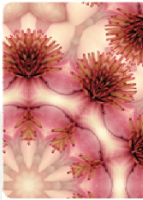


HTML and XHTML
2nd Edition

Tutorial 4

Creating Special Effects with CSS



New Perspectives on HTML and XHTML, Comprehensive

Objectives

- Work with CSS selectors
- Create styles for lists
- Create and apply class styles
- Create a rollover effect
- Create a drop cap using a pseudo-element
- Manage page layout with CSS
- Work with overflow and clipping styles

New Perspectives on HTML and XHTML, Comprehensive

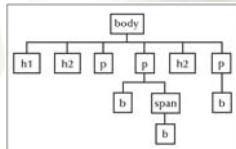
Objectives

- Create stacked objects
- Apply styles to various media
- Hide elements from printing
- Create and apply printer styles
- Create and prohibit page breaks for printing

New Perspectives on HTML and XHTML, Comprehensive

Working with Selector Patterns

- On a Web page, elements are nested within other elements, forming a hierarchical tree structure



```
graph TD; body --> h1; body --> h2; body --> p; body --> p; body --> h2; body --> p; p --> b; p --> span; span --> b;
```

New Perspectives on HTML and XHTML, Comprehensive

Working with Selector Patterns

- To take advantage of this tree structure, CSS allows you to create **contextual selectors** that express the location of an element within the hierarchy of elements

parent descendant {styles}

```
li b {color: blue}
li b, h2 {color: blue}
#notes b {color: blue}
* {color: blue}
p > b {color: blue}
```

New Perspectives on HTML and XHTML, Comprehensive

Working with Selector Patterns

- On occasion you might also need to select elements based on their attribute values

element[att] {styles}

```
a[href] {color: blue}
```

New Perspectives on HTML and XHTML, Comprehensive

Class Notes: Topic 3 – Creating Special Effects with CSS

Using Selector Patterns

- To apply a style to all elements in the document, use the * selector
- To apply a style to a single element, use the e selector, where e is the name of the element
- To apply a selector to a descendant element, f, use the e f selector, where e is the name of the parent element and f is an element nested within the parent
- To apply a selector to a child element, f, use the e > f selector, where e is the name of a parent element and f is an element that is a direct child of the parent
- To apply a selector to a sibling element, use the e + f selector, where e and f are siblings and f immediately follows e in the document tree

New Perspectives on HTML and XHTML, Comprehensive

7

Working with Selector Patterns

New Perspectives on HTML and XHTML, Comprehensive

8

Working with Selector Patterns

- Conditional comments** allow you to apply different HTML code for different versions of Internet Explorer

```
<!-- [if condition IE version]>
HTML code
<![endif]-->
```

New Perspectives on HTML and XHTML, Comprehensive

9

Applying Styles to Lists

- To specify the list marker displayed by the browser, you can apply the style `list-style-type: type`

list-style-type	Marker (s)
disc	●
circle	○
square	□
decimal	1, 2, 3, 4, ...
decimal-leading-zero	01, 02, 03, 04, ...
lower-roman	i, ii, iii, iv, ...
upper-roman	I, II, III, IV, ...
lower-alpha	a, b, c, d, ...
upper-alpha	A, B, C, D, ...
none	no marker displayed

New Perspectives on HTML and XHTML, Comprehensive

10

Applying Styles to Lists

New Perspectives on HTML and XHTML, Comprehensive

11

Applying Styles to Lists

- Most browsers place the list marker to the left of the block, lining up the markers with each list item

`list-style-position: position`

New Perspectives on HTML and XHTML, Comprehensive

12

Class Notes: Topic 3 – Creating Special Effects with CSS

Working with Classes

- The class attribute is used when you want to identify elements that share a common characteristic
`<elem class="class"> ... </elem>`
- You can use the class attribute to assign the same style to multiple elements sharing the same class value
`.class {styles}`

New Perspectives on HTML and XHTML, Comprehensive

13

Using Pseudo-Classes and Pseudo-Elements

- A **pseudo-class** is a classification of an element based on its current status, position, or use in the document
`selector:pseudo-class {styles}`

Pseudo-class	Description	Example
link	The link has not yet been visited by the user	a:link {color: red}
visited	The link has been visited by the user	a:visited {color: green}
active	The link is in the process of being activated by the user	a:active {color: yellow}
hover	The mouse pointer is hovering over the link	a:hover {color: blue}
focus	The element has received the focus of the keyboard or mouse pointer	input:focus {background-color: yellow}
first-child	The element is the first child of its parent	p:first-child {text-indent: 0}
lang	Specifies the language to be used with the element	q:lang(FR) {quotes: '«' '»'}

New Perspectives on HTML and XHTML, Comprehensive

14

Using Pseudo-Classes and Pseudo-Elements

```
#links ul {float: left; width: 200px;}
#links ul li {float: left; width: 100px; margin-left: 15px; padding-left: 0px;}
#links ul li a {float: left; width: 100px; margin-left: 15px; padding-left: 0px;}
#links a:visited {text-decoration: none;}
#links a:active {text-decoration: underline;}
#links a:active {text-decoration: none;}
```



New Perspectives on HTML and XHTML, Comprehensive

15

Using Pseudo-Classes and Pseudo-Elements

- **Pseudo-elements** are abstracted from what we know of an element's content, use, or position in the document
`selector:pseudo-element {styles}`

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

New Perspectives on HTML and XHTML, Comprehensive

16

Positioning Objects with CSS



New Perspectives on HTML and XHTML, Comprehensive

17

Positioning Objects with CSS

- Create div containers for each note
- Add a class attribute to apply a common set of styles
- Each note should have a unique id

```
<title>Samples from online scrapbooks</title>
<link href="scrap.css" rel="stylesheet" type="text/css" />
<style type="text/css">
  .notes {font-family: sans-serif; font-size: 8pt; color: brown;
  background-color: ivory; border: 3px inset rgb(212, 212, 212);
  width: 130px}
  .notes p {margin: 3px}
</style>
</head>
```

New Perspectives on HTML and XHTML, Comprehensive

18

Positioning Objects with CSS

- **CSS-P (CSS-Positioning)** became part of the specification for CSS2, and positioning styles were some of the first CSS2 styles to be adopted by browsers
position: type; top: value; right: value; bottom: value; left: value;
- **Absolute positioning** enables you to place an element at specific coordinates either on a page or within a containing element
position: absolute; left: 100px; top: 50px

New Perspectives on HTML and XHTML, Comprehensive

19

Positioning Objects with CSS

- **Relative positioning** is used to move an element relative to its default position on the page
position: relative; left: 100px; top: 50px
- You can fix an element at a specific spot in the document window while the rest of the page scrolls by setting the value of the position style to fixed
- You can assign the inherit position style to an element so that it inherits the position value of its parent element

New Perspectives on HTML and XHTML, Comprehensive

20

Positioning Objects with CSS

```
<style type="text/css">
.notes {font-family: aana-serif; font-size: 8pt; color: brown;
background-color: ivory; border: 3px inset rgb(211, 212, 212);
width: 130px;}
.notes p {margin: 5px;}
#note1 {position: absolute; left: 600px; top: 120px;}
#note2 {position: absolute; left: 170px; top: 400px;}
#note3 {position: absolute; left: 570px; top: 150px;}
</style>
```

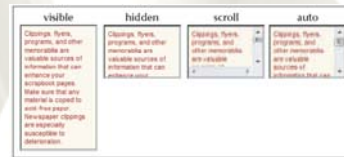


New Perspectives on HTML and XHTML, Comprehensive

21

Working with Overflow and Clipping

- If you want to force an element into a specified height and width, you have to define how the browser should handle a situation where content overflows the space allotted to the object
overflow: type

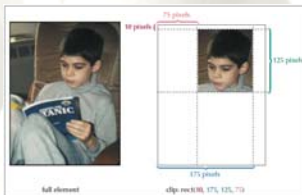


New Perspectives on HTML and XHTML, Comprehensive

22

Working with Overflow and Clipping

- The clip style allows you to define a rectangular region through which the element's content can be viewed
clip: rect(top, right, bottom, left)

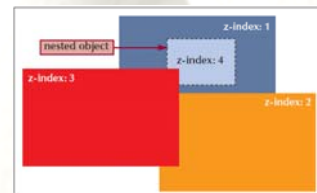


New Perspectives on HTML and XHTML, Comprehensive

23

Stacking Elements

- Elements placed using CSS positioning are stacked on top of elements that are not
- To specify a different stacking order, use the style:
z-index: value



New Perspectives on HTML and XHTML, Comprehensive

24

Working with Different Media

- By default, a style sheet is applied to all devices, and each device must determine how best to match the styles to its own requirements
`<style type="text/css" media="type">`

...

```
</style>
```

or

```
<link href="url" type="text/css" media="type" ... />
```

Working with Different Media

Value	Used for
all	All output devices (the default)
aural	Speech and sound synthesizers
braille	Braille tactile feedback devices
embossed	Paged Braille printers
handheld	Small or handheld devices with small screens, monochrome graphics, and limited bandwidth
print	Printers
projection	Projectors
screen	Computer screens
tty	Fixed-width devices like teletype machines and terminals
tv	Television-type devices with low resolution, color, and limited scrollability

Working with Different Media

- You can use a single style sheet broken down into different sections for each media type

```
@media screen { body {font-size: 1em} h1 {font-size: 2em} }
```

```
@media print { body {font-size: 12pt} h1 {font-size: 16pt} }
```

```
@media handheld { body {font-size: 8pt} h1 {font-size: 12pt} }
```

```
@media tv { body {font-size: 16pt} h1 {font-size: 24pt} }
```

Working with Different Media

- CSS uses **media groups** to describe how different media devices render content
 - Continuous or paged
 - Visual, aural, or tactile
 - Grid (for character grid devices) or bitmap
 - Interactive (for devices that allow user interaction) or static (for devices that allow no interaction)

Working with Different Media

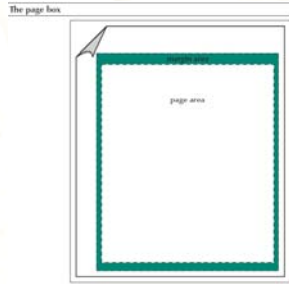
Media type	continuous/ paged	visual/aural/ tactile	grid/bitmap	interactive/ static
aural	continuous	aural	N/A	both
braille	continuous	tactile	grid	both
embossed	paged	tactile	grid	both
handheld	both	visual	both	both
print	paged	visual	bitmap	static
projection	paged	visual	bitmap	static
screen	continuous	visual	bitmap	both
tty	continuous	visual	grid	both
tv	both	visual, aural	bitmap	both

Hiding Elements

- CSS has two styles that you can use to keep an element from being displayed in the output: the display style and the visibility style
visibility: *type*
- The *type* attribute can be set to visible, hidden, collapse, or inherit

Using Print Styles

- CSS defines printed pages by extending the box model



Using Print Styles

- The general rule to create and define a page box is:
`@page {styles}`
- Printed media can vary in size and orientation
- The size style allows the Web author to define the default dimensions and orientation of the printed page
size: *width height orientation*

Working with Page Breaks

- When a document is sent to the printer, the printer decides the location of the page breaks unless that information is included as part of the print style
page-break-before: *type*
page-break-after: *type*

Working with Page Breaks

- The *type* style attribute has the following values:
 - Always
 - Avoid
 - Left
 - Right
 - Auto
 - Inherit

Working with Page Breaks

- A **widow** occurs when only a few ending lines of an element appear at the top of a page
- An **orphan** is just the opposite: it occurs when only a few beginning lines of an element appear at the bottom of a page
- The styles to control the appearance of widows and orphans in the printout are:
widow: *value*
orphan: *value*

Summary

- Worked with CSS selectors
- Created styles for lists
- Created and applied class styles
- Created a rollover effect
- Created a drop cap using a pseudo-element
- Managed page layout with CSS
- Worked with overflow and clipping styles

Summary

- Created stacked objects
- Applied styles to various media
- Hid elements from printing
- Created and applied printer styles
- Created and prohibited page breaks for printing