

CIS 4004: Web Based Information Technology Spring 2011

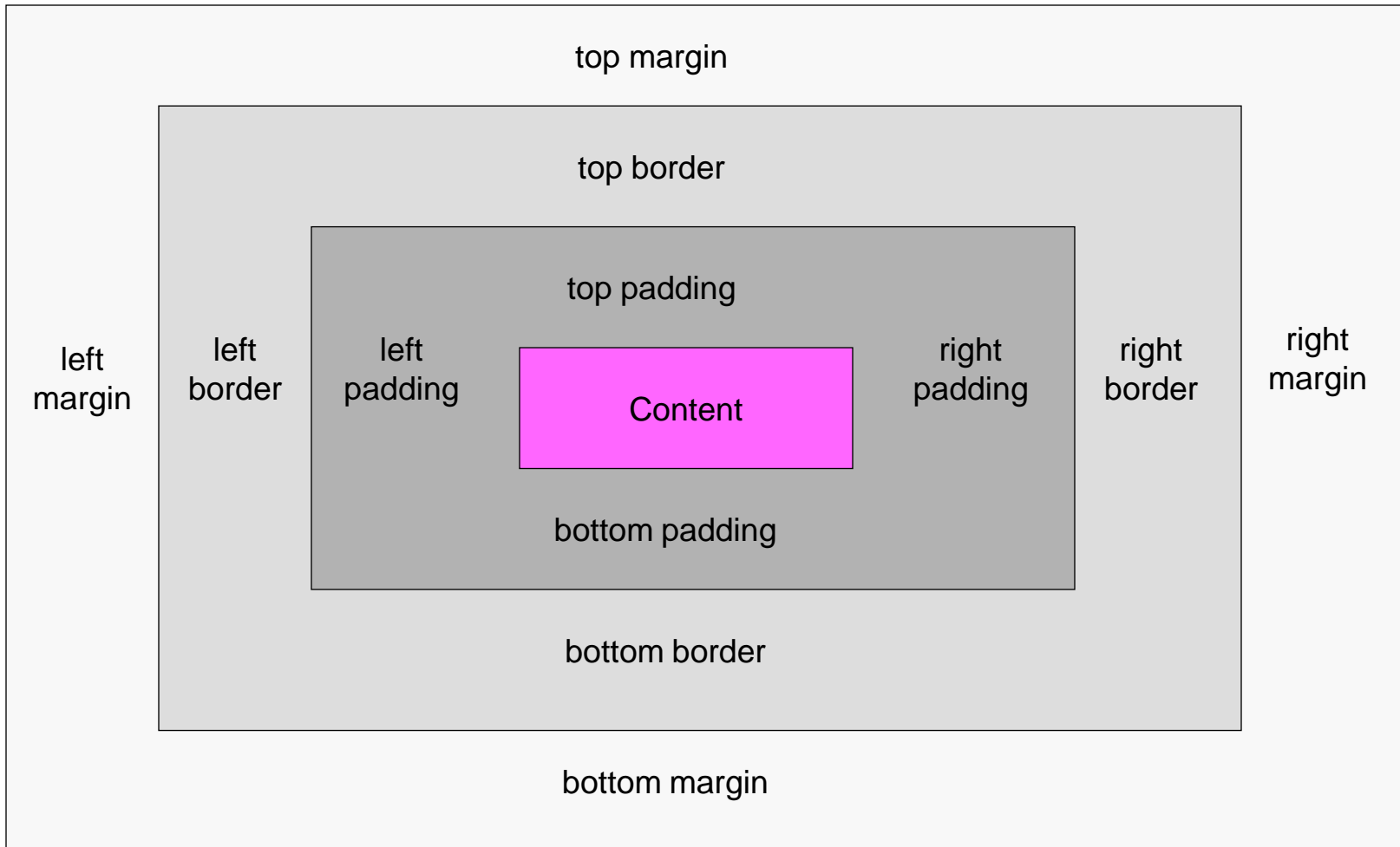
Cascading Style Sheets – Page Layout - Part 4

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
<http://www.cs.ucf.edu/courses/cis4004/spr2011>

Department of Electrical Engineering and Computer Science
University of Central Florida



The CSS Box Model



The `position` Property

- In the past two sections of notes, we've looked in detail at the box border, padding, and margins, as well as the float and clear properties.
- In this section of notes, we'll look more closely at the `position` property. The `position` property is at the heart of all CSS-based layouts. The `position` property determines the reference point for the positioning of each element box.
- There are four values for the `position` property: **`static`**, **`absolute`**, **`fixed`**, and **`relative`**.



The `position` Property

- We'll set up a running example demonstration XHTML/CSS to illustrate the `position` property.
- The basic XHTML is shown on the next page, with its rendering on the following page.
- Notice that the default `position` property for any element is `static`.
- In the running example, the third paragraph is a special paragraph (styled differently from the other paragraphs) so that we can see the difference in the various `position` property values.



```
C:\Courses\CIS 4004 - Web Based IT\Spring 2011\code\CSS-P\CSS-P - Part 4\static positioning demo.html - Notepad++
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
static positioning demo.html
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml">
5 <head>
6 <title>Static Positioning Demo</title>
7 <style type="text/css">
8 <!--
9     body {background-color:#FFC;}
10    p {border:1px solid #000;}
11    p#specialpara {color:red; background:#EEE;}
12    -->
13 </style>
14 </head>
15 <body>
16 <p>This is the first paragraph of the positioning demo. The objective is to
17 demonstrate the difference between the four position property values:
18 static, relative, absolute and fixed. The key to working with the position
19 property is to understand that every element is positioned with respect to
20 another element; which element that is can be changed by changing the value
21 of the position property.
22 </p>
23 <p>This is the second paragraph of the positioning demo. The objective is to
24 demonstrate the difference between the four position property values: static,
25 relative, absolute and fixed.
26 </p>
Hyper Text Mark nb char : 2126 nb line : 47 Ln:1 Col:1 Sel:0 UNIX ANSI INS
```



This is the first paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed. The key to working with the position property is to understand that every element is positioned with respect to another element; which element that is can be changed by changing the value of the position property.

This is the second paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed.

This is the third paragraph of the positioning demo. This paragraph has an ID so we can change its position value without affecting the other paragraphs. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed.

This is the fourth paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed. I've added a lot of extra text into this paragraph so that the layout is longer than the browser window to better demonstrate the effect of fixed positioning. I've added a lot of extra text into this paragraph so that the layout is longer than the browser window to better demonstrate the effect of fixed positioning. I've added a lot of extra text into this paragraph so that the layout is longer than the browser window to better demonstrate the effect of fixed positioning.

This version illustrates `static` positioning.



Static Positioning

- The default position for any XHTML element is `static`.
- With `static` positioning, each element is simply laid out one after the other (in normal flow), so the paragraphs in our demo appear under each other, with their default margin settings creating the space between them.
- To break away from this sequential (normal flow) layout of the elements provided by the default `static` positioning, you must change a box's position property to one of the other three possible values.



Relative Positioning

- Relative positioning allows you to use the `top`, `right`, `bottom` and `left` attributes to move the element with respect to the position in which it would appear using normal flow.
- In our running demo example, notice on the next page that we've changed the style for the special third paragraph to now have `position: relative`.

```
p#specialpara {position:relative;  
                top:30px;  
                left:20px;  
            }
```





```
1 <?xml version="1.0" encoding="UTF-8" stand
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1
4 <html xmlns="http://www.w3.org/1999/xhtml"
5 <head>
6 <title>Relative Positioning Demo</title>
7 <style type="text/css">
8 <!--
9     body {background-color:#FFC;}
10    p {border:1px solid #000; }
11    p#specialpara { position: relative; top: 30px; left: 20px; color:red; background:#EEE;}
12    -->
13 </style>
14 </head>
15 <body>
16 <p>This is the first paragraph of the positioning demo. The objectiv
17 demonstrate the difference between the four position property valu
18 static, relative, absolute and fixed. The key to working with the
19 property is to understand that every element is positioned with re
20 another element; which element that is can be changed by changing
21 of the position property.
```

The special paragraph now uses relative positioning. Its top is moved down by 30 pixels and to the right by 20pixels from where it would appear in normal rendering.



This is the first paragraph of the positioning demo. The objective is to demonstrate property values: static, relative, absolute and fixed. The key to working with the position element is positioned with respect to another element; which element that is can be determined by the position property.

This is the second paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed.

This is the third paragraph of the positioning demo. This paragraph has an ID so we can change its position value without affecting the other paragraphs. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed.

This is the fourth paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed. I've added a lot of extra text into this paragraph so that the layout is longer than the browser window to better demonstrate the effect of fixed positioning. I've added a lot of extra text into this paragraph so that the layout is longer than the browser window to better demonstrate the effect of fixed positioning. I've added a lot of extra text into this paragraph so that the layout is longer than the browser window to better demonstrate the effect of fixed positioning.

Notice that although the element moves relative to its original position, nothing else changes. The space occupied by the original static element is retained, as is the positioning of the other elements.



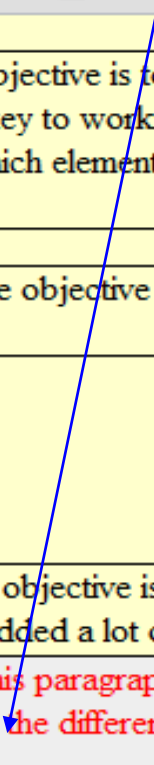
This is the first paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed. The key to working with relative positioning is that the element is positioned with respect to another element; which element is the reference element.

This is the second paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed.

This is the fourth paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed. I've added a lot of extra text into this paragraph so that the layout is longer than the other paragraphs.

This is the third paragraph of the positioning demo. This paragraph has an ID so we can change its position value without affecting the other paragraphs. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed.

An even more drastic relative position move, with the top set to 120 pixels. Notice that although the element moves relative to its original position, nothing else changes. The space occupied by the original static element is retained, as is the positioning of the other elements.



This is the first paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed. The key to working with the position property is to understand that every element is positioned with respect to another element; which element that is can be changed by changing the value of the position property.

This is the second paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed. This is the third paragraph of the positioning demo. This paragraph has an ID so we can change its position value without affecting the other paragraphs. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed.

This is the fourth paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed. I've added a lot of extra text into this paragraph so that the layout is longer than the browser window to better demonstrate the effect of fixed positioning. I've added a lot of extra text into this paragraph so that the layout is longer than the browser window to better demonstrate the effect of fixed positioning. I've added a lot of extra text into this paragraph so that the layout is longer than the browser window to better demonstrate the effect of fixed positioning.

Negative values also work which have the effect of moving an element up and to the left. In this case: `top` was set to `-40px` and `left` was set to `-20px`.



Relative Positioning

- The thing to remember about relative positioning is that if you move an element in this manner, you must allow space for it.
- Using the example on page 10, you might take the next step of adding a margin-top value of 30 pixels or greater to the fourth paragraph in order to move it down, thus preventing it from being overlapped by the repositioned third paragraph. (See next page.)



This is the first paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed. The key to working with relative positioning is to position an element with respect to another element; which element is the reference point for the relative property.

This is the second paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed.

This is the third paragraph of the positioning demo. This paragraph has an ID so we can change its position value without affecting the other paragraphs. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed.

This is the fourth paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed. I've added a lot of extra text into this paragraph so that the layout is longer than the browser window to better demonstrate the effect of fixed positioning. I've added a lot of extra text into this paragraph so that the layout is longer than the browser window to better demonstrate the effect of fixed positioning. I've added a lot of extra text into this paragraph so that the layout is longer than the browser window to better demonstrate the effect of fixed positioning.

The fourth paragraph is now styled to have a margin-top: 40px. Which moves it out from under the relocated third paragraph.

```
<!--  
  body {background-color:#FFC;}  
  p {border:1px solid #000;}  
  p#specialpara { position: relative; top: 30px; left: 20px; color: red;}  
  p#fourth {margin-top:40px; }  
-->
```



Absolute Positioning

- Absolute positioning is a whole different beast from `static` and `relative` positioning, since this type of positioning takes an element entirely out of normal flow.
- With absolute positioning, the default **positioning context** is the `body` of the document.
- In the running demo, we'll modify the special paragraph to be positioned absolutely.

```
p#specialpara {position:absolute;
                top:30px;
                left:20px;
            }
```



```
1 <?xml version="1.0" encoding="UTF-8" stand
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1
4 <html xmlns="http://www.w3.org/1999/xhtml"
5 <head>
6 <title>Absolute Positioning Demo</title>
7 <style type="text/css">
8 <!--
9     body {background-color:#FFC;}
10    p {border:1px solid #000; }
11    p#specialpara { position: absolute; top: 30px; left: 20px; color:red; background:#EEE;}
12    -->
13 </style>
14 </head>
15 <body>
16 <p>This is the first paragraph of the positioning demo. The objectiv
17 demonstrate the difference between the four position property valu
18 static, relative, absolute and fixed. The key to working with the
19 property is to understand that every element is positioned with re
20 another element; which element that is can be changed by changing
21 of the position property.
```

The special paragraph now uses absolute positioning. Its top is set at 30 pixels from the top of the body element and its left side is set at 20 pixels from the left side of the body element.

This is the first paragraph of the positioning demo. The objective is to demonstrate the difference between the four position

This is the third paragraph of the positioning demo. This paragraph has an ID so we can change its position value without affecting the other paragraphs. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed.

This is the second paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed.

This is the fourth paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed. I've added a lot of extra text into this paragraph so that the layout is longer than the browser window to better demonstrate the effect of fixed positioning. I've added a lot of extra text into this paragraph so that the layout is longer than the browser window to better demonstrate the effect of fixed positioning. I've added a lot of extra text into this paragraph so that the layout is longer than the browser window to better demonstrate the effect of fixed positioning.

Notice that the space previously occupied by the absolutely positioned element is gone. The absolutely positioned element has become entirely independent of the surrounding elements in the markup and is now positioned with respect to the <body> element.



Positioning Context

- The default positioning context of an absolutely positioned element is the `body` element.
- As the screen shot on the previous page illustrates, the offset provided by the `top` and `left` attribute values moves the element with respect to the `body` element – the top ancestor container in the markup hierarchy – not with respect to the element’s default position in the document flow (as is the case with `relative`).
- The next slide illustrates the same example with `top:50px` and `left:80px`.



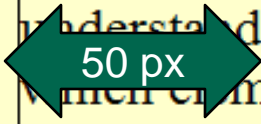
This is the first paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed. The key to working with the position property is to

understand which element has the position property.

This is the third paragraph of the positioning demo. This paragraph has an ID so we can change its position value without affecting the other paragraphs. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed.

This is the second paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed.

This is the fourth paragraph of the positioning demo. The objective is to demonstrate the difference between the four position property values: static, relative, absolute and fixed. I've added a lot of extra text into this paragraph so that the layout is longer than the browser window to better demonstrate the effect of fixed positioning. I've added a lot of extra text into this paragraph so that the layout is longer than the browser window to better demonstrate the effect of fixed positioning. I've added a lot of extra text into this paragraph so that the layout is



Positioning Context

- Because the absolutely element's positioning context is `body`, the element moves when the page is scrolled to retain its relationship to the `body` element, which also moves when the page scrolls.
- Before we look at how to use a different element than `body` as the positioning context for an absolutely positioned element, let's look at the last of the four positioning properties – `fixed` positioning.



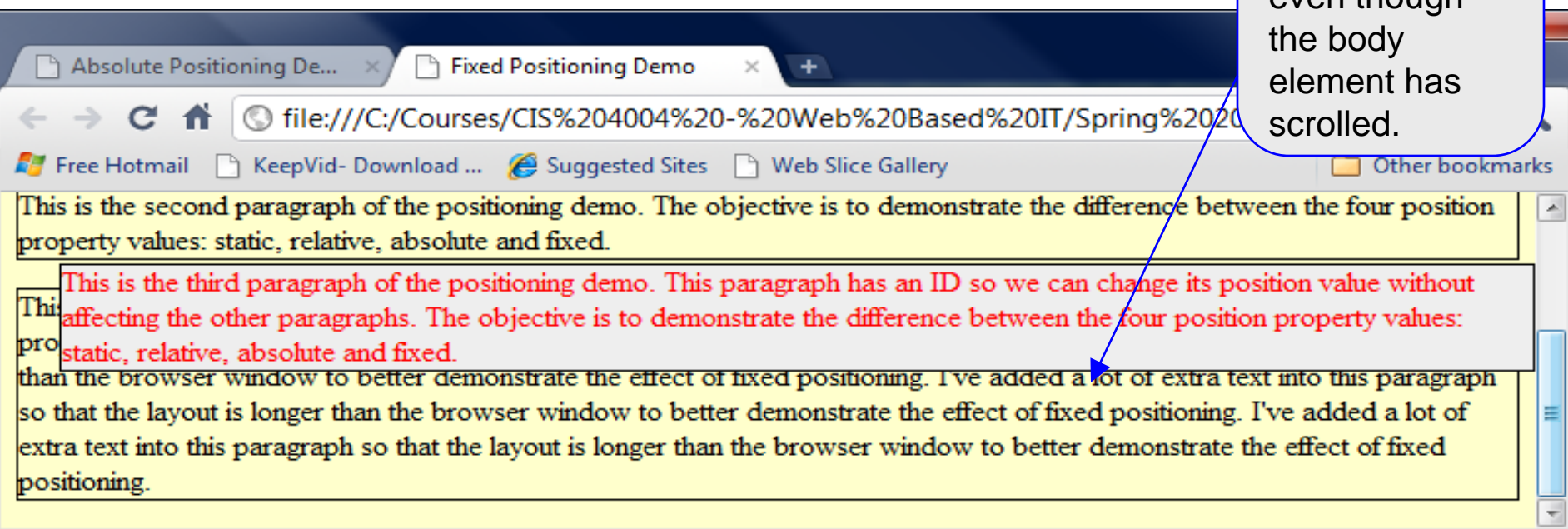
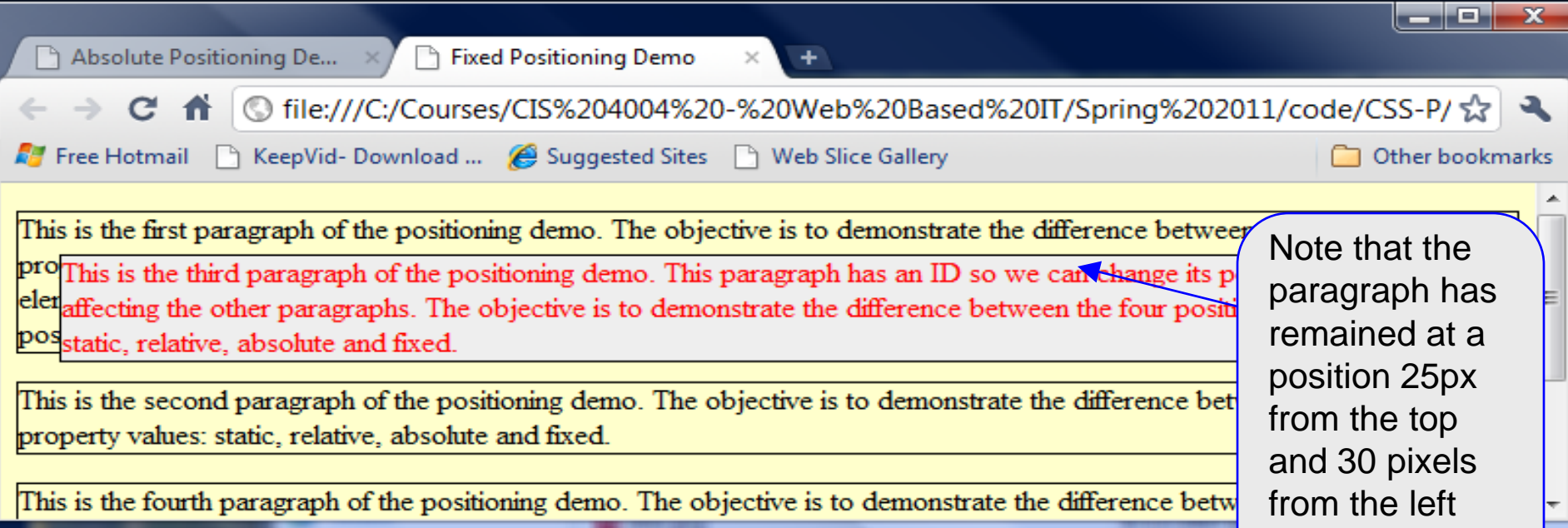
Fixed Positioning

- Fixed positioning is similar to absolute positioning, except that the element's positioning context is the viewport (the browser window or the screen of a handheld device, for example), so the element does not move when the page is scrolled.
- To really see this effect, you'll need to download the demo XHTML/CSS documents from this set of notes and pay particular attention to the fixed positioning example.



```
<?xml version="1.0" encoding="UTF-8" stand
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1
"http://www.w3.org/TR/xhtml1/DTD/xhtml1
<html xmlns="http://www.w3.org/1999/xhtml"
<head>
<title>Fixed Positioning Demo</title>
<style type="text/css">
<!--
body {background-color:#FFC;}
p {border:1px solid #000; }
p#specialpara { position: fixed; top: 25px; left: 30px; color:red; background:#EEE;}
-->
</style>
</head>
<body>
<p>This is the first paragraph of the positioning demo. The objective
demonstrate the difference between the four position property valu
static, relative, absolute and fixed. The key to working with the
property is to understand that every element is positioned with re
another element; which element that is can be changed by changing
of the position property.
```

The special paragraph now uses fixed positioning. Its top is set at 25 pixels from the top of the browser window and its left side is set at 30 pixels from the left side of the browser window.



Fixed Positioning

- This “nailed-to-the-browser” effect enables you to simulate the effect of what are now deprecated frames (recall the three flavors of XHTML: Strict, Transitional, and Frameset).
- For example, you can now create a navigation element that stays put on the page when the page scrolls without the problems that were associated with managing multiple documents in a frameset (the old way of doing this).
- **NOTE: the fixed position property does not work in IE6, but does work in IE7 and above.**



More On Positioning Context

- Now that we've seen all four types of positioning, let's go back and look at positioning context in more detail.
- Simply put, contextual positioning means that when you move an element using the attributes `top`, `right`, `bottom`, or `left`, you are moving that element with respect to another element. That other element is known as its **positioning context**.
- As we saw in the example on page 16, for absolute positioning, the default positioning context for an absolutely positioned element is `body`, unless you change it.



More On Positioning Context

- The body element is the containing element of all other elements in your markup, but you can use any ancestor element as a positioning context of another element by changing the ancestor's `position` value to `relative`.
- Consider the markup shown on the next page and its rendering on the following page.
- **QUESTION:** Why isn't the inner `<div>` 10 pixels down from the top of the outer `<div>` and 20 pixels to the left, as specified in the CSS?

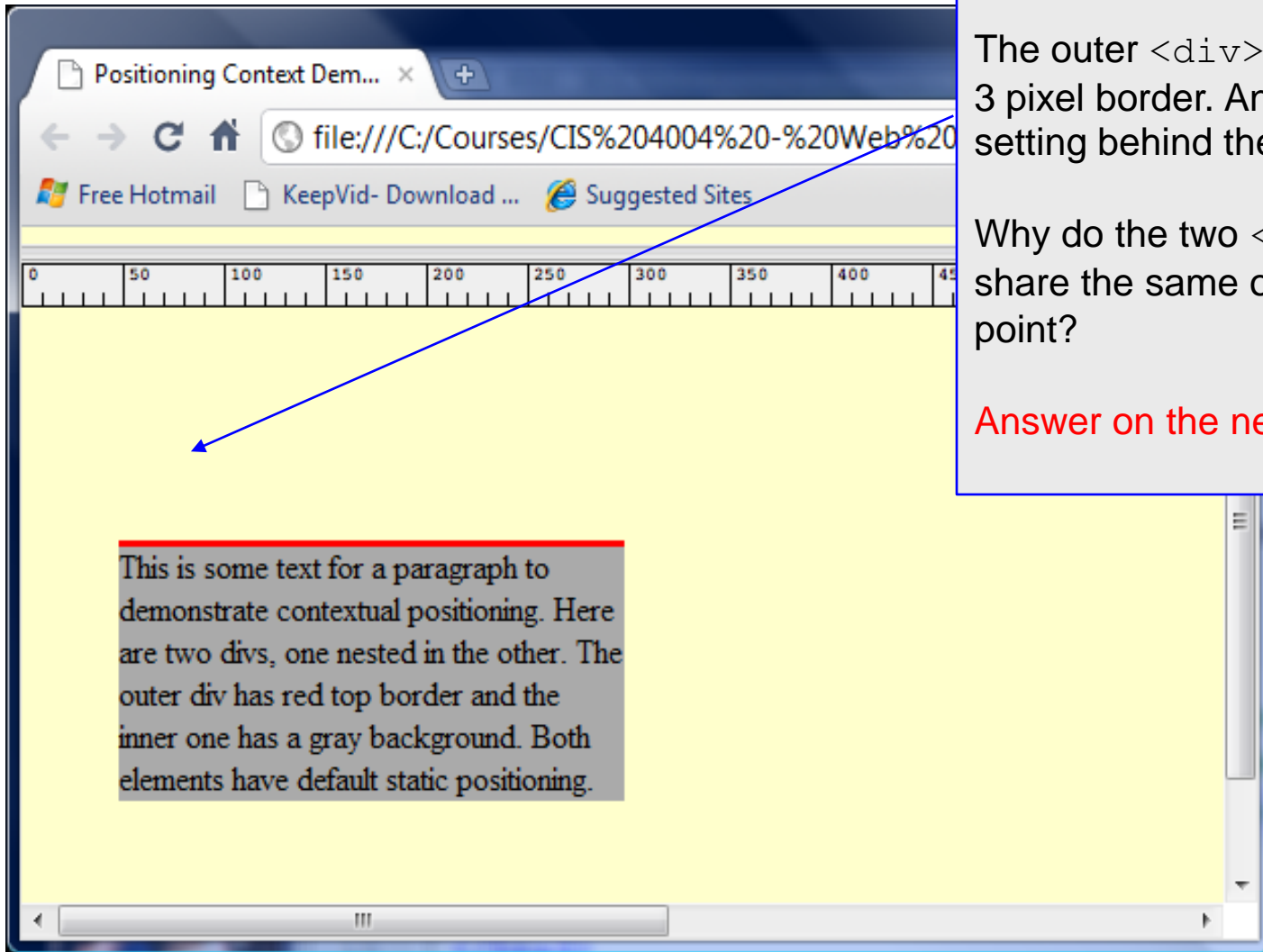


```
C:\Courses\CIS 4004 - Web Based IT\Spring 2011\code\CSS-P\CSS-P - Part 4\positioning context demo - version 1.html - Notepa...
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
static positioning demo.html relative positioning demo - version 1.html positioning context demo - version 1.html
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml">
5 <head>
6 <title>Positioning Context Demo - Version 1</title>
7 <style type="text/css">
8 <!--
9 body {background-color:#FFC;}
10 div#outer_div {width:250px; margin:100px 40px; border-top:3px solid red;}
11 div#inner_div {top:10px; left:20px; background:#AAA;}
12 -->
13 </style>
14 </head>
15 <body>
16 <div id="outer_div">
17 <div id="inner_div"> This is some text for a paragraph to demonstrate
18 positioning. Here are two divs, one nested in the other. The outer c
19 red top border and the inner one has a gray background. Both element
20 default static positioning.
21 </div>
22 </div>
23 </body>
24 </html>
```

Hyper Text Mark nb char : 827 nb line : 25 Ln:1 Col:1 Sel:0 UNIX ANSI INS



Positioning Context



The outer `<div>` has a solid red 3 pixel border. And it can be seen setting behind the inner `<div>`.

Why do the two `<div>` elements share the same origin (top-left) point?

Answer on the next page.



Positioning Context

- The answer to the question posed in the last example, is that the inner (and irrelevantly, the outer) `<div>` element has the default positioning of `static`. This means it is rendered in normal flow, and because the outer `<div>` has no content, the inner `<div>` starts in the same place.
- Only when you set an element to one of the other three positioning options – `relative`, `absolute`, or `fixed`, - do the `top`, `right`, `bottom`, and `left` attribute values actually do anything.
- To illustrate this fact, consider the modified markup shown on the next page, where the `left` and `top` attribute values have been reset for the inner `<div>`. Notice that since we left it with its default position it didn't move!

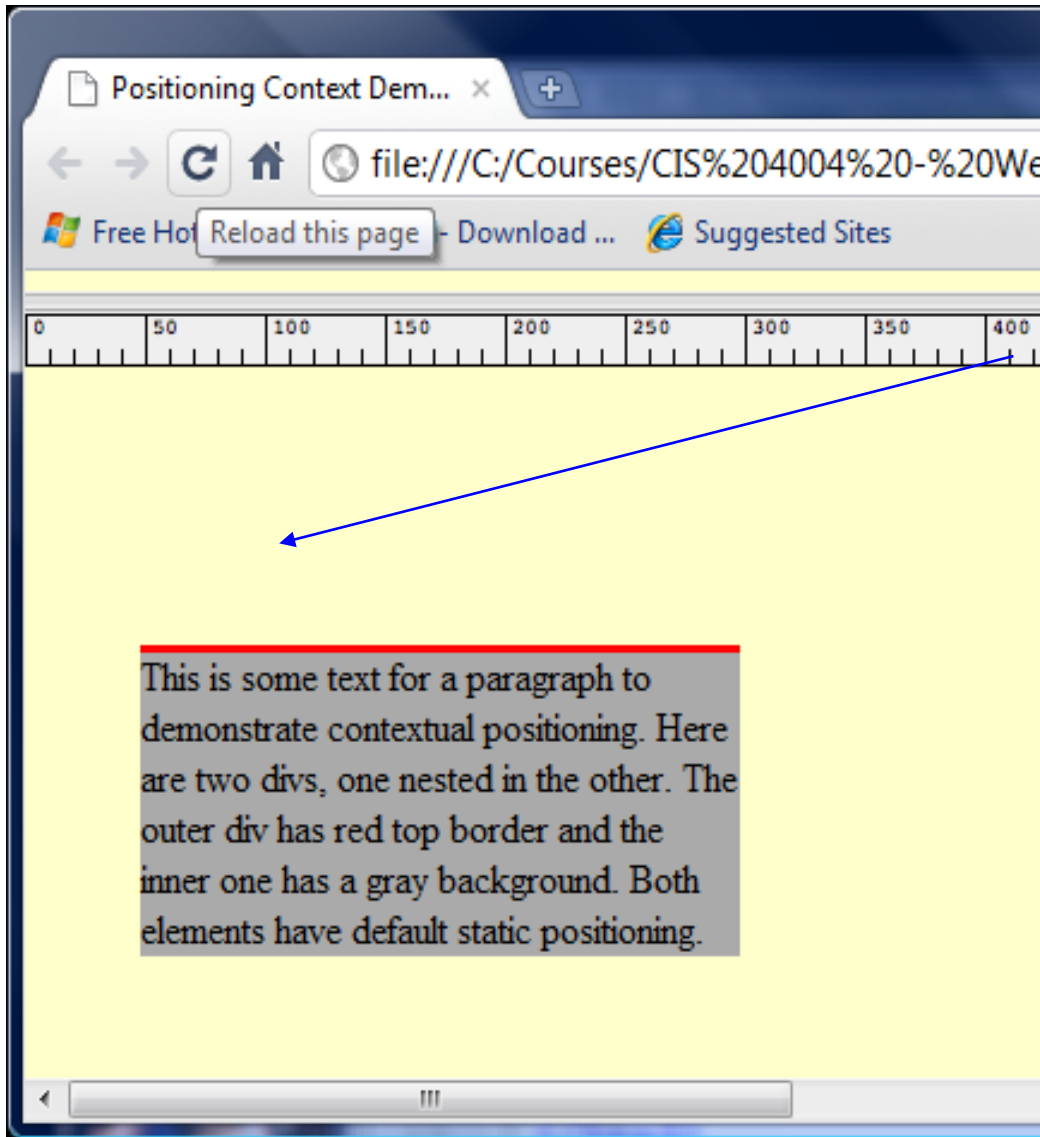


```

1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4  <html xmlns="http://www.w3.org/1999/xhtml">
5  <head>
6     <title>Positioning Context Demo - Version 2</title>
7     <style type="text/css">
8     <!--
9         body {background-color:#FFC;}
10        div#outer_div {width:250px; margin:100px 40px; border-top:3px solid red;}
11        div#inner_div {top:100px; left:200px; background:#AAA;}
12        #ruler {position:relative; left:-51px; top:0px; margin-bottom:5px;}
13    -->
14    </style>
15 </head>
16 <body>
17 
18 <div id="outer_div">
19 <div id="inner_div"> This is some text for a paragraph to demonstrate
20 positioning. Here are two divs, one nested in the other. The outer c
21 red top border and the inner one has a gray background. Both element

```





The two `<div>` elements are still in exactly the same relative positions, even though the `top` and `left` attribute values are quite a bit different between version 1 and version 2.

In version 1 `top = 10px` and `left = 20px`

In version 2 `top = 100px` and `left = 200px`



Positioning Context

- Now let's see what happens if we set the inner `<div>` element's `position` property to `absolute`.
- We'll modify the CSS to be:

```
body {background-color:#FFC;} 
```

```
div#outer_div {width:250px; margin:100px 40px;  
                border-top:3px solid red;}
```

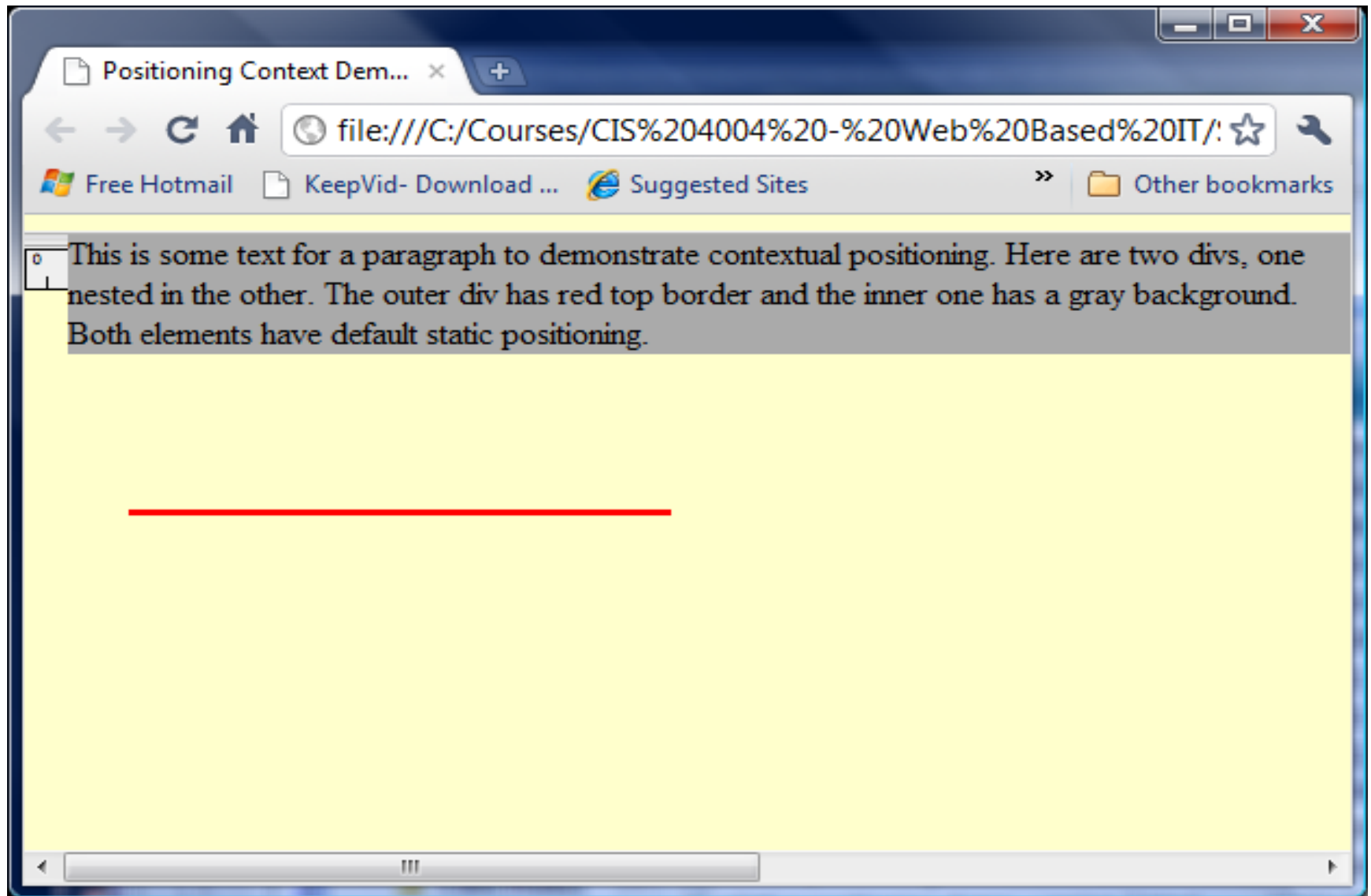
```
div#inner_div{position:absolute; top:10px;  
              left:20px; background-color:#AAA;}
```

- The inner `<div>` element is now absolutely positioned, but with respect to what? Where do you expect the inner `<div>` element to be positioned?




```
C:\Courses\CIS 4004 - Web Based IT\Spring 2011\code\CSS-P\CSS-P - Part 4\positioning context demo - version 3.html - Notepad++
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
positioning context demo - version 2.html positioning context demo - version 3.html
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml">
5 <head>
6 <title>Positioning Context Demo - Version 3</title>
7 <style type="text/css">
8 <!--
9     body {background-color:#FFC;}
10    div#outer_div {width:250px; margin:100px 40px; border-top:3px solid red;}
11    div#inner_div {position:absolute; top:10px; left:20px; background:#AAA;}
12    #ruler {position:relative; left:-58px; top:0px; margin-bottom:5px;}
13    -->
14 </style>
15 </head>
16 <body>
17 
18 <div id="outer_div">
19 <div id="inner_div"> This is some text for a paragraph to demonstrate conte
20 positioning. Here are two divs, one nested in the other. The outer div ha
21 red top border and the inner one has a gray background. Both elements ha
22 default static positioning.
Hyper Text Markup Lang nb char: 970 nb line: 27 Ln: 2 Col: 57 Sel: 0 UNIX ANSI INS
```





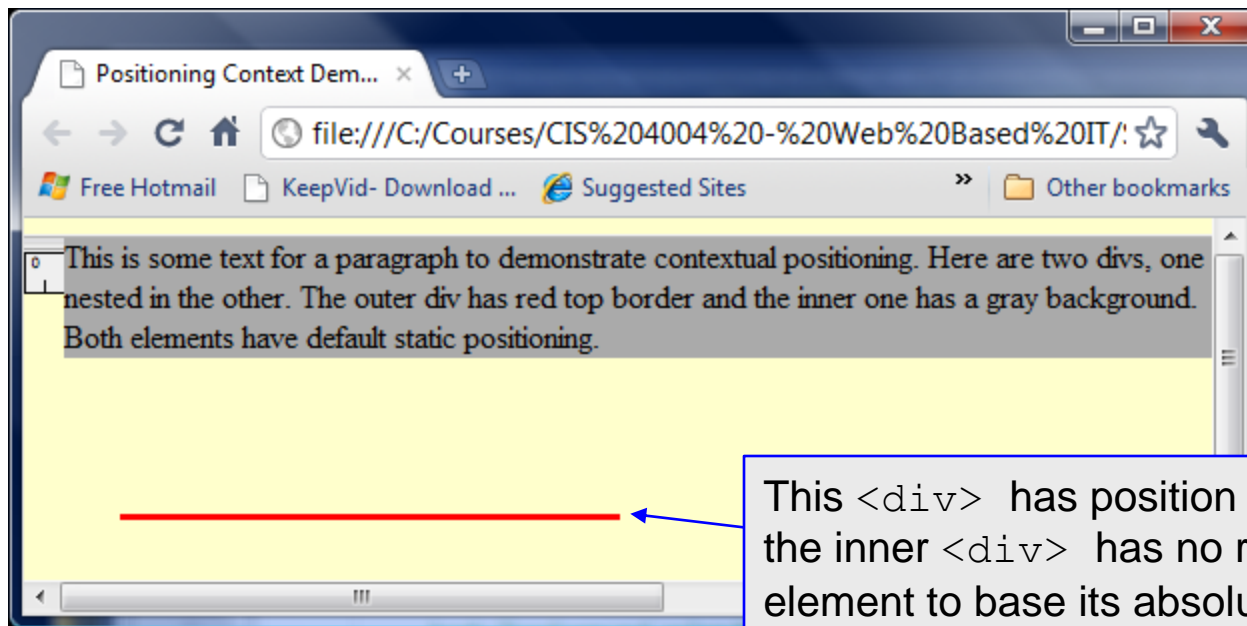
Positioning Context

- As you can see on the previous page, since there is no other relatively positioned element for the inner `<div>` to reference, it positions itself by default with respect to the `<body>` element (so it is overlaid over the ruler).
- The top border of the outer `<div>` is set to red so you can see where it is located. Its margins push it 50 pixels down and 40 pixels to the left of the top corner of the browser window.
- Because the inner `<div>`'s position property is set to absolute, it is positioned relative to the `<body>` element, because `<body>` is the default positioning context.



Positioning Context

- In other words, the inner `<div>` element entirely ignores its parent (the outer `<div>` element), and its `top` and `left` attributes offset it with respect to the `<body>` element, as shown in the rendering on pages 34 and below.



This `<div>` has `position static` by default. Thus the inner `<div>` has no relatively positioned element to base its absolute position on other than the default positioning context of the `<body>` element.



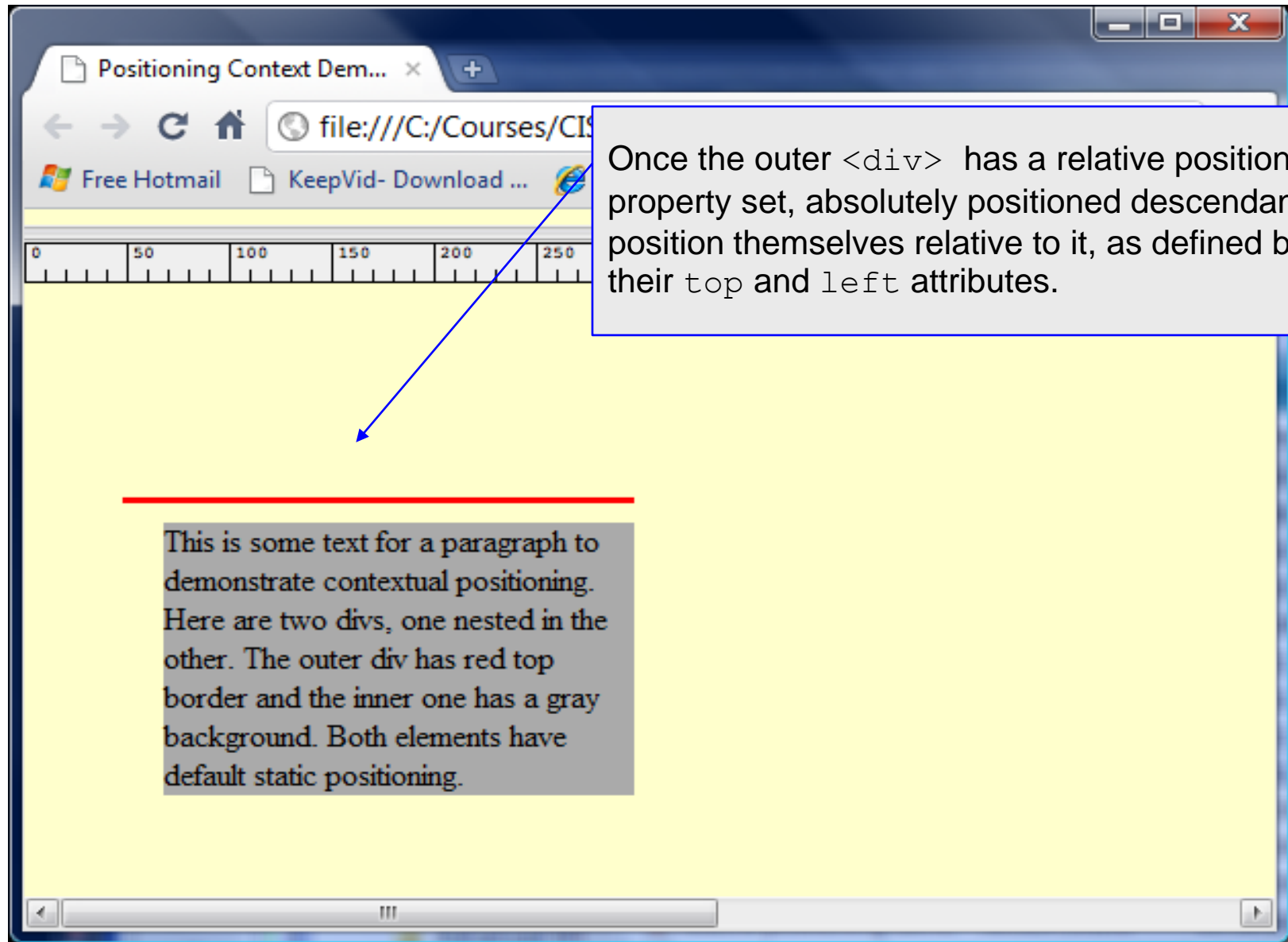
Positioning Context

- As the final example for explaining positioning context, let's now set the outer `<div>` element's `position` property to `relative`.
- This will now cause the positioning context of the absolutely positioned inner `<div>` element to become the outer `<div>` element in which it is nested.
- This means the setting the `top` and `left` attributes of the inner `<div>` element now positions it with respect to the outer `<div>` element.



```
C:\Courses\CIS 4004 - Web Based IT\Spring 2011\code\CSS-P\CSS-P - Part 4\positioning context demo - version 4.html - Notepad++
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
positioning context demo - version 3.html positioning context demo - version 4.html
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml">
5 <head>
6 <title>Positioning Context Demo - Version 4</title>
7 <style type="text/css">
8 <!--
9     body {background-color:#FFC;}
10    div#outer_div {position: relative; width:250px; margin:100px 40px; border-top:3px solid red;}
11    div#inner_div {position:absolute; top:10px; left:20px; background-color:#AAA;}
12    #ruler {position:relative; left:-58px; top:0px; margin-bottom:5px;}
13    -->
14 </style>
15 </head>
16 <body>
17 
18 <div id="outer_div">
19 <div id="inner_div"> This is some text for a paragraph to demonstrate content
20 positioning. Here are two divs, one nested in the other. The outer div has a
21 red top border and the inner one has a gray background. Both elements have
22 default static positioning.
Hyper Text Markup Language nb char: 996 nb line: 27 Ln: 17 Col: 54 Sel: 0 UNIX ANSI INS
```





Once the outer `<div>` has a relative positioning property set, absolutely positioned descendants position themselves relative to it, as defined by their `top` and `left` attributes.

This is some text for a paragraph to demonstrate contextual positioning. Here are two divs, one nested in the other. The outer div has red top border and the inner one has a gray background. Both elements have default static positioning.



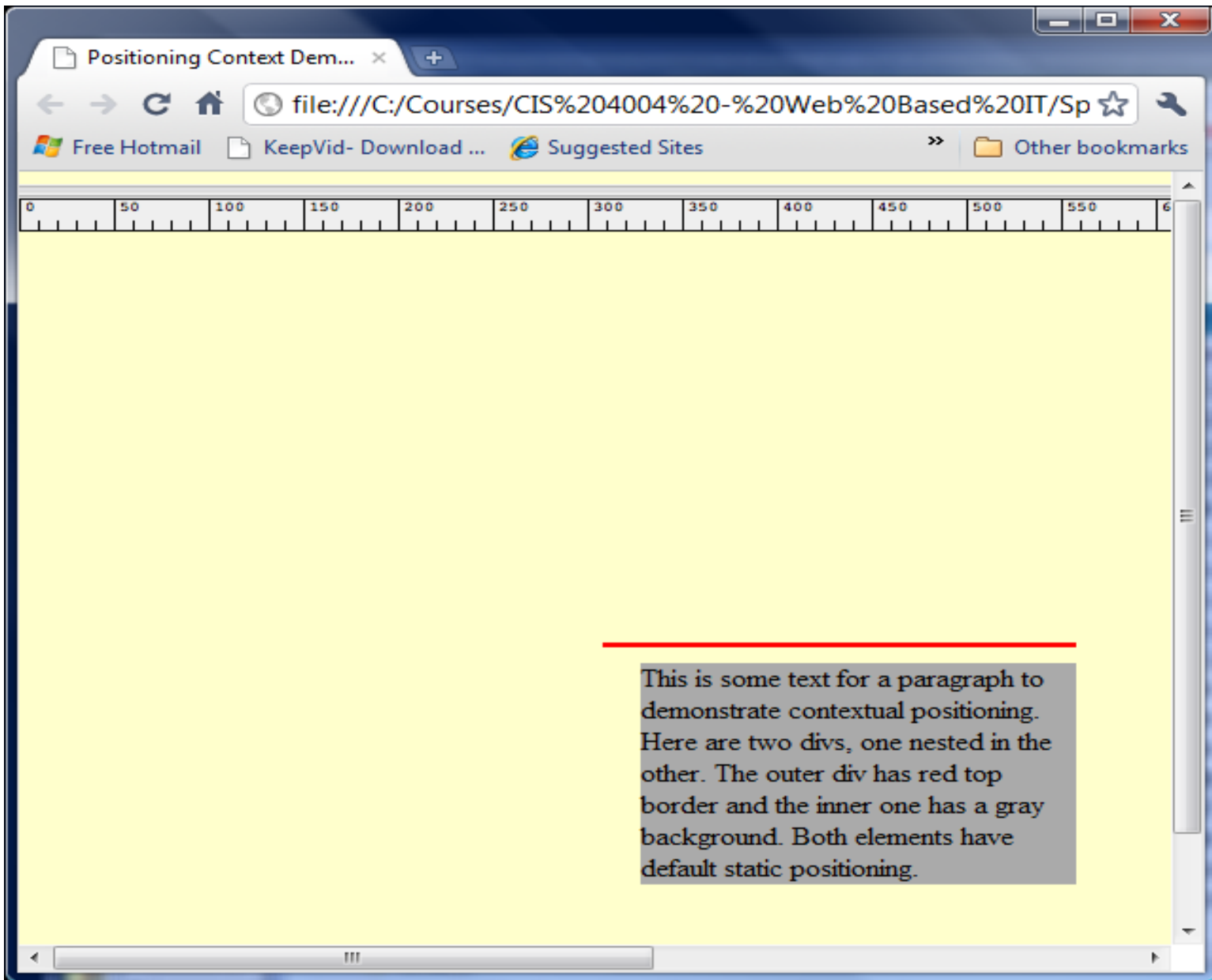
Positioning Context

- If you set the top and left attribute values of the outer `<div>` element to anything other than 0, the inner `<div>` would move to maintain its positioning relationship to the outer `<div>`, which is its positioning context.
- This last example more clearly illustrates this (it really is the last example this time).
- In this very last example, we'll reset the margins of the outer `<div>` element drastically from their original position. The thing to notice is how the inner `<div>` element move with respect to the new position of the outer `<div>`.




```
C:\Courses\CIS 4004 - Web Based IT\Spring 2011\code\CSS-P\CSS-P - Part 4\positioning context demo - version 5.html - Notepad++
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
positioning context demo - version 4.html positioning context demo - version 5.html
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml">
5 <head>
6 <title>Positioning Context Demo - Version 5</title>
7 <style type="text/css">
8 <!--
9     body {background-color:#FFC;}
10    div#outer_div {position: relative; width:250px; margin:250px 300px; border-top:3px solid red;}
11    div#inner_div {position:absolute; top:10px; left:20px; background-color:#AAA;}
12    #ruler {position:relative; left:-58px; top:0px; margin-bottom:5px;}
13    -->
14 </style>
15 </head>
16 <body>
17 
18 <div id="outer_div">
19 <div id="inner_div"> This is some text for a paragraph to demonstrate content
20 positioning. Here are two divs, one nested in the other. The outer div has
21 a red top border and the inner one has a gray background. Both elements have
22 default static positioning.
Hyper Text Markup Language nb char: 997 nb line: 27 Ln: 17 Col: 54 Sel: 0 UNIX ANSI INS
```





The `display` Property

- Just as every element has a `position` property, every element also has a `display` property.
- Although there are quite a number of `display` property values, the most commonly used elements have a default `display` property value of either **block** or **inline**.
- **Block elements**, such as paragraphs, headings, and lists, sit one above another when displayed in the browser.
- **Inline elements**, such as `anchor`, `span`, and `img`, sit side-by-side when they are displayed in the browser and only appear on a new line if there is insufficient room on the previous line.



The `display` Property

- The ability to change block elements to inline elements, and vice versa is a powerful capability that allows you, for example, to force an inline element to fill its containing element. We'll do this later with links when we create CSS drop-down menus.
- Changing an element's display property is done like this:

block by default

```
p {display: inline; }
```

inline by default

```
a {display: block; }
```



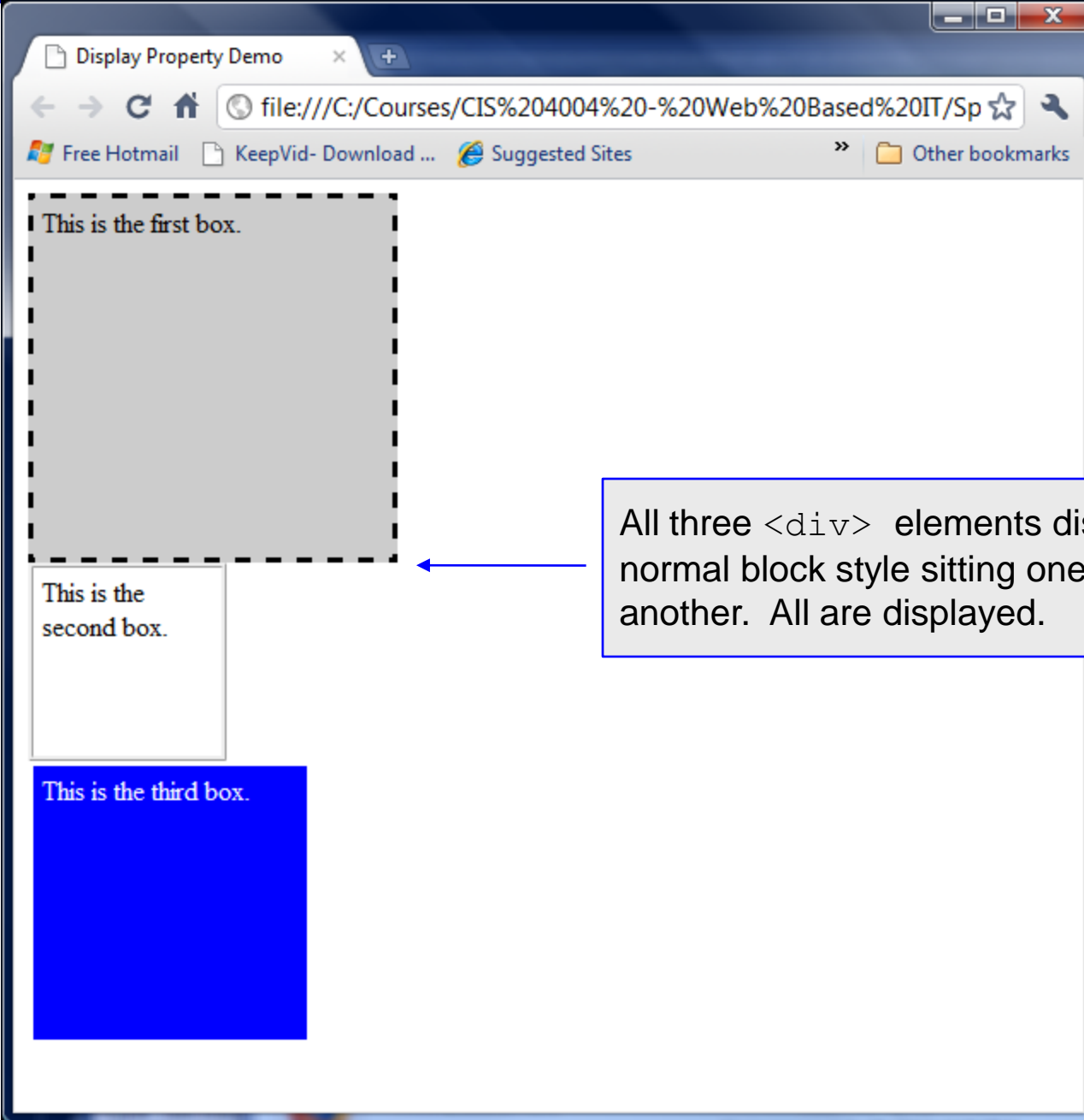
The `display` Property

- The other value for the `display` property that is worth discussing here is `none`.
- When an element's `display` property is set to `none`, that element, and any elements nested inside it, are not displayed on the page. Any space that was occupied by the element is removed; its as if the related markup did not exist.
- NOTE: This contrasts with the `visibility` property, which simply has the values `visible` or `hidden`. If an element's `visibility` is set to `hidden`, the element is hidden, but the space it occupied remains. We'll see more on this later.



```
C:\Courses\CIS 4004 - Web Based IT\Spring 2011\code\CSS-P\CSS-P - Part 4\display demo - version 1.html - Notepad++
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
positioning context demo - version 4.html positioning context demo - version 5.html display demo - version 1.html
1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml">
5 <head>
6 <title>Display Property Demo</title>
7 <style type="text/css">
8 <!--
9 .box1 {width:200px; height:200px; background-color:#cccccc; border:dashed; padding:5px;}
10 .box2 { width:100px; height:100px; background-color:#ffffff; border: ridge; padding:5px;}
11 .box3 {width:150px; height:150px; background-color:blue; border:solid; padding:5px; color:white;}
12 -->
13 </style>
14 </head>
15 <body>
16 <div class="box1">
17     This is the first box.
18 </div>
19 <div class="box2">
20     This is the second box.
21 </div>
22 <div class="box3">
23     This is the third box.
Hyper Text Markup Lang nb char: 805 nb line: 29 Ln:1 Col:1 Sel:0 Dos\Windows ANSI INS
```

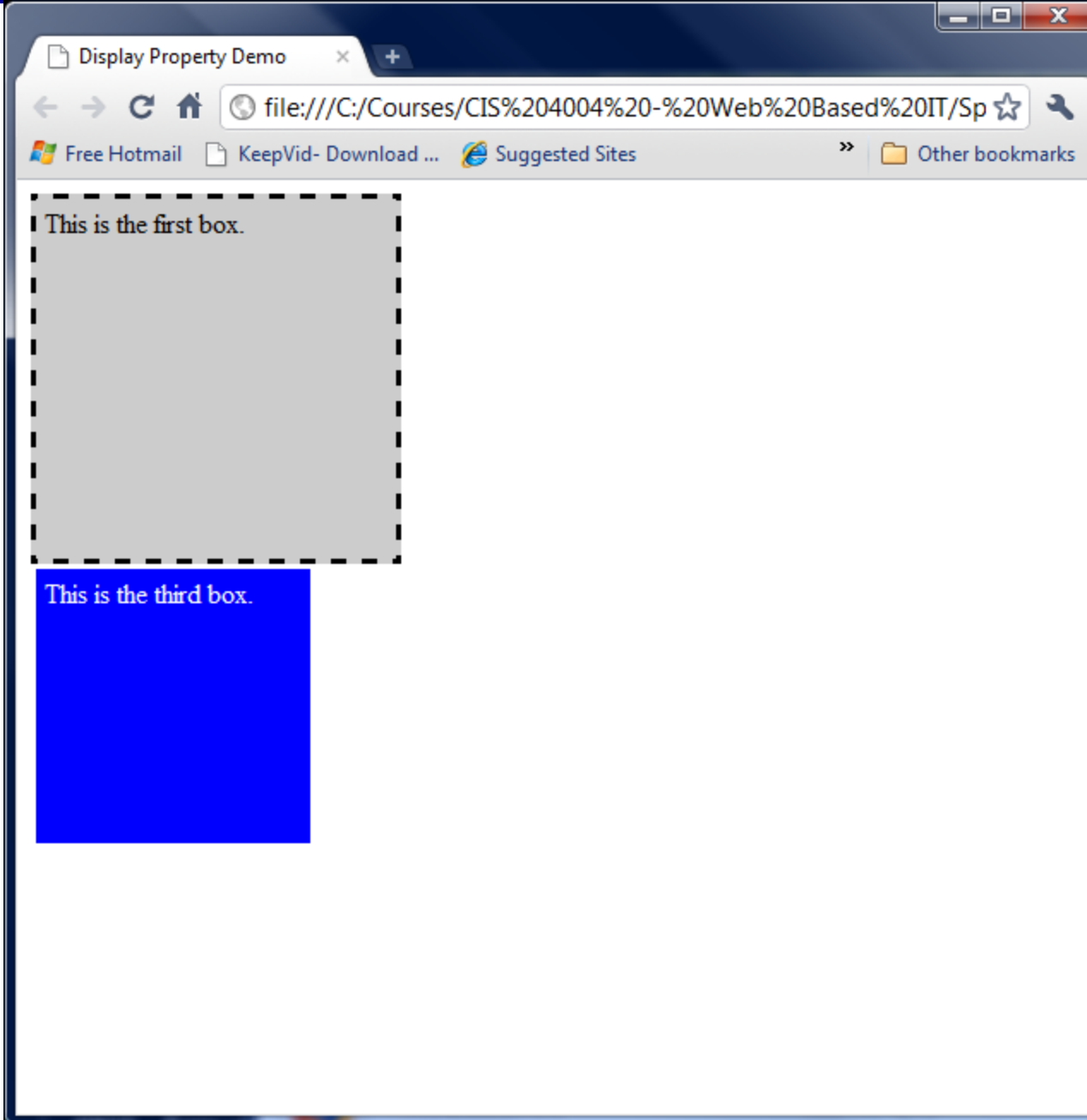




```
C:\Courses\CIS 4004 - Web Based IT\Spring 2011\code\CSS-P\CSS-P - Part 4\display demo - version 2.html - Notepad++
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
positioning context demo - version 4.html positioning context demo - version 5.html display demo - version 1.html display demo - version 2.html
1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml">
5 <head>
6 <title>Display Property Demo</title>
7 <style type="text/css">
8 <!--
9 .box1 {width:200px; height:200px; background-color:#cccccc; border:dashed; padding:5px;}
10 .box2 { display:none; width:100px; height:100px; background-color:#ffffff; border: ridge; padding:5px;}
11 .box3 {width:150px; height:150px; background-color:blue; border:solid; padding:5px; color:white;}
12 -->
13 </style>
14 </head>
15 <body>
16 <div class="box1">
17     This is the first box.
18 </div>
19 <div class="box2">
20     This is the second box.
21 </div>
22 <div class="box3">
23     This is the third box.
```

The <div> element styled using class box2 has its display property set to none. Notice on the next page that the space that would have been occupied by the second box has disappeared and box3 moves into that space.

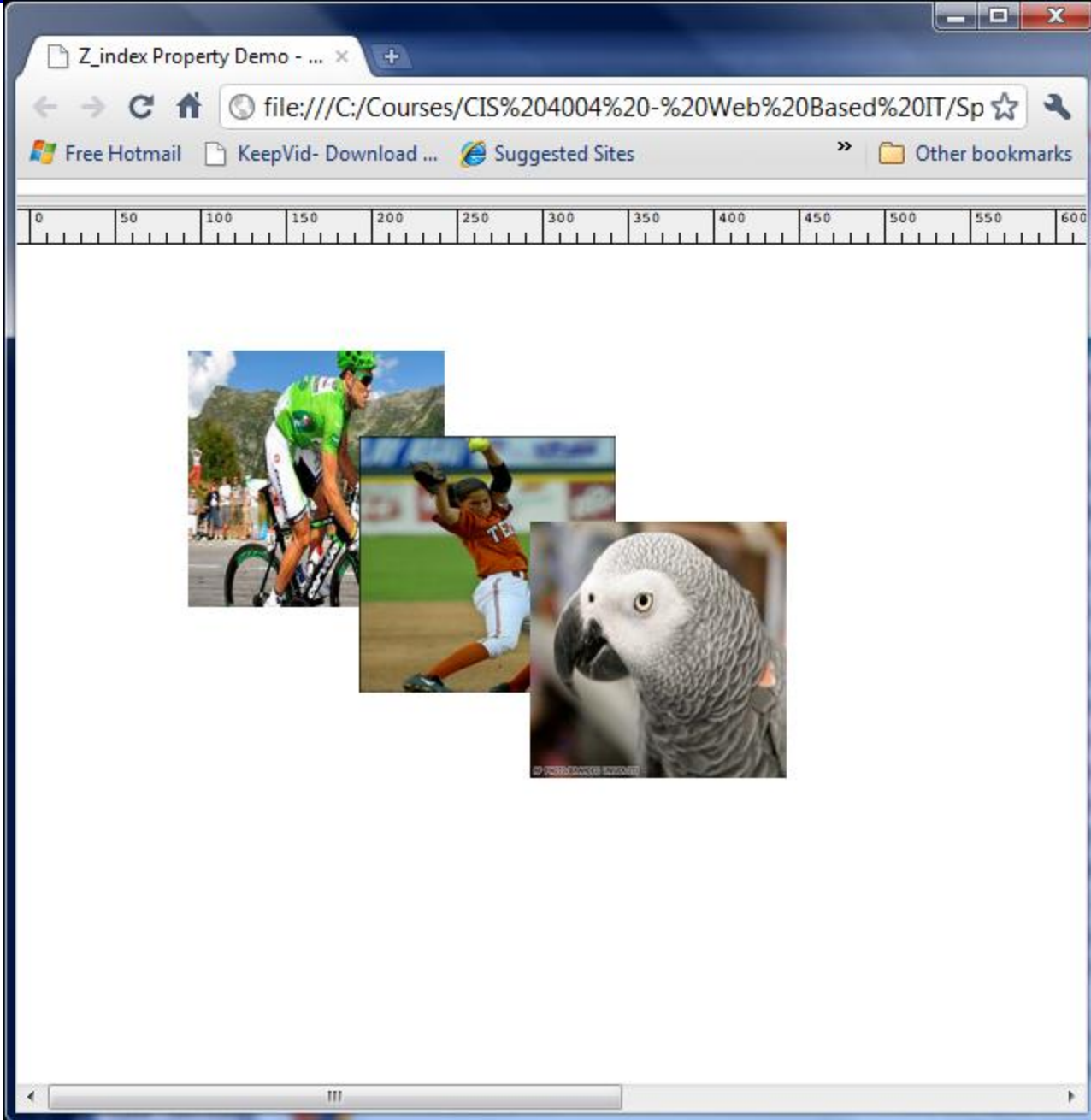




The `z-index` Property

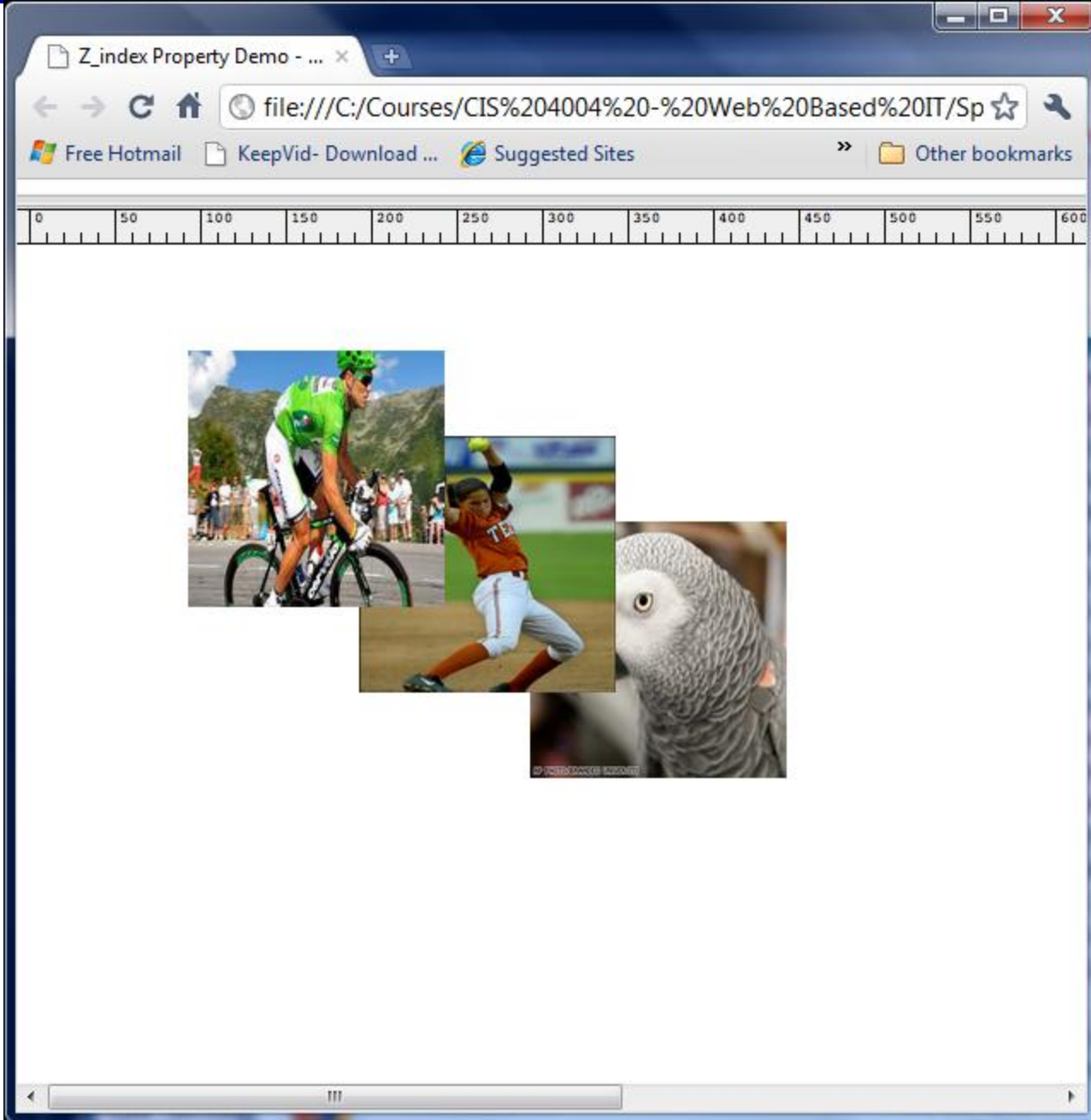
- The `z-index` property is used to modify the stacking order of elements on a Web page.
- When using only XHTML there is no easy way to “stack” elements other than configuring backgrounds for pages or tables.
- The `z-index` property provides flexibility in the display of elements.
- The default `z-index` value is “0”. Elements with higher `z-index` values will appear stacked on top of elements with lower `z-index` values rendered on the same position of the page.
- The Web page shown on the next page is configured using absolute positioning and `z-index` properties. The XHTML code is shown on the following page.





```
C:\Courses\CIS 4004 - Web Based IT\Spring 2011\code\CSS-P\CSS-P - Part 4\z-index property demo - version 1.html - Notepad++
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
positioning context demo - version 4.html positioning context demo - version 5.html display demo - version 1.html z-index property demo - version 1.html
4 <html xmlns="http://www.w3.org/1999/xhtml">
5 <head>
6 <title>Z_index Property Demo - version 1</title>
7 <style type="text/css">
8 <!--
9 #thor { position:absolute; left:100px; top:100px; z-index:1;}
10 #cat { position:absolute; left:200px; top:150px; z-index:2;}
11 #alex { position:absolute; left:300px; top:200px; z-index:3;}
12 #ruler {position:relative; left:-51px; top:0px; margin-bottom:5px; }
13 -->
14 </style>
15 </head>
16 <body>
17 <div>
18 
19 <div id="thor">
20 
21 </div>
22 <div id="cat">
23 
24 </div>
25 <div id="alex">
26 
27 </div>
Hyper Text Markup Language nb char : 986 nb line : 32 Ln: 20 Col: 50 Sel: 0 UNIX ANSI INS
```



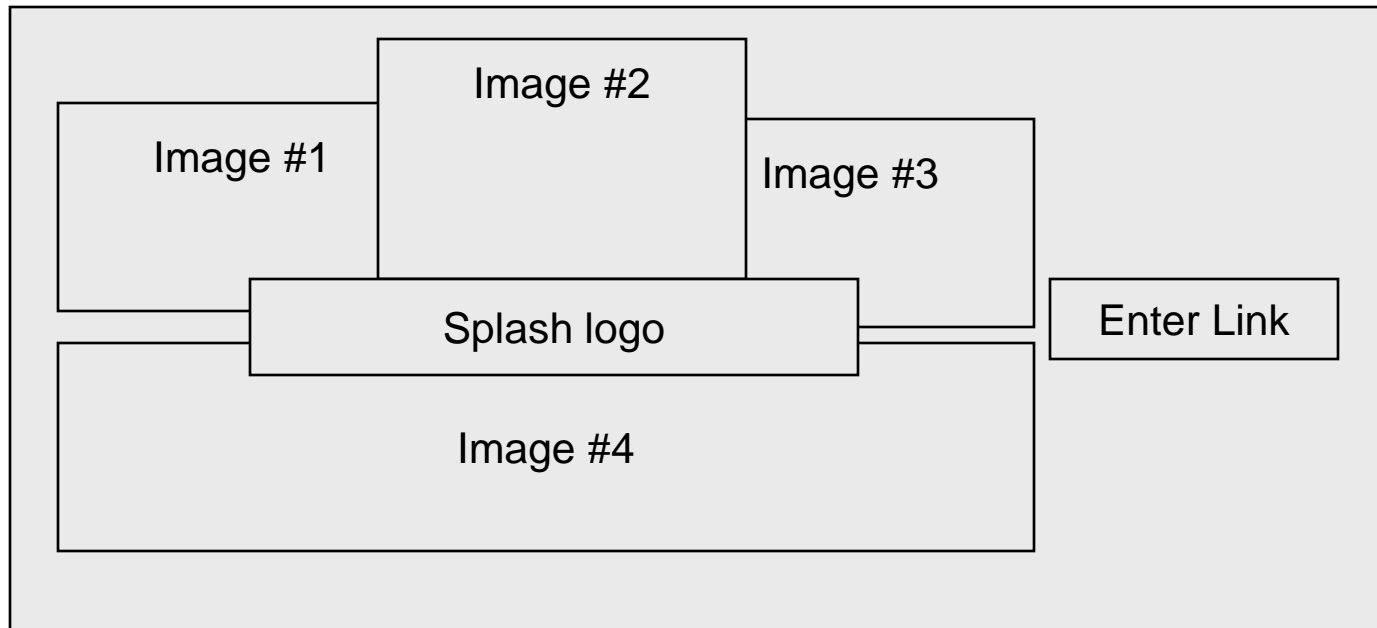


```
C:\Courses\CIS 4004 - Web Based IT\Spring 2011\code\CSS-P\CSS-P - Part 4\z-index property demo - version 2.html - Notepad++
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
z-index property demo - version 1.html z-index property demo - version 2.html
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml">
5 <head>
6 <title>Z_index Property Demo - version 2</title>
7 <style type="text/css">
8 <!--
9 #thor { position:absolute; left:100px; top:100px; z-index:3;}
10 #cat { position:absolute; left:200px; top:150px; z-index:2;}
11 #alex { position:absolute; left:300px; top:200px; z-index:1;}
12 #ruler {position:relative; left:-51px; top:0px; margin-bottom:5px; }
13 -->
14 </style>
15 </head>
16 <body>
17 <div>
18 
19 <div id="thor">
20 
21 </div>
22 <div id="cat">
23 
24 </div>
Hyper Text Markup Language nb char : 986 nb line : 32 Ln: 20 Col: 50 Sel: 0 UNIX ANSI INS
```

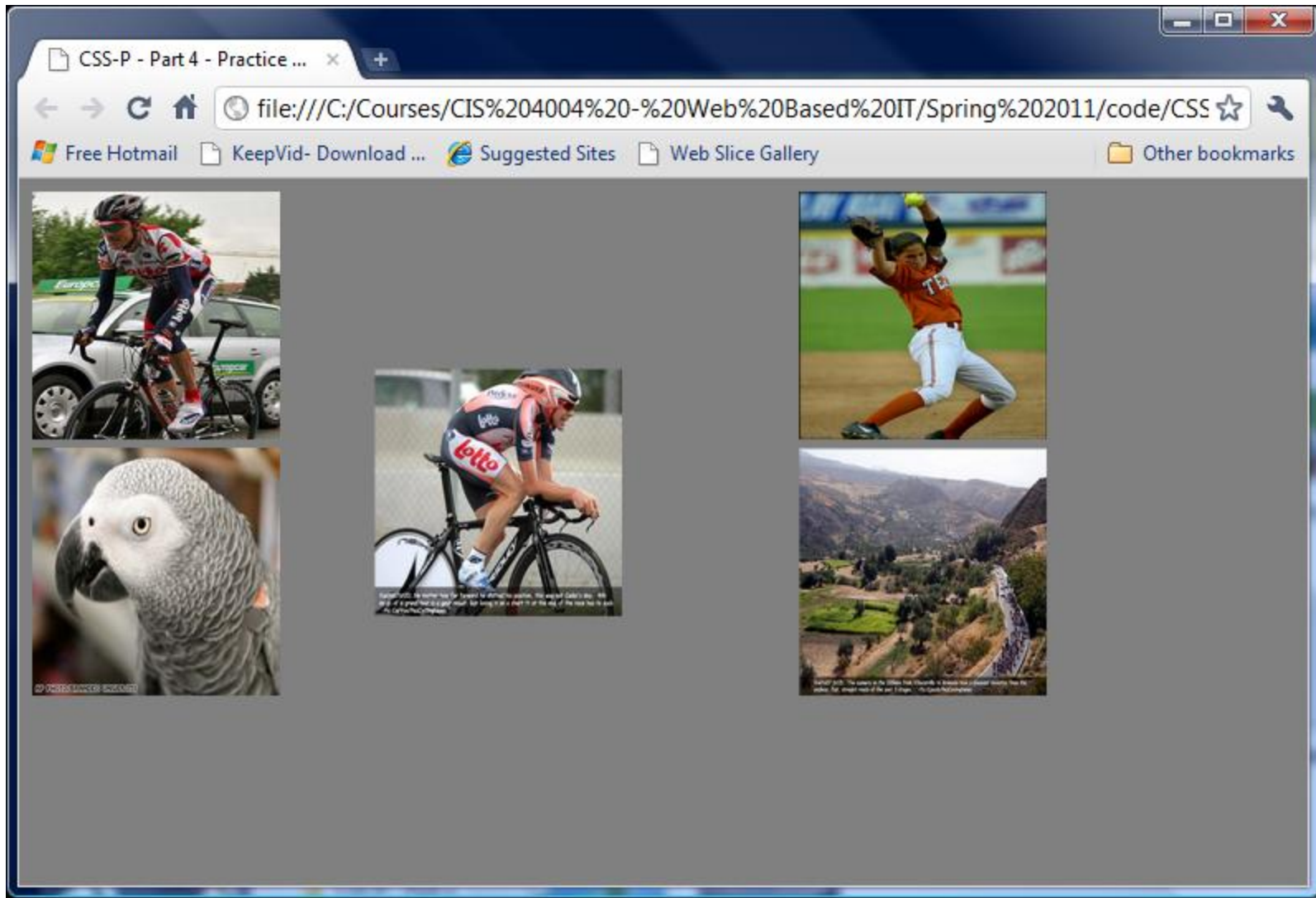


Practice Problems

1. Using five different image files, create a Web page that looks like the one shown on page 56. Use CSS-P to produce this rendering using a mixture of floating and absolute positioning.
2. Using XHTML and CSS, code a “splash page” for a Web site that looks like the one on page 57. The storyboard for this page is shown below. (A “splash page” is a client-server term for an application that displays an introductory (or splash) screen while the application loads. Splash pages, sometimes called splash screens, can set the tone or introduce a Web site.



Practice Problems





Professional Cycling 2011



[Enter Our Site](#)

