

CGS 3175: Internet Applications Fall 2007

Cascading Style Sheets – Page Layout

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
<http://www.cs.ucf.edu/courses/cgs3175/fall2007>

School of Electrical Engineering and Computer Science
University of Central Florida



Things to Try Yourself

10. Visit the following sites and determine what layout design they are using: ice, jello, or liquid:

- <http://www.nps.gov> (ice)
- <http://www.census.gov> (jello)
- <http://www.officedepot.com> (jello)
- <http://www.velonews.com> (jello)
- <http://wwwcfnews13.com> (ice)

11. Use one of the accessibility validators listed on the previous page and check your About Me page on Pegasus for accessibility issues.



CSS Page Layout Overview

- As we saw in the introductory notes, CSS is used to separate content from presentation in Web documents.
- CSS can also be used to separate the presentation (fonts, colors, and so on) from the page structure.
- Using CSS results in smaller Web page documents (XHTML files), which saves on bandwidth and eases Web site maintenance.
- Recall that CSS Level 1 (CSS1) was introduced by the W3C in 1996. CSS Level 2 (CSS2) was released in 1998.
- CSS2 builds on CSS1 and introduces new properties needed for positioning, commonly referred to as CSS-P.



CSS Page Layout Overview

- Even though CSS2 was introduced in the last millennium, commercial Web sites have only recently begun to utilize its features. We saw in the last section of notes that the National Security Agency utilizes CSS for their page layout, so to does TruGreen (lawn care at <http://trugreen.com>) and Wired at (www.wired.com – look at the source code for their home page – which DTD are they using?).
- The primary reason for this delay in acceptance of the CSS2 standards has been the lack of browser support for the W3C recommendations. Only very recently have the most current versions of commonly used browsers begun to support CSS2 reliably and consistently.
- Even so, you may expect to encounter differences in rendering when displaying pages coded using CSS-P in various browsers.
- By learning to develop Web pages using CSS-P, you will be on the very leading edge as web developers.
- As newer browsers are introduced the number of Web visitors who use older browsers is declining all the time. However, be aware that for specific target audiences this may not yet be true and may take longer to occur.



Advantages of Using CSS For Page Layout

- When CSS is used to configure page layout in addition to formatting, all the following advantages of using CSS for formatting are enhanced:
 - **Greater typography control.** This includes font size, line spacing, letter spacing, indents, margins, and element positioning without using tables.
 - **Style is separate from structure.** The formatting and page layout can be configured and stored separately from the content section of the Web page document. When the styles are modified, the XHTML remains intact. This means that if your client decides to change something as small as the background color or as potentially huge as the page layout, you may need to only change one file that contains the styles, instead of each Web page document.
 - **Potentially smaller documents.** Since both the formatting and page layout are separate from the document, the actual XHTML documents should be smaller.
 - **Easier site maintenance.** Again, if the styles or page layout need to be changed it may be possible to complete the modifications by changing only a single file – the style sheet.



Advantages of Using CSS For Page Layout (cont.)

- **Increased page layout control.** CSS used in conjunction with modern standards-compliant browsers provides a variety of positioning options (even down to the pixel level) along with an ability to overlap elements. This gives the Web developer more control over the layout compared to the traditional use of tables.
- **Increased accessibility.** Pages designed using tables for layout are easy to view with a traditional browser but can be very tedious when using a screen reader or other assistive technology. By reserving the use of tables for their intended purpose of organizing tabular information and using CSS for page layout – the pages become more accessible.
- **Ability to define styles for multiple media types.** Since presentation is separated from content, CSS can be used to set a separate style for printing, or possible use of a screen reader.
- **Support of the Semantic Web.** This is the W3C's vision of what the future Internet. In the semantic web information will have well-defined meaning that will better enable computers and humans to work in cooperation. A big step in this direction is using XHTML syntax and using CSS-P to separate styles from structure.



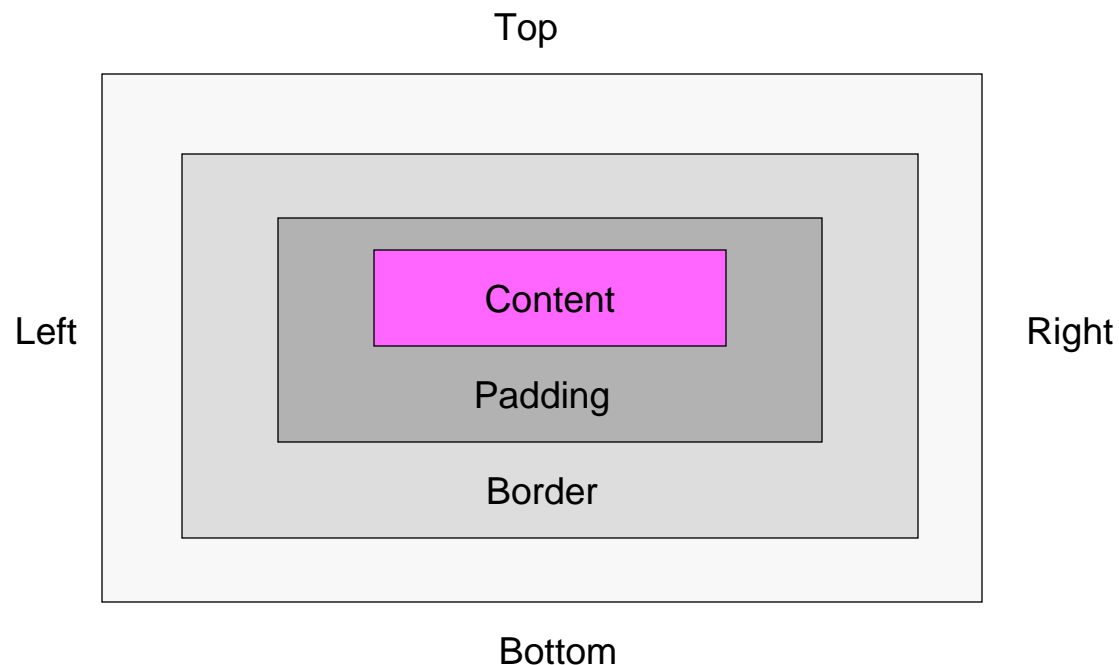
Disadvantages of Using CSS For Page Layout

- There are only a couple of disadvantages of using CSS-P:
 - **Lack of support by older browsers.** Older browsers do not support CSS-P, so for example, if you are developing an intranet site for a company that has standardized on Netscape 4.7 for their desktop, none of the advantages of CSS-P would be realized and it would be better to design the site using tables for the page layout. Note that as time passes, this will become less and less of a disadvantage to the use of CSS-P.
 - **Productivity drop.** Developers who are already proficient at developing Web pages using tables for the layout will see productivity drop (temporarily) as they learn about CSS-P techniques and properties.



The CSS Box Model

- The CSS Box Model is a crucial building block of CSS-P.
- Each element in a document is considered to be a rectangular box, as shown below. This box consists of a content area surrounded by padding, a border, and margins.

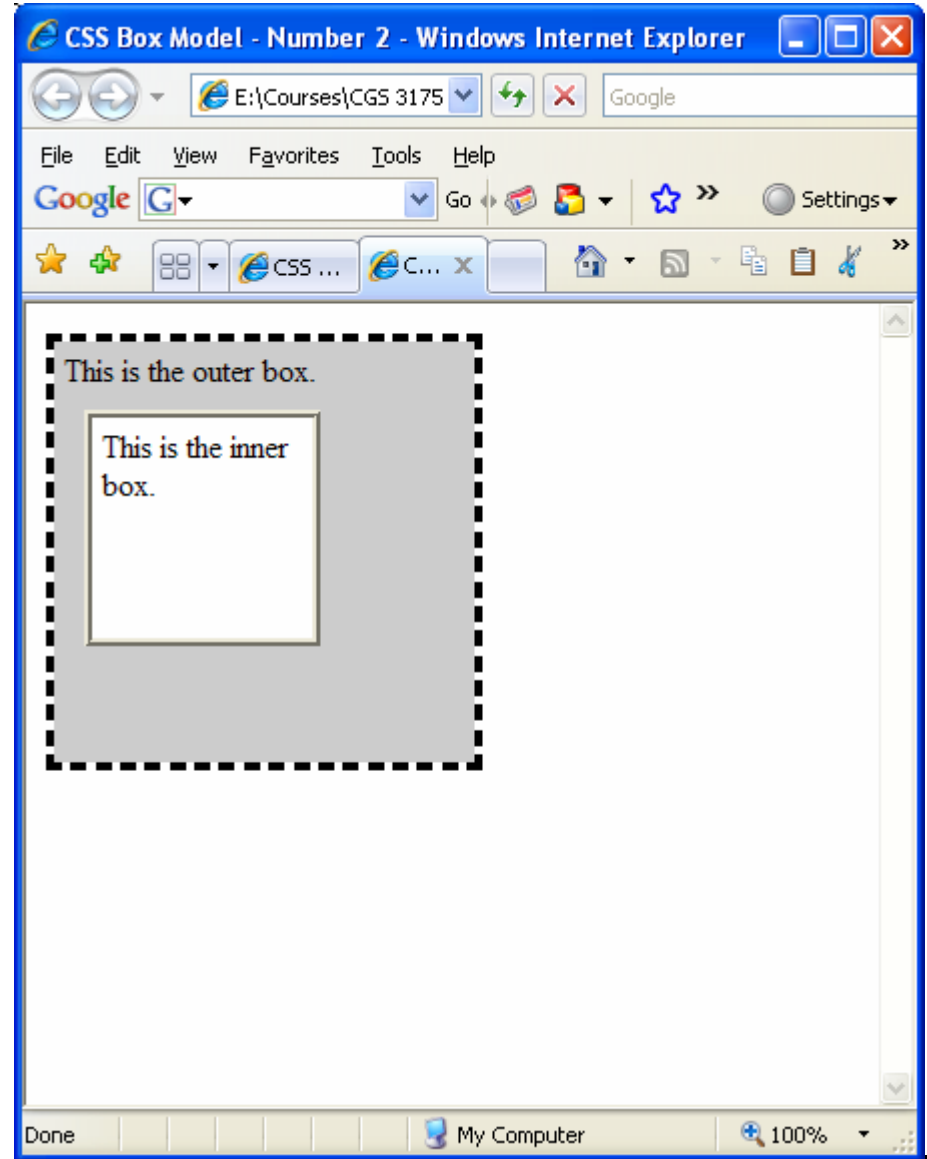
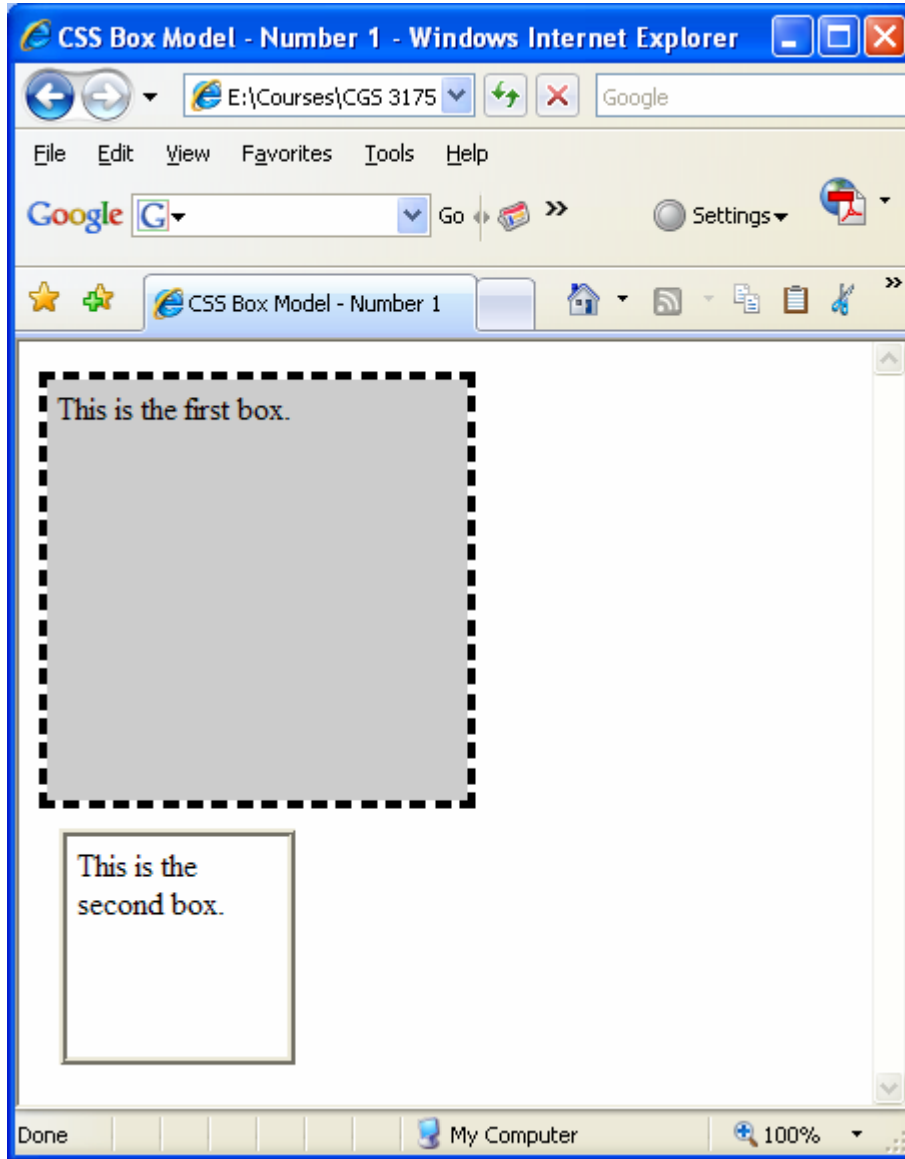


The CSS Box Model

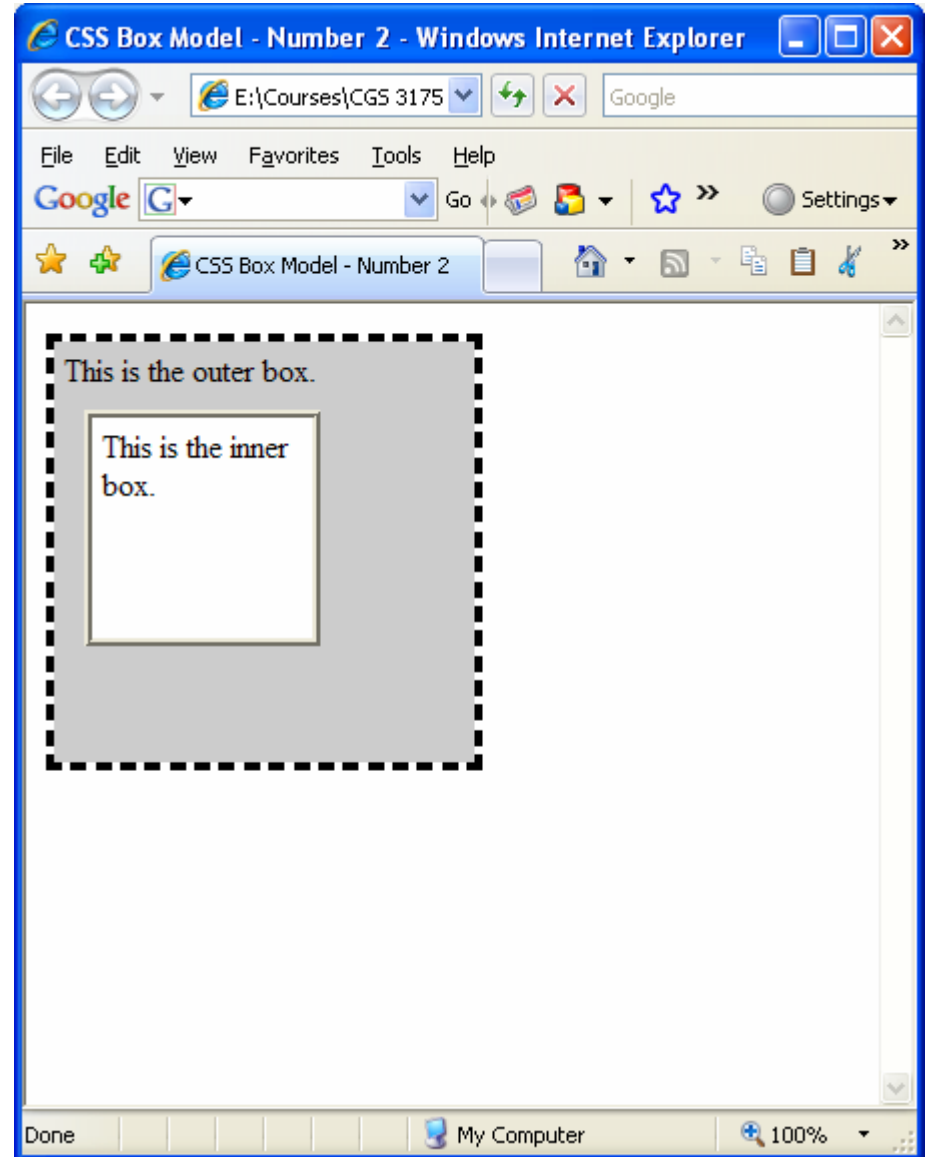
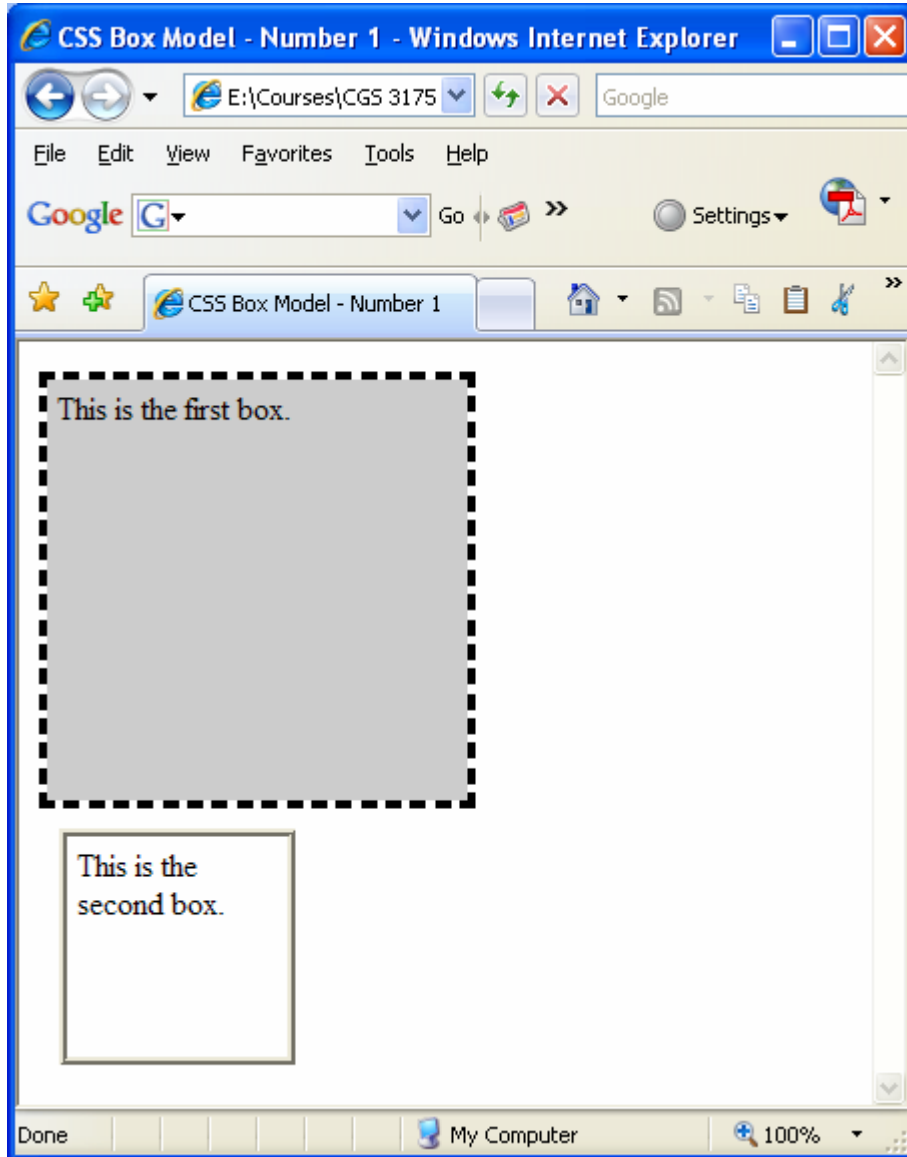
- **Content.** The content area can consist of a combination of text and Web page elements such as images, paragraphs, headings, lists, and so on. The screen shots on the next page show `<div>` elements containing text content. The **visible width** of the element on a Web page is the total of the content width, the padding width, and the border width.
- **Padding.** Padding is an area that is between the content and the border. The default padding value is 0. When configuring the background of an element, the background is applied to both the padding and the content area.
- **Border.** The border area is between the padding and the margin. The default border has a value of 0 and does not display.
- **Margin.** The margin determines the empty space between the element and any adjacent elements. The margin is always transparent. The solid line shown on the previous page that contains the margin area does not display on a Web page. Keep in mind that browsers often have default margin values set of the Web page document and for certain elements such as paragraphs, headings, forms, and so on. Use the margin property to override the default browser values.



The CSS Box Model



The CSS Box Model



Box Model – Example 1

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>CSS Box Model - Number 1</title>
<style type="text/css">
  .div1 {width:200px;
        height:200px;
        background-color:#cccccc;
        border:dashed;
        padding:5px;
  }
  .div2 {width:100px;
        height:100px;
        background-color:#ffffff;
        border: ridge;
        padding:5px;
        margin:10px;
  }
</style>
</head>
<body>
<div class="div1">
  This is the first box.
</div>
<div class="div2">
  This is the second box.
</div>
</body>
</html>
```

Internal style sheet

Note that the first <div> element ends before the second <div> element begins.



Box Model – Example 2

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
<title>CSS Box Model - Number 2</title>
<style type="text/css">

```

Note that the style sheet is exactly the same as for the first example. The page layout is quite a bit different from the first page's layout, but exactly the same CSS was used.

Internal style sheet

Note that the first <div> element encompasses the second <div> element.



CSS Positioning Properties

- The box examples on the previous pages illustrate how normal flow causes the browser to render the elements in the order that they appear in the XHTML source code.
- When using CSS for page layout there are times when you will want to specify the location of an element on the page – either the **absolute** pixel location, the location **relative** to where the element would normally display, or **floating** on the page.
- There may even be time when you will want to modify the way an element displays or cause an element to appear partially or completely over another element.



Relative and Absolute Positioning

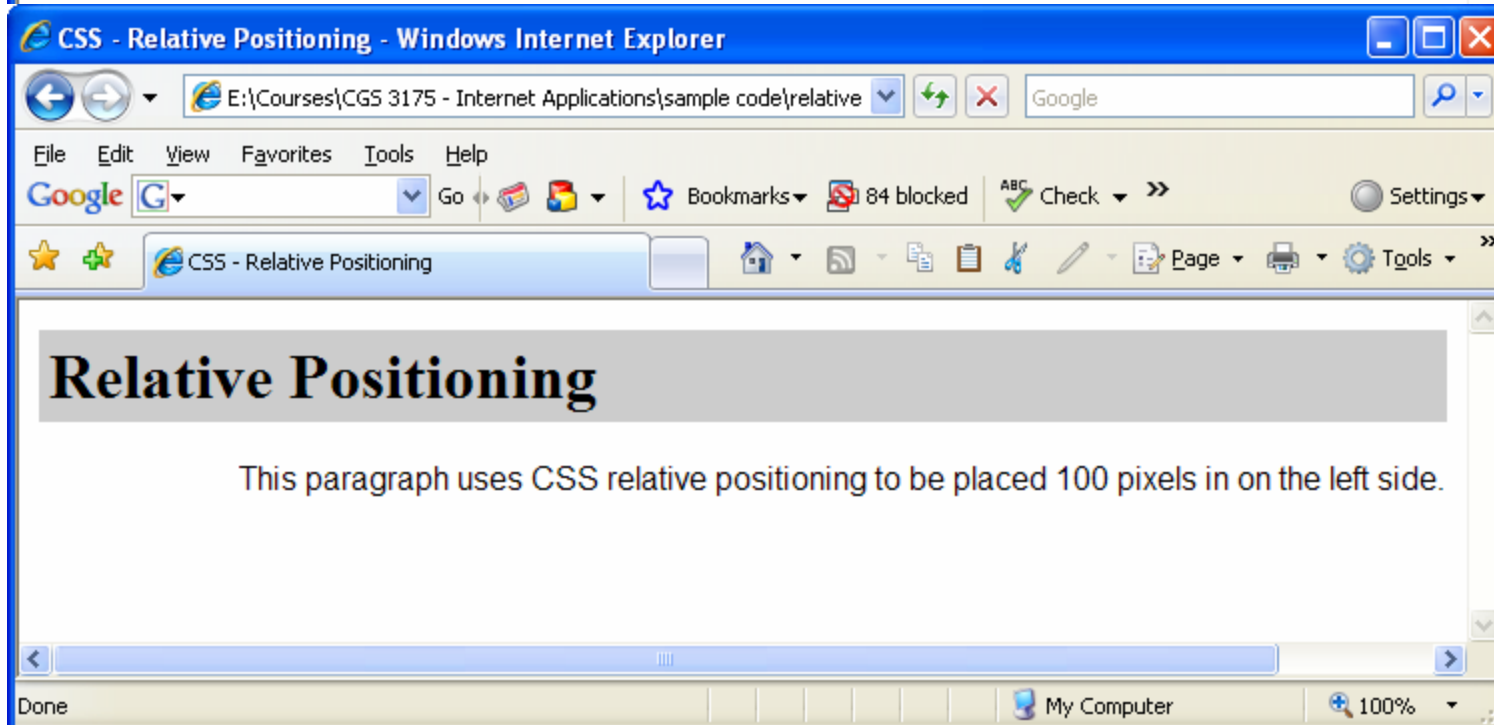
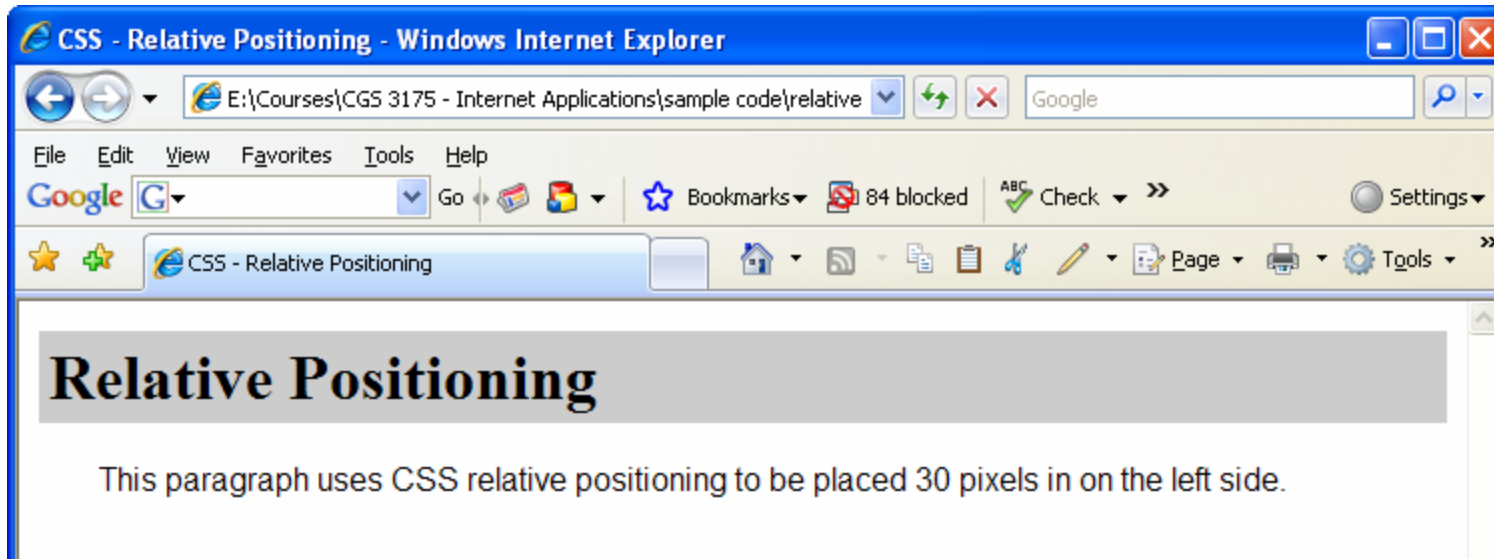
- **Relative positioning** is used to change the location of an element slightly, relative to where it would otherwise appear in normal flow.
- Use the `position:relative` property along with either a `left`, `right`, and/or `top` property.
 - The `left` property configures the position of the element in relation to the left side of the browser window.
 - The `right` property configures the position of the element in relation to the right side of the browser window.
 - The `top` property configures the position of the element in relation to the top of the document area in the browser window.



Relative Positioning

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>CSS - Relative Positioning</title>
<style type="text/css">
h1 {   background-color:#cccccc;
        padding:5px;
        color: #000000;
    }
#myContent {   position: relative;
                left:30px;
                font-family:Arial,sans-serif;
            }
</style>
</head>
<body>
<h1>Relative Positioning</h1>
<div id="myContent">
<p>This paragraph uses CSS relative positioning to be placed 30 pixels
in on the left side.</p>
</div>
</body>
</html>
```





Relative and Absolute Positioning

- **Absolute positioning** is used to specify the location of an element precisely in a browser window.
- Use the `position:absolute` property along with either a `left`, `right`, and/or `top` property need to configure the placement.
 - The `left` property configures the position of the element in relation to the left side of the browser window.
 - The `right` property configures the position of the element in relation to the right side of the browser window.
 - The `top` property configures the position of the element in relation to the top of the document area in the browser window.

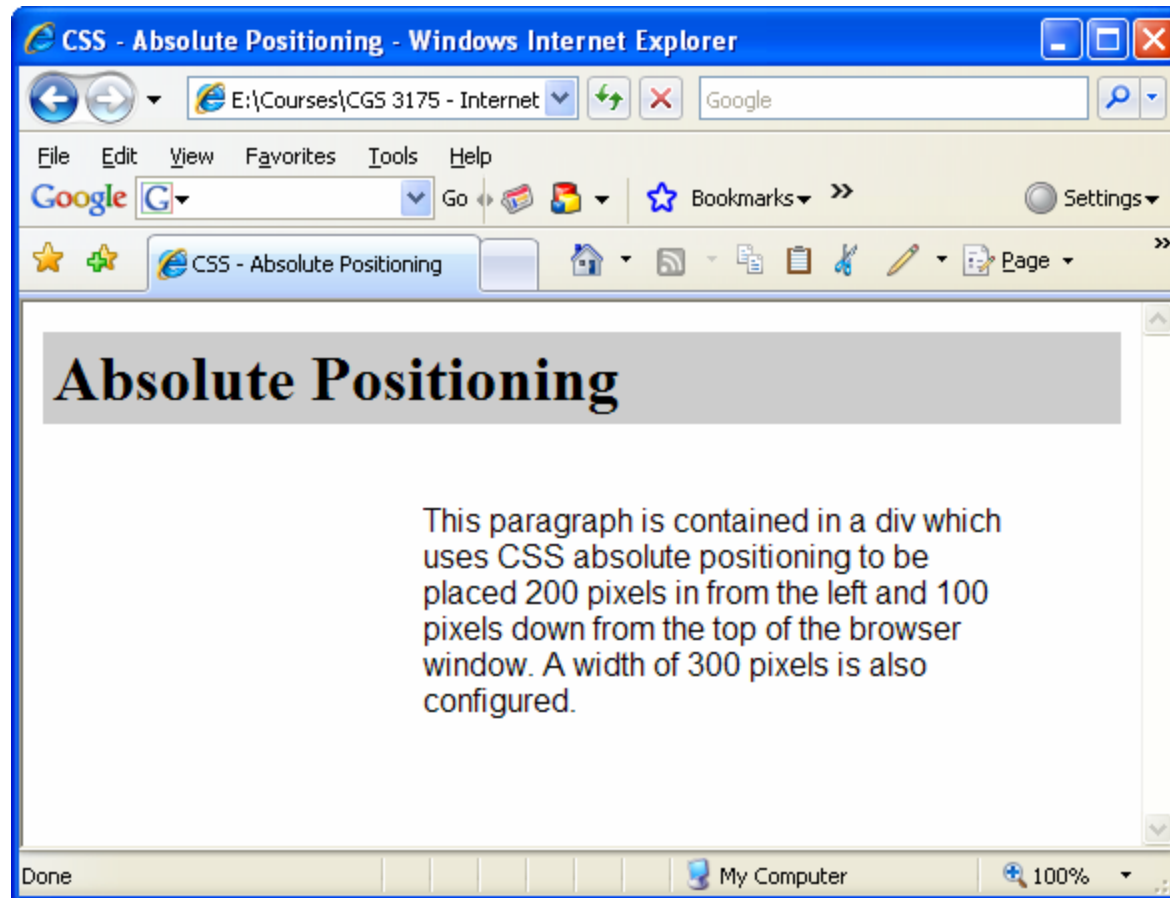


Absolute Positioning

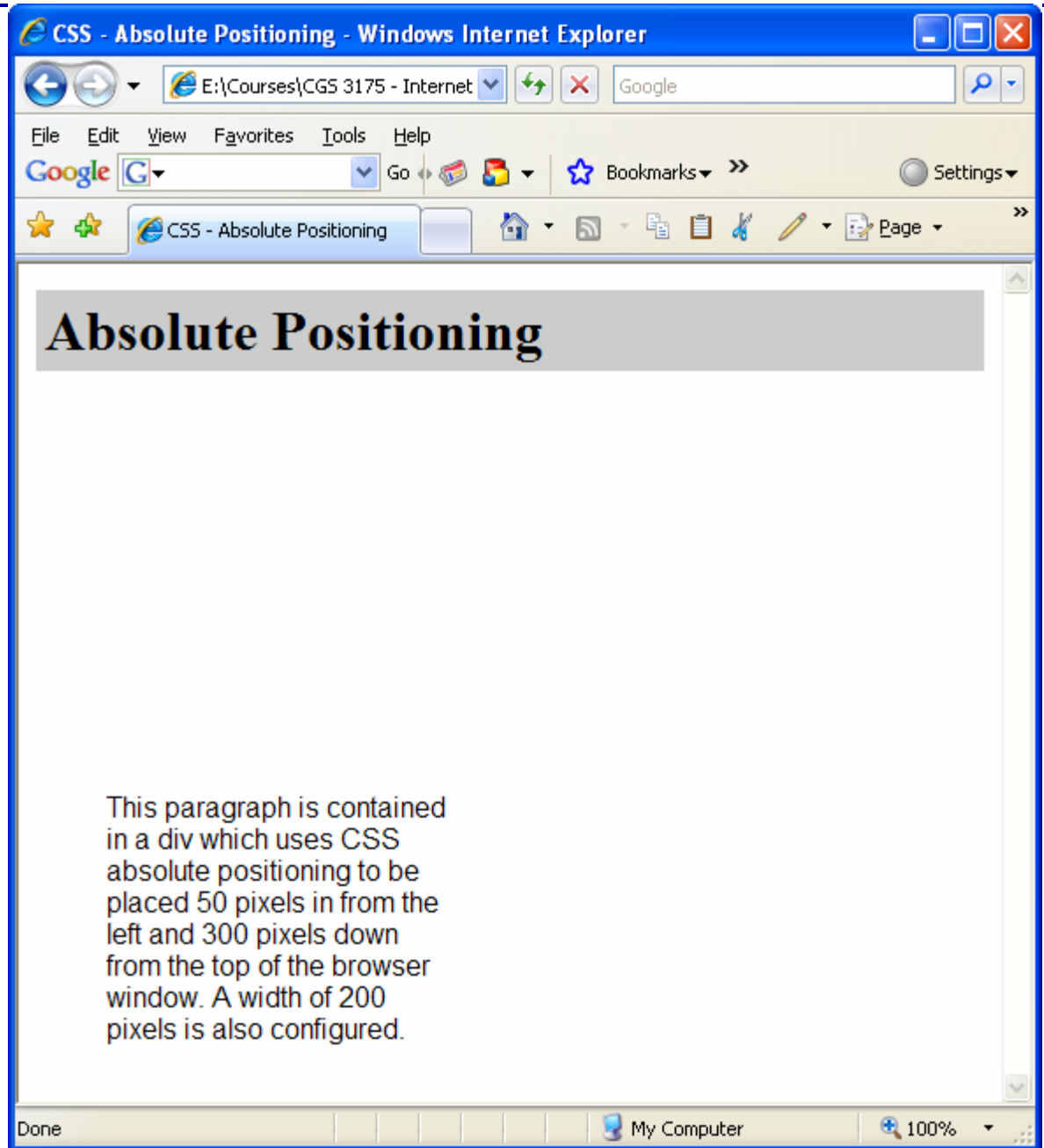
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>CSS - Absolute Positioning</title>
<style type="text/css">
h1 {   background-color:#cccccc;
        padding:5px;
        color: #000000;
    }
#content {position: absolute;
          left:200px;
          top:100px;
          font-family:Arial,sans-serif;
          width:300px;
    }
</style>
</head>
<body>
<h1>Absolute Positioning</h1>
<div id="content">
<p>This paragraph is contained in a div which uses CSS absolute
positioning to be placed 200 pixels in from the left and 100 pixels
down from the top of the browser window. A width of 300 pixels is also
configured.</p>
</div>
</body>
</html>
```



Absolute Positioning



Absolute Positioning



More On Absolute Positioning

- When working with absolute positioning it is important to be aware that elements not absolutely positioned will be rendered following normal flow by the browser. Elements that are absolutely positioned are rendered outside of normal flow.
- Consider the example shown on the next page.

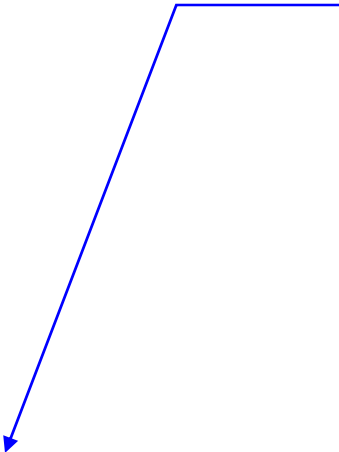


```

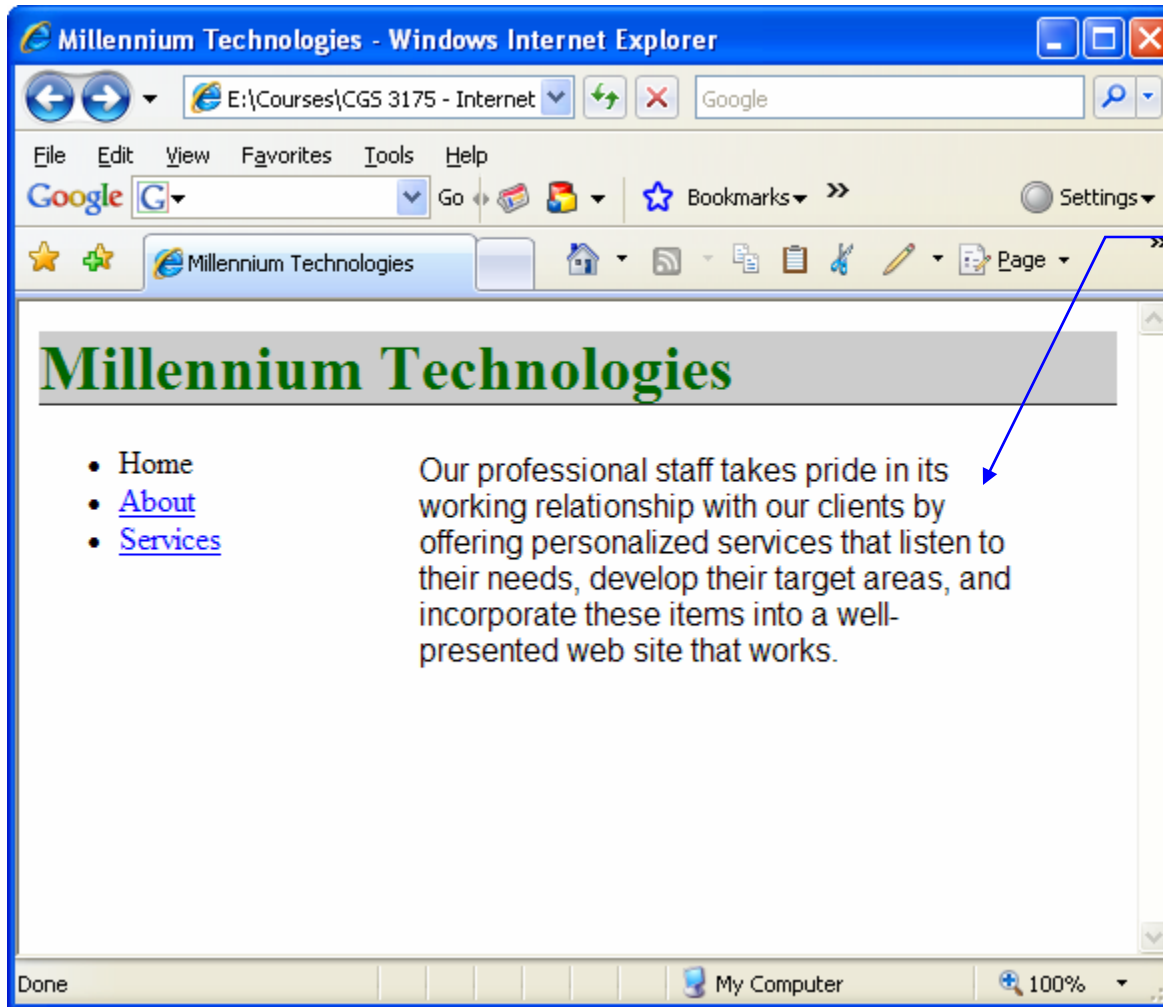
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Millennium Technologies</title>
<style type="text/css">
  h1 { border-bottom: 1px solid #333333;
        color: #006600;
        background-color:#cccccc
      }
  .content { position: absolute;
             left:200px;
             top:75px;
             font-family:Arial,sans-serif;
             width:300px;
           }
</style>
</head>
<body>
  <h1>Millennium Technologies</h1>
  <div class="content">
    <p>Our professional staff takes pride in its working relationship with our
      clients by offering personalized services that listen to their needs, develop
      their target areas, and incorporate these items into a well-presented web
      site that works. </p>
  </div>
  <ul>
    <li>Home</li>
    <li><a href="about.htm">About</a></li>
    <li><a href="services.htm">Services</a></li>
  </ul>
</body>
</html>

```

Will the paragraph display above the list or to the right of the list?



More On Absolute Positioning



The paragraph was set at an absolute position and thus rendered outside of the normal flow. So, the paragraph will appear to the right of the list. Even though the list was coded in the XHTML after the paragraph.



The `float` Property

- Elements that seem to float on the right or left side of either the browser window or another element are often configured using the `float` property.
- The browser renders these element using normal flow, and then shifts them as far as possible within their container (usually either the browser window or a `<div>`) to either the right or left. Other content will flow around the float.
- To stop this flow, use the `clear` property.



```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>CSS Float Property - Example 1</title>
<style type="text/css">
  h1 { background-color:#cccccc;
        padding:5px;
        color: #000000;
      }
  p { font-family:Arial,sans-serif;
      }
  #picture {float:right;
            margin: 0 0 5px 5px;
            border: solid;
          }
</style>
</head>
<body>
  <p></p>
  <h1>Floating an Image</h1>
  <p>The heading and paragraph follow normal flow. The racer pictured on the right is
  Axel Merckx. He is the son of world famous cyclist Eddy Merckx. This year Alex rode for
  the T-Mobile team in Germany.</p>
</body>
</html>

```

Style for the floating image.

 element with alternate text display.



The float Property

