CSS Mastery Errata

Many hands make light work; so if you spot an error that doesn't appear on this list, please let me know at errata@andybudd.com.

Page	Error
xix	In the sixth paragraph: Jason Andrew Andrew Santa Maria,, Dave Shea Should read: Jason Andrew Andrew Santa Maria, Dave Shea *(The duplicated middle name is not a typo)*
xxiii	Third paragraph reads: Lastly, for (X)HTML examples that contain repeating data, rather than writing out every line, the ellipse character (…) is used to denote code continuation: Should read: Lastly, for (X)HTML examples that contain repeating data, rather than writing out every line, the ellipsis character (…) is used to denote code continuation:
10	The sixth paragraph reads: The browser chooses which rendering method to use based on the existence of a DOCTYPE declaration and the DTD being used. If an XHTML document contains a fully formed DOCTYPE, it will normally be rendered in standards mode. For an HTML document, a strict DTD will usually cause the browser to render the page in standards mode, while a transitional DTD will cause it to render in quirks mode. A badly formed or nonexistent DOCTYPE will cause both HTML and XHTML documents to be rendered in quirks mode. Should read: The browser chooses which rendering method to use based on the existence of a DOCTYPE declaration and the DTD being used. If an XHTML document contains a fully formed DOCTYPE, it will normally be rendered in standards mode. For an HTML 4.01 document, a DOCTYPE containing a strict DTD will usually cause the page to render in standards mode. A DOCTYPE containing a transitional DTD and URI will also cause the page to render in standards mode, while a transitional DTD without a URI will cause the page to render in quirks mode. A badly formed or nonexistent DOCTYPE will cause both HTML and XHTML documents to be rendered in quirks mode.

10	Last paragraph begins: DOCTYPE switching is a hack used by browsers to distinguish legacy documents from more standards-complaint ones. Should read: DOCTYPE switching is a hack used by browsers to distinguish legacy documents from more standards-compliant ones.

11	Third code example on the page reads:
	`p {color: black;}`
	`a {text-decaration: underline;}`
	`h1 {font-weight: bold;}`
	Should read:
	`p {color: black;}`
	`a {text-decoration: underline;}`
	`h1 {font-weight: bold;}`

11	Last code example reads:
	`li a {text-decaration: none;}`
	Should read:
	`li a {text-decoration: none;}`

12	Second code example begins:
	`#mainContent h1 {text-size: 1.8em;}`
	`#secondaryContent h1 {text-size: 1.2em;}`
	Should read:
	`#mainContent h1 {font-size: 1.8em;}`
	`#secondaryContent h1 {font-size: 1.2em;}`

14	First paragraph ends: In the following example, the list items in the outer list will be bold while list items in the nested list will remain unaffected: Should read: In the following example, the list items in the outer list will be given a custom icon, while list items in the nested list will remain unaffected:

14	First code example begins:
	`#nav > li {font-weight: bold;}`
	Should read:
	`#nav > li { background: url(folder.png) no-repeat left top;}`

	Second code example reads:
	`#nav li {font-weight: bold;}`
	`#nav li * {font-weight: normal;}`
	Should read:
	`#nav li { background: url(folder.png) no-repeat left top;}`

	`#nav li * {background-image: none;}`
15	Third paragraph begins: One clever way of using the attribute selector is to capitalize on the fact that IE 6 and below does not support it. You can then apply one style to IE and another style to more standards-complaint browsers. Should read: One clever way of using the attribute selector is to capitalize on the fact that IE 6 and below does not support it. You can then apply one style to IE and another style to more standards-compliant browsers.
39	Last paragraph begins: Let's have a look at floating and clearing in a little more detail. Say you have a picture that you want to float to the right of a block of text. Should read: Let's have a look at floating and clearing in a little more detail. Say you have a picture that you want to float to the left of a block of text.

39-41	Code example reads:
	`.news img {`
	`  float: right;`
	`}`
	`.news p {`
	`  float: left;`
	`}`
	Should read:
	`.news img {`
	`  float: left;`
	`}`
	`.news p {`
	`float: right;`
	`}`

41	Second code example reads:
	`<div class="news">`
	`<img src="news-pic.jpg" />`
	`<p class="clear">Some text</p>`
	`</div>`
	Should read:
	`<div class="news clear">`
	`<img src="news-pic.jpg" />`
	`<p>Some text</p>`
	`</div>`

41	Fourth paragraph begins:

> Using the CSS method, you use the :after pseudo-class in combination with the content declaration to add some content after the specified element.

Should read:

> Using the CSS method, you use the :after pseudo-class in combination with the content declaration to add new content at the end of the specified element’s existing content.

42

Focus point reads:

> *This method works in most modern browsers but fails in Internet Explorer. Various workarounds are available, many of which are documented at* www.positioniseverything.net/easyclearing.html. *However, due to its complexity this method is considered an advanced technique and although it acts as a interesting proof of concept, it probably isn’t a day-to-day solution.*

Should read:

> *This method works in most modern browsers but fails in Internet Explorer 6 and below. Various workarounds are available, many of which are documented at* www.positioniseverything.net/easyclearing.html. *The most common of these involves using the Holly Hack (see chapter 8) to trick IE5-6 into applying “Layout” (see chapter 9) and incorrectly clearing the floats.*

```
.clear {
  display: inline-block;
}

/* Holly Hack Targets IE Win only \*/
* html .clear {height: 1%;}
.clear {display: block;}
/* End Holly Hack */
```

> *However, due to its complexity this method may not be suitable for everybody.*

50

Last paragraph begins:

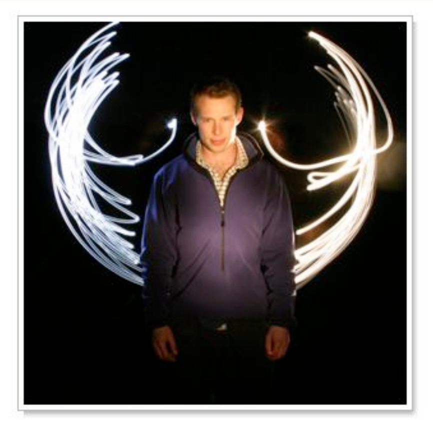
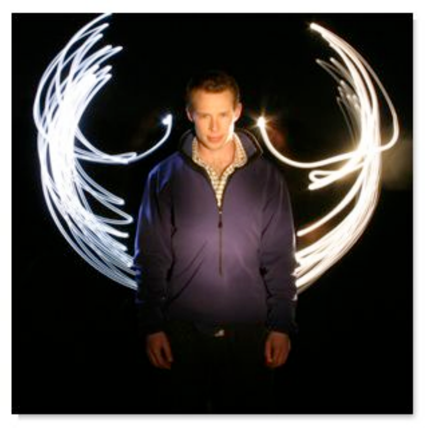
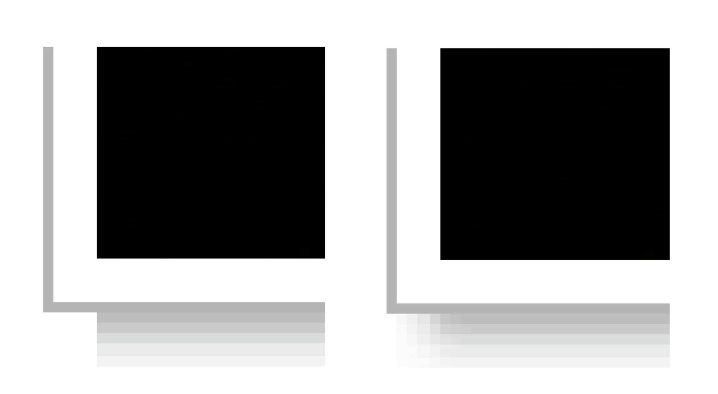
> First you apply the bottom-left.gif to the main box div and bottom-right.gif to the outer div. Next you apply top-left.gif to the inner div and finally top-right.gif to the header.

Should read:

> First you apply the bottom-left.gif to the main box div and bottom-right.gif to the outer div. Next you apply top-left.gif to the inner div and finally top-right.gif to the heading.

55

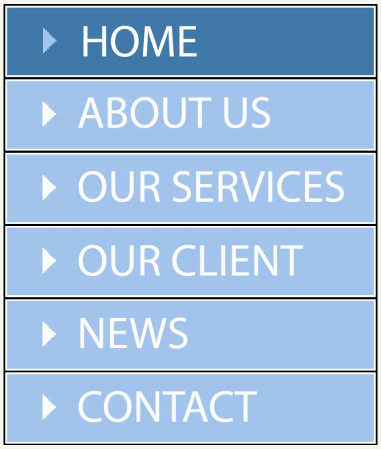
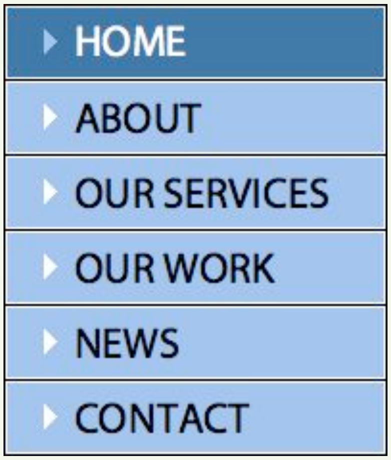
Figure 3-13 looks like this:

	 Should look like this: 
56	Figure 3-14 looks like this:  Should look like this:

61	Paragraph after the callout reads: And that is it. All modern browsers as well as IE 5.5 and above will display a nice, faded-corner drop shadow. Everything else will be presented with a hard-corrner drop shadow. Should read: And that is it. All modern browsers as well as IE 5.5 and above will display a nice, faded-corner drop shadow. Everything else will be presented with a hard-corner drop shadow.
78	Filename in figure 4-9 reads: button.over.gif Should read: button_over.gif

78	First code example contains:
	`a:hover {`
	`background: #369 url(images/button-over.gif) no-repeat left top;`
	`color: #fff;`
	`}`
	Should read:
	`a:hover {`
	`background: #369 url(images/button_over.gif) no-repeat left top;`
	`color: #fff;`
	`}`

79	First paragraph, last sentence reads: This method is known as the Pixy method after the nickname of its creator, Petr Stanícek. Should read: This method is known as the Pixy method after the nickname of its creator, Petr Staníček.

86	First code example reads: <ul> <li>Read emails</li> <li>Write chapter 5</li> <li>Go shopping</li> <li>Cook dinner</li> <li>Watch Scrubs</li> </ul> Should read: <ul> <li>Read emails</li> <li>Write book</li> <li>Go shopping</li> <li>Cook dinner</li> <li>Watch Scrubs</li> </ul>
87	Figure 5.2 looks like this:  HOME ABOUT US OUR SERVICES OUR CLIENT NEWS CONTACT Should look like this:  HOME ABOUT OUR SERVICES OUR WORK NEWS CONTACT
88	Third code example reads:

	`ul a {`
	`display: block;`
	`width: 200px;`
	`height: 40px;`
	`line-height: 40px;`
	`color: #000;`
	`text-decoration: none;`
	`background: #94B8E9 url(images/pixy-`
	`rollover.gif) no-repeat ➥`
	`left middle;`
	`text-indent: 50px;`
	`}`
	Should read:
	`ul a {`
	`display: block;`
	`width: 200px;`
	`height: 40px;`
	`line-height: 40px;`
	`color: #000;`
	`text-decoration: none;`
	`background: #94B8E9 url(images/pixy-`
	`rollover.gif) no-repeat ➥`
	`left center;`
	`text-indent: 50px;`
	`}`

116	Second Paragraph begins: CSS has a border-padding property that allows you to control the spacing between cells. Should read: CSS has a border-spacing property that allows you to control the spacing between cells.

138	In figure 7.2:
	`#subNav`
	Should read:
	`#mainNav`

139	Third code example reads:
	`#content h1, h2, p {`
	`  padding-right: 20px;`
	`}`
	Should read:
	`#content h1, #content h2, #content p {`
	`  padding-right: 20px;`
	`}`

141	First code example reads:
	`#secondaryContent h1, h2, p {`
	`padding-left: 20px;`
	`padding-right: 20px;`
	`}`
	Should read:
	`#secondaryContent h1, #secondaryContent h2, #secondaryContent p {`
	`padding-left: 20px;`
	`padding-right: 20px;`
	`}`

144	Fourth paragraph, last sentence reads: To combat this, it may be worth adding a max-width of 100% to the body tag. max-width isn’t currently supported by IE 6 and below, but it is supported by standards-complaint browsers such as Safari and Firefox. Should read: To combat this, it may be worth adding a max-width of 100% to the body tag. max-width isn’t currently supported by IE 6 and below, but it is supported by standards-compliant browsers such as Safari and Firefox.
148	Last sentence reads: However, on standards-complaint browsers, the image will never get larger than its actual size. Should read: However, on standards-compliant browsers, the image will never get larger than its actual size.
152	Last sentence on the page reads: In the next chapter, you will learn about some of the better-known hacks and how to use them responsiblhy. Should read: In the next chapter, you will learn about some of the better-known hacks and how to use them responsibly.

157	Last code example reads:
	`<!-- [if le IE 6]`
	`<style type="text/css">`
	`@import ("ie.css");`
	`</style>`
	`-->`
	Should read:
	`<!-- [if lt IE 6]`
	`<style type="text/css">`
	`@import ("ie.css");`
	`</style>`

	-->
165	Third paragraph begins: Similar to the !important hack is the underscore hack. By placing an underscore in front of a property, complaint browsers will no longer recognize that property and the declaration will be ignored. Should read: Similar to the !important hack is the underscore hack. By placing an underscore in front of a property, compliant browsers will no longer recognize that property and the declaration will be ignored.
184	Fourth paragraph begins: Luckily, IE 7 promises to be a much more complaint browser. Many of the better known IE bugs have been addressed, along with increased support for advanced CSS 2.1 selectors such as the child and attribute selectors. Should read: Luckily, IE 7 promises to be a much more compliant browser. Many of the better known IE bugs have been addressed, along with increased support for advanced CSS 2.1 selectors such as the child and attribute selectors.
189	Second sentence in the fifth paragraph reads: This is where the descendant selector comes in. Notice that #primaryContent is proceeded by #threeColLayout. Should read: This is where the descendant selector comes in. Notice that #primaryContent is preceded by #threeColLayout.

200	First code example reads:
	`.cbSide`
	`  width:160px;`
	`  margin: 5px 0 0 10px;`
	`  padding: 10px 0 5px 0;`
	`  background: #FFF;`
	`}`
	Should read:
	`.cbside`
	`  width:160px;`
	`  margin: 5px 0 0 10px;`
	`  padding: 10px 0 5px 0;`
	`  background: #FFF;`
	`}`

204	Last rule in last code example reads: `custom_background {`

	`background: #F762E0;`
	`}`
	Should read:
	`.custom_background {`
	`background: #F762E0;`
	`}`

212	First sentence of second paragraph reads: It is widely accepted that CSS declarations for links should be arranged as link, visited, hover, active (or LVHA—“LoVe Hate”). Should read: It is widely accepted that CSS declarations for links should be arranged as link, visited, hover, active (or LVHA—“LoVe HAte”).

225	Third code example on page reads:
	`<div id="container">`
	`<div id="masthead"”></div>`
	`<div id="content"></div>`
	`<div id="sidebar"></div>`
	`</div>`
	`<div id="footer"></div>`
	Third code example on page reads:
	`<div id="container">`
	`<div id="masthead"></div>`
	`<div id="content"></div>`
	`<div id="sidebar"></div>`
	`</div>`
	`<div id="footer"></div>`