



Formatting Your Web Page

“*Digital design is like painting, except the paint never dries.* —Neville Brody



This chapter covers

- Styling the text font, size, and style
- Aligning and indenting paragraphs
- Adding text and background colors

You now know how to display important and emphasized text, create links and headings, and display items in bulleted or numbered lists, but although these important techniques give your web page a bit of visual interest, they won't win you any design awards. To get people to sit up and take notice of your page, you need to concentrate on the CSS side of things for a bit, and that's what you'll do in this chapter. First, you'll learn a few ways to style your web page text, including specifying the typeface you want to use and setting the size of the text. You'll also learn how to apply bold to any text (not only important terms or keywords), as well as how to add italic to any text (not only emphasized words or alternative terms). From words and phrases, you jump to paragraphs, learning how to align text horizontally and indent paragraph text. The chapter closes on a colorful note as you learn how to apply CSS colors to text and to the page background.



Formatting Your Web Page

As you'll see, these basic CSS techniques are straightforward to learn and implement, but don't let their inherent simplicity fool you. These are powerful tools that you'll use over and over to make your pages look great and to give them your personal touch. Those design awards are right around the corner.

Styling Text

Each browser uses default styles to render text such as headings and paragraphs. Although some differences exist among browsers, for the most part these styles are rendered similarly in Google Chrome, Mozilla Firefox, Apple Safari, and so on. These styles are perfectly good design choices, but if you use these default styles, you run the risk of having your web page end up with a default look. That's the last thing you want as a web page designer, so one of your most important tasks is to override those defaults and specify your own text formatting.

Web typography is a huge, fascinating topic that you'll learn in depth in Chapter 14. For now, I'll keep things simple by focusing on four of the most important text-formatting features: typeface, type size, bolding, and italics.

Styling the Typeface

I like to describe fonts as the architecture of characters. When you examine a building, certain features and patterns help you identify the building's architectural style. A flying buttress, for example, usually is a telltale sign of a Gothic structure. Fonts, too, are distinguished by a unique set of characteristics, and those characteristics are embodied in the typeface.

A *typeface* is a distinctive design that's common to any related set of letters, numbers, and symbols. What's the difference between a typeface and a font? For all practical purposes, the two terms are interchangeable. For all impractical purposes, however, a *font* is a particular implementation of a typeface, meaning the typeface as rendered with a specific size, weight, and style. Helvetica is a typeface; Helvetica 16-point bold is a font.

The typeface design gives each character a shape and thickness that's unique to the typeface and difficult to classify. Five main categories serve to distinguish most typefaces you'll come across in your web-design career:



Serif—A *serif* (rhymes with *sheriff*) typeface contains fine cross strokes (called *feet*) at the extremities of each character. These subtle appendages give the typeface a traditional, classy look, but they can get lost when displayed on a screen at small sizes.

Playground

Sans serif—A *sans-serif* typeface doesn't contain cross strokes on the extremities of characters. These typefaces usually have a clean, modern look that's well suited to screen text, particularly at small sizes.

Playground

Monospace—A *monospace* typeface (also called a *fixed-width* typeface) uses the same amount of space for each character, so skinny letters such as *i* and *l* take up as much space as wider letters such as *m* and *w*.

Playground

Cursive—The *cursive* typefaces are designed to resemble handwritten pen or brush writing.

Playground

Fantasy—*Fantasy* typefaces usually are fanciful designs that have some extreme elements (such as being extra-thick).

Playground

USE IT

On a screen, *serif* usually works best for headings and other text set at large sizes; *sans serif* makes good body text; *monospace* works well for code listings; *cursive* is best for short bits of text that require elegance or playfulness; and *fantasy* should be used only when a special effect is required.

In CSS, you tell the web browser which typeface you want to apply to an element by using the `font-family` property. You have several ways to set the `font-family` value, but I begin by looking at the method that requires the least amount of work.

Lesson 4.1: Specifying a Generic Font

Covers: The `font-family` property and generic fonts

⇒ Online: wdpg.io/4-1-0

The simplest way to use `font-family` is to specify a *generic font*, which is a standard font implemented by all modern web browsers. There are five generic font families, and their names correspond to the five typeface categories discussed in the preceding section: *serif*, *sans serif*, *monospace*, *cursive*, and *fantasy*. The following example puts the `font-family` property through its paces.



Formatting Your Web Page

► Example

➡ Online: wdpg.io/4-1-1

This example shows you how to use the `font-family` property to apply the sans-serif generic font to the `h3` element and the serif generic font to the `p` element.

WEB PAGE	<p>The Web Design Playground</p> <p>Why work towards web design proficiency when you can play your way there?</p> <p>The h3 element</p> <p>The p element</p>
CSS	<pre>h3 { font-family: sans-serif; } p { font-family: serif; }</pre> <p>The h3 element gets the sans-serif generic font.</p> <p>The p element gets the serif generic font.</p>
HTML	<pre><h3>The Web Design Playground</h3> <p>Why work towards web design proficiency when you can play your way there?</p></pre>

Generic fonts are useful because they're supported by all web browsers, but with only five font families, they lack variety. If you'd like a bit more choice for your web page text, you need to access a broader collection of fonts.

Lesson 4.2: Specifying a System Font

Covers: The `font-family` property and system fonts

REMEMBER

Using quotation marks and capitalizing the first letter of each word in a system font name are optional, but they're good habits to get into because they make your code more readable.

➡ Online: wdpg.io/4-2-0

Besides the built-in generic fonts, each web browser can access the fonts that a site visitor has installed on her computer. Most computers have the serif typeface Times New Roman installed, for example, so your web page could use that typeface instead of the generic serif font. These installed typefaces are known as *system fonts*.

When you specify a system font, here are two things to keep in mind:

- If the font name includes one or more spaces, numbers, or punctuation characters other than a hyphen (-), surround the name with quotation marks:

```
font-family: "Times New Roman";
```



- Capitalize the first letter (or, for multiword names, capitalize the first letter of each word):

```
font-family: Georgia;
```

Note that it's perfectly legal to specify more than one font name as long as you separate the names with commas. In that case, the browser checks the fonts in the order in which they appear and uses the first one that's installed on the user's computer. This arrangement is useful because you can't be sure which system fonts each user has installed. In particular, it's good practice to include a similar generic font family after the system font. If you specify a serif system font such as Times New Roman or Georgia (or both), for example, include the serif generic font as the last item in the font-family value:

```
font-family: "Times New Roman", Georgia, serif;
```

The following example applies the Verdana system font to the div element, which (as you might recall from Chapter 2) is the element you use to divide the web page content into separate sections.

LEARN

To get the installation percentages for many popular system fonts, see <https://www.cssfontstack.com>.

REMEMBER

Some system fonts are installed on at least 90 percent of both Macs and Windows PCs. For sans-serif, these fonts are Arial, Arial Black, Tahoma, Trebuchet MS, and Verdana. For serif, these fonts are Georgia and Times New Roman. For monospace, this font is Courier New.

► Example

⇒ Online: wdpg.io/4-2-1

This example applies the Verdana system font to the div element and adds the sans-serif generic font as a backup.

WEB PAGE	<p><u>The clean, modern look of a sans serif typeface makes it ideal for web page text.</u></p> <p>↖ The div element</p>
CSS	<pre>div { font-family: Verdana, sans-serif; }</pre> <p>└ The div element gets the Verdana system font.</p>
HTML	<pre><div> The clean, modern look of a sans serif typeface makes it ideal for web page text. </div></pre>



Formatting Your Web Page

Lesson 4.3: Setting the Type Size

Covers: The font-size property

REMEMBER

You can specify font sizes in units other than pixels. I take you through all the available CSS units in Chapter 7.

⇒ Online: wdpg.io/4-3-0

In the same way that the web browser defines a default typeface for each element, it defines default type sizes, particularly for the heading elements h1 (largest) through h6 (smallest). Again, these defaults are usually reasonable, but I'm going to urge you to forget about the defaults and set up your own type sizes. Why? One of the secrets of good web design is assuming control of every aspect of the design, which is the only way to be sure that the web page looks the way you or your client wants it to look. One of your main jobs as a web page designer is to set your own type sizes not only for headings, but also for all your page elements, including body text, captions, sidebars, and navigation.

You specify the type size of an element by setting the font-size property equal to a value in pixels, which you indicate with the unit px. The example that follows tells the web browser to render all text that appears within a div element at a text size of 24 pixels. By comparison, the example also shows some text within a p element displayed in the default size, which in all modern browsers is 16 pixels.

► Example

⇒ Online: wdpg.io/4-3-1

This example formats the div element with a text size of 24 pixels.

WEB PAGE	<p>From Milan to Markup ← The h1 element</p> <p>The strange-but-true story of one woman's epic journey from fashion designer to web geek. ← The div element</p> <p>Hyperia Marcupala always loved design, but one day she discovered she'd rather work with pixels than pleats. ← The p element</p>
CSS	<pre>div { font-size: 24px; }</pre> <p>← The div element is given a font size of 24px.</p>



HTML

```
<h1>From Milan to Markup</h1>

<div>
The strange-but-true story of one woman's epic journey from
fashion designer to web geek.
</div>

<p>
Hyperia Marcupala always loved design, but one day she discovered
she'd rather work with pixels than pleats.
</p>
```

Working with Text Styles

When you have your typeface picked out and your page elements set up with different type sizes, you're well on your way to making typographically pleasing web pages. But to make your pages stand out from the herd, you need to know two more CSS properties related to styling text. The next couple of sections take you through these styles.

Lesson 4.4: Making Text Bold

Covers: The font-weight property

⇒ Online: wdpg.io/4-4-0

In Chapter 2, you learned that you can display text as bold by using the `` tag or the `` tag. You use these tags when the affected text has semantic significance: The `strong` element is for important text, whereas the `b` element is for keywords. But what if you have text that doesn't fit into either of these semantic categories, but you want it to appear bold anyway for the sake of appearance? In that case, you can turn to the CSS property font-weight. Table 4.1 lists the weights and keywords you can assign to this property.

USE IT

Nonsemantic uses for bold text include a title used at the beginning of each item in a bulleted list, the lead words or the lead sentence in a paragraph, and contact information.

BEWARE

Not all the values in Table 4.1 work in all systems. If whatever typeface you're using doesn't support one or more of the weights, specifying that weight won't have any effect.



Formatting Your Web Page

FAQ

When would I ever use the normal (or 400) value? When you're working with an element that defaults to bold styling, such as a heading. To prevent such an element from appearing with bold text, assign its `font-weight` property a value of normal (or 400).

► **Table 4.1** Possible Values for the `font-weight` Property

Weight	Keyword	Description
100		Thin text
200		Extra light text
300		Light text
400	normal	Regular text
500		Medium text
600		Semibold text
700	bold	Bold text
800		Extra-bold text
900		Black text

The following example gives you a taste of what bold text looks like by applying the weights 100, 400, and 700 to several span elements. (Recall from Chapter 2 that you use span to create an inline container that applies to a word or three.)

► Example

➡ Online: wdpg.io/4-4-1

This example demonstrates the weights 100, 400, and 700 of the Calibri typeface by applying each weight to a separate span element.

WEB PAGE



CSS

```
span {  
  font-family: Calibri, sans-serif;  
  font-size: 5em;  
}
```

HTML

```
<span style="font-weight: 100">A</span>  
<span style="font-weight: 400">A</span>  
<span style="font-weight: 700">A</span>
```

The span elements apply the various weights to the letter A.



Lesson 4.5: Making Text Italic

Covers: The font-style property

➡ Online: wdpg.io/4-5-0

As you learned in Chapter 2, you can display text in italics semantically by using the `` tag when you want to emphasize text or the `<i>` tag when you want to format alternative text. If you have text that isn't semantic, but you want it to appear italic anyway, use the CSS property `font-style`, and set it to the value `italic`. Here's an example:

USE IT

Nonsemantic uses for italic text include pull quotes, the lead words or the lead sentence in a paragraph, and article metadata (such as the author name and date).

► Example ➡ Online: wdpg.io/4-5-1

This example applies the italic font style to the `span` element. There are two instances: the `` that's nested within the `h1` element and the `span` that's nested at the beginning of the `div` element.

WEB PAGE	<h2>Italic Text: A History</h2> <p>The first use of italics came in 1500 when Aldus Manutius of the Aldine Press wanted a typeface that resembled the handwritten humanist script that was then in common use. He asked his typesetter Francesco Griffo to make the typeface, which Manutius first used in the frontispiece of a book of the letters of Catherine of Siena. He produced the first book set entirely in italics the next year.</p>
CSS	<pre>body { font-family: Georgia, serif; } span { font-style: italic; } div { font-size: 1.25em; }</pre> <p>The <code>span</code> element is formatted as italic.</p>
HTML	<pre><h1>Italic Text: <u>A History</u></h1> <div> <u>The first use of italics came in 1500</u> when Aldus Manutius of the Aldine Press wanted a typeface that resembled the handwritten humanist script that was then in common use. He asked his typesetter Francesco Griffo to make the typeface, which Manutius first used in the frontispiece of a book of the letters of Catherine of Siena. He produced the first book set entirely in italics the next year. </div></pre> <p>The first span instance</p> <p>The second span instance</p>



Formatting Your Web Page

REMEMBER

When you're working with an element that defaults to italic styling, such as `cite` or `var` (see Chapter 16), you can prevent that element from appearing with italic text by assigning the keyword `normal` to its `font-style` property.

USE IT

For most web page text blocks, left-aligned text is easiest to read. Centered text is useful for page titles and subtitles. Use justified alignment when you want your text to have a more elegant look.

BEWARE

The web browser justifies text by adding spaces between words in a line. If your text block is narrow or includes one or more long words, you can end up with large, unsightly gaps in the text.

Styling Paragraphs

When (or perhaps I should say *if*) people think of typography, they tend to look at individual letters or letter combinations. That's important, for sure, but it's only the "trees" view of typography. If you want your web pages to look their best, you also need to take in the "forest" view, which encompasses the larger text blocks on the page, including titles, subtitles, headings, and especially paragraphs. As you see in the next couple of sections, paying attention to important styling touches such as alignment and indents can go a long way toward changing your pages from drab to fab.

Lesson 4.6: Aligning Paragraphs Horizontally

Covers: The `text-align` property

⇒ Online: wdpg.io/4-6-0

To control how a paragraph or block of text is aligned horizontally—that is, with respect to the left and right page margins—use the CSS `text-align` property, which takes any of the keywords shown in Table 4.2.

► Table 4.2 Possible Values for the `text-align` Property

Keyword	Description
<code>left</code>	Aligns the left edge of the text block with the left margin; the right edge of the text block is not aligned (and so is said to be <i>ragged</i>); this is the default in languages that read left to right.
<code>right</code>	Aligns the right edge of the text block with the right margin; the left edge of the text block is not aligned (ragged); this is the default in languages that read right to left.
<code>center</code>	Centers each line of the text block between the left and right margin; both the left and right edges of the text block are ragged.
<code>justify</code>	Aligns the left edge of the text block with the left margin and the right edge of the text block with the right margin.

“The four modes of alignment (centered, justified, flush left, and flush right) form the basic grammar of typographic composition. —Ellen Lupton

The following example tries each of the four `text-align` values.



► **Example** ➞ Online: wdpg.io/4-6-1

This example shows the four alignment styles at work: centered for the title and subtitle, and left, right, and fully justified text blocks.

WEB PAGE	<div> <div>Centered</div> <div>Aligning Web Page Text</div> <div>Notes From the Field</div> <div>Left-aligned</div> <div> <p>We read text (in English, anyway) from left to right. This means that when we get to the end of each line, to continue we must jump down one line and then scan to the beginning of that line. That leap-and-scan is most easily made when we "know" where the next line begins. That's why left-justified text is the easiest alignment to read.</p> <p>Compare the left-justified text block above with this right-justified paragraph. In this case, when you reach the end of each line, jumping down to the next isn't a problem, but because the left side of the text block is set ragged, the beginning of each line isn't in a predictable place, which makes right-justified text a tad more difficult to read.</p> <p>Many books are set with justified paragraphs because it looks more elegant without the right-ragged edges. However, pro book designers use sophisticated layout software to manage things like hyphenation (particularly if the text includes a long word such as <i>honorificabilitudinitatibus</i>). These aren't available for the web, so it's often best not to justify.</p> </div> <div>Right-aligned</div> <div>Justified</div> </div>
CSS	<pre>h1, h2 { text-align: center; }</pre> <div>Centers the h1 and h2 elements</div>
HTML	<pre><h1>Aligning Web Page Text</h1> <h2>Notes From the Field</h2> <div style="text-align: left;"> We read text (in English, anyway) from left to right. This means that when we get to the end of each line, to continue we must jump down one line and then scan to the beginning of that line. That leap-and-scan is most easily made when we "know" where the next line begins. That's why left-justified text is the easiest alignment to read. </div> <div style="text-align: right;"> Compare the left-justified text block above with this right- justified paragraph. In this case, when you reach the end of each line, jumping down to the next isn't a problem, but because the left side of the text block is set ragged, the beginning of each line isn't in a predictable place, which makes right-justified text a tad more difficult to read. </div> <div style="text-align: justify;"> Many books are set with justified paragraphs because it looks more elegant without the right-ragged edges. However, pro book designers use sophisticated layout software to manage things like hyphenation (particularly if the text includes a long word such as <i>honorificabilitudinitatibus</i>). These aren't available for the web, so it's often best not to justify. </div></pre> <div>Left-aligns the first paragraph</div> <div>Right-aligns the second paragraph</div> <div>Justifies the third paragraph</div>



Lesson 4.7: Indenting Paragraph Text

Covers: The text-indent property

LEARN

Some browsers support the `text-align-last` property, which sets the alignment of the last line in a text block when the `text-align` property is set to justify. Possible values include left, right, center, and justify. See <http://caniuse.com/#feat=css-text-align-last> to follow the support for this property.

REMEMBER

A commonly used value for a paragraph indent is 16px.

BEWARE

If you want to create an outdent for a text block, make sure that the block has a left margin that's wide enough to accommodate the outdented text. See Chapter 7 to learn how to set the left margin for a text block.

⇒ Online: wdpg.io/4-7-0

You can indent paragraph text by using the CSS `text-indent` property, which takes either of the values shown in Table 4.3. Note that the indent applies only to the beginning of the first line of the text block.

► Table 4.3 Values You Can Apply to the `text-indent` Property

Value	Description
length	A numeric value entered with a unit, such as px.
percentage	A percentage value. The computed indent is the width of the text block multiplied by the percentage.

As with most things typographical, much debate exists about whether text blocks should be indented. Some typographers eschew indents because they believe that nonindented text is more aesthetically pleasing; others embrace indents because they believe that indented text is more readable. Whichever side you end up on, you should keep the following points in mind:

- Never indent the first paragraph of the page or the first paragraph after a heading. The purpose of an indent is to separate the paragraph from the one above it, but that doesn't apply to the first paragraph.
- If you indent your paragraphs, you don't need to add space between paragraphs.
- If you don't indent your paragraphs, you should add some margin or padding between the paragraphs for readability. See Chapter 6 to find out how to set the margins and padding.



Using paragraph spacing and indents together squanders space and gives the text block a flabby, indefinite shape. —Ellen Lupton



► Example

⇒ Online: wdpg.io/4-7-1

This example displays the three possible indent styles: flush (the first paragraph); a positive indent (second paragraph); and a negative indent (third paragraph), which is usually called an outdent or a hanging indent.

WEB PAGE	<p>Flush → The first word of the first line is the critical word of that particular body of text. Let it start flush, at least. —William Addison Dwiggins</p> <p>Indented → Typographers generally take pleasure in the unpredictable length of the paragraph while accepting the simple and reassuring consistency of the paragraph indent. —Robert Bringhurst</p> <p>Outdented → OUTDENTS work well when dramatic effect is desired. They sometimes have a second emphasis factor, such as a style or case change, that contrasts with the body text. —Kristin Cullen</p>
HTML	<p><code><div></code> → The text-indent property isn't set, so the first line is flush with the rest of the paragraph. The first word of the first line is the critical word of that particular body of text. Let it start flush, at least. —William Addison Dwiggins <code></div></code></p> <p><code><div style="text-indent: 16px;"></code> → Indents the first line of the paragraph Typographers generally take pleasure in the unpredictable length of the paragraph while accepting the simple and reassuring consistency of the paragraph indent. —Robert Bringhurst <code></div></code></p> <p><code><div style="text-indent: -64px;"></code> → Outdents the first line of the paragraph Outdents work well when dramatic effect is desired. They sometimes have a second emphasis factor, such as a style or case change, that contrasts with the body text. —Kristin Cullen <code></div></code></p>



Formatting Your Web Page

REMEMBER










In each grayscale keyword, you can replace the word gray with the word grey, and the result will be the same color for all browsers (except Internet Explorer 7 and earlier). The keywords darkgray and darkgrey produce the same shade, for example.

Working with Colors

By default, most web browsers display the page by using black text on a white background. That combination is certainly readable but not interesting. Our marvelous eyes are capable of distinguishing millions of colors, so a palette of only black and white seems wrong somehow. Fortunately, CSS enables you to put your designer eyes to good use by offering several methods for accessing any of the 16 or so million colors that are available in the digital realm. Alas, most of those methods are a bit complicated, so I'm going to put them off until later (see Chapter 13).

For now, you get access to colors using the keywords that CSS defines. Table 4.4 lists the keywords for a few common colors.

► **Table 4.4** The CSS Keywords for Nine Common Colors

Keyword	Color
red	
lime	
blue	
yellow	
magenta	
cyan	
black	
gray	
white	

There are more than 140 defined keywords in all, so you shouldn't have any trouble finding the right shade (or shades) for your next web project. I've put the complete list of color keywords on the Web Design Playground at wdpg.io/colorkeywords. Figure 4.1 shows a partial list.



Color	Keyword	RGB Value		lightpink	#ffb6c1		pink	#ffc0cb
	crimson	#dc143c		lavenderblush	#fff0f5		palevioletred	#db7093
	hotpink	#ff69b4		deeppink	#ff1493		mediumvioletred	#c71585
	orchid	#da70d6		thistle	#d8bfd8		plum	#dda0dd
	violet	#ee82ee		magenta	#ff00ff		fuchsia	#ff00ff
	darkmagenta	#8b008b		purple	#800080		rebeccapurple	#663399
	mediumorchid	#ba55d3		darkviolet	#9400d3		darkorchid	#9932cc
	indigo	#4b0082		blueviolet	#8a2be2		mediumpurple	#9370db
	mediumslateblue	#7b68ee		slateblue	#6a5acd		darkslateblue	#483d8b
	lavender	#e6e6fa		ghostwhite	#f8f8ff		blue	#0000ff
	mediumblue	#0000cd		midnightblue	#191970		darkblue	#00008b
	navy	#000080		royalblue	#4169e1		cornflowerblue	#6495ed
	lightsteelblue	#b0c4de		lightslategray	#778899		slategray	#708090
	dodgerblue	#1e90ff		aliceblue	#f0f8ff		steelblue	#4682b4
	lightskyblue	#87cefa		skyblue	#87ceeb		deepskyblue	#00bfff
	lightblue	#add8e6		powderblue	#b0e0e6		cadetblue	#5f9ea0
	azure	#f0ffff		lightcyan	#e0ffff		paleturquoise	#afeeee

► **Figure 4.1** To see a complete list of the CSS color keywords on the Web Design Playground, surf to wdpg.io/colorkeywords.

Lesson 4.8: Applying Color to Text

Covers: The color property

➡ Online: wdpg.io/4-8-0

Several CSS properties have a color component, including borders, backgrounds, and shadows. You learn about all those properties and more in this book (including backgrounds in the next section), but so far you know about text, so I'll start there. Here's the general CSS syntax for applying color to a text element:

```
selector {
  color: keyword;
}
```

The text item to which you want the color applied

The color property and its value

The selector can be an HTML element, such as an `h1` heading or `p` element, or it can be any of the CSS selectors that you see in Chapter 7. The real work is done by the `color` property and its associated value, which can be any of the CSS color keywords (or any of the other color values supported by CSS, which you learn about in Chapter 17).

The following example shows the `color` definition for purple `h1` text.



Formatting Your Web Page

► Example

⇒ Online: wdpg.io/4-8-1

This example uses a keyword to assign the color purple to the `h1` element.

WEB PAGE	<h1>Royalty: A History</h1> ← <code><h1></code>
CSS	<pre>h1 { color: purple; }</pre> <p>Specify the <code>h1</code> element to style.</p> <p>Use the keyword <code>purple</code> to set the color property value.</p>
HTML	<pre><h1>Royalty: A History</h1></pre>

Lesson 4.9: Applying Color to a Background

Covers: The `background-color` property

PLAY

Style the `a` element to display the link text as yellow. Then add a second rule that displays the link text red and underlined when you hover the mouse over the link.

⇒ Online: wdpg.io/4-8-4

⇒ Online: wdpg.io/4-9-0

So far, I've looked only at setting the color of the foreground—the web page text—but you can use CSS to apply a color to a background. This color could be the background of the entire page (that is, the `body` element), a heading, a paragraph, a link, or part of a page such as a `<div>` or `` tag.

Here's the general CSS syntax for applying a background color to a web page item:

```
selector {  
  background-color: keyword;  
}
```

The item to which you want the background color applied

The `background-color` property and its value

The selector can be an HTML element or any of the CSS selectors that you learn about in Chapter 7. The key is the `background-color` property and its associated value, which can be any of the color keywords you learned about earlier.

The following example shows a web page with a Table of Contents sidebar that has a black background and white text. The example also gives you a partial look at the HTML and CSS used to set it up.



► Example

⇒ Online: wdpg.io/4-9-1

This example shows a web page with a Table of Contents sidebar that has a black background and white text.

WEB PAGE

Using Colors Effectively

“There are only 3 colors, 10 digits, and 7 notes; its what we do with them that's important.” —Jim Rohn

“Some colors reconcile themselves to one another, others just clash.”
—Edvard Munch

“All colors are the friends of their neighbors and the lovers of their opposites.” —Marc Chagall

Color Psychology

When selecting colors, think about the psychological impact that your scheme will have on your users. Studies have shown that “cool” colors such as blue and gray evoke a sense of dependability and trust. Use these colors for a more business-like appearance. For pages that require a little more excitement, “warm” colors such as red, yellow, and orange can evoke a festive, fun atmosphere. For a safe, comfortable ambiance, try using brown and yellow. For an environmental touch, use green and brown.

Color Schemes

<div>

Table of Contents

Color Psychology
Color Schemes
Color Caveats
A Few Examples
Best Practices
CSS and Color

CSS

```
div {
  background-color: black;
  color: white;
  float: right;
  font-size: 16px;
  font-weight: bold;
  margin-left: 0.5em;
  padding: 0 10px 5px 10px;
  text-align: left;
}
```

The background-color property sets the <div> background to black.

These properties apply various styles to the <div>.

The color property sets <div> text to white.

HTML

```
<div>
  <h3>Table of Contents</h3>
  Color Psychology<br>
  Color Schemes<br>
  Color Caveats<br>
  A Few Examples<br>
  Best Practices<br>
  CSS and Color
</div>
```

The <div> tag and its associated HTML and text



Formatting Your Web Page

PLAY

How would you modify the CSS in this example to display the Table of Contents sidebar with light gray text on a purple background?

⇒ Online: wdpg.io/4-9-2

PLAY

Write a CSS rule that styles links with blue text and a yellow background.

⇒ Online: wdpg.io/4-9-4

Summary

- You can use the `font-family` property to assign a typeface to a page element. This typeface can be one of the five generic fonts—`serif`, `sans-serif`, `monospace`, `cursive`, or `fantasy`—or a system font that's already installed on the user's computer.
- Use the `font-size` property to control the size of your text elements.
- Use the `font-weight` property to apply bolding nonsemantically.
- Use the `font-style` property to apply italics nonsemantically.
- Use `text-align` to set the horizontal alignment, such as centering headings and left-aligning text.
- Use `text-indent` to indent or outdent the first line of a text block.
- To color an element's text, use the `color` property.
- To color an element's background, use the `background-color` property.