



# Adding Structure to Your Page

“Organizing is what you do before you do something, so that when you do it, it is not all mixed up. —A. A. Milne



## This chapter covers

- Dividing page text into paragraphs and sections
- Adding numbered lists
- Building bulleted lists

You learned in Chapter 2 that you can create an effective web page by typing some text and then using headings and elements such as `strong` and `em` to make the text more readable and easier to understand. Headings in particular are crucial page devices, not only because they help the reader see where one part of the page ends and another begins, but also because they give the reader a general sense of the page hierarchy. All this falls under the general rubric of page structure, and that's the focus of this chapter.



## Adding Structure to Your Page

Thinking about the structure of your web page is important, because a wall of unstructured text is difficult to scan and read, as well as difficult to style. When you add structure (such as the headings from Chapter 2 and the paragraphs, sections, containers, and lists that you learn about in this chapter), each of those substructures is seen by the browser as a separate entity to which you can apply many style properties. As a rule, the more structured your page, the greater the control you have over how it looks. Fortunately, as you see in this chapter, HTML comes with several useful and straightforward tools for adding structure to a page.

### HTML Elements for Structuring Page Text

If you work with a word processor, you know that almost all documents have a structure: a title, possibly a subtitle, one or more topic headings, and one or more paragraphs within each topic. This makes the document easy to browse and comfortable to read because the structure guides readers and enables them to focus on the text. You can get those same advantages in your web pages by taking advantage of the various structural elements offered by HTML. I'll begin with one of the most common structures: the paragraph.

#### Lesson 3.1: Working with Paragraphs

Covers: The `p` element

#### PLAY

Using the Web Design Playground, modify the first five lines in this example so that the text snippets Line 1, Line 2, Line 3, and Line 4 each appear in a separate paragraph.

⇒ Online: [wdpg.io/3-1-2](http://wdpg.io/3-1-2)

⇒ Online: [wdpg.io/3-1-0](http://wdpg.io/3-1-0)

I mentioned in Chapter 2 that web browsers ignore whitespace, including carriage returns created by pressing Enter or Return, which is normally how you'd separate text into paragraphs in a text editor or word processor. The most common way to create a paragraph in HTML is to place a `<p>` (for paragraph) tag at the beginning of the text and a closing `</p>` tag at the end of the text.

The following example shows you both the wrong and right ways to create paragraphs.

**► Example**⇒ Online: [wdpg.io/3-1-1](http://wdpg.io/3-1-1)

In this example, you can see that the web browser ignores the whitespace created by the carriage returns but happily renders text into paragraphs when you use the `p` element.

WEB PAGE	<p>Line 1 Line 2 Line 3 Line 4</p> <p>Paragraph 1</p> <p>Paragraph 2</p> <p>Paragraph 3</p> <p>Paragraph 4</p>	<p>The web browser ignores the carriage returns.</p> <p>The browser uses the <code>&lt;p&gt;</code> tags to render the text into separate paragraphs.</p>
HTML	<pre>Line 1 Line 2 Line 3 Line 4  &lt;p&gt;Paragraph 1&lt;/p&gt; &lt;p&gt;Paragraph 2&lt;/p&gt; &lt;p&gt;Paragraph 3&lt;/p&gt; &lt;p&gt;Paragraph 4&lt;/p&gt;</pre>	<p>Wrong: Separate lines created by pressing only Enter/Return</p> <p>Right: Paragraphs created by using the <code>p</code> element</p>



# Adding Structure to Your Page

## Lesson 3.2: Inserting Line Breaks

Covers: The `br` element

### USE IT

Use a line break for poems, lyrics, addresses, contact information, or programming statements, or to show a sample of HTML or CSS code.

### PLAY

Render the poem “Break, Break, Break,” by Alfred Lord Tennyson, correctly by adding line breaks to each line that isn’t the end of a stanza.

⇒ Online: [wdpg.io/3-2-2](http://wdpg.io/3-2-2)

⇒ Online: [wdpg.io/3-2-0](http://wdpg.io/3-2-0)

When you separate page text into paragraphs, the web browser renders this text by (among other things) creating a bit of space between paragraphs. This space is normally what you want because that vertical gap gives the reader a visual clue as to where one paragraph ends and the next one begins, as well as a chance to take a quick breather between sections of text. This space isn’t *always* what you want, however. If your page text is a poem, for example, you almost certainly don’t want paragraphs between lines. The same is true if your text is programming code or song lyrics.

When you want to start a new line but don’t want to have any space between the two lines, you need the `br` (short for *line break*) element. As you can see in the following example, the web browser renders the `br` element by inserting a carriage return and beginning the next line immediately below the previous one.

### ▶ Example

⇒ Online: [wdpg.io/3-2-1](http://wdpg.io/3-2-1)

In this example, you can see that the web browser renders the `br` element by inserting a carriage return and beginning the next line immediately below the previous one.

WEB PAGE	<h3>Contact Info</h3> <p>Manning Publications Co. PO Box 761 Shelter Island, NY 11964 support@manning.com 203-626-1510</p>
HTML	<pre>&lt;h3&gt;Contact Info&lt;/h3&gt; Manning Publications Co.&lt;br&gt; PO Box 761&lt;br&gt; Shelter Island, NY 11964&lt;br&gt; support@manning.com&lt;br&gt; 203-626-1510</pre> <p>The <code>&lt;br&gt;</code> tag tells the browser to start the following text on a new line.</p>



### Lesson 3.3: Dividing Web Page Text

Covers: The `div` element

⇒ Online: [wdpg.io/3-3-0](http://wdpg.io/3-3-0)

In Chapter 11, I show you the HTML5 sectioning elements, including `<section>` and `<article>`. These elements enable you to structure your page semantically by designating containers as sections and articles within those sections, as well as headers, footers, navigation, and more. Not all text falls neatly into any of the HTML5 semantic categories, however. For text that requires a container but for which none of the semantic elements (including the `p` element) is appropriate, HTML offers the `div` (short for *division*) element. The `<div>` tag and its corresponding `</div>` end tag create a simple container for text. The web browser applies no inherent formatting to the text, including not rendering any space between consecutive `div` elements, as you see in the following example.

#### PLAY

The `br` element is often a poor choice for structuring page text because it doesn't provide a container for the text, so you can't style the text. On the Web Design Playground, replace the `br` elements with `div` elements.

⇒ Online: [wdpg.io/3-3-2](http://wdpg.io/3-3-2)

#### ► Example

⇒ Online: [wdpg.io/3-3-1](http://wdpg.io/3-3-1)

This example uses the `div` element to divide a web page into two text blocks.

WEB PAGE	<h2>Shortening</h2> <p>The <b>shortening</b> process is based mostly on a kind of linguistic laziness called <b>clipping</b> that causes us to lop off great chunks of words. For example, we end up with <i>fridge</i> from <i>refrigerator</i> and <i>flu</i> from <i>influenza</i>. Often we clip everything after the first syllable: <i>dis</i> (from <i>disrespect</i>) and <i>gym</i> (from <i>gymnasium</i>).</p> <p>A related process is the creation of <b>acronyms</b>, which form a pronounceable word using the first letters of each word in a phrase. For example, <i>UNICEF</i> from United Nations International Children's Emergency Fund, and <i>NATO</i> from North Atlantic Treaty Organization.</p>
HTML	<pre> &lt;h2&gt;Shortening&lt;/h2&gt; &lt;div&gt; The &lt;b&gt;shortening&lt;/b&gt; process is based mostly on a kind of linguistic laziness called &lt;b&gt;clipping&lt;/b&gt; that causes us to lop off great chunks of words. For example, we end up with &lt;i&gt;fridge&lt;/i&gt; from &lt;i&gt;refrigerator&lt;/i&gt; and &lt;em&gt;flu&lt;/em&gt; from &lt;i&gt;influenza&lt;/i&gt;. Often we clip everything after the first syllable: &lt;i&gt;dis&lt;/i&gt; (from &lt;i&gt;disrespect&lt;/i&gt;) and &lt;i&gt;gym&lt;/i&gt; (from &lt;i&gt;gymnasium&lt;/i&gt;). &lt;/div&gt; &lt;div&gt; A related process is the creation of &lt;b&gt;acronyms&lt;/b&gt;, which form a pronounceable word using the first letters of each word in a phrase. For example, &lt;i&gt;UNICEF&lt;/i&gt; from United Nations International Children's Emergency Fund, and &lt;i&gt;NATO&lt;/i&gt; from North Atlantic Treaty Organization. &lt;/div&gt; </pre> <p>The <code>&lt;div&gt;</code> and <code>&lt;/div&gt;</code> tags divide your web page text into blocks.</p>



## Adding Structure to Your Page

### REMEMBER

Elements such as `div` and `p` are known as *block-level elements* because they create a boxlike container that begins on a new line and within which the content (such as text) flows. Elements such as `span` are known as *inline elements* because each one creates a container that exists within some larger element and flows with the rest of the content in that larger element.

### Lesson 3.4: Creating Inline Containers

Covers: The `span` element

⇒ Online: [wdpg.io/3-4-0](http://wdpg.io/3-4-0)

Elements such as `div` and `p` are important because they provide containers in which you add and style text. Sometimes, however, you want to style just a subset of the text within such a container. You may want to apply a font effect or color to a few words or to a sentence, for example. In that case, you can create an *inline container* by surrounding the text with the `<span>` tag and its `</span>` end tag. The following example creates several inline containers, and a CSS property is defined for the `span` element to apply a yellow background to each container.

#### ► Example

⇒ Online: [wdpg.io/3-4-1](http://wdpg.io/3-4-1)

This example creates several inline containers, and a CSS property is defined for the `span` element to apply a yellow background to each container.

WEB PAGE

Throughout this document, screen items that you click and text that you type appear with a yellow background. Here are some examples:

- Click the **File** menu and then click **Save**.
- Set the number of copies and then click **Print**.
- Click **Search**, type **blockquote**, and then press **Enter**.

CSS

```
span {  
    background-color: yellow;  
}
```

This style definition tells the browser to apply yellow as the background color to all the `span` elements.



HTML

```
<p>
Throughout this document, screen items that you click and text
that you type appear with a <span>yellow background</span>. Here
are some examples:
</p>

<ul>
  <li>Click the <span>File</span> menu and then click
<span>Save</span>.</li>
  <li>Set the number of copies and then click <span>Print</span>.</
li>
  <li>Click <span>Search</span>, type <span>blockquote</span>,
and then press <span>Enter</span>.</li>
</ul>
```

The `<span>` and `</span>` tags  
create inline containers.

### Lesson 3.5: Adding a Visual Break between Blocks

Covers: The `hr` element

⇒ Online: [wdpg.io/3-5-0](http://wdpg.io/3-5-0)

As I mention earlier, the `p` element automatically adds whitespace between paragraphs, and for other block-level elements such as `div`, you can use CSS to create your own vertical spacing between blocks. Sometimes, however, you want a more direct or more emphatic visual indicator of a break between blocks. In such a case, you can insert the `hr` (short for *horizontal rule*) element. As you can see in the following example, the web browser displays a horizontal line across the page. If you don't want the line to extend the width of its container, you can use the `width` CSS property and set it to the width (measured in, say, pixels or a percentage) you prefer.

### PLAY

To get some practice with the `span` element, try the exercises on the Web Design Playground.

⇒ Online: [wdpg.io/3-4-2](http://wdpg.io/3-4-2)



## Adding Structure to Your Page

### ► Example

➡ Online: [wdpg.io/3-5-1](http://wdpg.io/3-5-1)

*This example shows that when you add the `hr` element, the web browser displays a horizontal line across the page.*

WEB PAGE	<p><b>Word Origins: Introduction</b></p> <p>In a cynical world where attention spans are 140-characters long and where much of the populace is obsessed with the low-brow goings-on of Kim or Miley or Kylie, one amazing fact rises above the muck: it's rare to meet someone who isn't in some way interested in words and language. From slang-slinging youngsters to crossword-solving oldsters, from inveterate punsters to intrepid neologists, some aspect of language appeals to everyone.</p> <hr/> <p>Is there one slice of the language pie that everyone likes? Probably not. People are just too complex to like any one thing universally. However, in my own admittedly limited experience (I haven't met every person in the world), I have yet to come across a person who doesn't appreciate a good story about the origins of a word or phrase.</p>
HTML	<pre>&lt;h2&gt;Word Origins: Introduction&lt;/h2&gt; &lt;div&gt;In a cynical world where attention spans are 140-characters long and where much of the populace is obsessed with the low- brow goings-on of Kim or Miley or Kylie, one amazing fact rises above the muck: it's rare to meet someone who isn't in some way interested in words and language. From slang-slinging youngsters to crossword-solving oldsters, from inveterate punsters to intrepid neologists, some aspect of language appeals to everyone.&lt;/div&gt; &lt;hr&gt; &lt;div&gt;Is there one slice of the language pie that everyone likes? Probably not. People are just too complex to like any one thing universally. However, in my own admittedly limited experience (I haven't met every person in the world), I have yet to come across a person who doesn't appreciate a good story about the origins of a word or phrase.&lt;/div&gt;</pre> <p>The <code>&lt;hr&gt;</code> tag inserts a horizontal line between two text blocks.</p>

I should note here that many web-design gurus recommend that instead of using the `hr` element to get a horizontal line between two blocks, you should add a bottom border to the top block or a top border to the bottom block. See Chapter 7 for more info on styling borders.

### Organizing Text Into Lists

It's tough to surf the web these days and not come across a list or three in your travels—a top-ten list, a best-of list, a point-form summary of an event, or any of a thousand other variations on the list theme. A list is often the perfect way to display certain types of information, such as a series of steps or an unordered collection of items.





HTML offers these two list types:

- A *numbered list* (sometimes called an *ordered list*) presents its items in numeric order, with each item's number on the left and the item text indented to the right.
- A *bulleted list* (sometimes called an *unordered list*) presents its items in the order you specify, with each item having a bullet (usually, a small dot) on the left and the item text indented to the right.

### Lesson 3.6: Adding a Numbered List

Covers: The `ol` element

⇒ Online: [wdpg.io/3-6-0](http://wdpg.io/3-6-0)

If the things you want to display have an inherent numeric order, such as you might find in the steps of a procedure or the elements in a series, a numbered list is the way to go. The good news is that you don't have to enter the numbers yourself, because the browser takes care of them for you automatically. The first item in the list is given the number 1, the second is given 2, and so on. If you insert or delete items, the browser adjusts all the list numbers as needed to keep everything in numeric order.

You start to construct a numbered list by creating a container that consists of the `<ol>` tag (short for *ordered list*) and its closing `</ol>` tag. Between those tags, you add one or more `<li>` (short for *list item*) tags followed by the item text and the closing `</li>` tag:

```
<li>Item text</li>
```

The browser displays the item with a number on the left (the value of which is determined by the item's position in the list), followed by item text, which is indented from the number, and the entire item is indented from the left margin of whatever element contains it.

The following example shows a basic numbered list and the HTML tags and text used to create it.

### REMEMBER

*Although this type of list is used far less than numbered and bulleted lists, you should also be aware of description list, which is a list of terms and descriptions. The entire list uses the `<dl>` and `</dl>` tags as a container; you specify each term within the `<dt>` and `</dt>` tags and each description within the `<dd>` and `</dd>` tags.*

### USE IT

*Use a numbered list for any collection that must appear in sequential, numeric order. Examples are the steps the reader must follow in a how-to procedure, the tasks involved in a recipe, the sections in a document (particularly a contract or other legal document), or the items in a ranking such as a top-ten list.*



## Adding Structure to Your Page

### ► Example

➡ Online: [wdpg.io/3-6-1](http://wdpg.io/3-6-1)

This example shows how to use a numbered list to set up a top-ten list.

WEB PAGE

### Top 10 Modern Words of Unknown Origin

1. *jazz* (1909)
2. *jive* (1928)
3. *bozo* (1920)
4. *dork* (1964)
5. *pizzazz* (1937)
6. *humongous* (1970)
7. *gismo* (1943)
8. *zit* (1966)
9. *reggae* (1968)
10. *mosh* (1987)

HTML

```
<h3>Top 10 Modern Words of Unknown Origin</h3>
<ol>
  <li><em>jazz</em> (1909)</li>
  <li><em>jive</em> (1928)</li>
  <li><em>bozo</em> (1920)</li>
  <li><em>dork</em> (1964)</li>
  <li><em>pizzazz</em> (1937)</li>
  <li><em>humongous</em> (1970)</li>
  <li><em>gismo</em> (1943)</li>
  <li><em>zit</em> (1966)</li>
  <li><em>reggae</em> (1968)</li>
  <li><em>mosh</em> (1987)</li>
</ol>
```

Use the `<ol>` and `</ol>` tags as the container for the numbered list.

Within the container, enter each item's text between the `<li>` and `</li>` tags.

### PLAY

To get some practice with the `ol` and `li` elements, try the exercises on the Web Design Playground.

➡ Online: [wdpg.io/3-6-2](http://wdpg.io/3-6-2)

By default, the numbers used in the list are standard decimal values (1, 2, 3, and so on). You can change the number type by specifying the `list-style-type` CSS property. Table 3.1 lists the most common numbered-list values for this property.



► **Table 3.1** Common Numbered-List Values for the list-style-type CSS Property

Value	Description	Example Numbers
decimal	Decimal numbers	1, 2, 3, 4, ...
decimal-leading-zero	Decimals numbers with a leading 0	01, 02, 03, 04, ...
lower-alpha	Lowercase letters	a, b, c, d, ...
upper-alpha	Uppercase letters	A, B, C, D, ...
lower-roman	Lowercase Roman numerals	i, ii, iii, iv, ...
upper-roman	Uppercase Roman numbers	I, II, III, IV, ...
lower-greek	Lowercase Greek letters	α, β, γ, δ, ...
upper-greek	Uppercase Greek letters	Α, Β, Γ, Δ, ...

## LEARN

Quite a few values for list-style-type are associated with various other languages, such as Chinese, Hebrew, and Japanese. See the following page for the complete list: <https://developer.mozilla.org/en-US/docs/Web/CSS/list-style-type>.

## Lesson 3.7: Adding a Bulleted List

Covers: The ul element

⇒ Online: [wdpg.io/3-7-0](http://wdpg.io/3-7-0)

If the items you want to display have no inherent numeric order, such as you might find in a to-do list or a set of characteristics, a bulleted list is the way to go. Each item appears in its own paragraph, preceded by a bullet (usually, a black dot). You don't have to enter the bullets manually because the browser adds them automatically.

You start building a bulleted list by creating a container that consists of the `<ul>` (short for *unordered list*) tag and its closing `</ul>` tag. Between these tags, as with a numbered list, you add one or more `<li>` tags, followed by the item text and the closing `</li>` tag:

```
<li>Item text</li>
```

The browser displays the item with a bullet on the left, followed by item text, which is indented from the bullet, and the entire item is indented from the left margin of the element that contains it.

The following example shows a basic bulleted list and its underlying HTML tags and text.

## USE IT

Use a bulleted list for any collection of items that are related in some way but don't have to appear in numeric order. Examples include a to-do list or grocery list, a set of traits or properties associated with an object, or a collection of prerequisites for a course.



## Adding Structure to Your Page

### ► Example

➡ Online: [wdpg.io/3-7-1](http://wdpg.io/3-7-1)

This example shows you how to build a bulleted list.

WEB PAGE

This “ordinary linguistic competence” manifests as various mechanisms that people use to forge new words:

- Combining
- Shortening
- Shifting
- Borrowing
- Onomatopoeia
- Mistakes
- Retronyms
- Ex Nihilo

HTML

```
<p>This <q>ordinary linguistic competence</q> manifests as various mechanisms that people use to forge new words:</p>
```

```
<ul>
```

```
<li>Combining</li>
<li>Shortening</li>
<li>Shifting</li>
<li>Borrowing</li>
<li>Onomatopoeia</li>
<li>Mistakes</li>
<li>Retronyms</li>
<li>Ex Nihilo</li>
```

```
</ul>
```

Within the container, enter each item's text between the `<li>` and `</li>` tags.

Use the `<ul>` and `</ul>` tags as the container for the bulleted list.

### PLAY

To get some practice with the `ul` and `li` elements, try the exercises on the Web Design Playground.

➡ Online: [wdpg.io/3-7-2](http://wdpg.io/3-7-2)

By default, the bullets used in the list are filled circles (•). You can change the number type by specifying the `list-style-type` CSS property. Table 3.2 lists the available bulleted-list values for this property.

► Table 3.2 The Bulleted-List Values for the `list-style-type` CSS Property

Value	Description	Example Bullet
disc	Filled circle	●
circle	Unfilled circle	○
square	Filled square	■



## Summary

- To structure your page text into paragraphs, use the `p` element.
- To separate one line from the next, insert the `br` element to add a line break.
- Use the `div` element to divide the page into text blocks.
- You can create an inline container for text by surrounding the text with the `<span>` and `</span>` tags.
- Use the `hr` element to separate text blocks with a horizontal rule.
- Use the `<ol>` tag to create a numbered list and the `<ul>` tag to create a bulleted list. In both cases, you use the `<li>` tag to designate each item in the list.