

# **Cascading Style Sheets, part III**

---

Web Design in a Nutshell, Third Edition

Chapters 19, 20, and 21

---

# Objectives

---

- Understand the CSS visual formatting model
  - Use the CSS box model
  - Use the margin properties
  - Use the padding properties
  - Use the border properties
  - Use the special box properties
  - Apply special box properties
-

# The CSS Visual Formatting Model

---

- Describes how the element content boxes should be displayed by the browser
    - Based on the hierarchical structure of the HTML document and element display type
    - Elements fall into three display type categories: Block, Inline and lists
-

# The CSS Visual Formatting Model

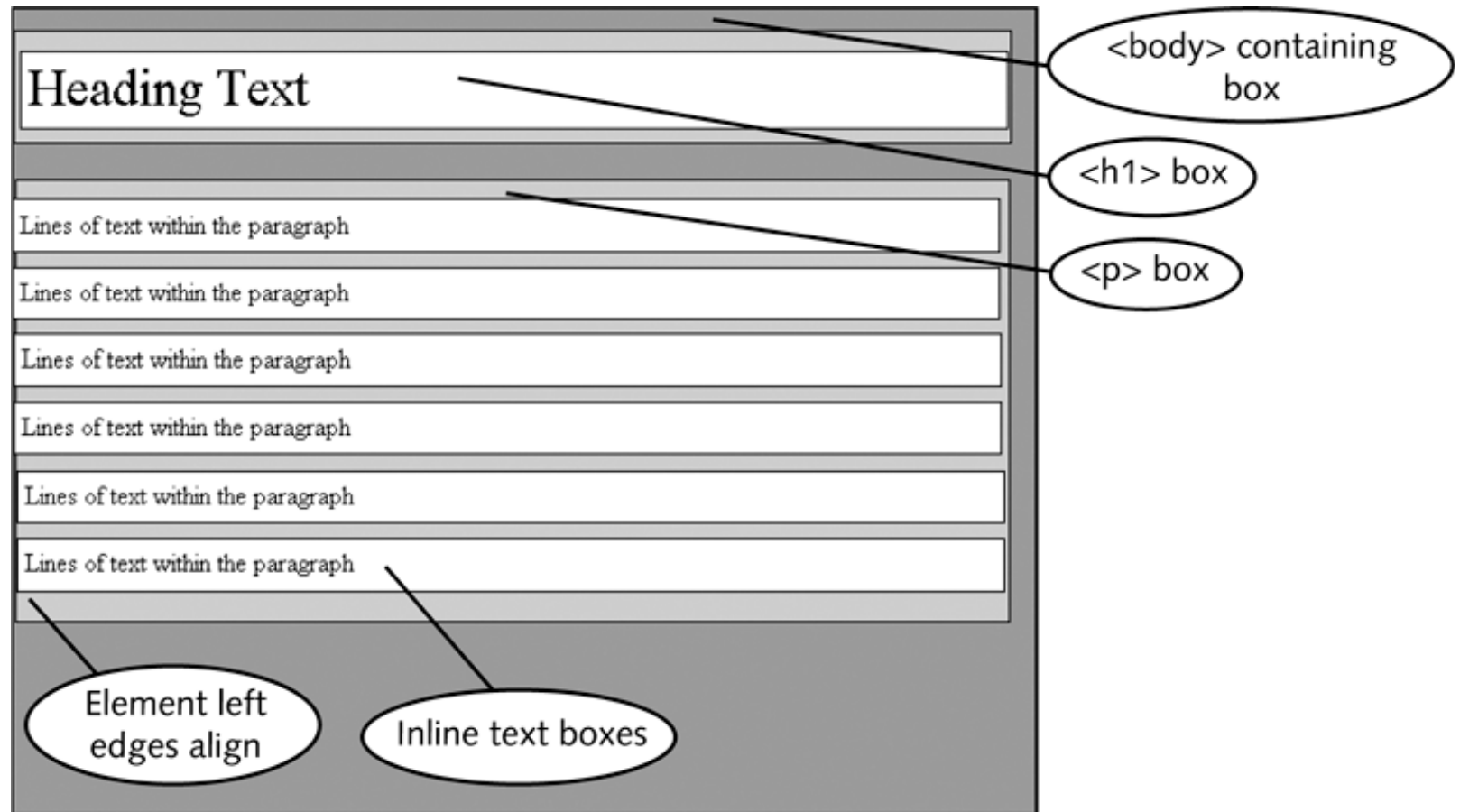


Figure 9-1 The CSS visual formatting model

# Specifying Element Display Type

---

- The CSS **display** property determines the display type of an element.
- The following style rule changes the default display type for an `<h1>` element from block to inline:

```
h1 {display: inline;}
```

---

# Specifying Element Display Type

---

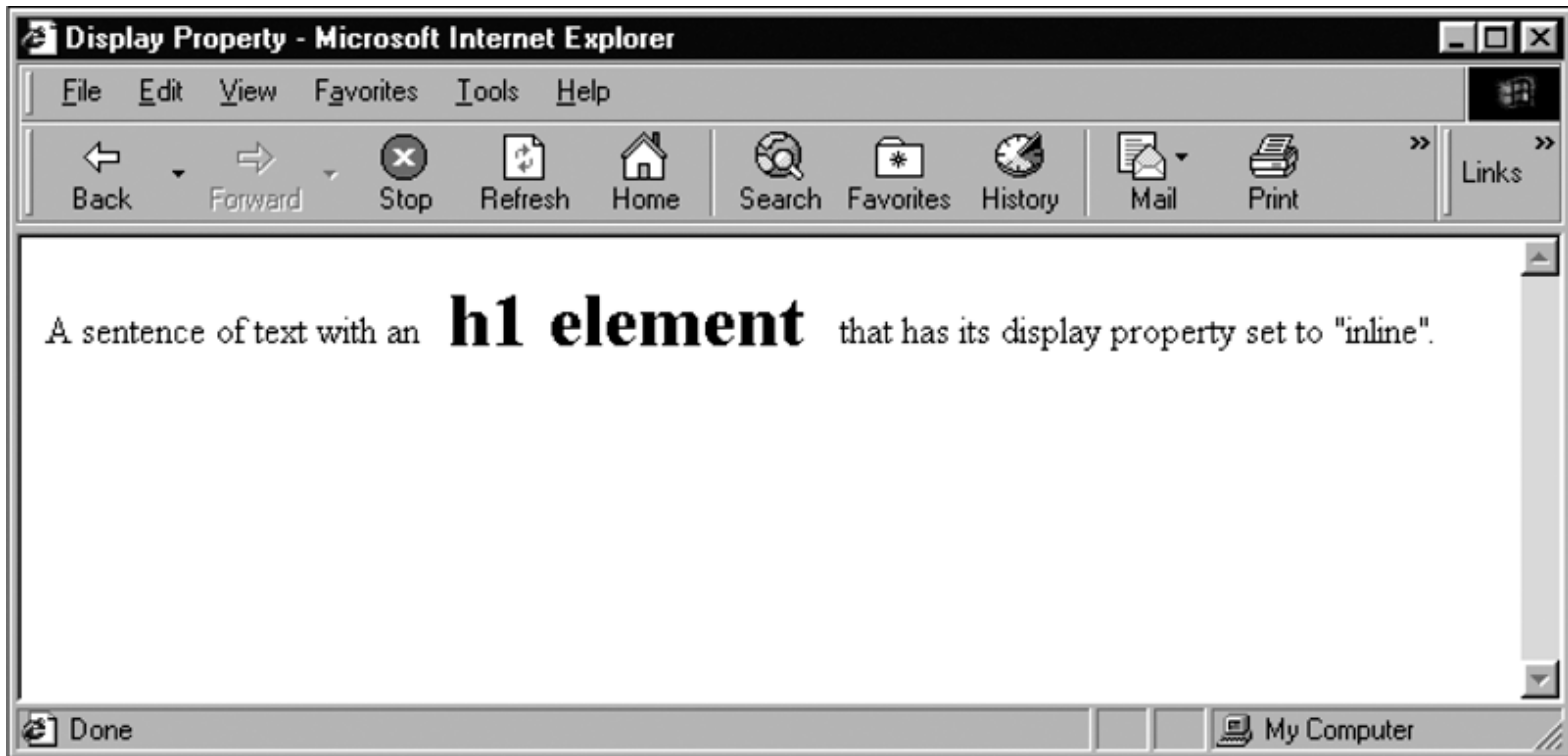


Figure 9-2 Manipulating the display property

---

# Specifying Element Display Type

---

- The CSS **display** property can be used in different style sheets to display or hide content.
- For instance, you might hide a block on a stylesheet meant for printing:

```
#header { display: none; }
```

---

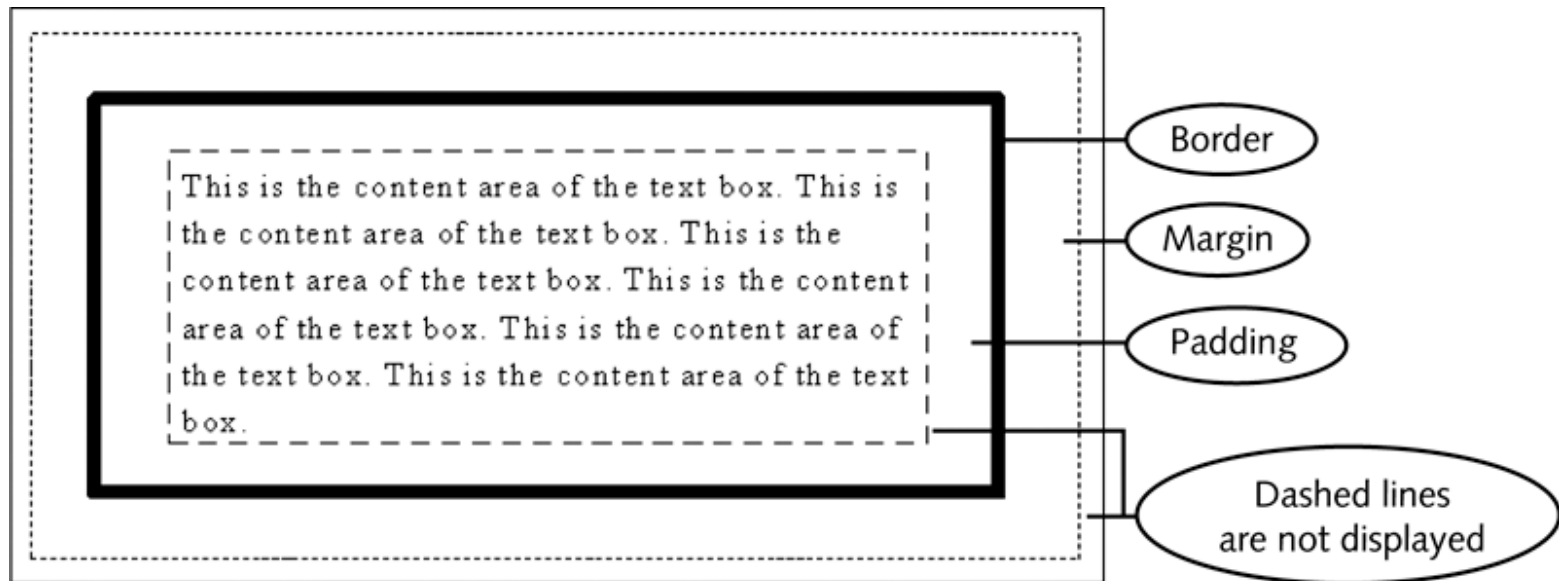
# Using the CSS Box Model

---

- Describes the rectangular boxes that contain content on a Web page
  - Each block-level element created is displayed as a box containing content in the browser window
  - Each content box can have margins, borders and padding (specified individually)
-

# CSS Box Model

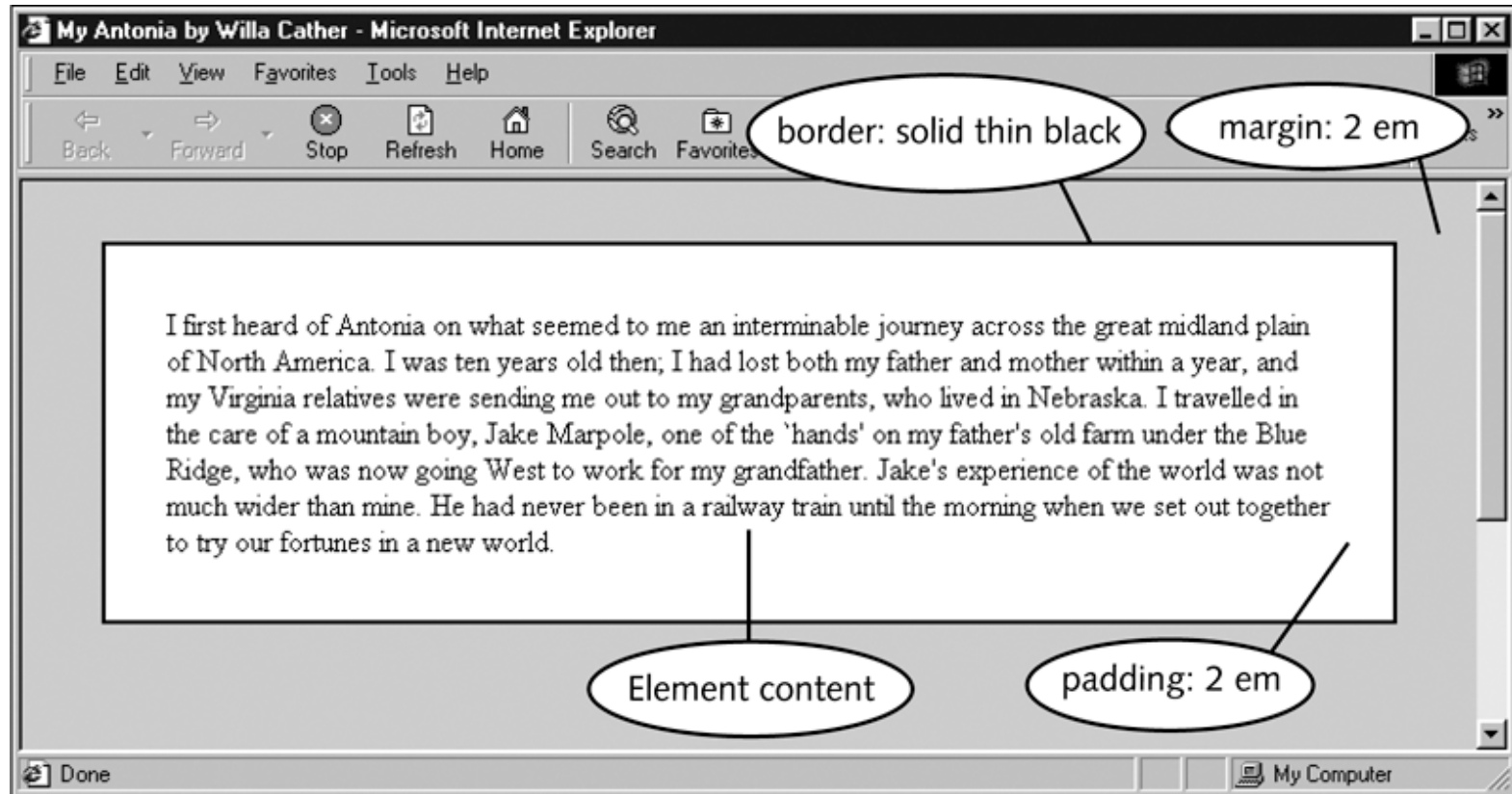
---



**Figure 9-3** The CSS box model

---

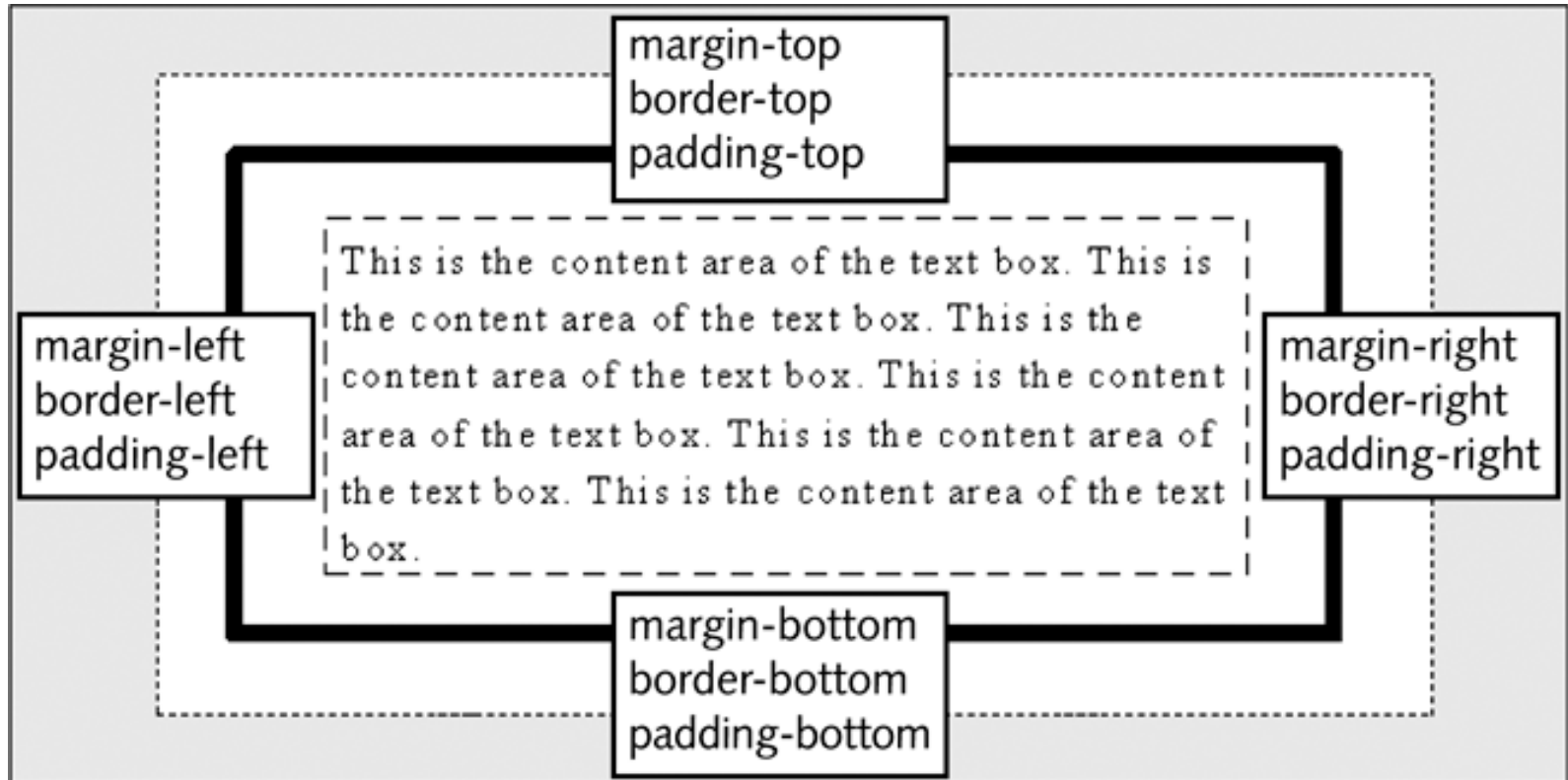
# CSS Box Model



**Figure 9-4** The CSS box model areas in a <p> element

# CSS Box Model

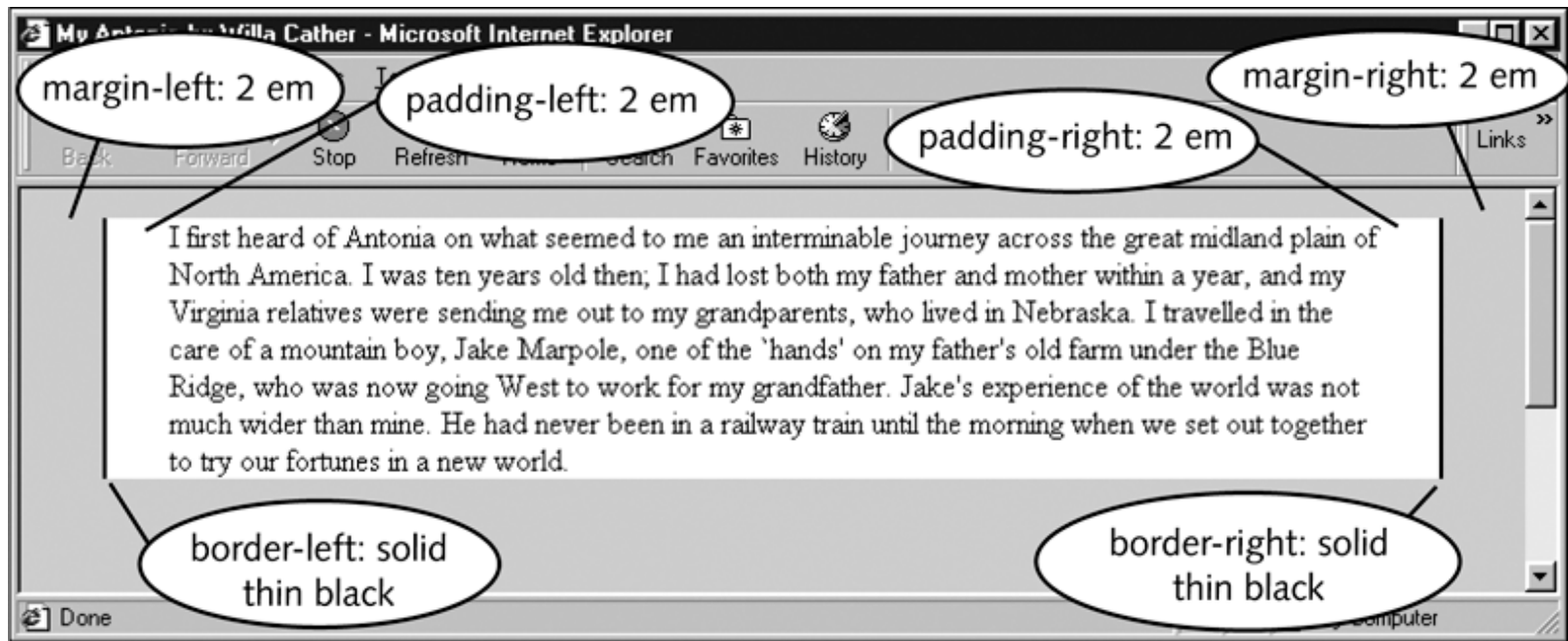
---



**Figure 9-5** The CSS box model individual sides

---

# CSS Box Model



**Figure 9-6** The CSS box model individual sides in a <p> element

# Measurement Values

---

- Margin, border and padding properties allow two types of measurement:
    - **Length**
      - Absolute or relative values, e.g. 1em, 2px, 12pt
    - **Percentage**
      - Based on width of containing box, e.g. 25%
-

# The Margin Property

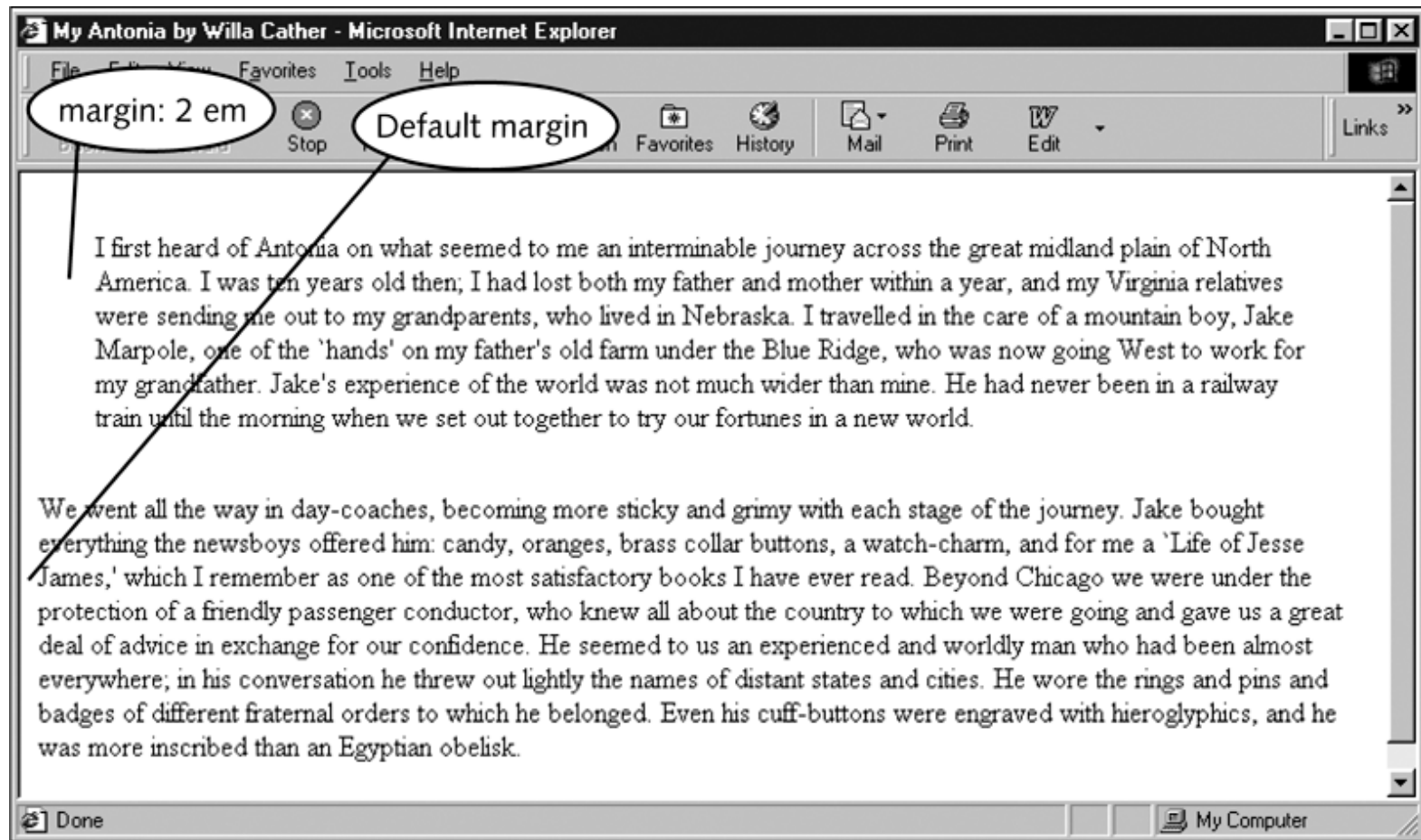
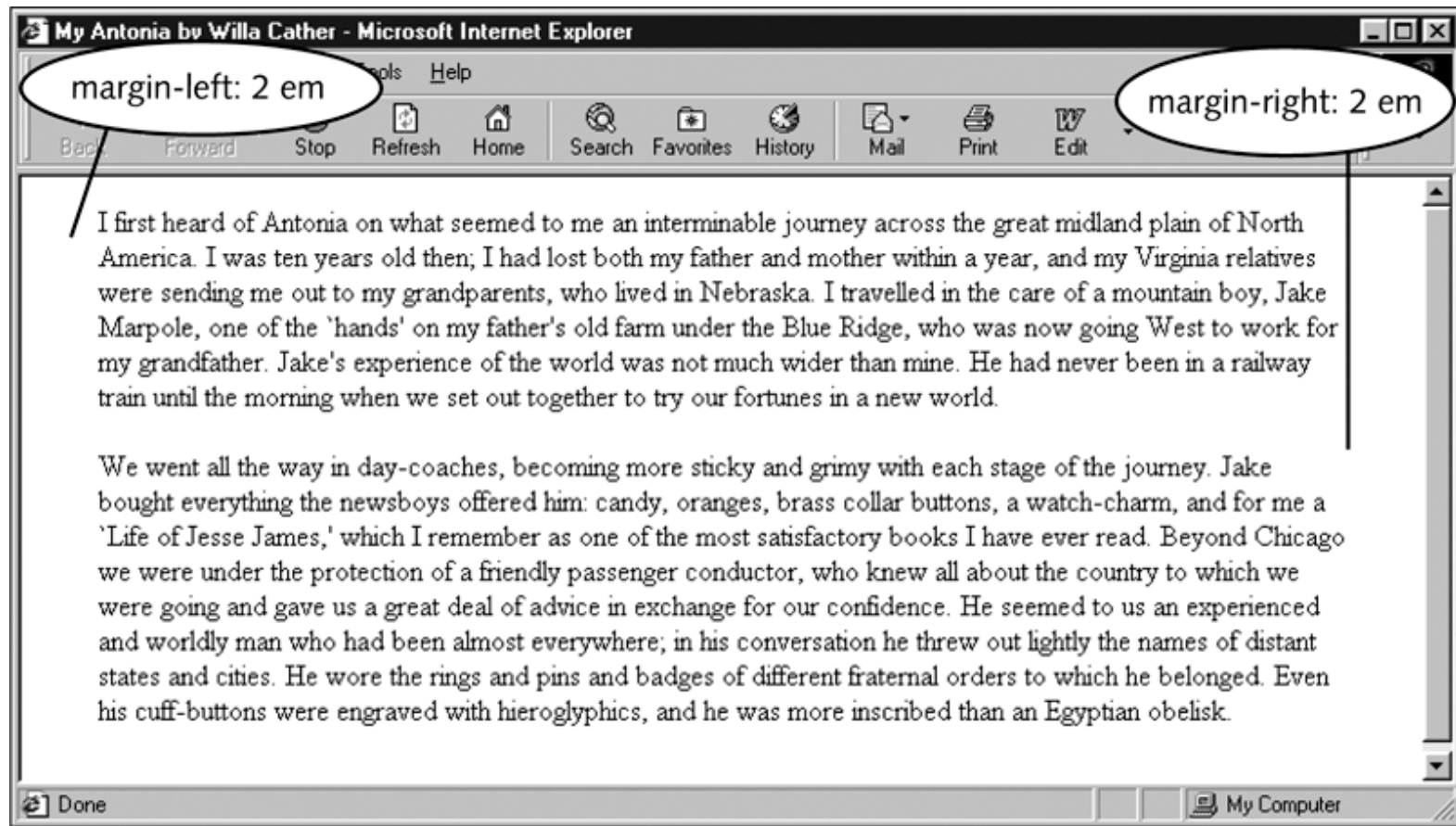


Figure 9-7 Using the margin property

# Left, Right, Top and Bottom margins



**Figure 9-8** Using the individual margin properties

# Negative Margins

---

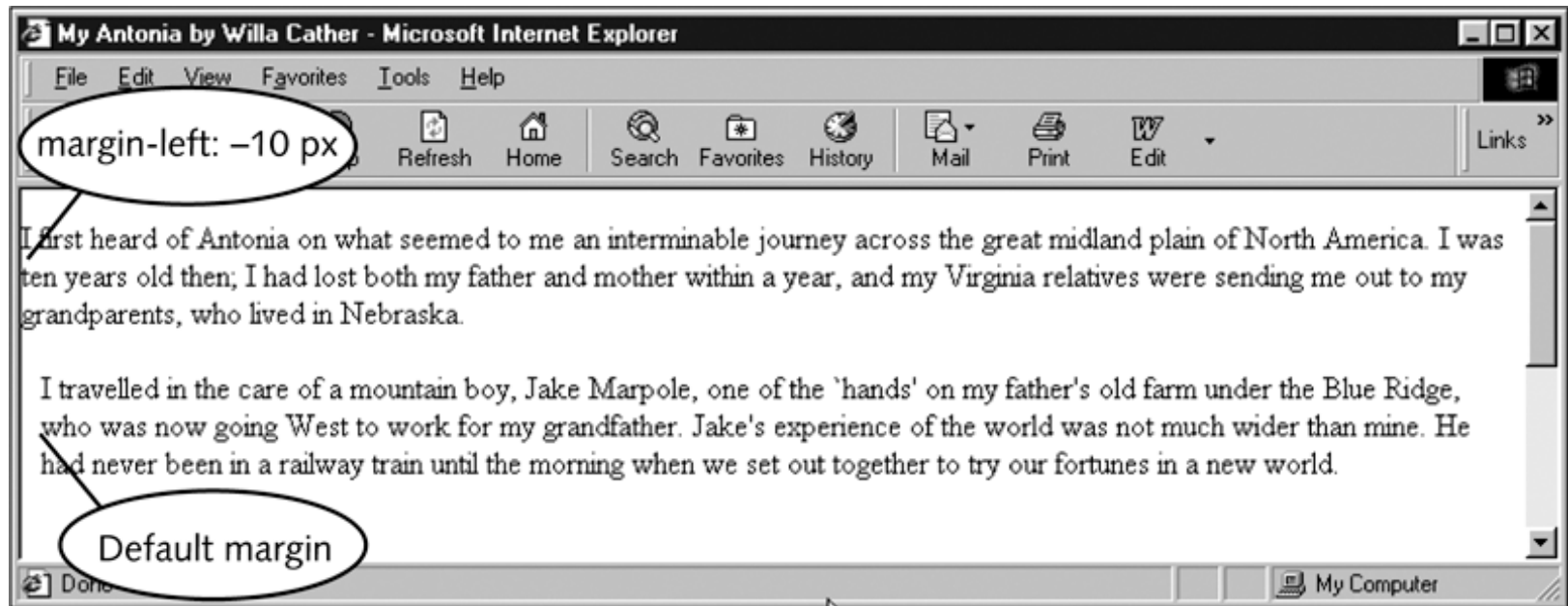
- Negative margins can be set to achieve special effects
  - Example: Hide some text, but leave it on the page to help with accessibility and search engine rankings

**#header h1 {margin-left: -1em;}**

- Can also be used to remove the default margins from other elements
-

# Negative Margins

---



**Figure 9-9** A `<p>` element with negative left margin

---

# Negative Margins

---

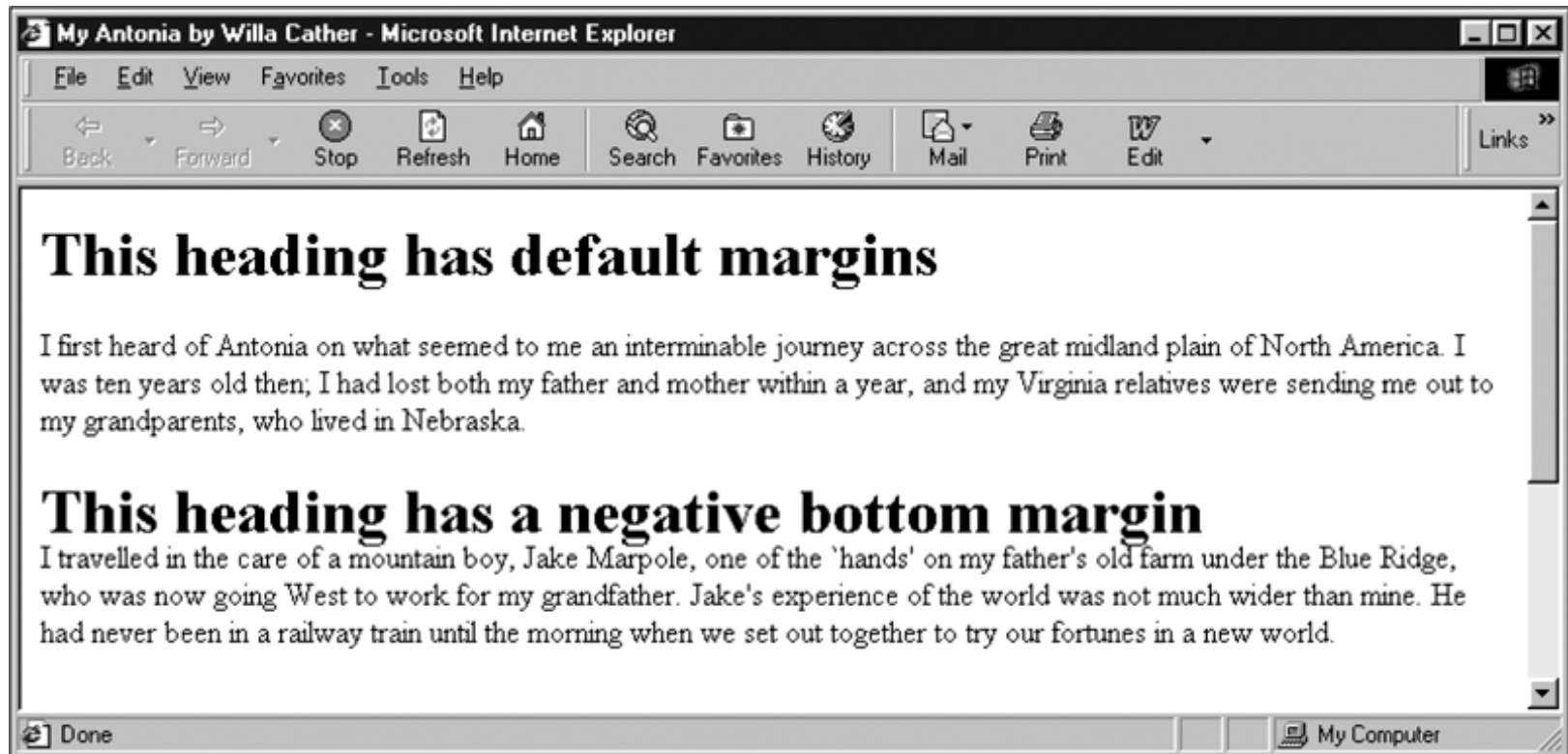


Figure 9-10 An `<h1>` element with negative bottom margin

---

# The Padding Property

---

- Control the padding area in the box model
    - Area between the element content and the border
  - Padding area inherits the background color of the element
  - If a border is added to an element, padding should be adjusted to increase legibility and enhance the presentation
-

# The Padding Property

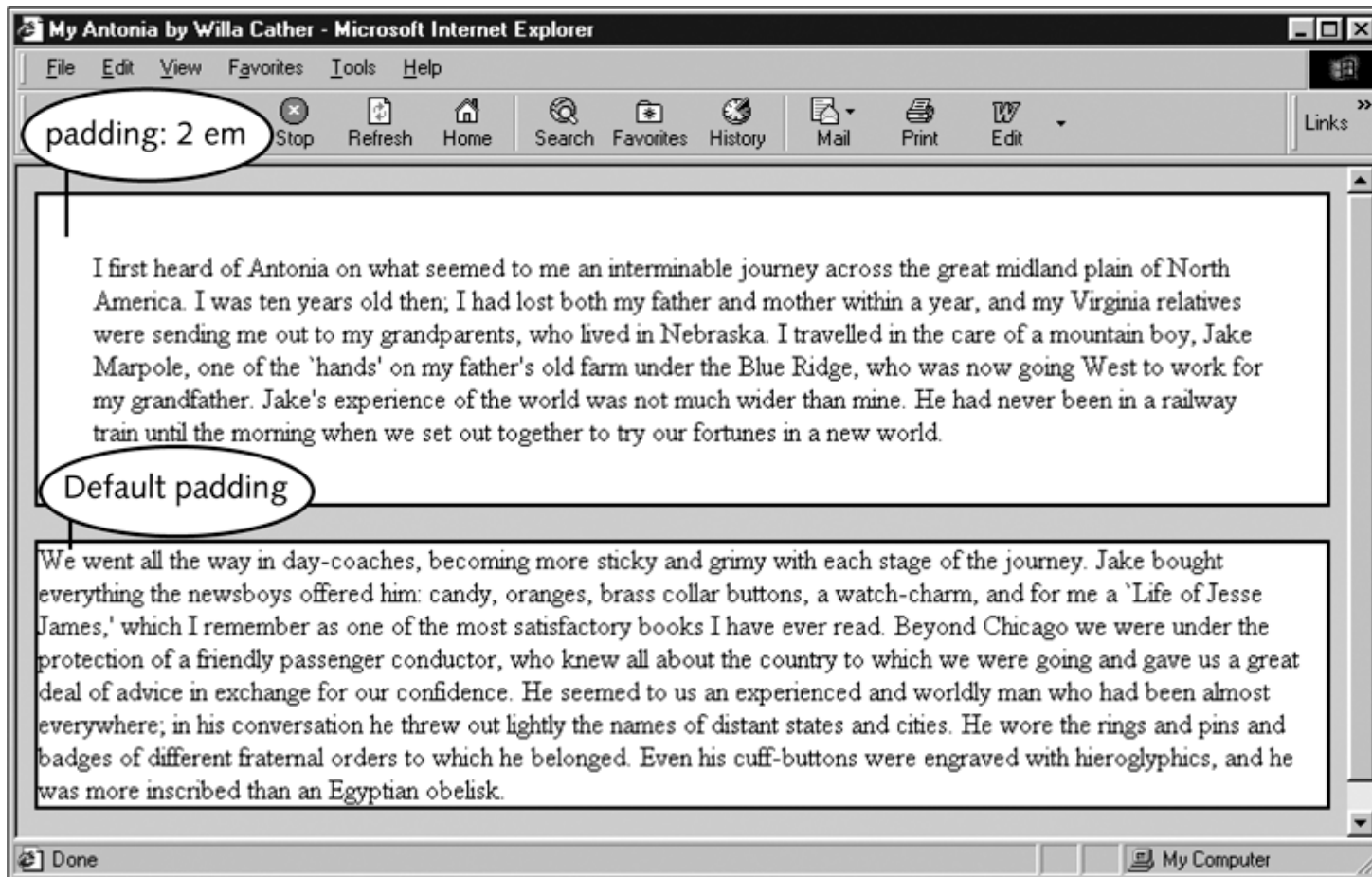
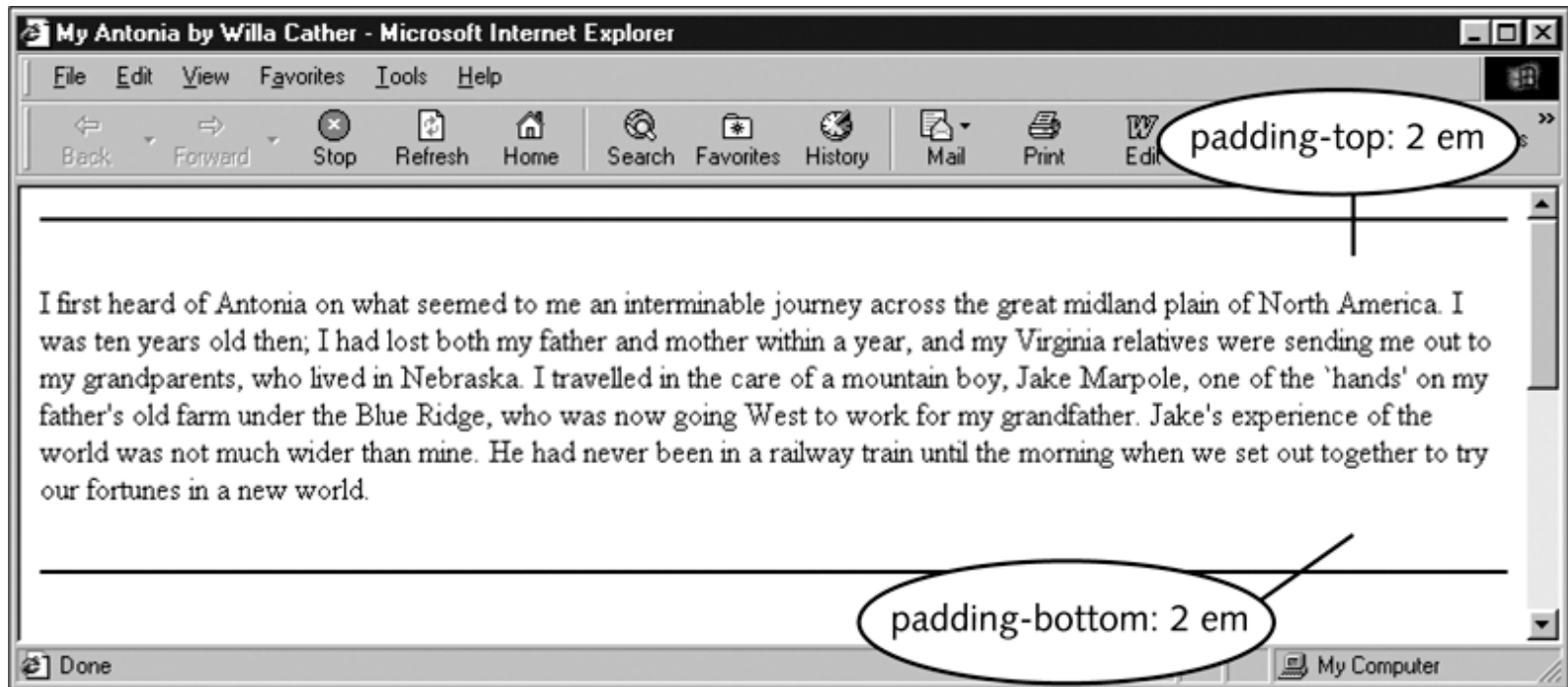


Figure 9-12 Default padding and 2-em padding

# Left, Right, Top and Bottom padding



**Figure 9-13** Using the individual padding properties

# Border Properties

---

- Control the appearance of borders around elements
    - Area between the margin and the padding
  - There are five basic border properties:
    - border
    - border-left
    - border-right
    - border-top
    - border-bottom
-

# Specifying Border Style

---

- Border style keywords:
  - **none** — no border on the element (default)
  - **dotted** — dotted border
  - **dashed** — dashed border
  - **solid** — solid line border
  - **double** — double line border
  - **groove** — 3-D embossed border
  - **ridge** — 3-D outward extended border
  - **inset** — 3-D inset border (entire box)
  - **outset** — 3-D outset (entire box)
-

# Specifying Border Style

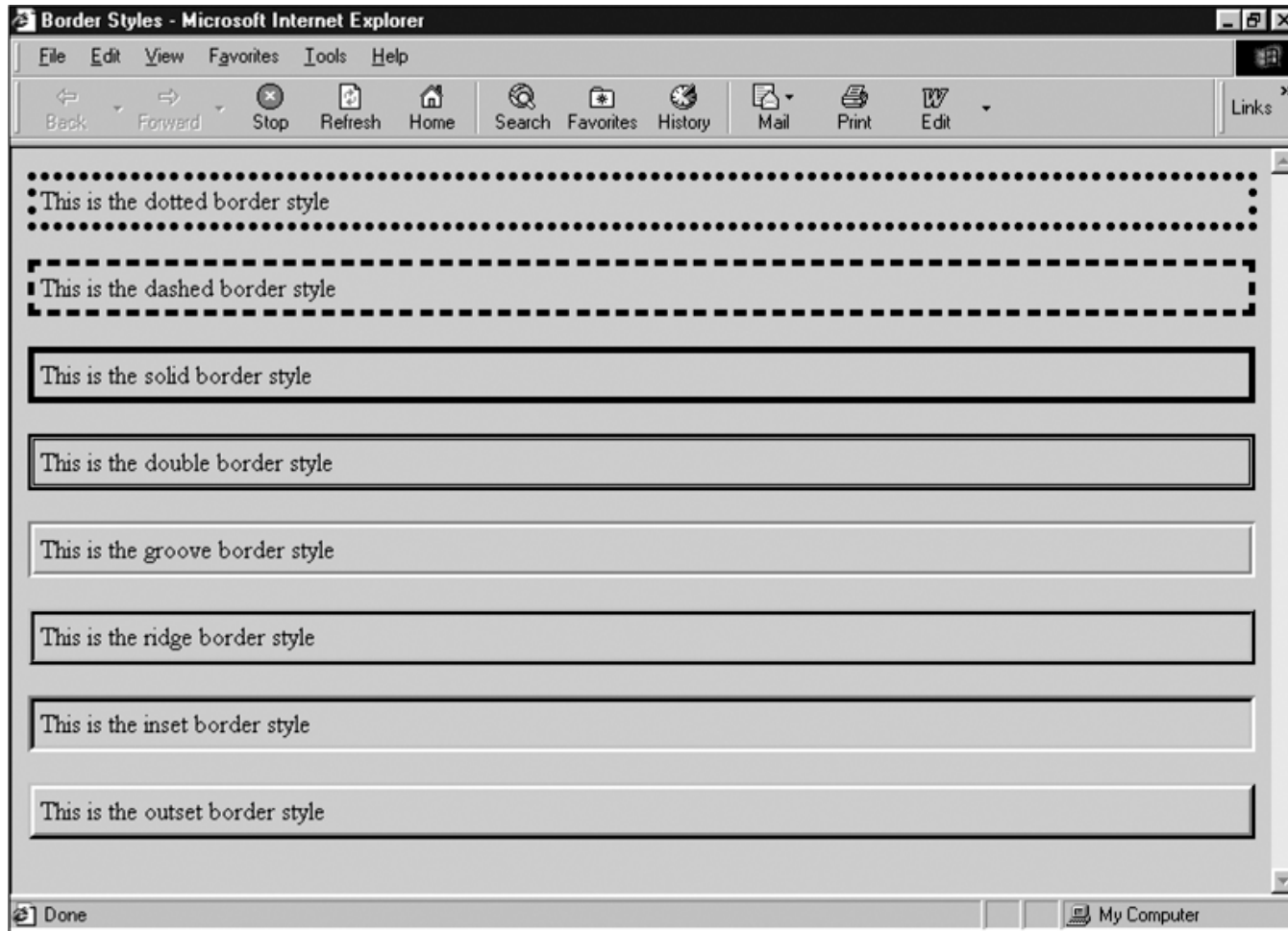
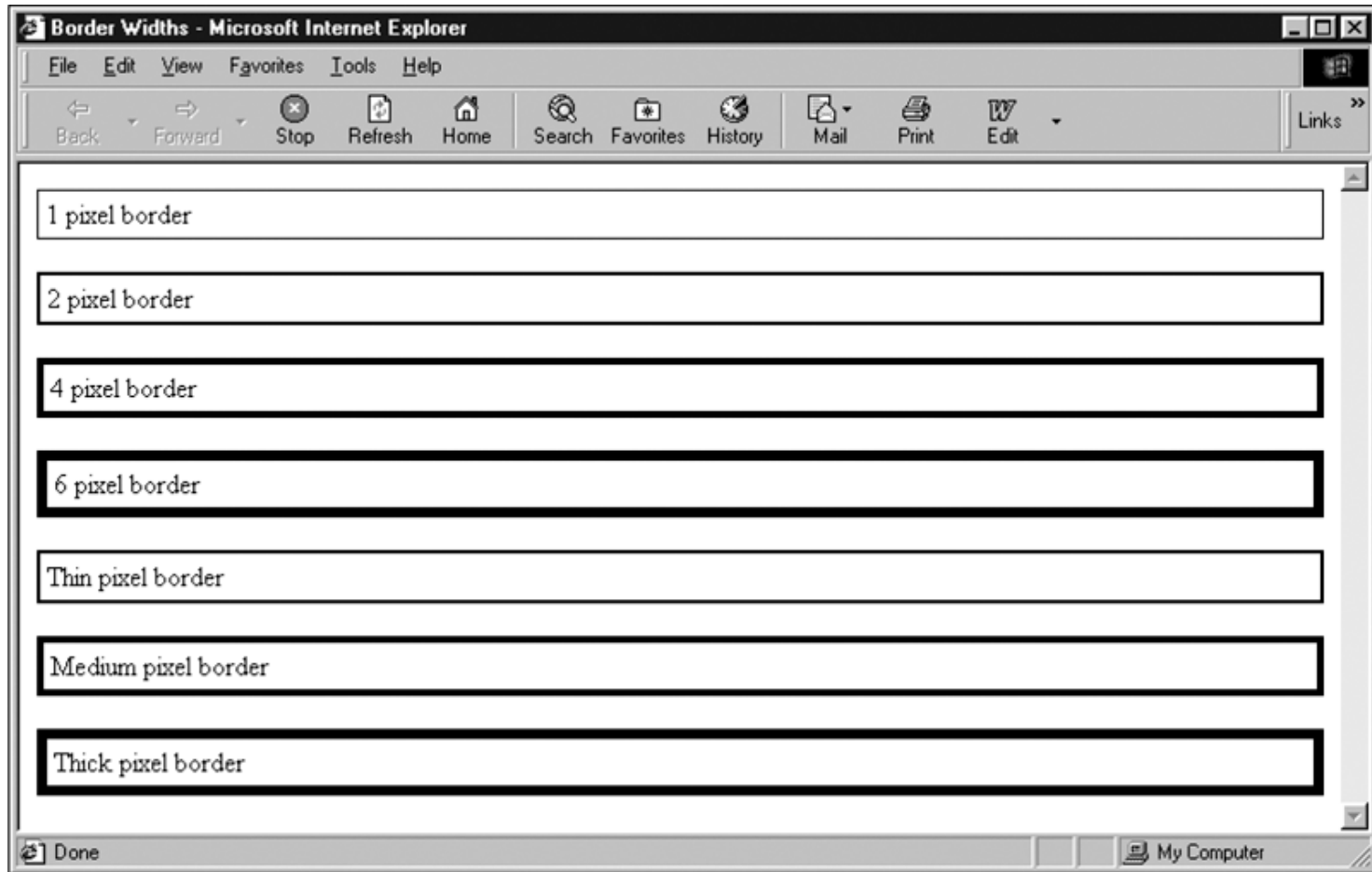


Figure 9-14 Different border styles

# Specifying Border Style



**Figure 9-15** Different border widths

# Specifying Border Color

---

- Allows setting of element border color
- To set a border color, use the property as shown in the following rule:

```
p {  
    border-color: red;  
    border-width: 1px;  
    border-style: solid;  
}
```

---

# Using the Special Box Properties

---

- Allow floating of images or boxes to the left or right of content on the page
  - Special box properties:
    - **width**
    - **height**
    - **float**
    - **clear**
-

# Width

---

- Sets the horizontal width of an element
- Width is not intended for normal block-level elements, but you can use it to create floating text boxes or with images.

```
div {width: 200px;}
```

---

# Height

---

- Sets the vertical height of an element
- Like width, height is not intended for normal block-level elements, but you can use it to create floating text

**div {height: 150px;}**

---

# Float

---

- The float property lets you position an element to the left or right edge of its parent element.
  - Float is most commonly used for `<img>` elements, allowing alignment of an image to the left or right of text.
  - You can also use the float property to align a text box to the left or right edge of its parent.
-

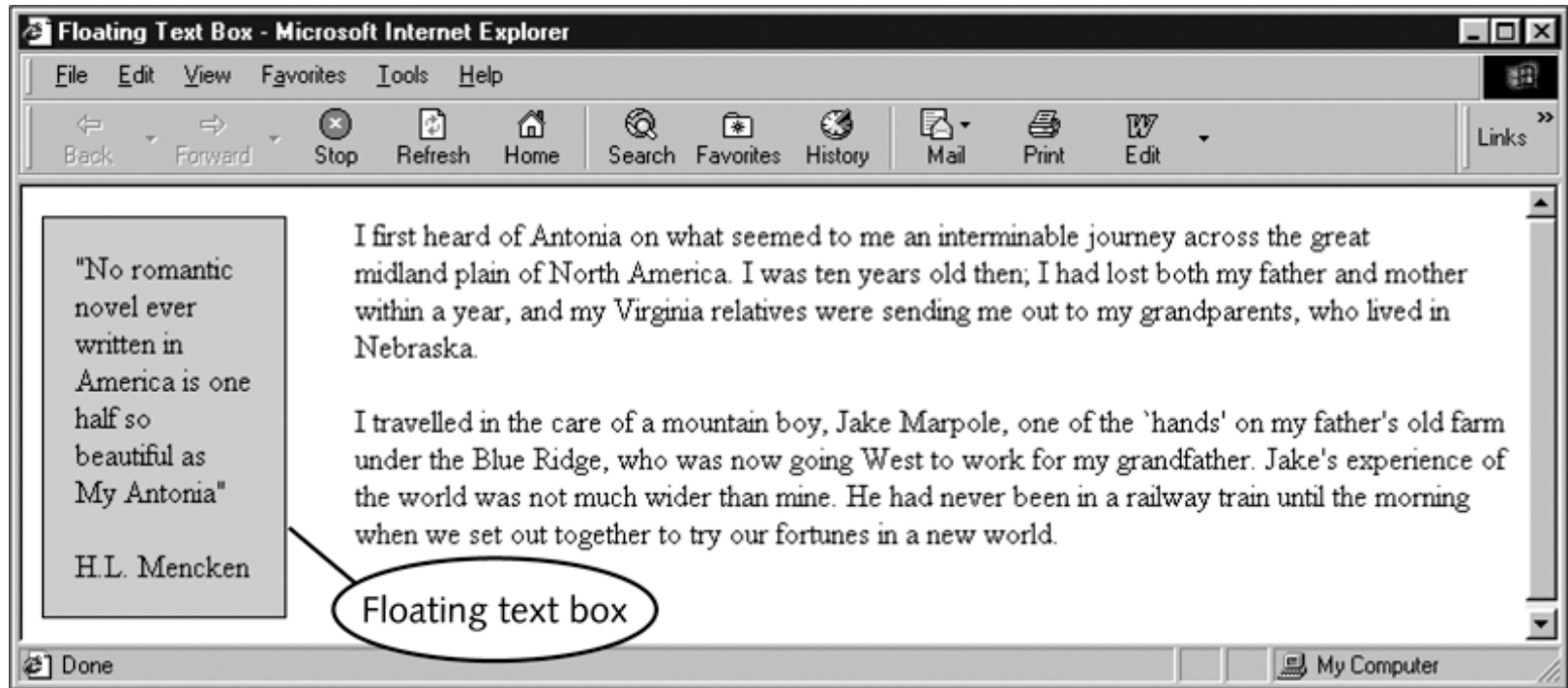
# Float

---

- Float property can be used to float a text box to the left or right of text
  - Advantage: tables aren't required, just a single simple CSS style rule is all that is required
-

# Using the Float Property

---



**Figure 9-16** A floating text box

---

# Clear

---

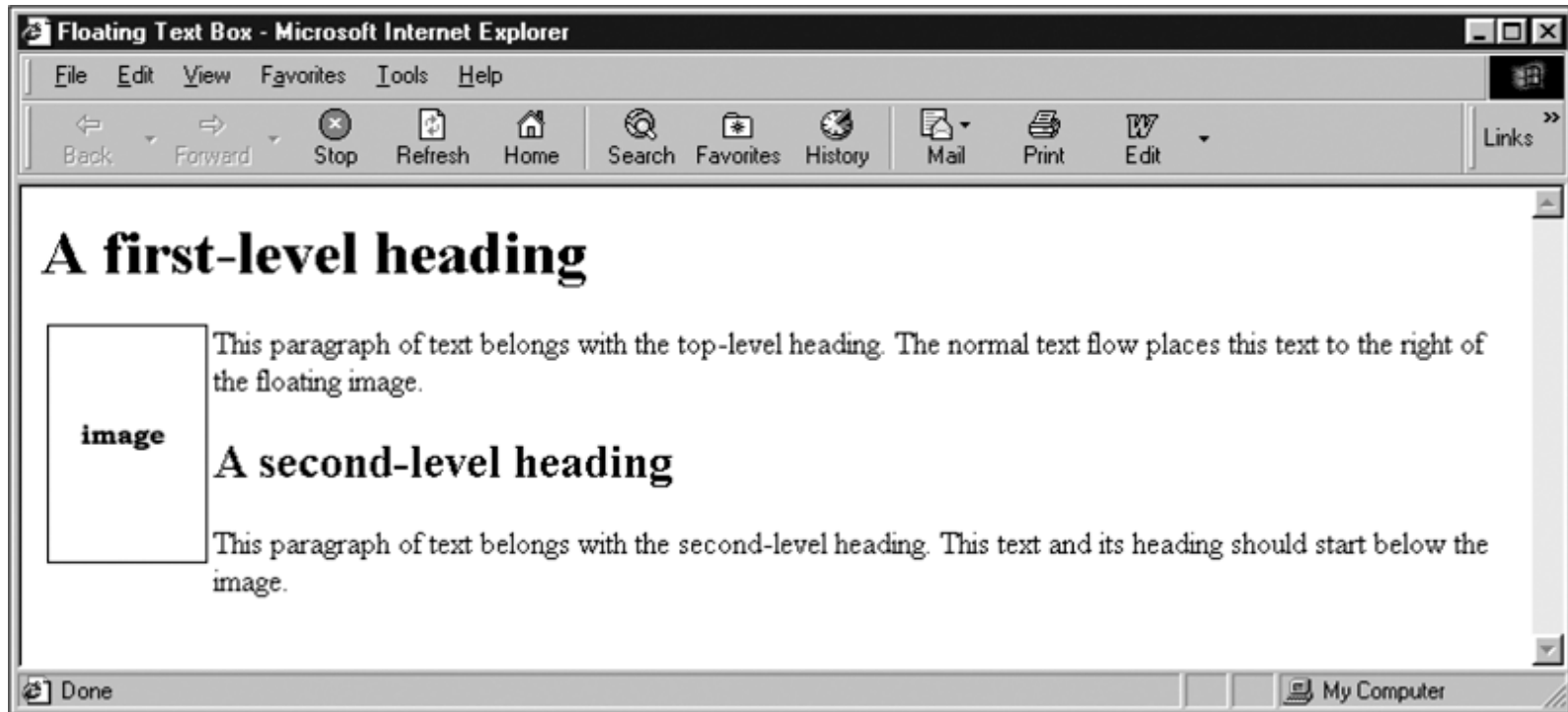
- Controls the flow of text around floated elements
- You only use the clear property when you are using the float property
- Clear lets you force text to appear beneath a floated element, rather than next to it

**`<h2 style="clear: left;">`**

---

# Normal Text Flow

---

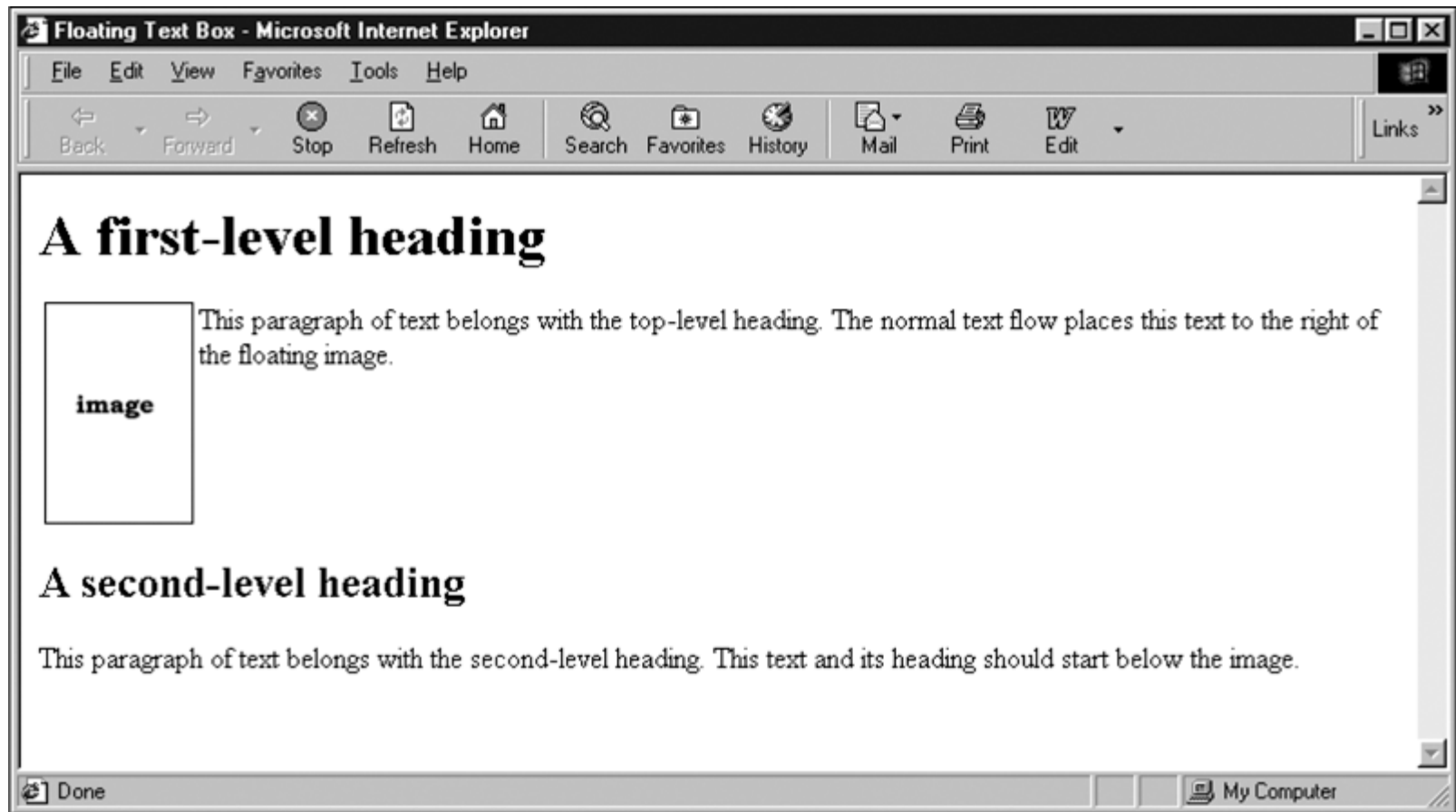


**Figure 9-18** Normal text flow around a floating image

---

# Using Clear

---



**Figure 9-19** Using the clear property

---

# CSS and Images

---

- Images can be used as the background of a div or other element

```
#header { background: #4c9f2c url(images/bg.gif); }
```

- Images are like any other block element in that they have margins and can be floated left or right:

```
#header img { margin-left: 10px; float: right; }
```

- Images can have borders:

```
#header img { border: 1px solid #000; }
```

---

# CSS and Images

---

**Table 7-4** CSS properties that replace <img> attributes

Deprecated Attribute	Equivalent CSS Property
align	float allows you to flow text around an image or other object; for example: <code>img {float: left;}</code>
border	border lets you set a border on an image or remove the border from a linked image
vspace and hspace	The padding or margin properties set white space around an image. You can control individual sides of the image, or apply white space around the entire image.

# Color and Backgrounds

---

- Web Colors are best defined using hexadecimal values that represent Red, Green, and Blue values, aka RGB, as opposed to the CMYK or Cyan, Magenta, Yellow and Black colors used for printing
- A handful of colors — **red**, **orange**, **blue**, etc. — are supported, but results can vary by browser
- Colors are used to set foreground and background colors

```
body { color: black; background-color: white}
```

---

# Hex Colors

---

- HTML uses hexadecimal numbers to express RGB color values
- Hexadecimal color values are six-digit numbers; the first two define the red value, the second two define the green, and the third two define the blue
- Browser safe hexadecimal colors are always made up of the following 2-digit color values: 00, 33, 66, 99, CC, and FF

# CSS Background Properties

---

- Cascading Style Sheets allow you more control over backgrounds than standard HTML
- To apply a background image, use the `<body>` element as the selector with the background property
- The CSS background-repeat property allows you to set how a background image tiles tiling completely across the page, e.g. repeat-x, repeat-y, no-repeat.
- You can also specify fixed position backgrounds

# Using Backgrounds

---

- One of the simplest ways to work with hexadecimal color is to specify a background color for your pages
- Use the CSS background shortcut to set the background-color, background-image, background-repeat and background-position properties with body as the selector

```
body { background: #CCC url(images/  
bg.gif) repeat-x top; }
```

# CSS and Images

---

- Images can be used as the background of a div or other element

```
#header { background: #ff6600 url(images/bg.gif); }
```

- Images are like any other block element in that they have margins and can be floated left or right:

```
#header img { margin-left: 10px; float: right; }
```

- Images can have borders:

```
#header img { border: 1px solid #000; }
```

---

# Summary

---

- Margins are transparent; padding has the same background color as the content;
  - Border properties allow the addition of borders to individual sides or all sides of an element
  - Special box properties allow the creation of floating images or text boxes
  - Always design for readability
-

# Summary

---

- Web colors are usually expressed in hexadecimal values representing red, green and blue.
  - CSS Allows for more flexibility than html when it comes to background images
  - Using the body tag as a selector in your style sheet will allow you to set a background color or image for the entire page
  - Look at your designs on multiple devices, e.g. Mac, Windows, Linux, etc.
-