PES UNIVERSITY
CELEBRATING 50 YEARS

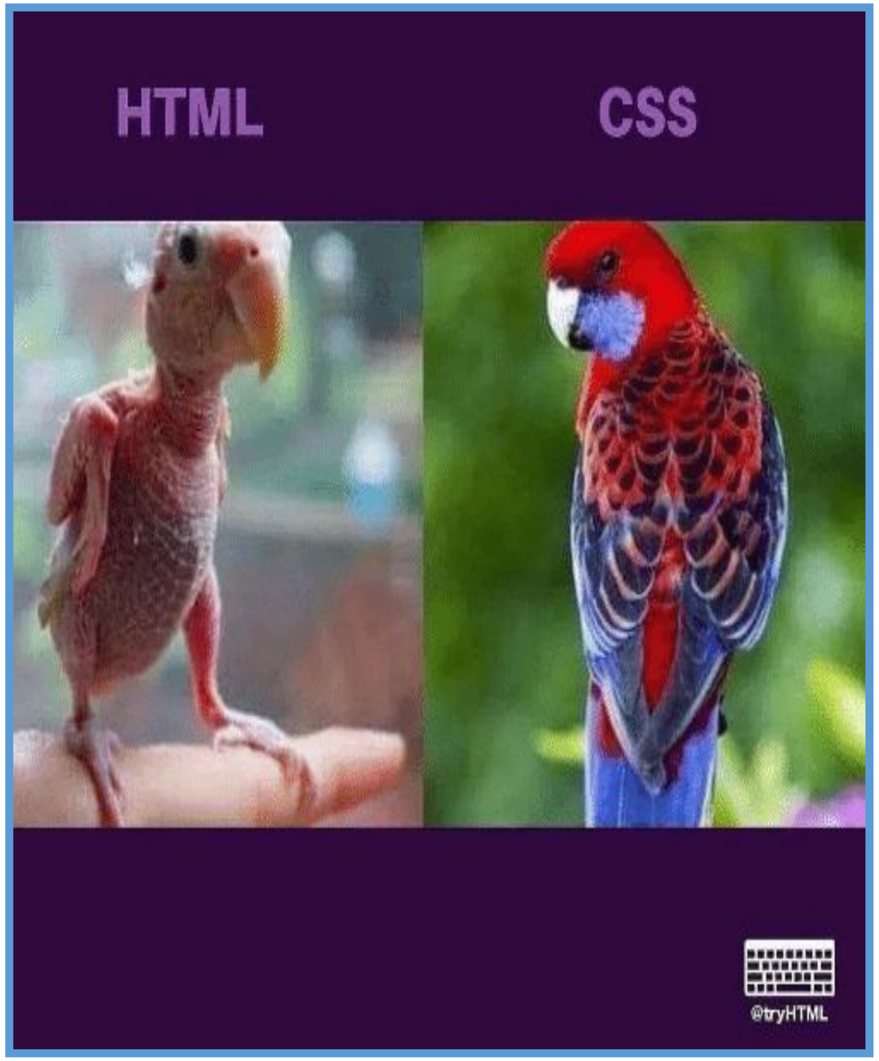
WEB TECHNOLOGIES

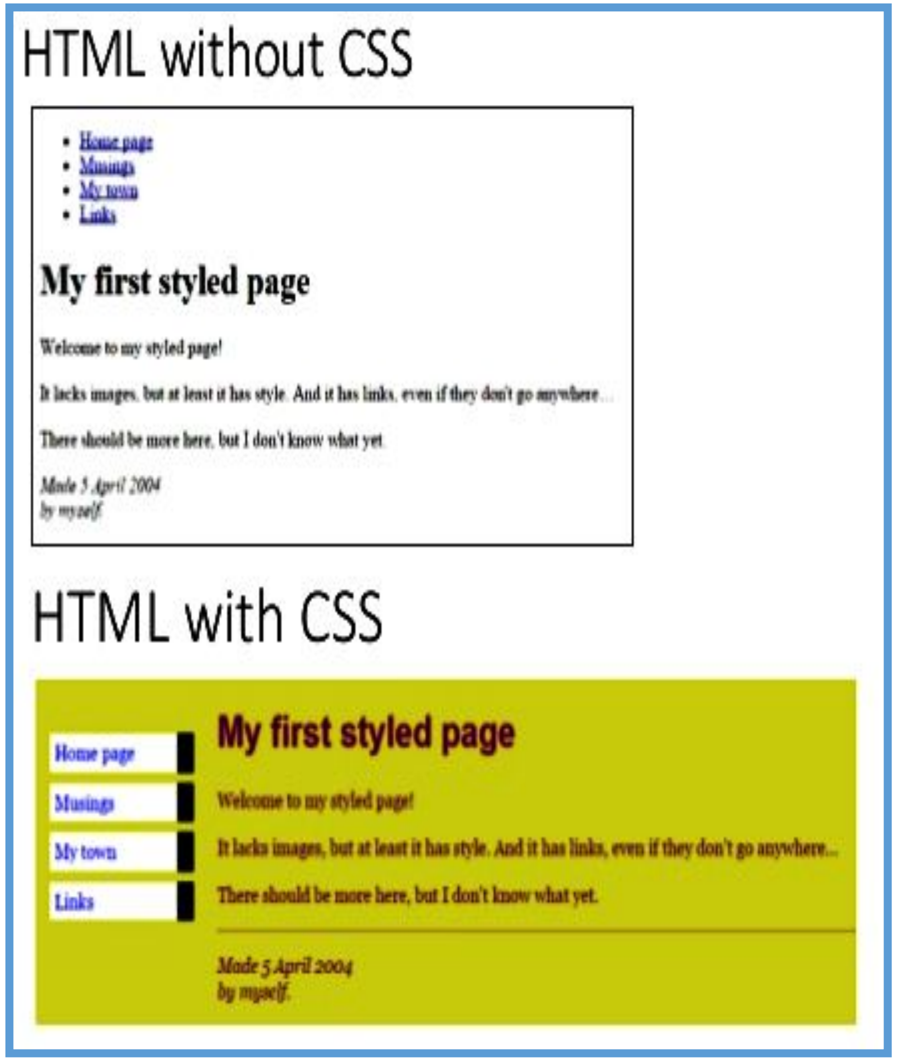
CSS – Cascading Style Sheet and Selectors

Prof.Vinay Joshi

Department of Computer Science and Engineering

Introduction




HTML without CSS

- Home page
- Musings
- My town
- Links

My first styled page

Welcome to my styled page!

It lacks images, but at least it has style. And it has links, even if they don't go anywhere...

There should be more here, but I don't know what yet.

Made 5 April 2004
by myself.

HTML with CSS

Home page
Musings
My town
Links

My first styled page

Welcome to my styled page!

It lacks images, but at least it has style. And it has links, even if they don't go anywhere...

There should be more here, but I don't know what yet.

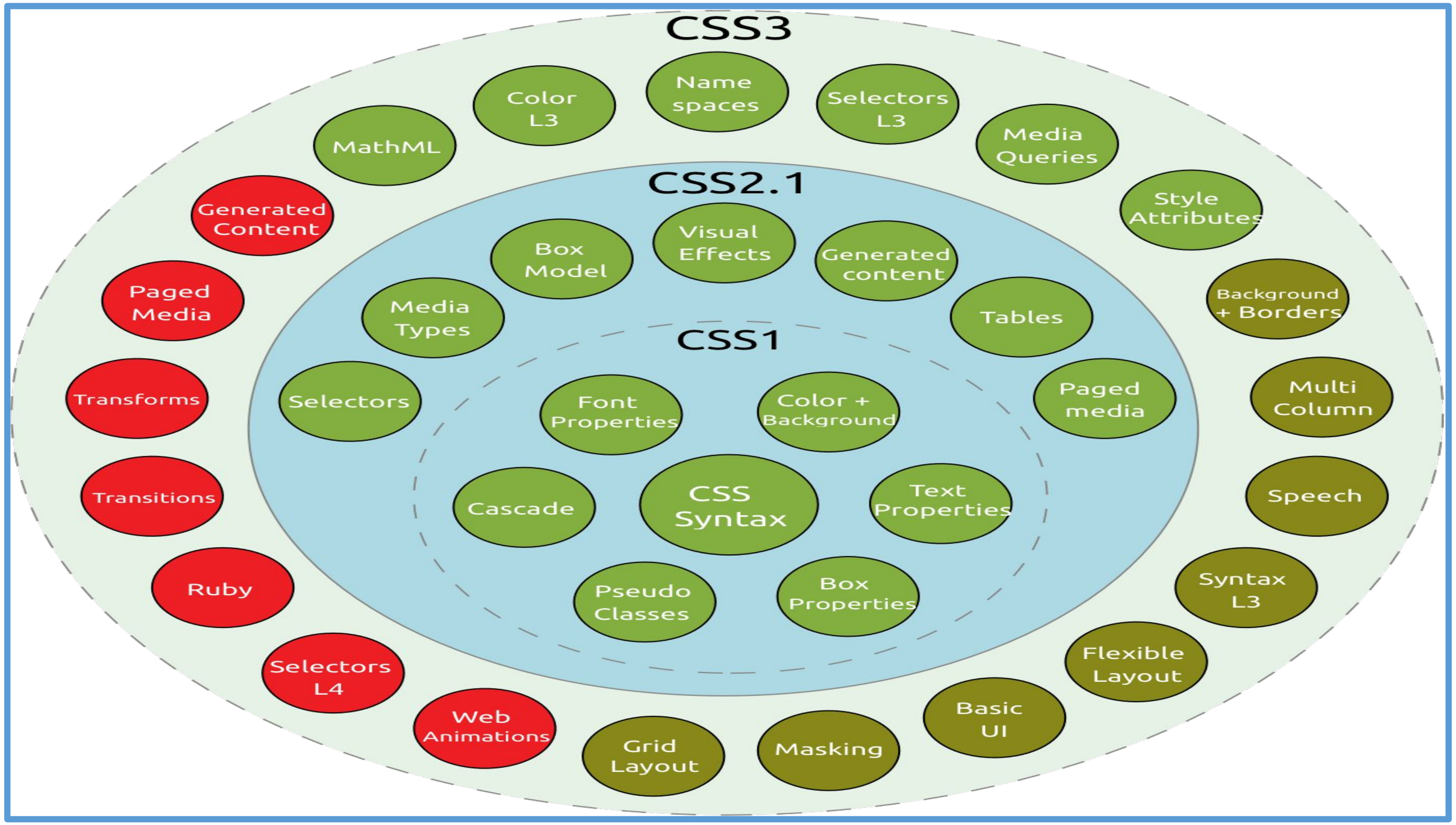
Made 5 April 2004
by myself.

Cascading Style Sheets

- The CSS1 specification was developed in 1996
- CSS2 was released in 1998
- CSS3 was released in 1999 and presentation-style properties were added in it that allows you to build a presentation from documents.
- Allows you to specify the *presentation* of elements on a web page (e.g., fonts, spacing, sizes, colors, positioning) *separately* from the document's *structure and content* (section headers, body text, links, etc.).
- Simplifies maintaining and modifying web pages, especially on large-scale websites.

CSS – Cascading Style Sheets and Selectors

Introduction



CSS – Cascading Style Sheets and Selectors

Three ways to include CSS

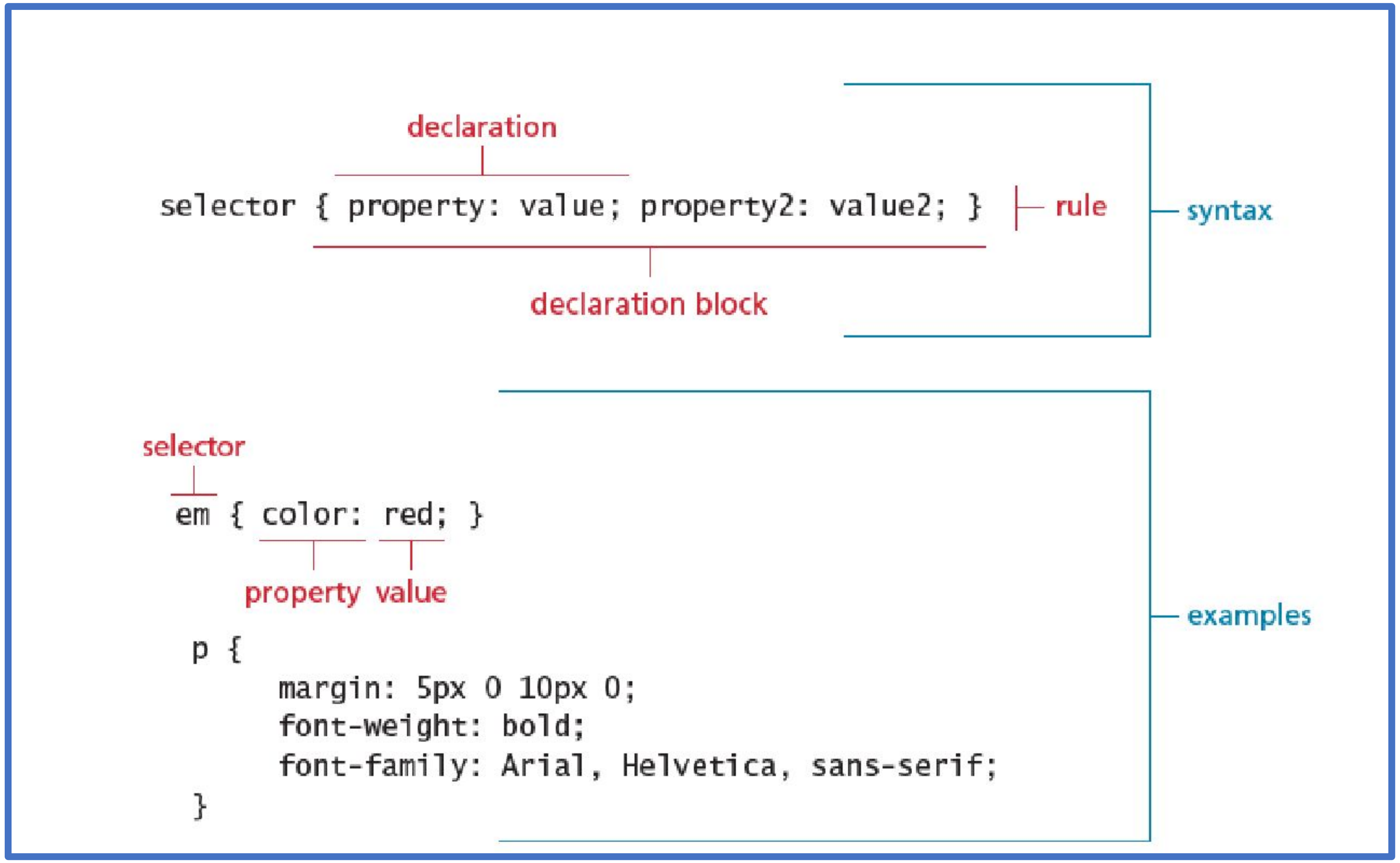
1) Inline Style - CSS is placed directly into the HTML element.

1) Internal Style Sheet /Embedded Style Sheet - CSS is placed into a separate area within the <head> section of a web page using <style> tag.

1) External Style Sheet - CSS is placed into a separate file and "connected" to a web page.

CSS – Cascading Style Sheets and Selectors

Syntax



```
selector { property: value; property2: value2; }
```

```
em { color: red; }
```

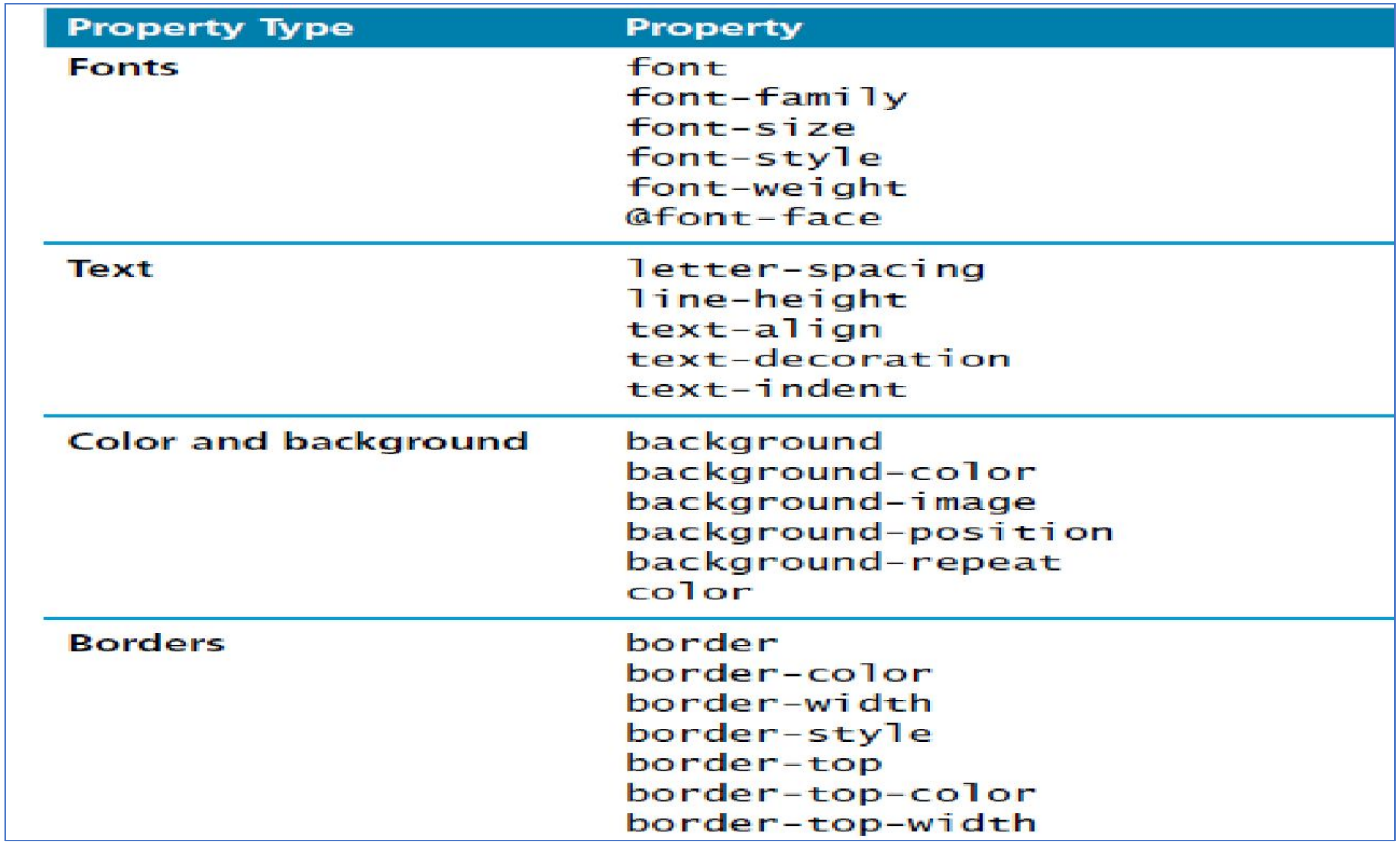
```
p {
     margin: 5px 0 10px 0;
     font-weight: bold;
     font-family: Arial, Helvetica, sans-serif;
}
```

Syntax

- Every CSS rule begins with a **selector**.
- The selector identifies which element or elements in the HTML document will be affected by the declarations in the rule

CSS – Cascading Style Sheets and Selectors

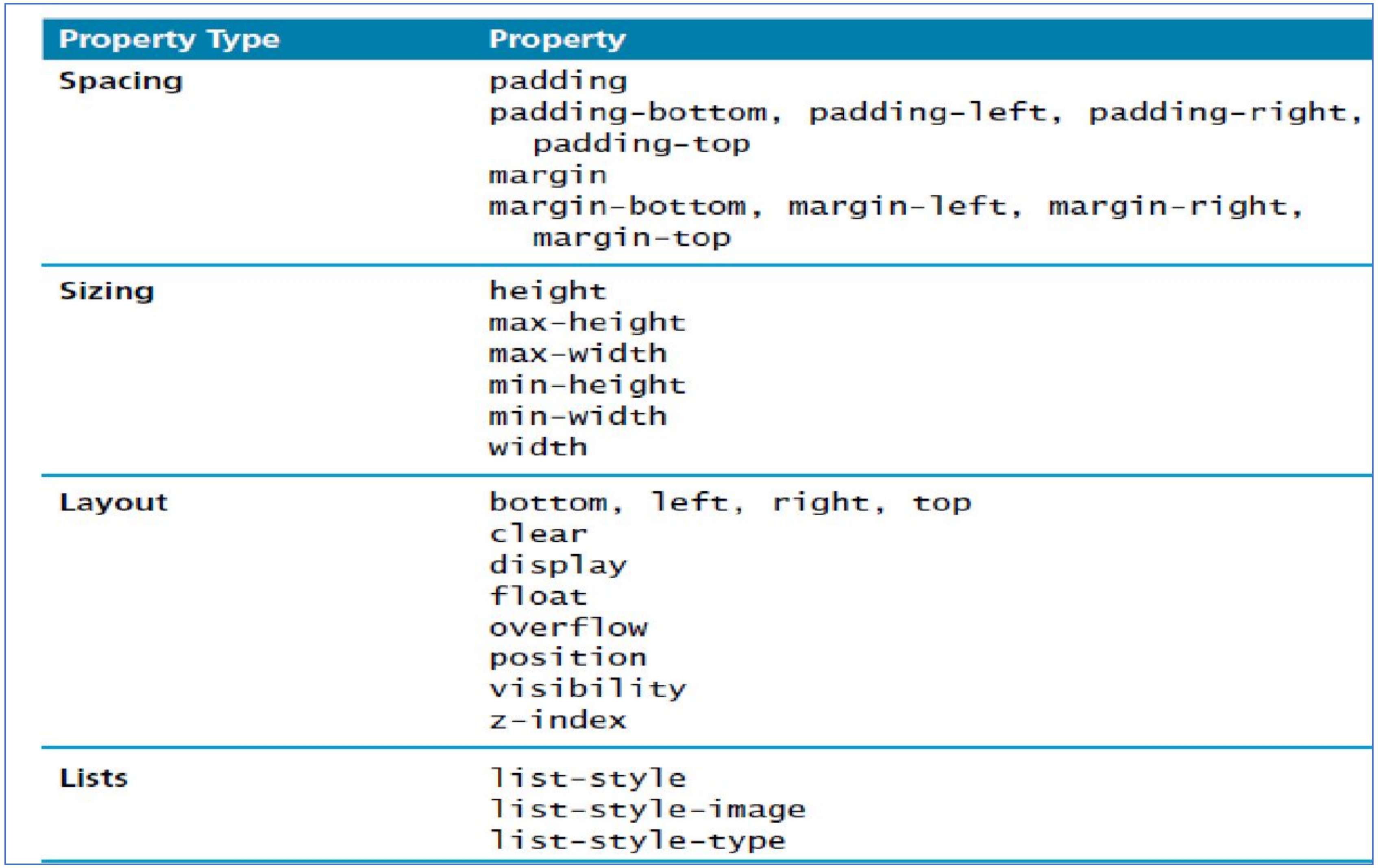
Properties



Property Type	Property
Fonts	font font-family font-size font-style font-weight @font-face
Text	letter-spacing line-height text-align text-decoration text-indent
Color and background	background background-color background-image background-position background-repeat color
Borders	border border-color border-width border-style border-top border-top-color border-top-width

CSS – Cascading Style Sheets and Selectors

Properties



Property Type	Property
Spacing	padding padding-bottom, padding-left, padding-right, padding-top margin margin-bottom, margin-left, margin-right, margin-top
Sizing	height max-height max-width min-height min-width width
Layout	bottom, left, right, top clear display float overflow position visibility z-index
Lists	list-style list-style-image list-style-type

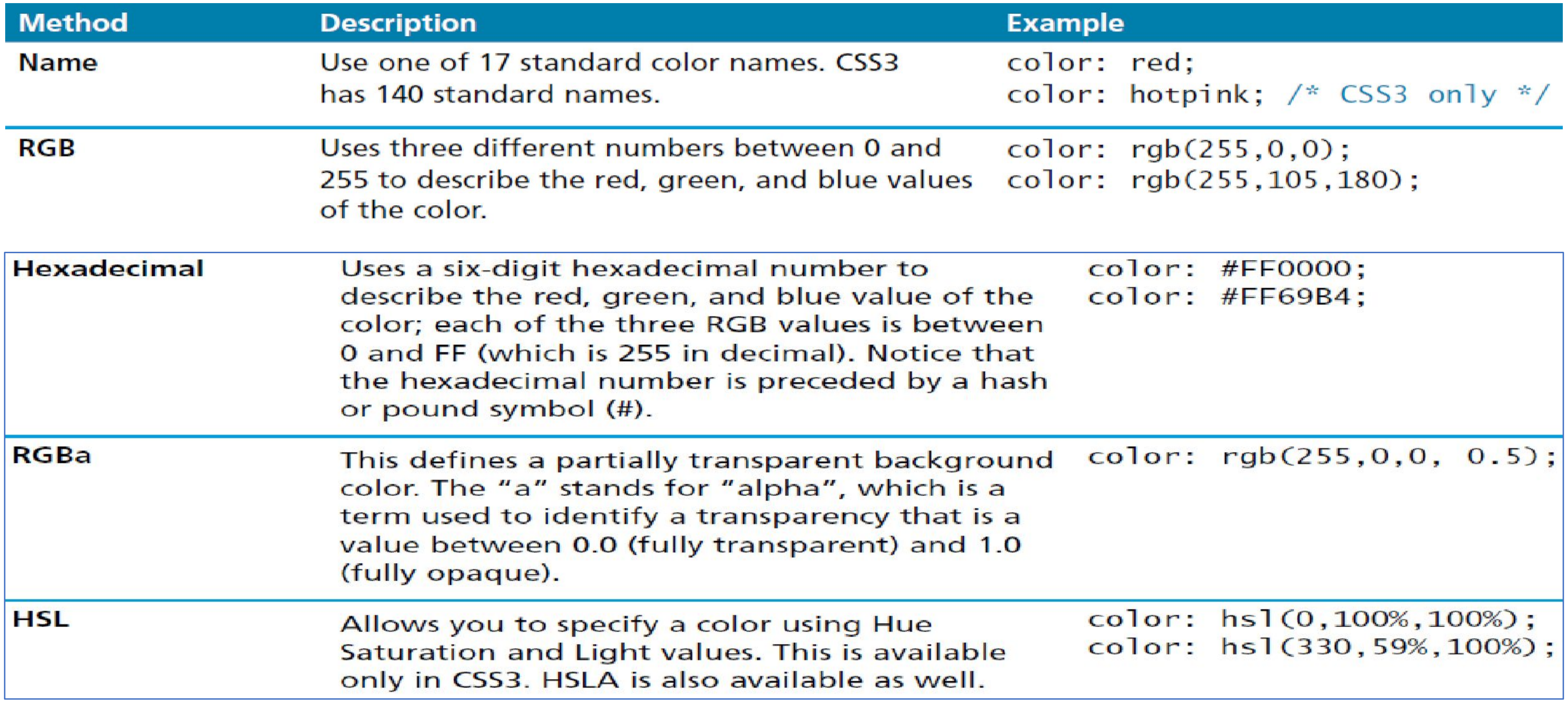
CSS – Cascading Style Sheets and Selectors

Properties

- Some property values are from a predefined list of keywords.
- Others are values such as length measurements, percentages, numbers without units, color values, and URLs.

CSS – Cascading Style Sheets and Selectors

Color values in CSS



Method	Description	Example

Name	Use one of 17 standard color names. CSS3 has 140 standard names.	`color: red;`
		`color: hotpink; /* CSS3 only */`

RGB	Uses three different numbers between 0 and 255 to describe the red, green, and blue values of the color.	`color: rgb(255,0,0);`
		`color: rgb(255,105,180);`

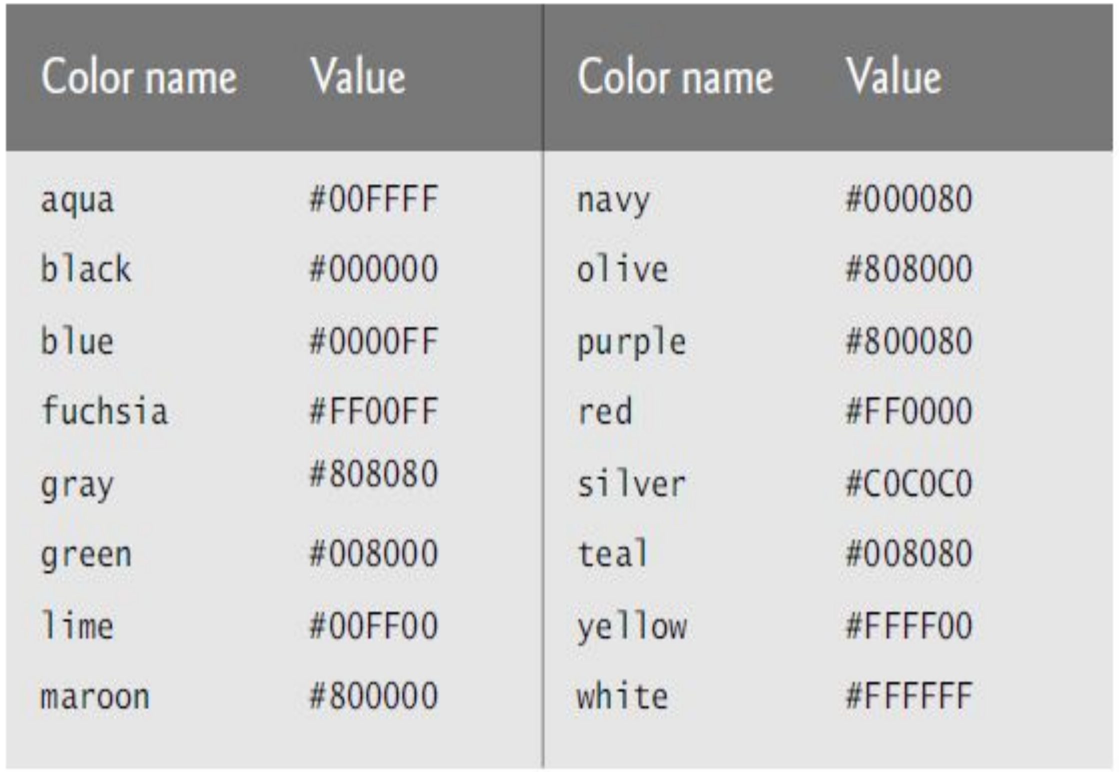
Hexadecimal	Uses a six-digit hexadecimal number to describe the red, green, and blue value of the color; each of the three RGB values is between 0 and FF (which is 255 in decimal). Notice that the hexadecimal number is preceded by a hash or pound symbol (#).	`color: #FF0000;`
		`color: #FF69B4;`

RGBa	This defines a partially transparent background color. The “a” stands for “alpha”, which is a term used to identify a transparency that is a value between 0.0 (fully transparent) and 1.0 (fully opaque).	`color: rgb(255,0,0, 0.5);`

HSL	Allows you to specify a color using Hue Saturation and Light values. This is available only in CSS3. HSLA is also available as well.	`color: hsl(0,100%,100%);`
		`color: hsl(330,59%,100%);`

Inline Styles

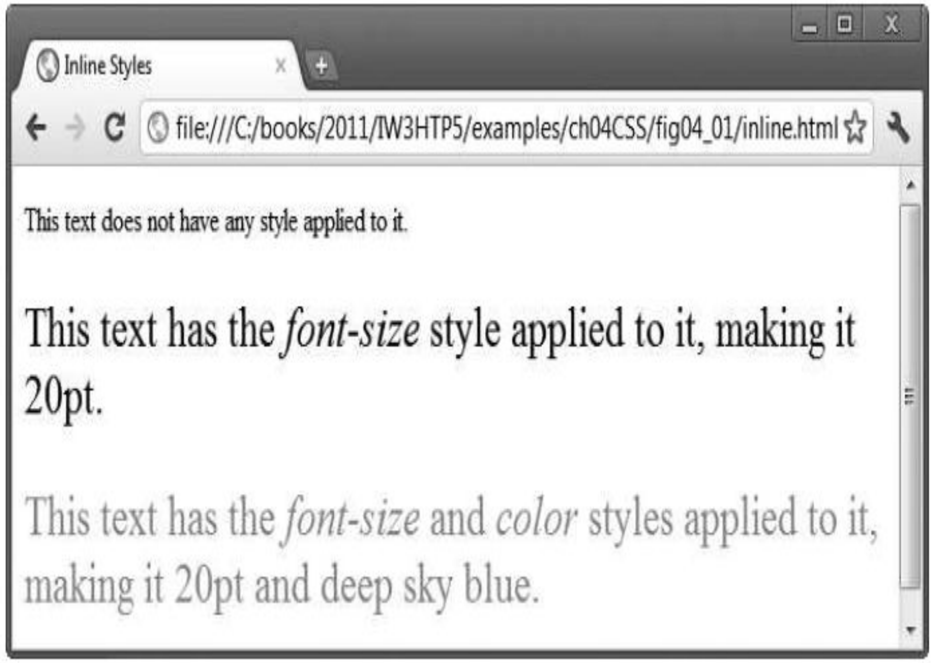
- **Inline styles** declare an individual element’s format using theHTML5 attribute **style**.
- Inline styles *override* any other styles applied.
- Figure contains the HTML standard color set.



Color name	Value	Color name	Value
aqua	#00FFFF	navy	#000080
black	#000000	olive	#808000
blue	#0000FF	purple	#800080
fuchsia	#FF00FF	red	#FF0000
gray	#808080	silver	#C0C0C0
green	#008000	teal	#008080
lime	#00FF00	yellow	#FFFF00
maroon	#800000	white	#FFFFFF

Fig. 4.2 | HTML standard colors and hexadecimal RGB values.

```
<!DOCTYPE html>

<!-- Fig. 4.1: inline.html -->
<!-- Using inline styles -->
<html>
   <head>
      <meta charset = "utf-8">
      <title>Inline Styles</title>
   </head>
   <body>
      <p>This text does not have any style applied to it.<

      <!-- The style attribute allows you to declare -->
      <!-- inline styles. Separate multiple -->
      <!-- style properties with a semicolon. -->
      <p style = "font-size: 20pt;">This text has the
         <em>font-size</em> style applied to it, making it
      </p>

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


Inline Styles
file:///C:/books/2011/IW3HTP5/examples/ch04CSS/fig04_01/inline.html

This text does not have any style applied to it.

This text has the *font-size* style applied to it, making it 20pt.

This text has the *font-size* and *color* styles applied to it, making it 20pt and deep sky blue.

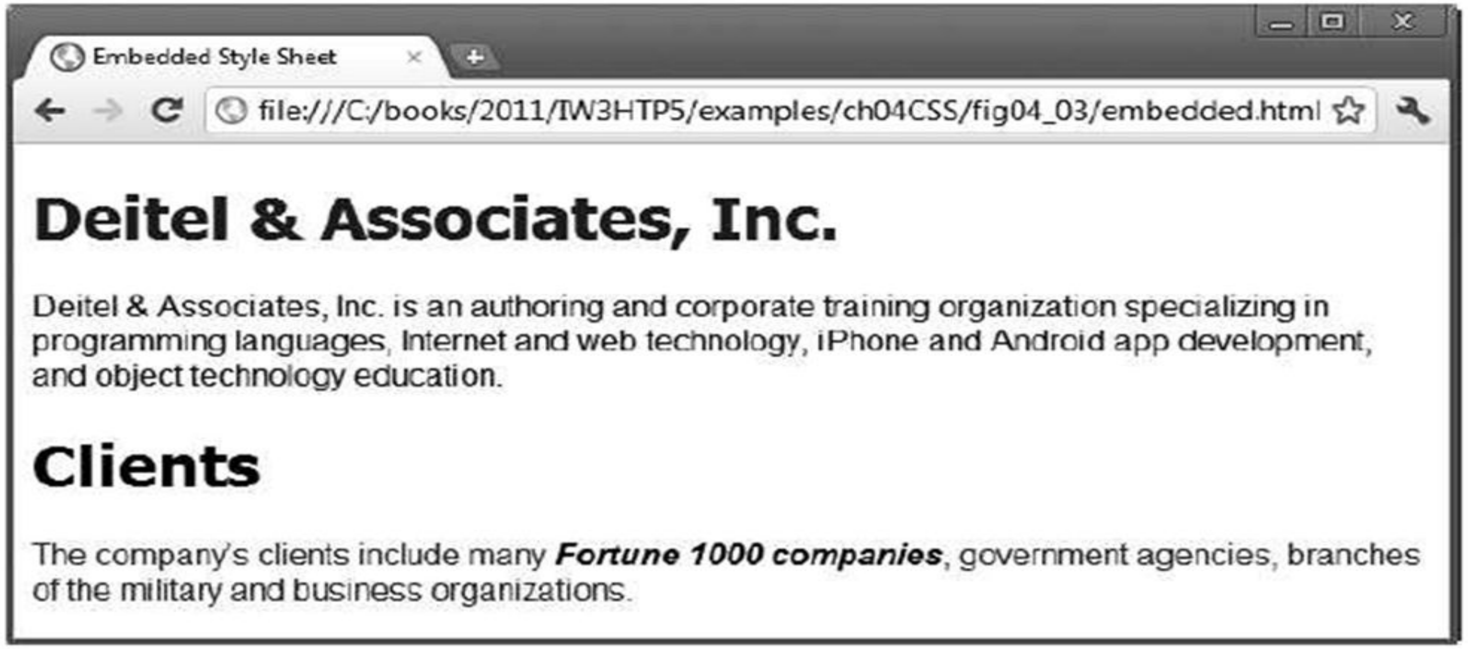
Fig. 4.1 | Using inline styles.

Embedded Style Sheets

- A second technique for using style sheets is **embedded style sheets.**
- enable you to *embed* a CSS3 document in an HTML5 document’s head section.

```
<!DOCTYPE html>

<!-- Fig. 4.3: embedded.html -->
<!-- Embedded style sheet. -->
<html>
   <head>
      <meta charset = "utf-8">
      <title>Embedded Style Sheet</title>

      <!-- this begins the style sheet section -->
      <style type = "text/css">
         em       { font-weight: bold;
                    color: black; }
         h1       { font-family: tahoma, helvetica, sans-serif; }
         p        { font-size: 12pt;
                    font-family: arial, sans-serif; }
         .special { color: purple; }
      </style>
   </head>
   <body>
      <!-- this attribute applies the .special style class -->
      <h1 class = "special">Deitel & Associates, Inc.</h1>

```

```
        <p>Deitel & Associates, Inc. is an authoring and
           corporate training organization specializing in
           programming languages, Internet and web technology,
           iPhone and Android app development, and object
           technology education.</p>

        <h1>Clients</h1>
        <p class = "special"> The company's clients include many
           <em>Fortune 1000 companies</em>, government agencies,
           branches of the military and business organizations.</p>
     </body>
  </html>
```


Embedded Style Sheet

file:///C:/books/2011/IW3HTP5/examples/ch04CSS/fig04_03/embedded.html

Deitel & Associates, Inc.

Deitel & Associates, Inc. is an authoring and corporate training organization specializing in programming languages, Internet and web technology, iPhone and Android app development, and object technology education.

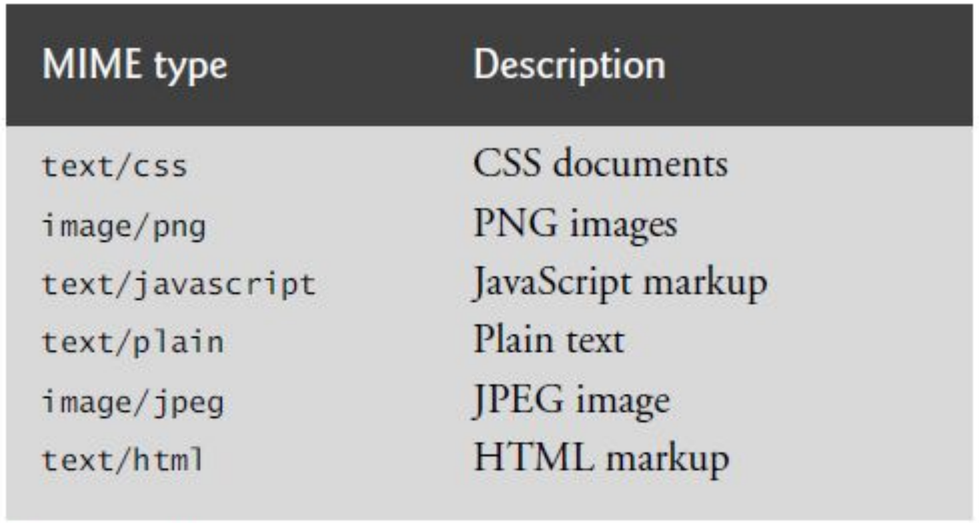
Clients

The company's clients include many ***Fortune 1000 companies***, government agencies, branches of the military and business organizations.

Fig. 4.3 | Embedded style sheet. (Part 2 of 2.)

The style Element and MIME Types

- The style element defines the *embedded style sheet*.
- Styles placed in the head apply to matching elements wherever they appear in the body.
- The style element's type attribute specifies the **MIME (Multipurpose Internet Mail Extensions) type** that describes the style element's content.
- Figure 4.4 lists some common MIME types used.



MIME type	Description
`text/css`	CSS documents
`image/png`	PNG images
`text/javascript`	JavaScript markup
`text/plain`	Plain text
`image/jpeg`	JPEG image
`text/html`	HTML markup

Fig. 4.4 | A few common MIME types.

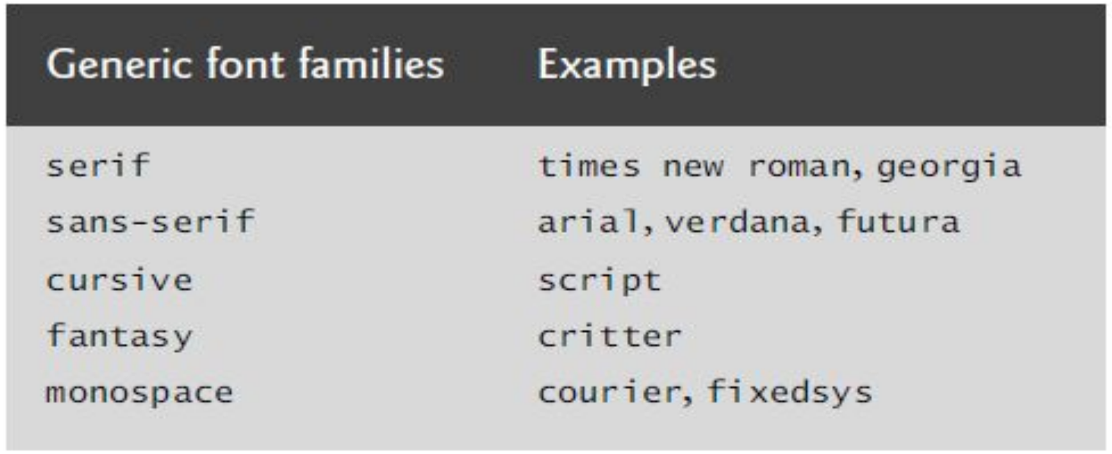
- first rule begins with the selector em, which selects all **em elements** in the document.
- An **em element** indicates that its contents should be *emphasized*.
- Browsers usually render em elements in an *italic* font.
- Each rule’s body is enclosed in *curly braces* ({ and }).
- The property name is followed by a colon (:) and the property value. Multiple properties are separated by semicolons (;).
- The **font-weight** property possible values are bold, normal (the default), bolder (bolder than bold text) and lighter (lighter than normal text).

Style Classes

- Line 17 declares a selector for a **style class** named special.
- Style-class declarations are preceded by a period (.).

font-family Property

- The **font-family** property (line 14) specifies the name of the font to use.
- CSS allows you to specify a comma-separated list of fonts to use for a particular style.



Generic font families	Examples
`serif`	`times new roman, georgia`
`sans-serif`	`arial, verdana, futura`
`cursive`	`script`
`fantasy`	`critter`
`monospace`	`courier, fixedsys`

Fig. 4.5 | Generic font families.

font-size Property

- Property **font-size** (line 15) specifies a 12-point font.
- Generally, *relative font-size values are preferred over points, because an author does not know the specific measurements of each client's display*.
- Relative values permit more flexible viewing of web pages.
- A user may view a web page on a handheld device with a small screen.

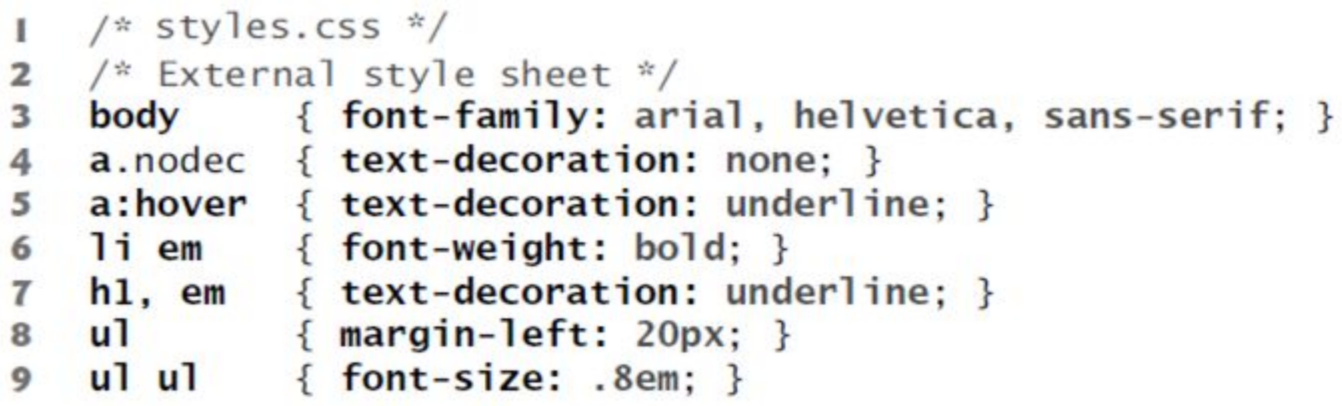
Applying a Style Class

- Line 22 uses the HTML5 attribute **class** in an h1 element to apply a style class.
- When the browser renders the h1 element, the text appears on screen with the properties of both an h1 element and the .special style class applied.

CSS – Cascading Style Sheets and Selectors

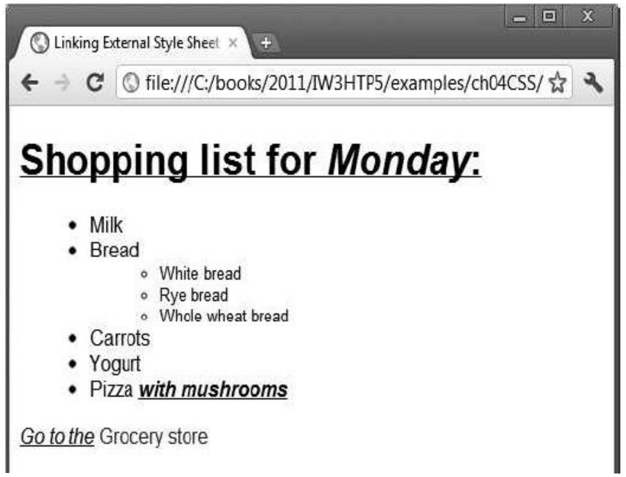
External Style Sheets

- With **external style sheets** (i.e., *separate* documents that contain only CSS rules), you can provide a *uniform look and feel* to all pages in the website (or to a portion of it).
- You can also *reuse* the same external style sheet across multiple websites.
- When changes to the styles are required, you need to modify only a single CSS file to make style changes across *all* the pages that use those styles.
- Linking done using <link rel=“stylesheet” type=“text/css” href=“styles.css” />



```
/* styles.css */
/* External style sheet */
body      { font-family: arial, helvetica, sans-serif; }
a.nodec   { text-decoration: none; }
a:hover   { text-decoration: underline; }
li em     { font-weight: bold; }
h1, em    { text-decoration: underline; }
ul        { margin-left: 20px; }
ul ul     { font-size: .8em; }
```

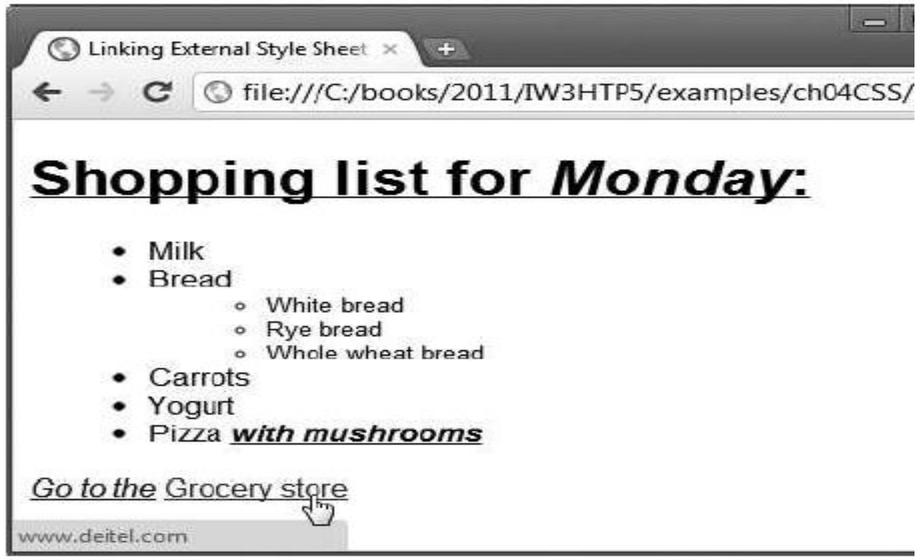
```html
<!DOCTYPE html>

<!-- Fig. 4.8: external.html -->
<!-- Linking an external style sheet. -->
<html>
   <head>
      <meta charset = "utf-8">
      <title>Linking External Style Sheets</title>
      <link rel = "stylesheet" type = "text/css"
         href = "styles.css">
   </head>
   <body>
      <h1>Shopping list for <em>Monday</em>:</h1>

      <ul>
         <li>Milk</li>
         <li>Bread
            <ul>
               <li>White bread</li>
               <li>Rye bread</li>
               <li>Whole wheat bread</li>
            </ul>
         </li>
         <li>Carrots</li>
         <li>Yogurt</li>
         <li>Pizza <em>with mushrooms</em></li>
      </ul>

      <p><em>Go to the</em>
         <a class = "nodec" href = "http://www.deitel.com">
            Grocery store</a>
      </p>
   </body>
</html>
```


Linking External Style Sheet
file:///C:/books/2011/IW3HTP5/examples/ch04CSS/

Shopping list for *Monday*:

- Milk
- Bread
  - White bread
  - Rye bread
  - Whole wheat bread
- Carrots
- Yogurt
- Pizza ***with mushrooms***

Go to the Grocery store


Linking External Style Sheet
file:///C:/books/2011/IW3HTP5/examples/ch04CSS/
www.deitel.com

Shopping list for *Monday*:

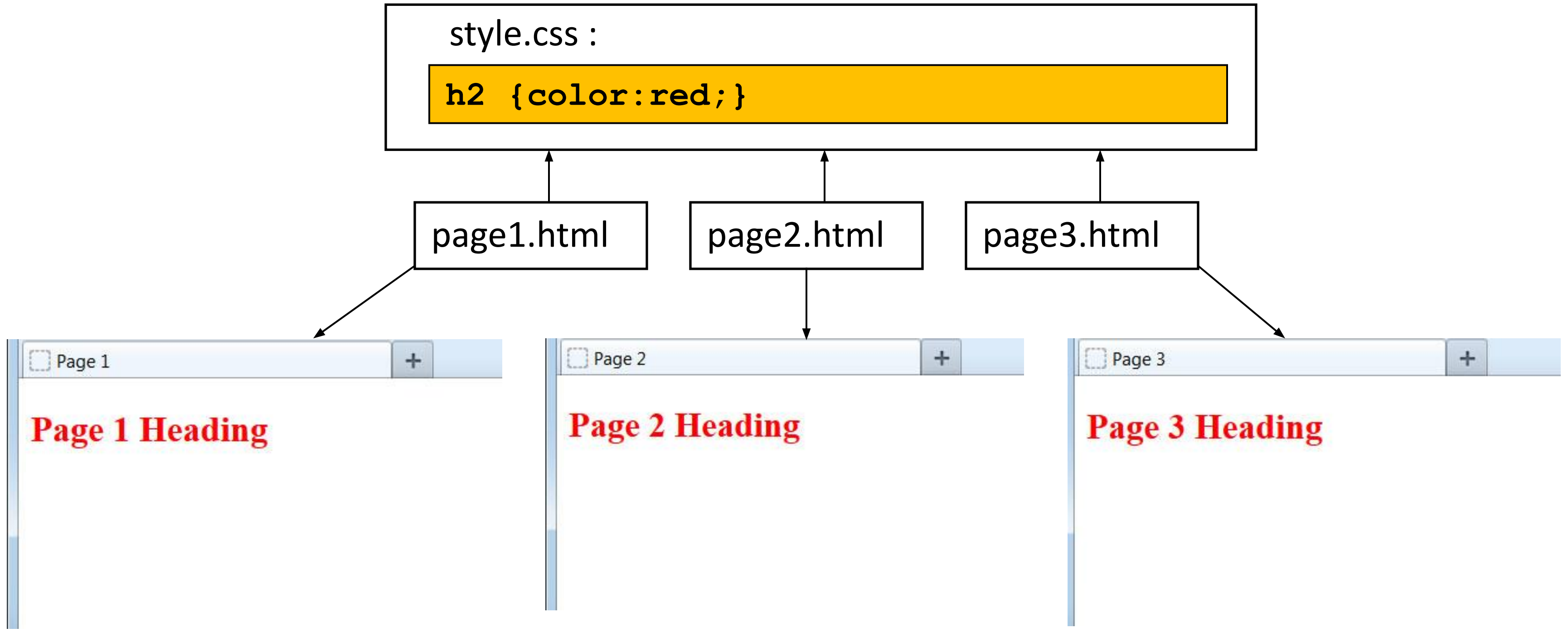
- Milk
- Bread
  - White bread
  - Rye bread
  - Whole wheat bread
- Carrots
- Yogurt
- Pizza ***with mushrooms***

Go to the Grocery store

CSS – Cascading Style Sheets and Selectors

Benefits of External Style Sheets

The real power of using an external style sheet is that multiple web pages on our site can link to the same style sheet:



Styles declared in an external style sheet will affect all matching elements on <u>all web pages that link to the style sheet</u>. By editing the external style sheet, we can make site-wide changes (even to hundreds of pages) instantly.

Internal vs. External Style Sheets

Internal Style Sheets

- are appropriate for very small sites, especially those that have just a single page.
- might also make sense when each page of a site needs to have a completely different look.

External Style Sheets:

- are better for multi-page websites that need to have a uniform look and feel to all pages.
- make for faster-loading sites (less redundant code).
- allow designers to make site-wide changes quickly and easily.

External style sheets create the furthest separation between content and presentation. Hence, the best option when creating a new site.

Conflicts

- Same formatting rules can be defined in all three locations at the same time.
- For example, a paragraph element could contain an inline style (color:red) but the internal style sheet (color:blue) and the external style sheet (color:green) give conflicting instructions to the web browser.
- Web browsers need a consistent way of "settling" this disagreement.

Cascading in CSS

- We use the term cascading because there is an established order of priority to resolve these formatting conflicts:
  1) Inline style (highest priority)
  2) Internal style sheet (second priority)
  3) External style sheet (third priority)
  4) Web browser default (only if not defined elsewhere)

For each XHTML element, the browser will check to see which inline styles are defined, then those styles in the internal style sheet, and finally those styles in the external sheet. For all conflicts, it will use this priority system to determine which format to display on the page.

In the prior example, the paragraph would display as red, because the inline style "outranks" all the others.

CSS – Cascading Style Sheets and Selectors

Cascading in CSS

- We use the term cascading because there is an established order of priority to resolve these formatting conflicts:
  1) Inline style (highest priority)
  2) Internal style sheet (second priority)
  3) External style sheet (third priority)
  4) Web browser default (only if not defined elsewhere)

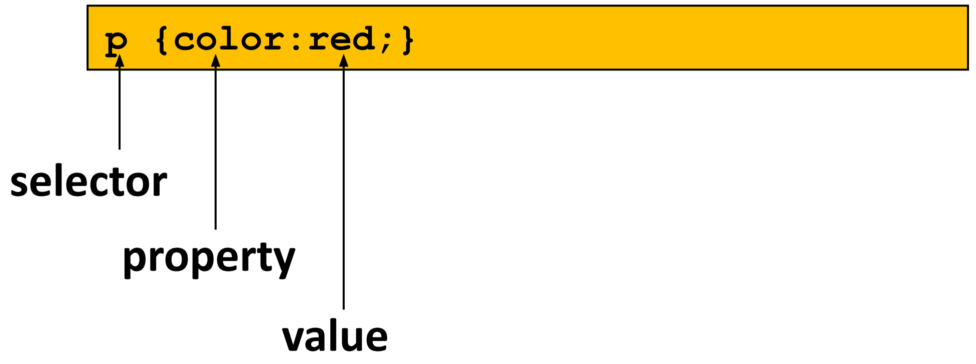
For each XHTML element, the browser will combine all the styles defined at different levels. For all conflicts, it will use the above priority system to determine which format to display on the page.

In the prior example, the paragraph would display as red, because the inline style "outranks" all the others.

CSS – Cascading Style Sheets and Selectors

CSS Terminologies

The correct syntax of a CSS declaration is: **selector {property:value;}**



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

A semicolon must be placed after each CSS declaration. Omitting this semicolon is the single most common mistake made by those learning CSS.

Setting Multiple Properties

We can define as many properties as we wish for a selector:

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

In this example, all text within paragraph elements will show in red italics that is centered on the page.

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

Just as with HTML, browsers ignore space characters in CSS code. Many designers take advantage of this fact by placing the opening and closing curly brackets on their own dedicated lines. Each of the property and value pairings are placed on their own indented line, with a space after the colon. This makes the code far easier to read.

CSS – Cascading Style Sheets and Selectors

How the browser handles styling?

- A web browser will process all CSS code it encounters, even if it is from all three methods.
- For example, an external style sheet could define the font of a heading, an internal style sheet could specify the font size of the heading, and an inline style could italicize the heading. All three would be applied.
- Sometimes a browser will receive conflicting instructions from the CSS code. For example, what if each of the above CSS sources specified a different color for the heading text?

Browsers need a consistent way of settling these formatting conflicts in a consistent fashion. That is where the "cascade" of cascading style sheets comes into effect.

CSS Selectors

Select the Elements to Apply CSS Rules

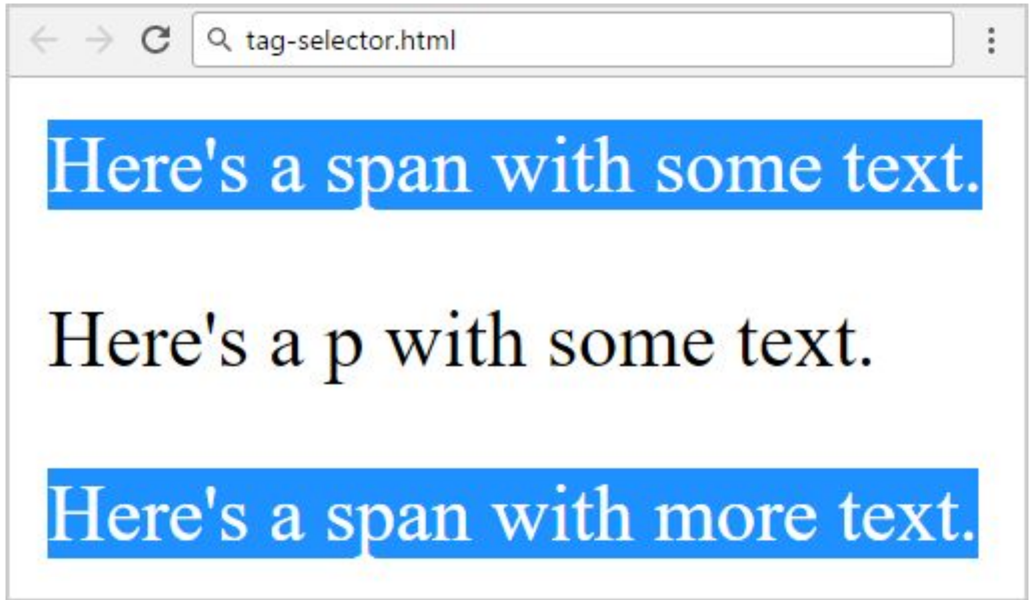
Selectors

- Primary Selectors
  - Element selector
  - Class Selectors
  - ID selectors

- Nested Selector
- Multiple Selector
- Pseudo-selector
- Attribute selector

Primary Selectors: Select by Tag – ELEMENT SELECTORS

```
<span>Here's a span with some text.</span>
  <p>Here's a p with some text.</p>
<span>Here's a span with more text.</span>
```


tag-selector.html

Here's a span with some text.

Here's a p with some text.

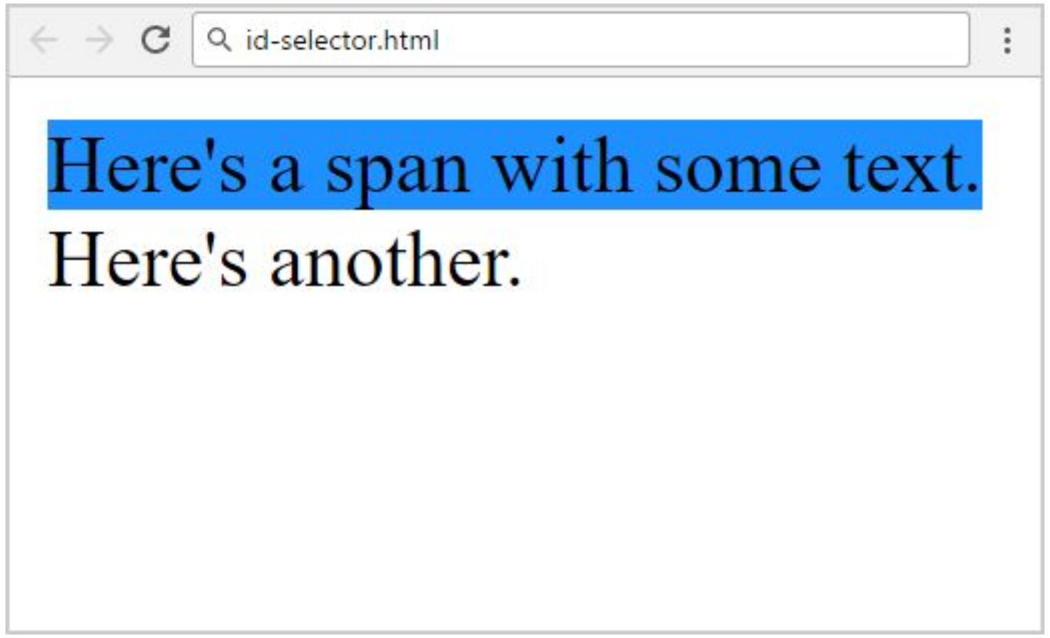
Here's a span with more text.

Select all <span> elements

```
span {
  background: DodgerBlue;
  color: #ffffff;
}
```

Primary Selectors: Select by ID

```
<span id="top">Here's a span with some text.</span>
<span>Here's another.</span>
```



id-selector.html

Here's a span with some text.
Here's another.

Select <span> with id="top"

```
span#top {
  background: DodgerBlue;
}
```

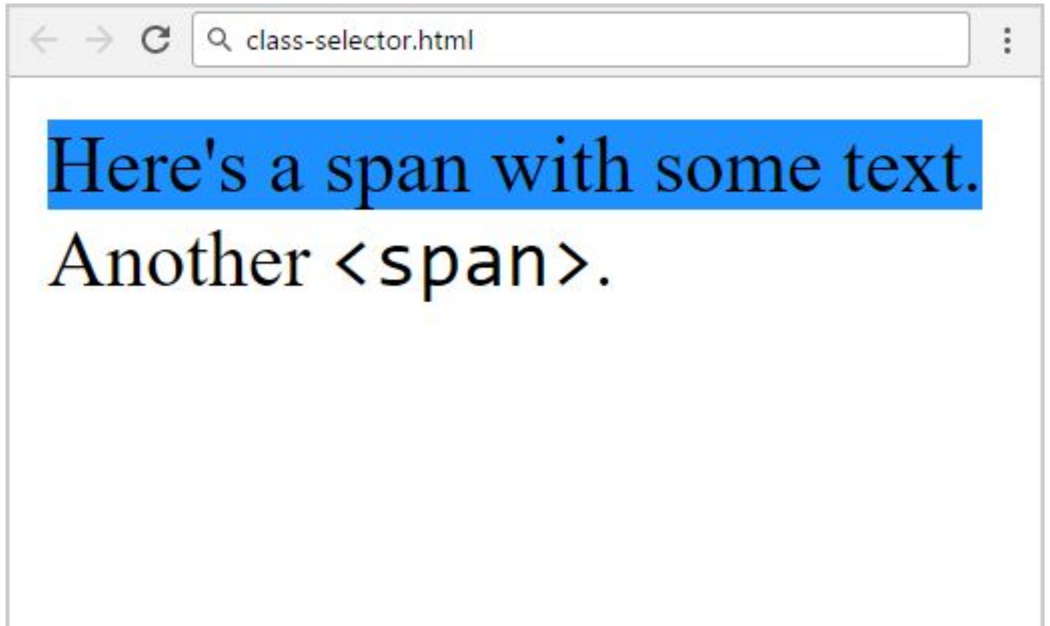
Primary Selectors: Select by Class

The *.class* selector selects elements with a specific class attribute.

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

Primary Selectors: Select by Class

```
<span class="sky">Here's a span with some text.</span>
<span>
  Another <span class="code">&lt;span&gt;</span>.
</span>
```



class-selector.html

Here's a span with some text.
Another <span>.

```
span.sky {
  background: DodgerBlue;
}
.code {
  font-family:
    Consolas;
}
```

<span> with class="sky"

Elements with class="code"

Pseudo-class selector

- **What are Pseudo-classes?**
- A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- Style an element when a user hovers over it
- Style visited and unvisited links differently
- Style an element when it gets focus.

Syntax:

selector:pseudo-class {   property:value; }

- The **order** of these pseudo-class elements is important.
- **The :Link, :Visited, :Focus, :Hover-** Order of pseudo-classes.

Pseudo-Element selector

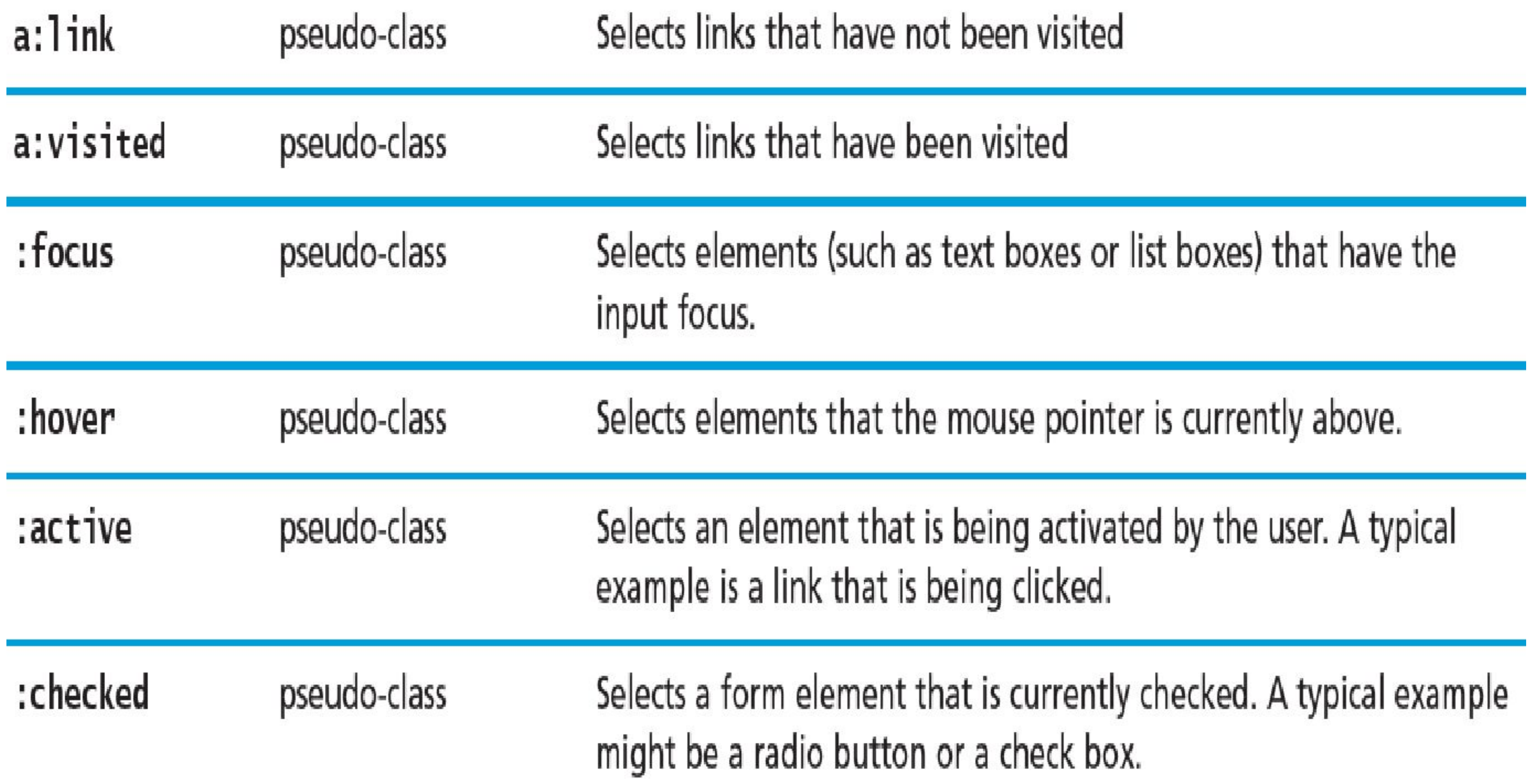
- A CSS pseudo-element is used to style specified parts of an element.

For example, it can be used to:

- Style the first letter, or line, of an element
- Insert content before, or after, the content of an element
- Syntax:

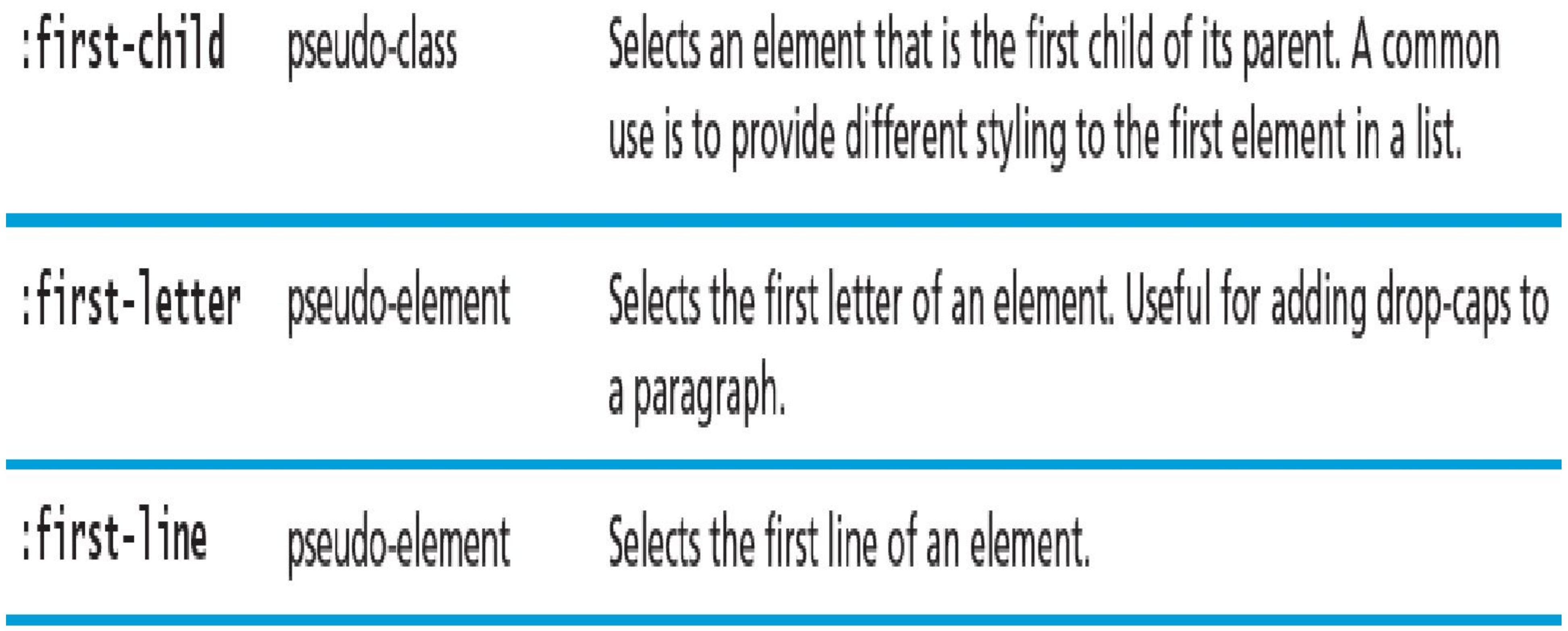
```
selector:pseudo-element {   property:value; }
```

Common Pseudo-element & Pseudo-class selectors



`a:link`	pseudo-class	Selects links that have not been visited
`a:visited`	pseudo-class	Selects links that have been visited
`:focus`	pseudo-class	Selects elements (such as text boxes or list boxes) that have the input focus.
`:hover`	pseudo-class	Selects elements that the mouse pointer is currently above.
`:active`	pseudo-class	Selects an element that is being activated by the user. A typical example is a link that is being clicked.
`:checked`	pseudo-class	Selects a form element that is currently checked. A typical example might be a radio button or a check box.

Common Pseudo-element & Pseudo-class selectors



:first-child	pseudo-class	Selects an element that is the first child of its parent. A common use is to provide different styling to the first element in a list.
:first-letter	pseudo-element	Selects the first letter of an element. Useful for adding drop-caps to a paragraph.
:first-line	pseudo-element	Selects the first line of an element.

Attribute Selectors

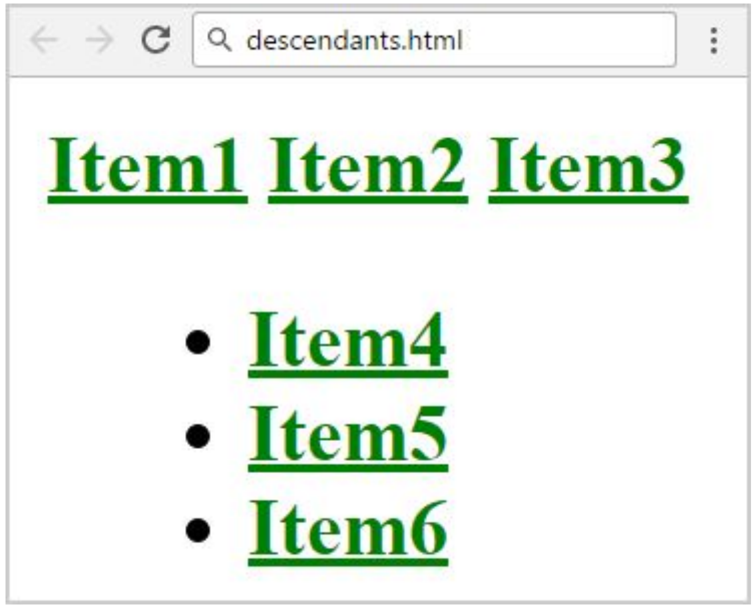
Selector	Matches	Example
[]	A specific attribute.	`[title]` Matches any element with a title attribute
[=]	A specific attribute with a specific value.	`a[title="posts from this country"]` Matches any `<a>` element whose title attribute is exactly "posts from this country"
[~=]	A specific attribute whose value matches at least one of the words in a space-delimited list of words.	`[title~="Countries"]` Matches any `title` attribute that contains the word "Countries"
[^=]	A specific attribute whose value begins with a specified value.	`a[href^="mailto"]` Matches any `<a>` element whose href attribute begins with "mailto"
[*=]	A specific attribute whose value contains a substring.	`img[src*="flag"]` Matches any `<img>` element whose src attribute contains somewhere within it the text "flag"
[$=]	A specific attribute whose value ends with a specified value.	`a[href$=".pdf"]` Matches any `<a>` element whose href attribute ends with the text ".pdf"

Nested Selectors: Descendant

```
div.items a {
  color: green;
  font-weight: bold;
}
```

`<a>` inside `<div class="items">`

```
<div class="items">
  <a href="#">Item1</a>
  <a href="#">Item2</a>
  <a href="#">Item3</a>
  <ul>
    <li><a href="#">Item4</a></li>
    <li><a href="#">Item5</a></li>
    <li><a href="#">Item6</a></li>
  </ul>
</div>
```



Item1 Item2 Item3

- Item4
- Item5
- Item6

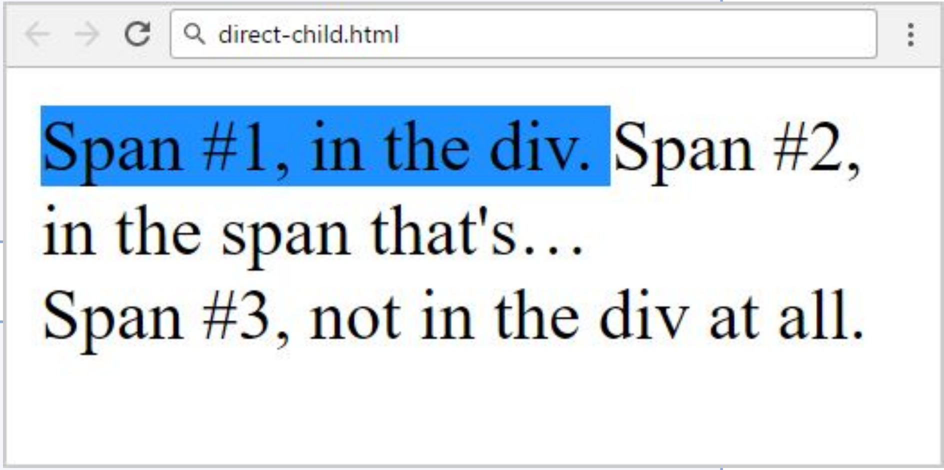
Nested Selectors: Direct Child

```css
div > span {
  background: DodgerBlue;
}
span { background: #fff; }
```

<span> directly contained in a <div>

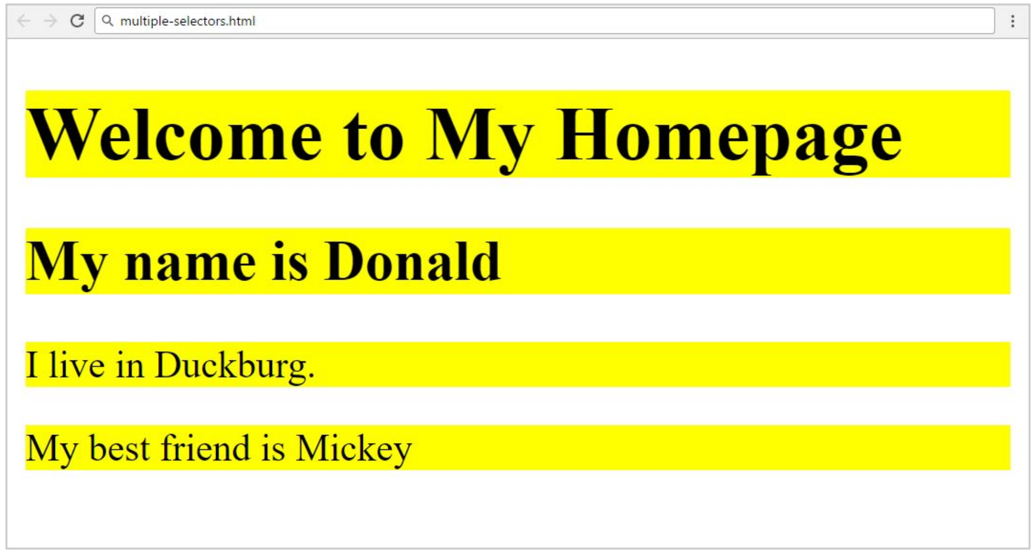
```html
<div>
  <span>Span #1, in the div.
    <span>Span #2, in the span that's…</span>
  </span>
</div>
<span>Span #3, not in the div at all.</span>
```

Direct child of <div>


direct-child.html

Span #1, in the div. Span #2, in the span that's…
Span #3, not in the div at all.

Multiple Selectors (Element, Element)



Welcome to My Homepage

My name is Donald

I live in Duckburg.

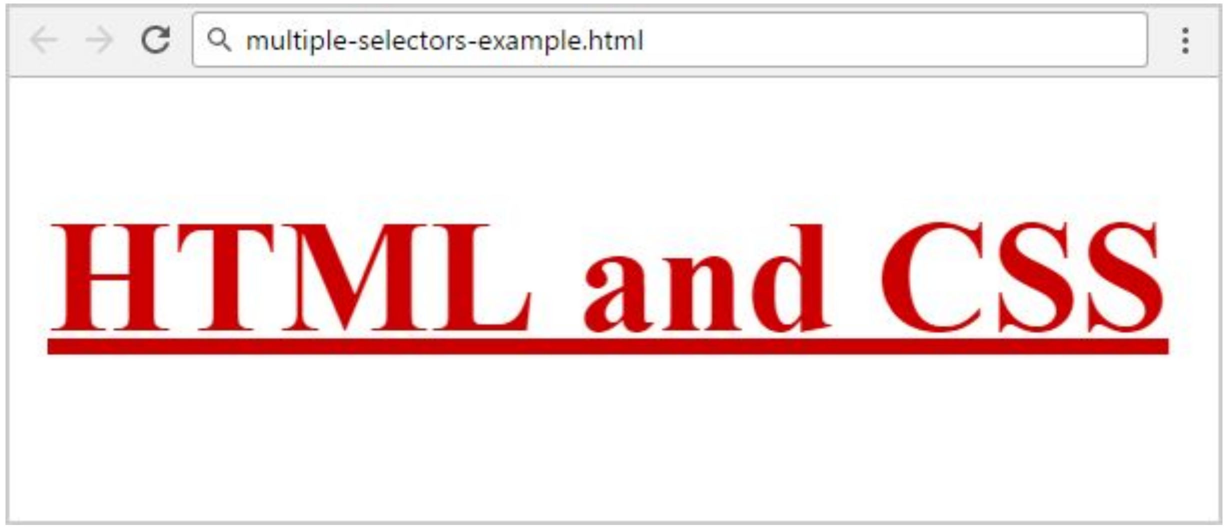
My best friend is Mickey

```
<h1>Welcome…</h1>
<h2>My name is…</h2>
<p>I live in Duckburg.</p>
<p>My best friend is…</p>
```

```
h1, h2, p {
  background: yellow;
}
```

Combining Multiple Selectors

```
h1#header.intro {
  text-decoration:
    underline;
  color: #C00;
}
```


multiple-selectors-example.html

HTML and CSS

```
<h1 id="header" class="intro">HTML and CSS</h1>
```

Pseudo Selectors

Relative to Element Content or State

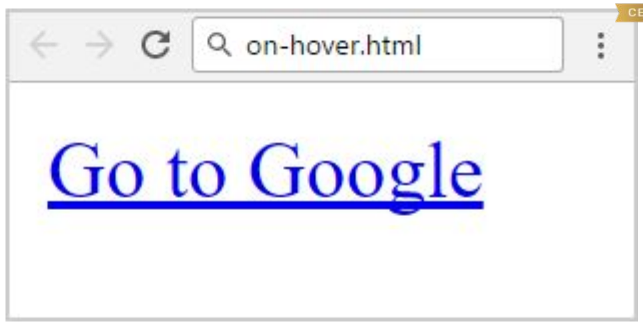
```
html, body, div { padding: 0; margin: 0; font-size: 100%; }

/* Start of "Micro clearfix" */

.cf { zoom: 1; }
.cf:before,
.cf:after { content: ""; display: table; }
.cf:after { clear: both; }
```

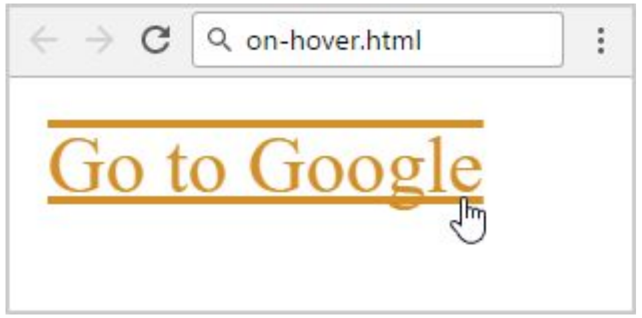
```
/head>
body>
   <h1>Micro clearfix demo</h1>
   <div class="container cf">
       <div class="section">Float</div>
       <div class="section">Float</div>
   </div>
```

Pseudo Selector (:hover)

```
<a href="#">Go to Google</a>
```

```
a:hover {
   color: #D48F29;
   text-decoration:
      underline overline;
}
```

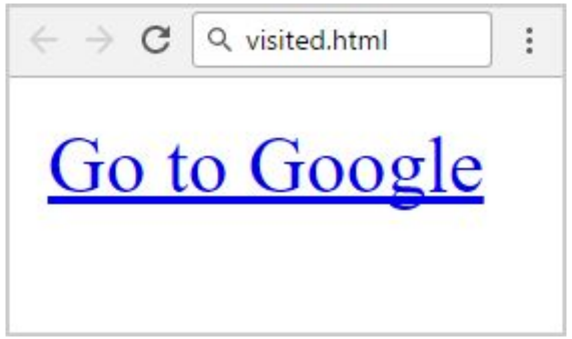


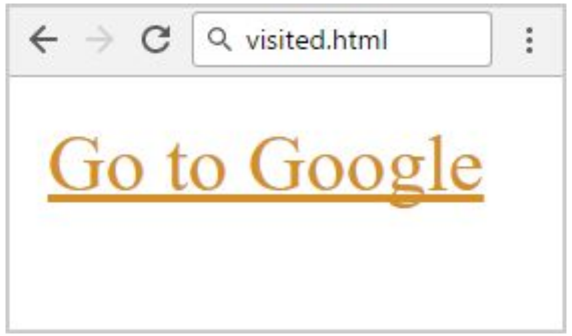


Pseudo Selector (:visited)

```
<a href="#">Go to Google</a>
```

```
a:visited {
  color:#D48F29;
}
```

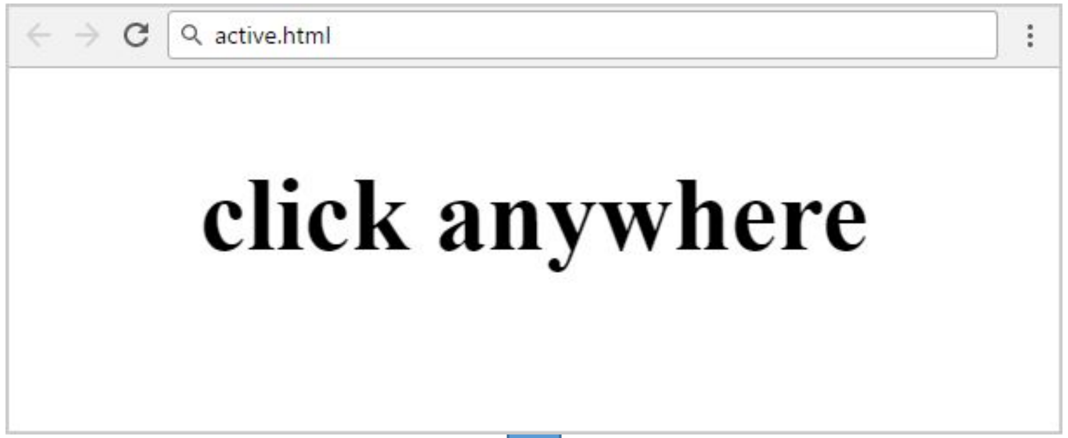




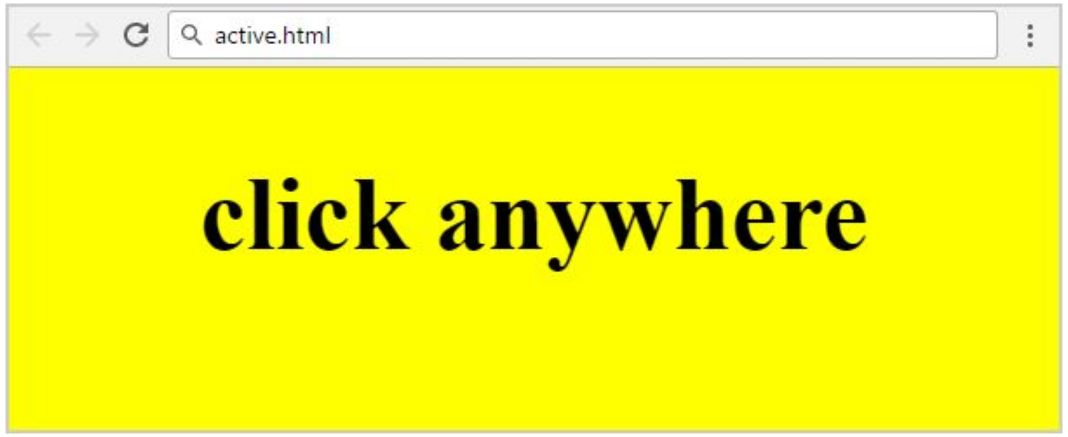
Pseudo Selector (:active)

```
<h2>click anywhere</h2>
```

```
html:active {
  background: yellow;
}
```

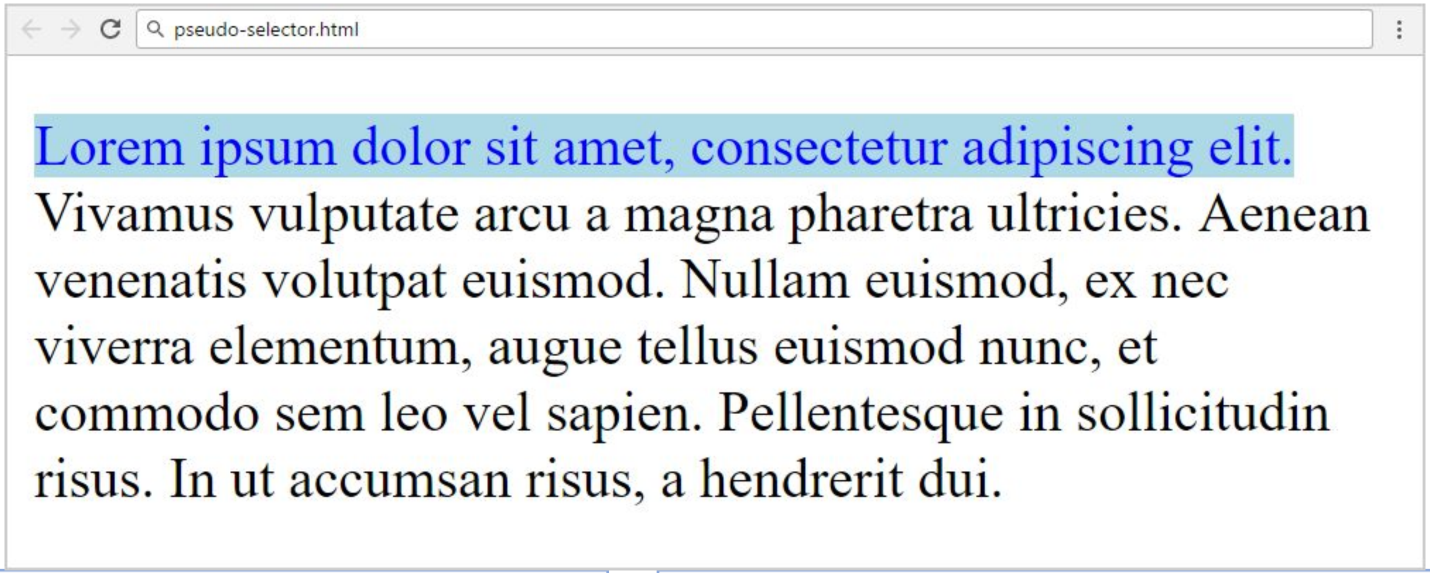

active.html

click anywhere


active.html

click anywhere

Pseudo Selector (:first-line)


pseudo-selector.html

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus vulputate arcu a magna pharetra ultricies. Aenean venenatis volutpat euismod. Nullam euismod, ex nec viverra elementum, augue tellus euismod nunc, et commodo sem leo vel sapien. Pellentesque in sollicitudin risus. In ut accumsan risus, a hendrerit dui.

```
<p>Lorem ipsum dolor
sit amet, consectetur
adipiscing elit.
</p>
```

```
p:first-line {
  color:blue;
  background-color: lightblue;
}
```

PES UNIVERSITY
CELEBRATING 50 YEARS

THANK YOU

Prof.Vinay Joshi

Department of Computer Science and Engineering