> also. </p>
    </body>
</html>
```



Example: Selector p>b

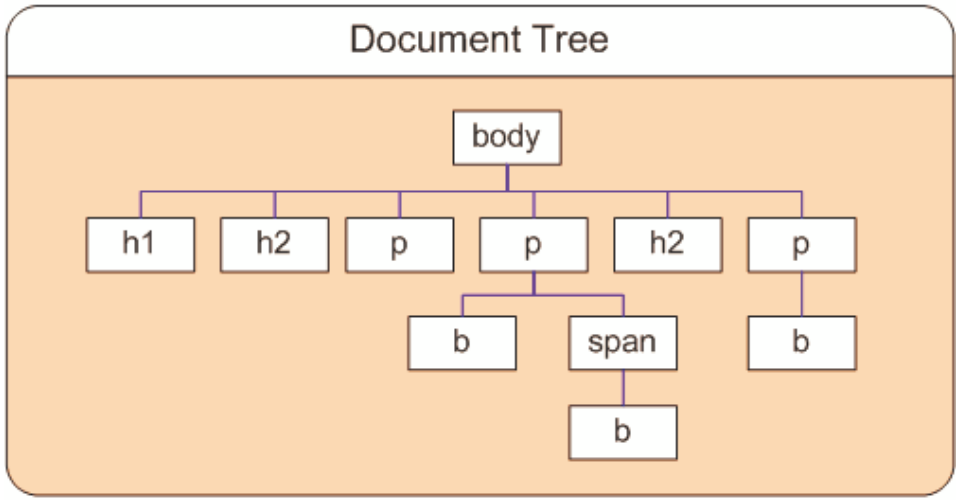


Selector p>b example

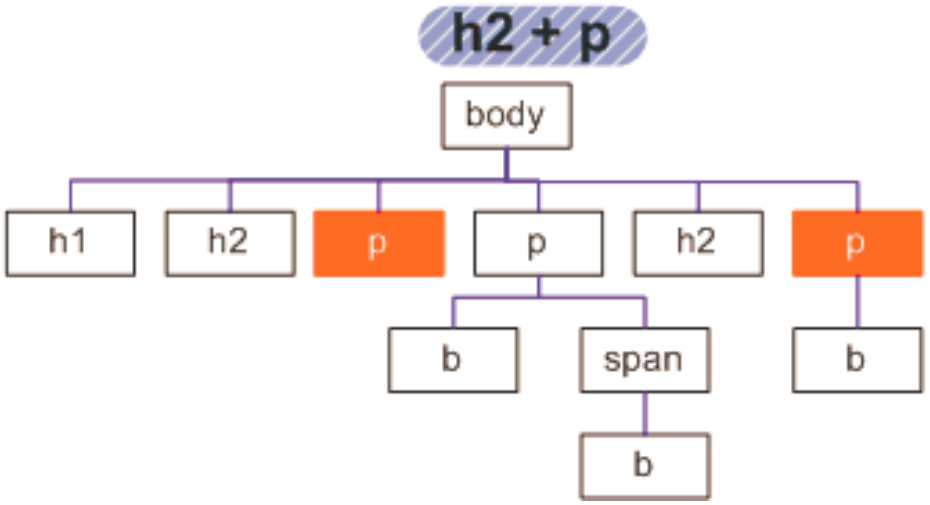


Example: Selector h2+p

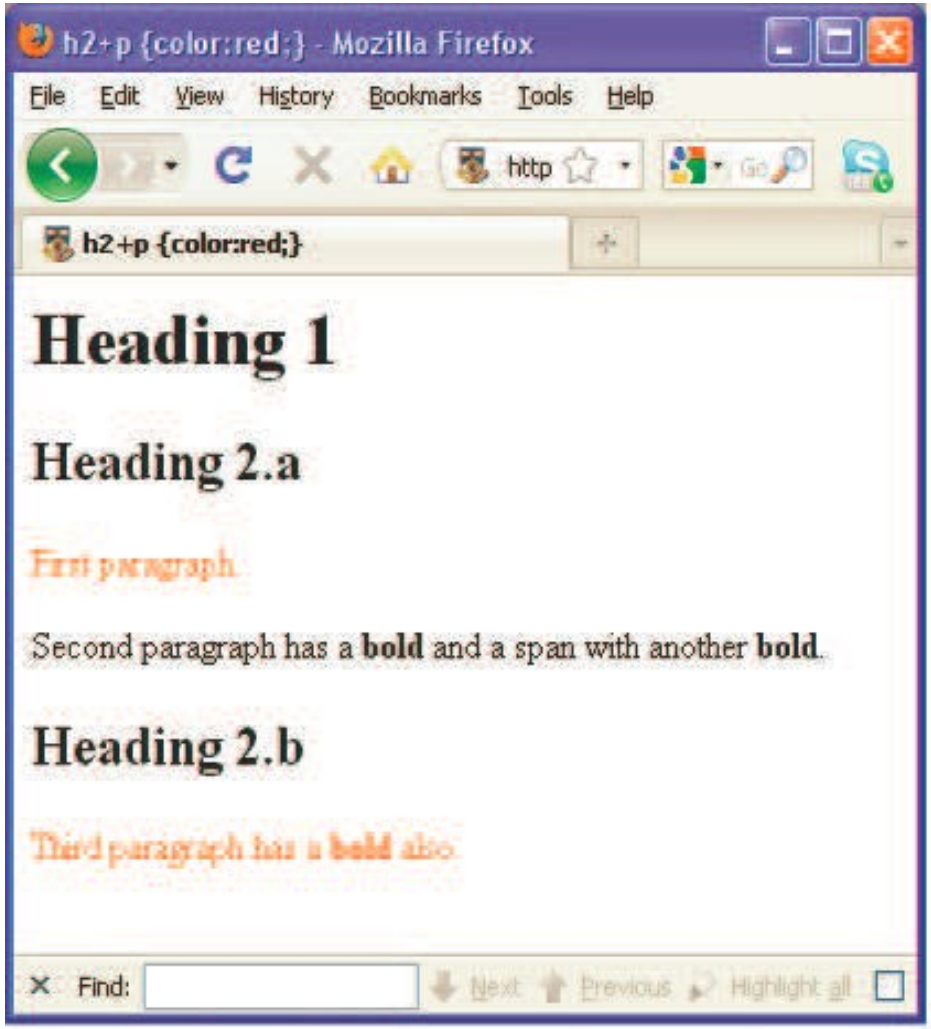
```
<html>
    <head>
        <style type="text/css">
        <!–
          h2+p {color:red;}
        -->
        </style>
    </head>
    <body>
        <h1> Heading 1</h1>
        <h2> Heading 2.a</h2>
        <p> First paragraph. </p>
        <p> Second paragraph has a <b>bold</b>
          and a <span>span with another <b>bold</b></span>. </p>
        <h2> Heading 2.b</h2>
        <p> Third paragraph has a <b>bold</b> also. </p>
    </body>
</html>
```



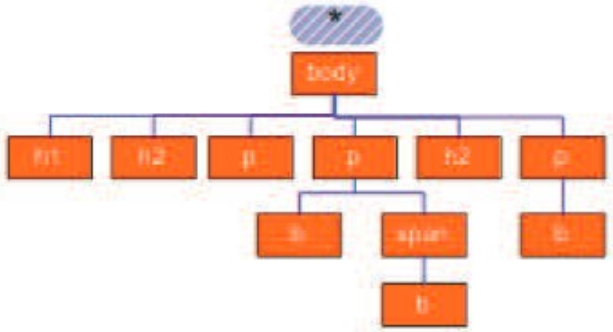
Example: Selector h2+p

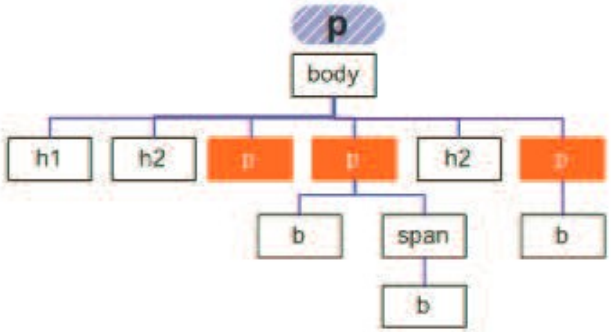


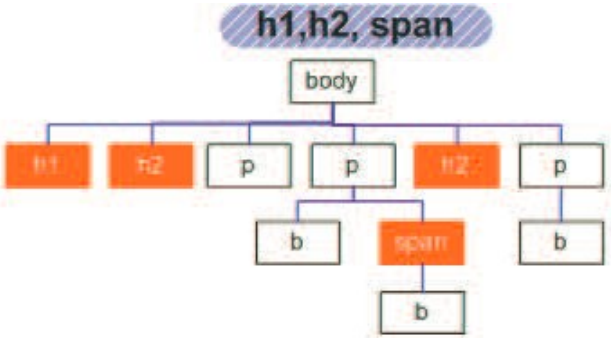
Selector h2+p example

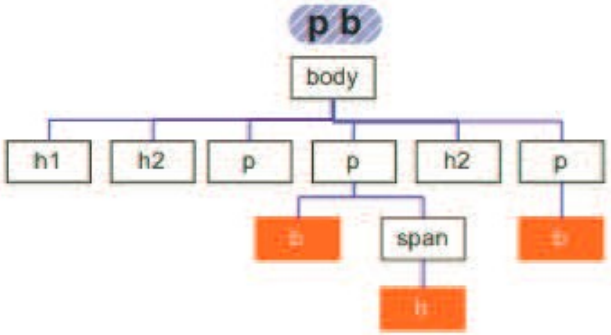


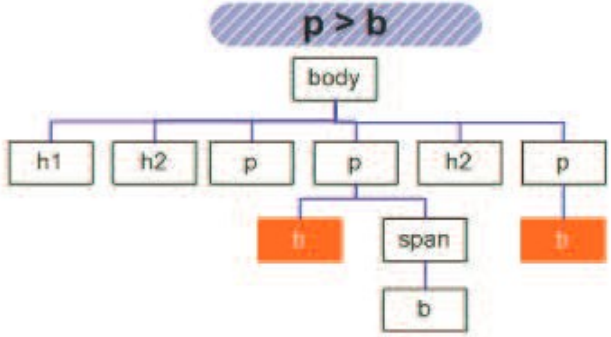
Examples Summary

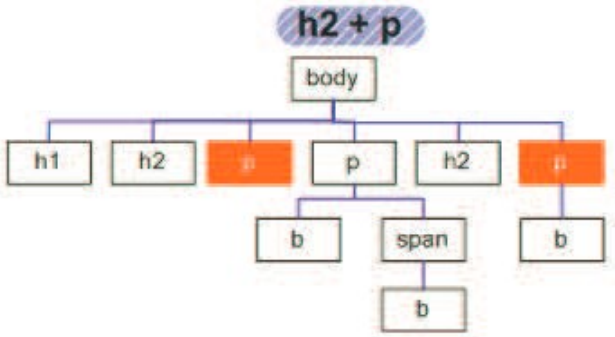












Code Summary

- Any selector example * {color:red;}
- Selector p example p {color:red;}
- Selector h1,h2,span ex. h1,h2,span {color:red;}
- Selector p b example p b {color:red;}
- Selector p>b example p>b {color:red;}
- Selector h2+p example h2+p {color:red;}

Element CLASS selector

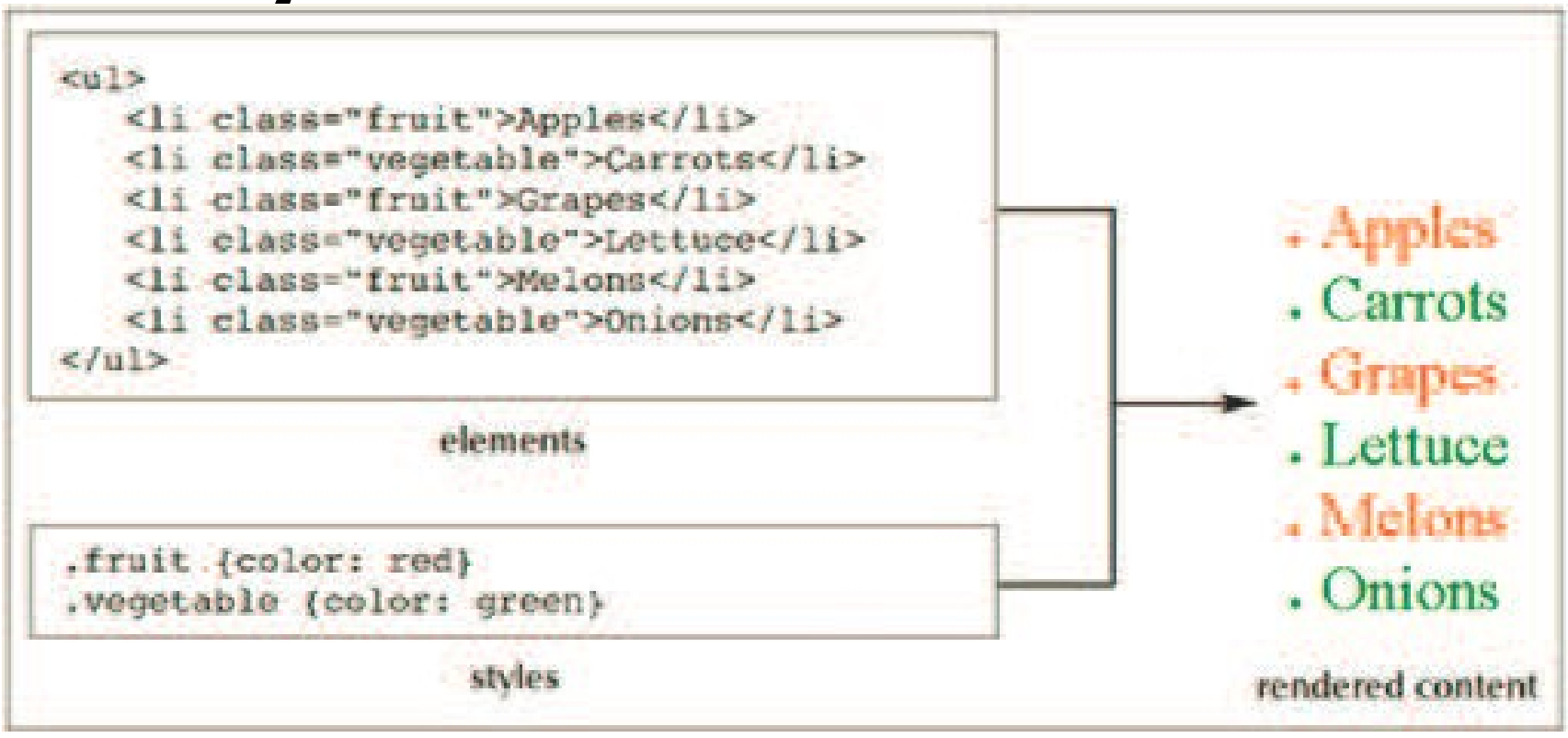
- Elements can be selected on the basis of their **class**:

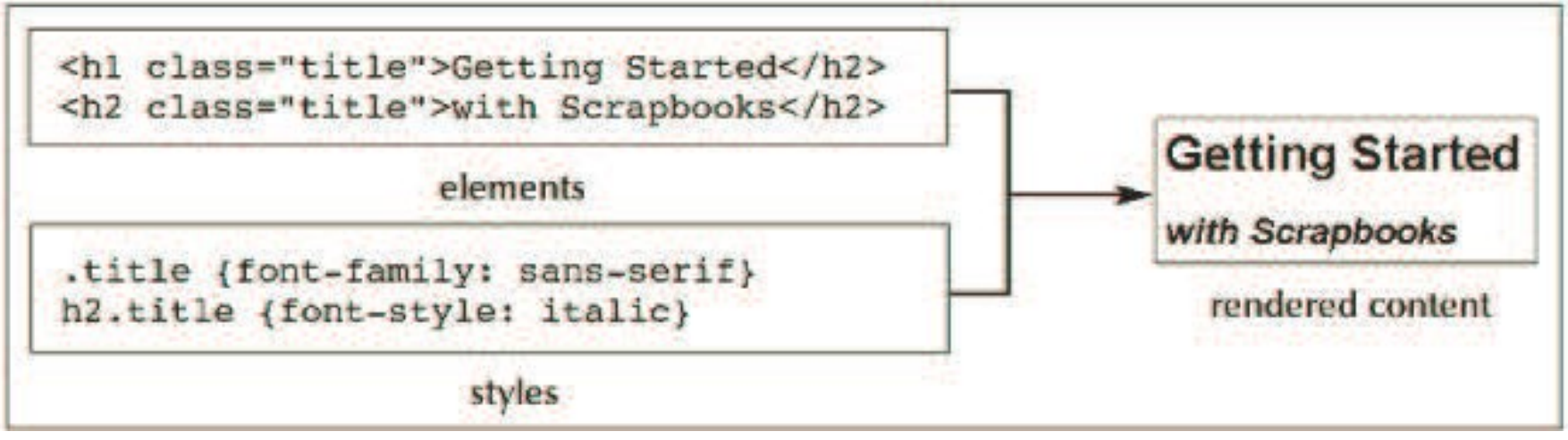
```
<html><head>
<title>CLASS selector example</title>
<style type="text/css">
        .important {font-size:larger}
        .trivial {font-size:smaller}
</style></head>
<body><h2>Warning</h2>
        <p class="important">Important text</p>
        <p class="trivial">Less important text</p>
</body>
</html>
```

What this CSS will do?

Think about an example where class might be useful.

Style for a class of elements





Element ID selector

- Styles can be applied to elements with a specific **id**

```
<html><head>
<title>ID selector example</title>
<style type="text/css">

        p {font-family: "Times New Roman", serif}
        #special {font-family: Courier, sans-serif}

</style></head>
<body>
        <p>Any element may have a ID attribute.</p>
        <p id="special">So long as it is unique.</p>
</body>
</html>
```

What this CSS will do?

Pseudo-elements selectors

- **Rollover effects** can be created using pseudo-classes
- **Pseudo-elements** are elements based on information about an element's content, use or position

Pseudo-element	Description	Example
first-letter	The first letter of the element text	p:first-letter {font-size: 14pt}
first-line	The first line of the element text	p:first-line {text-transform: uppercase}
before	Content to be placed directly before the element (CSS2)	p:before {content: "Special!"}
after	Content to be placed directly after the element (CSS2)	p:after {content: "eof"}

Pseudo-element selectors

- CSS has two pseudo-elements: first-letter and first-line.

```
<html>
    <head><title>The style of a news letter</title>
        <style type="text/css">
            p.special:first-line {font-variant:small-caps}
            p.special:first-letter {font-size: 300%; float:left}
        </style>
    </head>
    <body>
        <h1>Student Club News</h1>
        <p class="special">The first line of this news letter is in
            small capital letters and the first letter is 300% larger (
            a drop cap).</p> …
```

What this CSS will do?

The resulting page …

Types of selectors (extra)

- Contextual selectors
  - **h1 em {color: blue}** /*the **** within the **<h1>***/
  - **body > p {line-height: 1.3}** /***<p>** directly within the **<body>***/
  - **h1+h2 {margin-top: -5mm}** /* **<h2>** directly after an **<h1>** */
  - **div > p:first-child {text-indent: 0}**

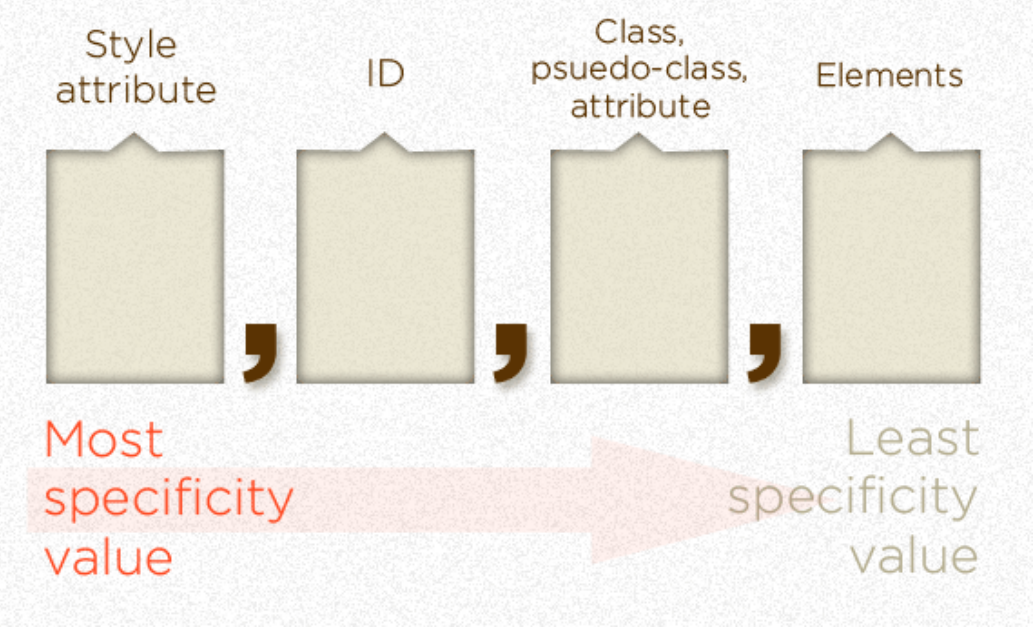
    /***<p>** that is the first child of a ** ***/
- Selection on attribute
  - **a[href] {border: solid}** /* selection by existence */
  - **span[class="example"] {color:blue}** /* selection by value */
  - **div[status~="important"] {z-order:2}**

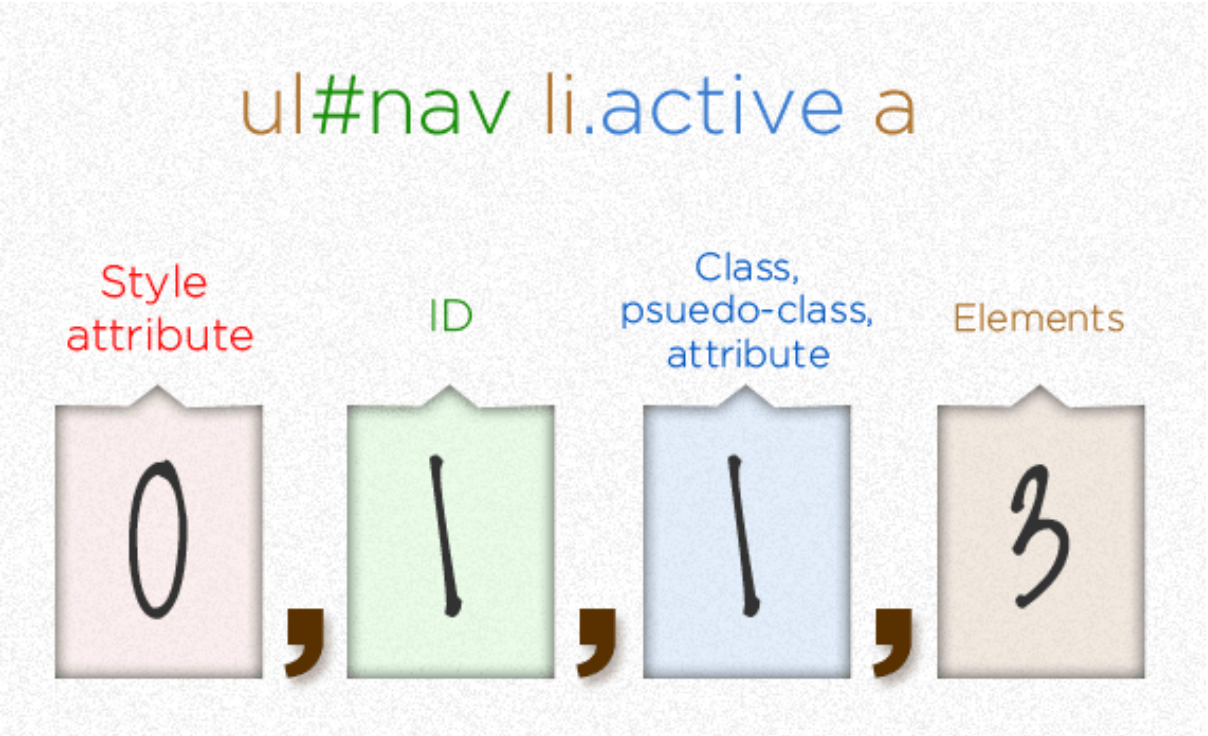
    /* selection by value: from comma-separated list */

Calculating CSS Specificity Value

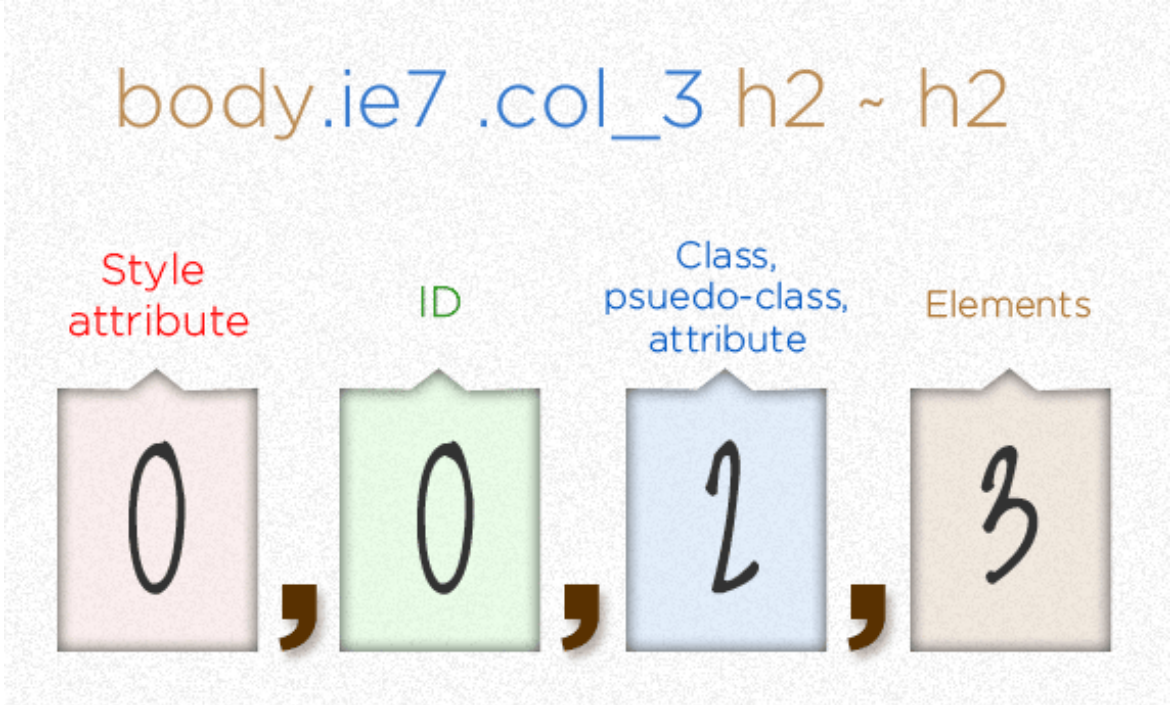
In other words:

- If the element has inline styling, that automatically wins (1,0,0,0 points)
- For each ID value, apply 0,1,0,0 points
- For each class value (or pseudo-class or attribute selector), apply 0,0,1,0 points
- For each element reference, apply 0,0,0,1 point





Sample Calculations



https://flukeout.github.io/

http://davidshariff.com/quiz/