

CSS

The Cottage Garden

The *cottage garden* is a distinct style of garden that uses an informal design, dense plantings, and a mixture of ornamental and edible plants.

The Cottage Garden originated in [England](#) and its history can be traced back for centuries, although they were re-invented in 1870's England, when stylized versions were formed as a reaction to the more structured and rigorously maintained [English estate gardens](#).

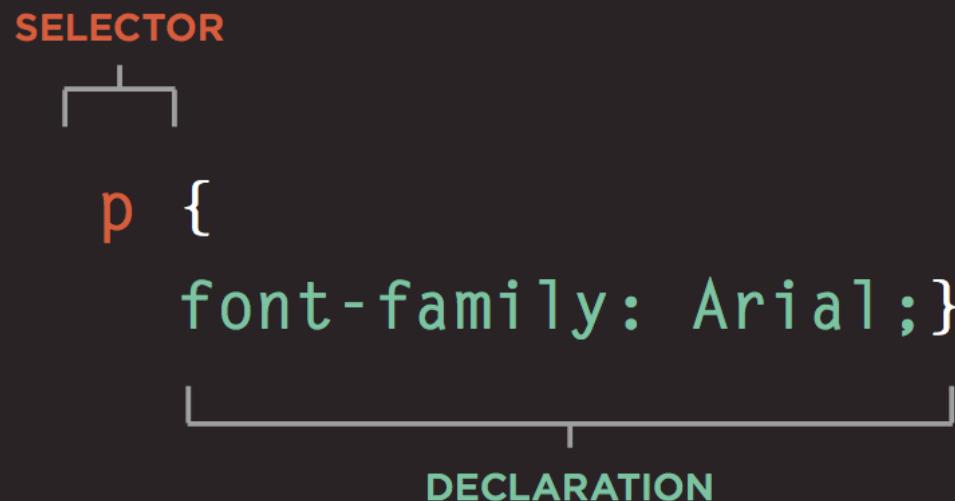
The earliest cottage gardens were more practical than their modern descendants, with an emphasis on vegetables and herbs, along with some fruit trees.

In this example, block level elements are shown with red borders, and inline elements have green borders.

The `<body>` element creates the first box, then the `<h1>`, `<h2>`, `<p>`, `<i>`, and `<a>` elements each create their own boxes within it.

Using CSS, you could add a border around any of the boxes, specify its width and height, or add a background color. You could also control text inside a box — for example, its color, size, and the typeface used.

CSS works by associating rules with HTML elements. These rules govern how the content of specified elements should be displayed. A CSS rule contains two parts: a **selector** and a **declaration**.



This rule indicates that all <p> elements should be shown in the Arial typeface.

Selectors indicate which element the rule applies to. The same rule can apply to more than one element if you separate the element names with commas.

Declarations indicate how the elements referred to in the selector should be styled. Declarations are split into two parts (a property and a value), and are separated by a colon.

CSS declarations sit inside curly brackets and each is made up of two parts: a **property** and a **value**, separated by a colon. You can specify several properties in one declaration, each separated by a semi-colon.

```
h1, h2, h3 {  
    font-family: Arial;  
    color: yellow;}
```



The diagram shows a horizontal bracket spanning the entire declaration. Below the bracket, the word "PROPERTY" is centered under the first part of the declaration, and the word "VALUE" is centered under the second part.

This rule indicates that all `<h1>`, `<h2>` and `<h3>` elements should be shown in the Arial typeface, in a yellow color.

Properties indicate the aspects of the element you want to change. For example, color, font, width, height and border.

Values specify the settings you want to use for the chosen properties. For example, if you want to specify a color property then the value is the color you want the text in these elements to be.

This example uses two documents: the HTML file (`example.html`) and a separate CSS file (`example.css`). The fifth line of HTML uses the `<link>` element to indicate where the CSS file is located.

On the next page, you will see how CSS rules can also be placed in your HTML pages and we will discuss when you might want to do this.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Introducing CSS</title>
    <link href="css/example.css" type="text/css"
          rel="stylesheet" />
  </head>
  <body>
    <h1>From Garden to Plate</h1>
    <p>A <i>potager</i> is a French term for an
       ornamental vegetable or kitchen garden ... </p>
    <h2>What to Plant</h2>
    <p>Plants are chosen as much for their functionality
       as for their color and form ... </p>
  </body>
</html>

body {
  font-family: Arial, Verdana, sans-serif;}
h1, h2 {
  color: #ee3e80;}
p {
  color: #665544;}
```

chapter-10/using-external-css.html

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Using External CSS</title>
    <link href="css/styles.css" type="text/css"
      rel="stylesheet" />
  </head>
  <body>
    <h1>Potatoes</h1>
    <p>There are dozens of different potato
      varieties. They are usually described as
      early, second early and maincrop.</p>
  </body>
</html>
```

chapter-10/styles.css

CSS

```
body {
  font-family: arial;
  background-color: rgb(185,179,175);}
h1 {
  color: rgb(255,255,255);}
```

Potatoes

There are dozens of different potato varieties. They are usually described as early, second early and maincrop potatoes.

RESULT

HTML

CSS

chapter-10/using-internal-css.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Using Internal CSS</title>
    <style type="text/css">
      body {
        font-family: arial;
        background-color: rgb(185,179,175;}
      h1 {
        color: rgb(255,255,255);}
    </style>
  </head>
  <body>
    <h1>Potatoes</h1>
    <p>There are dozens of different potato
      varieties. They are usually described as
      early, second early and maincrop.</p>
  </body>
</html>
```

RESULT

Potatoes

There are dozens of different potato varieties. They are usually described as early, second early and maincrop potatoes.

<p style="color:red;">

SELECTOR

MEANING

EXAMPLE

UNIVERSAL SELECTOR

Applies to all elements in the document

*** { }**
Targets all elements on the page

TYPE SELECTOR

Matches element names

h1, h2, h3 { }
Targets the `<h1>`, `<h2>` and `<h3>` elements

CLASS SELECTOR

Matches an element whose `class` attribute has a value that matches the one specified after the period (or full stop) symbol

.note { }
Targets any element whose `class` attribute has a value of `note`
p.note { }
Targets only `<p>` elements whose `class` attribute has a value of `note`

ID SELECTOR

Matches an element whose `id` attribute has a value that matches the one specified after the pound or hash symbol

#introduction { }
Targets the element whose `id` attribute has a value of `introduction`

CHILD SELECTOR

Matches an element that is a direct child of another

li>a { }
Targets any `<a>` elements that are children of an `` element (but not other `<a>` elements in the page)

DESCENDANT SELECTOR

Matches an element that is a descendent of another specified element (not just a direct child of that element)

p a { }
Targets any `<a>` elements that sit inside a `<p>` element, even if there are other elements nested between them

ADJACENT SIBLING SELECTOR

Matches an element that is the next sibling of another

h1+p { }
Targets the first `<p>` element after any `<h1>` element (but not other `<p>` elements)

GENERAL SIBLING SELECTOR

Matches an element that is a sibling of another, although it does not have to be the directly preceding element

h1~p { }
If you had two `<p>` elements that are siblings of an `<h1>` element, this rule would apply to both

chapter-10/css-selectors.html

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS Selectors</title>
  </head>
  <body>
    <h1 id="top">Kitchen Garden Calendar</h1>
    <p id="introduction">Here you can read our handy guide about what to do when.</p>
    <h2>Spring</h2>
    <ul>
      <li><a href="mulch.html">Spring mulch vegetable beds</a></li>
      <li><a href="potato.html">Plant out early potatoes</a></li>
      <li><a href="tomato.html">Sow tomato seeds</a></li>
      <li><a href="beet.html">Sow beet seeds</a></li>
      <li><a href="zucchini.html">Sow zucchini seeds</a></li>
      <li><a href="rhubarb.html">Deadhead rhubarb flowers</a></li>
    </ul>
    <p class="note">
      This page was written by
      <a href="mailto:ivy@example.org">ivy@example.org</a> for
      <a href="http://www.example.org">Example</a>.
    </p>
    <p>
      <a href="#top">Top of page</a>
    </p>
  </body>
</html>
```

SELECTOR	MEANING	EXAMPLE
EXISTENCE	[]	<code>p[class]</code> Targets any <p> element with an attribute called class
EQUALITY	[=]	<code>p[class="dog"]</code> Targets any <p> element with an attribute called class whose value is dog
SPACE	[~=]	<code>p[class~="dog"]</code> Targets any <p> element with an attribute called class whose value is a list of space-separated words, one of which is dog
PREFIX	[^=]	<code>p[attr^"d"]</code> Targets any <p> element with an attribute whose value begins with the letter "d"
SUBSTRING	[*=]	<code>p[attr*do"]</code> Targets any <p> element with an attribute whose value contains the letters "do"
SUFFIX	[=\$=]	<code>p[attr\$g"]</code> Targets any <p> element with an attribute whose value ends with the letter "g"

If there are two or more rules that apply to the same element, it is important to understand which will take precedence.

LAST RULE

If the two selectors are identical, the latter of the two will take precedence. Here you can see the second *i* selector takes precedence over the first.

SPECIFICITY

If one selector is more specific than the others, the more specific rule will take precedence over more general ones. In this example:

h1 is more specific than *
p b is more specific than p
p#intro is more specific than p

IMPORTANT

You can add !important after any property value to indicate that it should be considered more important than other rules that apply to the same element.

Understanding how CSS rules cascade means you can write simpler style sheets because you can create generic rules that apply to most elements and then override the properties on individual elements that need to appear differently.

chapter-10/cascade.html

HTML

```
<h1>Potatoes</h1>
<p id="intro">There are <i>dozens</i> of different
<b>potato</b> varieties.</p>
<p>They are usually described as early, second early
and maincrop potatoes.</p>
```

CSS

```
* {
    font-family: Arial, Verdana, sans-serif;}
h1 {
    font-family: "Courier New", monospace;}
i {
    color: green;}
i {
    color: red;}
b {
    color: pink;}
p b {
    color: blue !important;}
p b {
    color: violet;}
p#intro {
    font-size: 100%;}
p {
    font-size: 75%;}
```

RESULT

Potatoes

There are *dozens* of different **potato** varieties.

They are usually described as early, second early and maincrop potatoes.

HTML

chapter-10/inheritance.html

```
<div class="page">
  <h1>Potatoes</h1>
  <p>There are dozens of different potato
  varieties.</p>
  <p>They are usually described as early, second
  early and maincrop potatoes.</p>
</div>
```

CSS

```
body {
  font-family: Arial, Verdana, sans-serif;
  color: #665544;
  padding: 10px;}
.page {
  border: 1px solid #665544;
  background-color: #efefef;
  padding: inherit;}
```

RESULT

Potatoes

There are dozens of different potato varieties.

They are usually described as early, second early and maincrop potatoes.

If you specify the `font-family` or `color` properties on the `<body>` element, they will apply to most child elements. This is because the value of the `font-family` property is **inherited** by child elements. It saves you from having to apply these properties to as many elements (and results in simpler style sheets).

You can compare this with the `background-color` or `border` properties; they are **not** **inherited** by child elements. If these were inherited by all child elements then the page could look quite messy.

You can force a lot of properties to inherit values from their parent elements by using `inherit` for the value of the properties. In this example, the `<div>` element with a `class` called `page` inherits the padding size from the CSS rule that applies to the `<body>` element.

COLOR

```
/* color name */
h1 {
    color: DarkCyan;
}
/* hex code */
h2 {
    color: #ee3e80;
}
/* rgb value */
p {
    color: rgb(100,100,90);
```

RESULT

Marine Biology

The Composition of Seawater

Almost anything can be found in seawater. This includes dissolved materials from Earth's crust as well as materials released from organisms. The most important components of seawater that influence life forms are salinity, temperature, dissolved gases (mostly oxygen and carbon dioxide), nutrients, and pH. These elements vary in their composition as well as in their influence on marine life.

CSS

chapter-11/background-color.html

```
body {  
    background-color: rgb(200,200,200);}  
h1 {  
    background-color: DarkCyan;}  
h2 {  
    background-color: #ee3e80;}  
p {  
    background-color: white;}
```

RESULT

Marine Biology

The Composition of Seawater

Almost anything can be found in seawater. This includes dissolved materials from Earth's crust as well as materials released from organisms. The most important components of seawater that influence life forms are salinity, temperature, dissolved gases (mostly oxygen and carbon dioxide), nutrients, and pH. These elements vary in their composition as well as in their influence on marine life.

RGB VALUES

Values for red, green, and blue are expressed as numbers between 0 and 255.



rgb(102,205,170)

This color is made up of the following values:
102 red
205 green
170 blue

HEX CODES

Hex values represent values for red, green, and blue in hexadecimal code.



#66cdaa

The value of the red, 102, is expressed as 66 in hexadecimal code. The 205 of the green is expressed as cd and the 170 of the blue equates to aa.

COLOR NAMES

Colors are represented by predefined names. However, they are very limited in number.



MediumAquaMarine

There are 147 color names supported by browsers (this color is MediumAquaMarine). Most consider this to be a limited color palette, and it is hard to remember the name for each of the colors so (apart from white and black) they are not commonly used.

HUE

Hue is near to the colloquial idea of color. Technically speaking however, a color can also have saturation and brightness as well as hue.



SATURATION

Saturation refers to the amount of gray in a color. At maximum saturation, there would be no gray in the color. At minimum saturation, the color would be mostly gray.



BRIGHTNESS

Brightness (or "value") refers to how much black is in a color. At maximum brightness, there would be no black in the color. At minimum brightness, the color would be very dark.



CSS

chapter-11/opacity.html

```
p.one {  
    background-color: rgb(0,0,0);  
    opacity: 0.5;}  
p.two {  
    background-color: rgb(0,0,0);  
    background-color: rgba(0,0,0,0.5);}
```

RESULT**RESULT IN OLDER BROWSER**

HUE

Hue is the colloquial idea of color. In HSL colors, hue is often represented as a color circle where the angle represents the color, although it may also be shown as a slider with values from 0 to 360.



SATURATION

Saturation is the amount of gray in a color. Saturation is represented as a percentage. 100% is full saturation and 0% is a shade of gray.



LIGHTNESS

Lightness is the amount of white (lightness) or black (darkness) in a color. Lightness is represented as a percentage. 0% lightness is black, 100% lightness is white, and 50% lightness is normal. Lightness is sometimes referred to as *luminosity*.



Please note that lightness is a different concept to brightness. Graphic design software (such as Photoshop and GIMP) have color pickers that use hue, saturation, and brightness — but brightness only adds black, whereas lightness offers both white and black.

```
body {  
    background-color: #C8C8C8;  
    background-color: hsl(0,0%,78%);}  
p {  
    background-color: #ffffff;  
    background-color: hsla(0,100%,100%,0.5);}
```

RESULT

Marine Biology

The Composition of Seawater

Almost anything can be found in seawater. This includes dissolved materials from Earth's crust as well as materials released from organisms. The most important components of seawater that influence life forms are salinity, temperature, dissolved gases (mostly oxygen and carbon dioxide), nutrients, and pH. These elements vary in their composition as well as in their influence on marine life.

TEXT

SERIF

Serif fonts have extra details on the ends of the main strokes of the letters. These details are known as serifs.

im

In print, serif fonts were traditionally used for long passages of text because they were considered easier to read.

SANS-SERIF

Sans-serif fonts have straight ends to letters, and therefore have a much cleaner design.

im

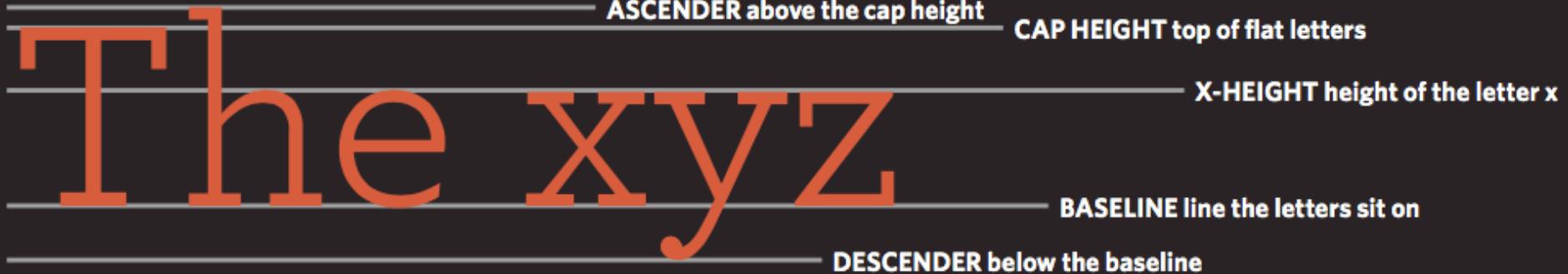
Screens have a lower resolution than print. So, if the text is small, sans-serif fonts can be clearer to read.

MONOSPACE

Every letter in a monospace (or fixed-width) font is the same width. (Non-monospace fonts have different widths.)

im

Monospace fonts are commonly used for code because they align nicely, making the text easier to follow.



WEIGHT

Light

Medium

Bold

Black

STYLE

Normal

Italic

Oblique

STRETCH

Condensed

Regular

Extended

The font weight not only adds emphasis but can also affect the amount of white space and contrast on a page.

Italic fonts have a cursive aspect to some of the lettering. Oblique font styles take the normal style and put it on an angle.

In condensed (or narrow) versions of the font, letters are thinner and closer together. In expanded versions they are thicker and further apart.

When choosing a typeface, it is important to understand that a browser will usually only display it if it's installed on that user's computer.

SERIF

Serif fonts have extra details on the end of the main strokes of the letters.

EXAMPLES:

Georgia

Times

Times New Roman

SANS-SERIF

Sans-serif fonts have straight ends to letters and therefore have a much cleaner design.

EXAMPLES:

Arial

Verdana

Helvetica

MONOSPACE

Every letter in a monospace typeface is the same width. (Non-monospace fonts have different widths.)

EXAMPLES:

Courier

Courier New

CURSIVE

Cursive fonts either have joining strokes or other cursive characteristics, such as handwriting styles.

EXAMPLES:

Comic Sans MS

Monotype Corsiva

FANTASY

Fantasy fonts are usually decorative fonts and are often used for titles. They're not designed for long bodies of text.

EXAMPLES:

Impact

Haettenschweller

Browsers are supposed to support at least one typeface from each of the groups above. For this reason, it is common to add the generic font name after your preferred choice of typefaces.

For example, if you wanted serif type, you could write the following:
`font-family: Georgia, Times, serif;`

```
<!DOCTYPE html>
<html>
  <head>
    <title>Font Family</title>
    <style type="text/css">
      body {
        font-family: Georgia, Times, serif;
      }
      h1, h2 {
        font-family: Arial, Verdana, sans-serif;
      }
      .credits {
        font-family: "Courier New", Courier,
        monospace;
      }
    </style>
  </head>
  <body>
    <h1>Briards</h1>
    <p class="credits">by Ivy Duckett</p>
    <p class="intro">The <a class="breed"
      href="http://en.wikipedia.org/wiki/
      Briard">briard</a>, or berger de brie, is
      a large breed of dog traditionally used as
      a herder and guardian of sheep...</p>
  </body>
</html>
```

```
body {  
    font-family: Arial, Verdana, sans-serif;  
    font-size: 12px;}  
h1 {  
    font-size: 200%;}  
h2 {  
    font-size: 1.3em;}
```

RESULT

Briards

by Ivy Duckett

The [briard](#), or berger de brie, is a large breed of dog traditionally used as a herder and guardian of sheep.

Breed History

The briard, which is believed to have originated in France, has been bred for centuries to herd and to protect sheep. The breed was used by the French Army as sentries, messengers and to search for wounded soldiers because of its fine sense of hearing. Briards were used in the First World War almost to the point of extinction. Currently the population of briards is slowly recovering. Charlemagne, Napoleon, Thomas Jefferson and Lafayette all owned briards.

SIXTEEN PIXEL SCALE

h1	32px
h2	24px
h3	18px
body	16px

=

h1	200%
h2	150%
h3	133%
body	100%

=

h1	2em
h2	1.5em
h3	1.125em
body	100%
p	1em

BOLD font-weight

The `font-weight` property allows you to create bold text. There are two values that this property commonly takes:

normal

This causes text to appear at a normal weight.

bold

This causes text to appear bold.

In this example, you can see that the element whose `class` attribute has a value of `credits` has been bolded.

You might wonder why there is a normal weight. This is because if, for example, you created a rule for the `<body>` element indicating that all text inside the body should appear bold, you might need an option that allows the text in certain instances to appear normal weight. So it is essentially used as an "off switch."

chapter-12/font-weight.html

CSS

```
.credits {  
    font-weight: bold;}
```

RESULT

Briards

by Ivy Duckett

The [briard](#), or berger de brie, is a large breed of dog traditionally used as a herder and guardian of sheep.

Breed History

The briard, which is believed to have originated in France, has been bred for centuries to herd and to protect sheep. The breed was used by the French Army as sentries, messengers and to search for wounded soldiers because of its fine sense of hearing. Briards were used in the First World War almost to the point of extinction. Currently the population of briards is slowly recovering. Charlemagne, Napoleon, Thomas Jefferson and Lafayette all owned

ITALIC font-style

CSS

chapter-12/font-style.html

```
.credits {  
    font-style: italic;}
```

RESULT

Briards

by Ivy Duckett

The briard, or berger de brie, is a large breed of dog traditionally used as a herder and guardian of sheep.

Breed History

The briard, which is believed to have originated in France, has been bred for centuries to herd and to protect sheep. The breed was used by the French Army as sentries, messengers and to search for wounded soldiers because of its fine sense of hearing. Briards were used in the First World War almost to the point of extinction. Currently the population of briards is slowly recovering. Charlemagne, Napoleon, Thomas Jefferson and Lafayette all owned briards.

If you want to create italic text, you can use the `font-style` property. There are three values this property can take:

normal

This causes text to appear in a normal style (as opposed to italic or oblique).

italic

This causes text to appear italic.

oblique

This causes text to appear oblique.

In this example, you can see that the credits have been italicized.

Italic fonts were traditionally stylized versions of the font based on calligraphy, whereas an oblique version would take the normal version and put it on an angle.

It is not unusual for the browser to fail to find an italic version of a typeface, in which case it will use an algorithm to place the normal version of the type on a slant, which means that a lot of italic text online is actually oblique.

UPPERCASE & LOWERCASE *text-transform*

The `text-transform` property is used to change the case of text giving it one of the following values:

uppercase

This causes the text to appear uppercase.

lowercase

This causes the text to appear lowercase.

capitalize

This causes the first letter of each word to appear capitalized.

In this example, the `<h1>` element is uppercase, the `<h2>` element is lowercase, and the credits are capitalized. In the HTML, the word *by* in the credits had a lowercase *b*.

If you do utilize the uppercase option, it is worth looking at the `letter-spacing` property to increase the gap between each letter as shown on page 284. This will help improve readability.

`chapter-12/text-transform.html`

CSS

```
h1 {  
    text-transform: uppercase;}  
h2 {  
    text-transform: lowercase;}  
.credits {  
    text-transform: capitalize;}
```

RESULT

BRIARDS

By Ivy Duckett

The briard, or berger de brie, is a large breed of dog traditionally used as a herder and guardian of sheep.

breed history

The briard, which is believed to have originated in France, has been bred for centuries to herd and to protect sheep. The breed was used by the French Army as sentries, messengers and to search for wounded soldiers because of its fine sense of hearing. Briards were used in the First World War almost to the point of extinction. Currently the population of briards is slowly recovering. Charlemagne, Napoleon, Thomas Jefferson and Lafayette all owned briards.

UNDERLINE & STRIKE

text-decoration

CSS

```
.credits {  
    text-decoration: underline;  
}  
  
a {  
    text-decoration: none;  
}
```

chapter-12/text-decoration.html

RESULT

Briards

by Ivy Duckett

The briard, or berger de brie, is a large breed of dog traditionally used as a herder and guardian of sheep.

Breed History

The briard, which is believed to have originated in France, has been bred for centuries to herd and to protect sheep. The breed was used by the French Army as sentries, messengers and to search for wounded soldiers because of its fine sense of hearing. Briards were used in the First World War almost to the point of extinction. Currently the population of briards is slowly recovering. Charlemagne, Napoleon, Thomas Jefferson and Lafayette all owned briards.

The `text-decoration` property allows you to specify the following values:

none

This removes any decoration already applied to the text.

underline

This adds a line underneath the text.

overline

This adds a line over the top of the text.

line-through

This adds a line through words.

blink

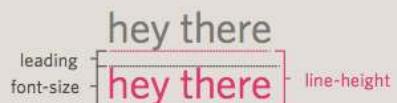
This animates the text to make it flash on and off (however this is generally frowned upon, as it is considered rather annoying).

In this example, the credits have been underlined. Also, the name of the breed (which is a link) is not underlined, which it would be by default because it is a link.

This property is commonly used by designers to remove the underlines that browsers place under links. Pages 290-291 show how to add or remove an underline when a user hovers over a link.

LEADING line-height

Leading (pronounced *ledding*) is a term typographers use for the vertical space between lines of text. In a typeface, the part of a letter that drops beneath the baseline is called a **descender**, while the highest point of a letter is called the **ascender**. Leading is measured from the bottom of the descender on one line to the top of the ascender on the next.



In CSS, the `line-height` property sets the height of an entire line of text, so the difference between the `font-size` and the `line-height` is equivalent to the leading (as shown in the diagram above).

Increasing the `line-height` makes the vertical gap between lines of text larger.

chapter-12/line-height.html

CSS

```
p {  
    line-height: 1.4em;}
```

RESULT

Briards

by Ivy Duckett

The [briard](#), or berger de brie, is a large breed of dog traditionally used as a herder and guardian of sheep.

Breed History

The briard, which is believed to have originated in France, has been bred for centuries to herd and to protect sheep. The breed was used by the French Army as sentries, messengers and to search for wounded soldiers because of its fine sense of hearing. Briards were used in the First World War almost to the point of extinction. Currently the population of briards is slowly recovering. Charlemagne, Napoleon, Thomas Jefferson and Lafayette all owned briards.

RESULT MINUS CSS

Briards

by Ivy Duckett

The [briard](#), or berger de brie, is a large breed of dog traditionally used as a herder and guardian of sheep.

Breed History

The briard, which is believed to have originated in France, has been bred for centuries to herd and to protect sheep. The breed was used by the French Army as sentries, messengers and to search for wounded soldiers because of its fine sense of hearing. Briards were used in the First World War almost to the point of extinction. Currently the population of briards is slowly recovering. Charlemagne, Napoleon, Thomas Jefferson and Lafayette all owned briards.

LETTER & WORD SPACING

letter-spacing, word-spacing

CSS

chapter-12/letter-and-word-spacing.html

```
h1, h2 {  
    text-transform: uppercase;  
    letter-spacing: 0.2em;}  
.credits {  
    font-weight: bold;  
    word-spacing: 1em;}
```

RESULT

BRIARDS

by Ivy Duckett

The briard, or berger de brie, is a large breed of dog traditionally used as a herder and guardian of sheep.

BREED HISTORY

The briard, which is believed to have originated in France, has been bred for centuries to herd and to protect sheep. The breed was used by the French Army as sentries, messengers and to search for wounded soldiers because of its fine sense of hearing. Briards were used in the First World War almost to the point of extinction. Currently the population of briards is slowly recovering. Charlemagne, Napoleon, Thomas Jefferson and Lafayette all owned briards.

RESULT MINUS CSS

Briards

by Ivy Duckett

The briard, or berger de brie, is a large breed of dog traditionally used as a herder and guardian of sheep.

Breed History

The briard, which is believed to have originated in France, has been bred for centuries to herd and to protect sheep. The breed was used by the French Army as sentries, messengers and to search for wounded soldiers because of its fine sense of hearing. Briards were used in the First World War almost to the point of extinction. Currently the population of briards is slowly recovering. Charlemagne, Napoleon, Thomas Jefferson and Lafayette all owned briards.

Kerning is the term typographers use for the space between each letter. You can control the space between each letter with the `letter-spacing` property.

It is particularly helpful to increase the kerning when your heading or sentence is all in uppercase. If your text is in sentence (or normal) case, increasing or decreasing the kerning can make it harder to read.

You can also control the gap between words using the `word-spacing` property.

When you specify a value for these properties, it should be given in ems, and it will be added on top of the default value specified by the font.

The default gap between words is set by the typeface (often around 0.25em), and it is unlikely that you would need to change this property regularly. If the typeface is bold or you have increased the space between letters, then a larger gap between words can increase readability.

ALIGNMENT

text-align

The `text-align` property allows you to control the alignment of text. The property can take one of four values:

left

This indicates that the text should be left-aligned.

right

This indicates that the text should be right-aligned.

center

This allows you to center text.

justify

This indicates that every line in a paragraph, except the last line, should be set to take up the full width of the containing box.

When you have several paragraphs of text, it is considered easiest to read if the text is left-aligned.

chapter-12/text-align.html

CSS

```
h1 {  
    text-align: left;}  
p {  
    text-align: justify;}  
.credits {  
    text-align: right;}
```

Briards

RESULT

by Ivy Duckett

The briard, or berger de brie, is a large breed of dog traditionally used as a herder and guardian of sheep.

Breed History

The briard, which is believed to have originated in France, has been bred for centuries to herd and to protect sheep. The breed was used by the French Army as sentries, messengers and to search for wounded soldiers because of its fine sense of hearing. Briards were used in the First World War almost to the point of extinction. Currently the population of briards is slowly recovering. Charlemagne, Napoleon, Thomas Jefferson and Lafayette all owned briards.

CSS3: DROP SHADOW

text-shadow

CSS

```
p.one {  
background-color: #eeeeee;  
color: #666666;  
text-shadow: 1px 1px 0px #000000;}  
p.two {  
background-color: #dddddd;  
color: #666666;  
text-shadow: 1px 1px 3px #666666;}  
p.three {  
background-color: #cccccc;  
color: #ffffff;  
text-shadow: 2px 2px 7px #111111;}  
p.four {  
background-color: #bbbbbb;  
color: #cccccc;  
text-shadow: -1px -2px #666666;}  
p.five {  
background-color: #aaaaaa;  
color: #ffffff;  
text-shadow: -1px -1px #666666;}
```

chapter-12/text-shadow.html

RESULT

The briard is known as a heart wrapped in fur.
The briard is known as a heart wrapped in fur.
The briard is known as a heart wrapped in fur.
The briard is known as a heart wrapped in fur.
The briard is known as a heart wrapped in fur.

The `text-shadow` property has become commonly used despite lacking support in all browsers.

It is used to create a drop shadow, which is a dark version of the word just behind it and slightly offset. It can also be used to create an embossed effect by adding a shadow that is slightly lighter than the text.

The value of this property is quite complicated because it can take three lengths and a color for the drop shadow.

The first length indicates how far to the left or right the shadow should fall.

The second value indicates the distance to the top or bottom that the shadow should fall.

The third value is optional and specifies the amount of blur that should be applied to the drop shadow.

The fourth value is the color of the drop shadow.

The `text-shadow` property has become very popular but at the time of writing it was not supported in any versions of Internet Explorer (currently IE9). Other browser makers introduced it in Firefox 3.1, Safari 3, Chrome 2 and Opera 9.5.

BOXES

BOX DIMENSIONS

width, height

By default a box is sized just big enough to hold its contents. To set your own dimensions for a box you can use the `height` and `width` properties.

The most popular ways to specify the size of a box are to use pixels, percentages, or ems. Traditionally, pixels have been the most popular method because they allow designers to accurately control their size.

When you use percentages, the size of the box is relative to the size of the browser window or, if the box is encased within another box, it is a percentage of the size of the containing box.

When you use ems, the size of the box is based on the size of text within it. Designers have recently started to use percentages and ems more for measurements as they try to create designs that are flexible across devices which have different-sized screens.

In the example on the right, you can see that a containing `<div>` element is used which is 300 pixels wide by 300 pixels high. Inside of this is a paragraph that is 75% of the width and height of the containing element. This means that the size of the paragraph is 225 pixels wide by 225 pixels high.

chapter-13/width-height.html

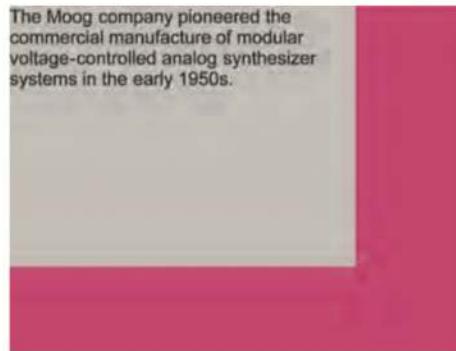
HTML

```
<div>
  <p>The Moog company pioneered the commercial
  manufacture of modular voltage-controlled
  analog synthesizer systems in the early
  1950s.</p>
</div>
```

CSS

```
div.box {
  height: 300px;
  width: 300px;
  background-color: #bbbbaa;}
p {
  height: 75%;
  width: 75%;
  background-color: #0088dd;}
```

RESULT



OVERFLOWING CONTENT

overflow

HTML

chapter-13/overflow.html

```
<h2>Fender Stratocaster</h2>
<p class="one">The Fender Stratocaster or "Strat"
    is one of the most popular electric guitars of
    all time, and its design has been copied by many
    guitar makers. It was designed by Leo... </p>
<h2>Gibson Les Paul</h2>
<p class="two">The Gibson Les Paul is a solid body
    electric guitar that was first sold in 1952.
    The Les Paul was designed by Ted McCarty... </p>
```

CSS

```
p.one {
    overflow: hidden;}
p.two {
    overflow: scroll;}
```

RESULT



The `overflow` property tells the browser what to do if the content contained within a box is larger than the box itself. It can have one of two values:

hidden

This property simply hides any extra content that does not fit in the box.

scroll

This property adds a scrollbar to the box so that users can scroll to see the missing content.

On the left, you can see two boxes whose contents expand beyond their set dimensions. The first example has the `overflow` property with a value of `hidden`. The second example has the `overflow` property with a value of `scroll`.

The `overflow` property is particularly handy because some browsers allow users to adjust the size of the text to appear as large or as small as they want. If the text is set too large then the page can become an unreadable mess. Hiding the overflow on such boxes helps prevent items overlapping on the page.

BORDER, MARGIN & PADDING

Every box has three available properties that can be adjusted to control its appearance:

1

BORDER

Every box has a border (even if it is not visible or is specified to be 0 pixels wide). The border separates the edge of one box from another.

2

MARGIN

Margins sit outside the edge of the border. You can set the width of a margin to create a gap between the borders of two adjacent boxes.

3

PADDING

Padding is the space between the border of a box and any content contained within it. Adding padding can increase the readability of its contents.

If you specify a width for a box, then the borders, margin, and padding are added to its width and height.



WHITE SPACE & VERTICAL MARGIN

WITH MARGIN & PADDING

Moog

Moog synthesisers were created by Dr. Robert Moog under the company name Moog Music. Popular models include the Moog Modular, Minimoog, Micromoog, Moog Rogue, and Moog Source.

ARP

ARP Instruments Inc. was set up by Alan Peralman, and was the main competitor for Moog during the 1970's. Popular models include the Arp 2600 and the ARP Odyssey.

Sequential Circuits

Sequential Circuits Inc was founded by Dave Smith, and the company was pivotal in the creation of MIDI. Famous models include the Prophet 5, Prophet 600, and Pro-One.

WITHOUT MARGIN & PADDING

Moog

Moog synthesisers were created by Dr. Robert Moog under the company name Moog Music. Popular models include the Moog Modular, Minimoog, Micromoog, Moog Rogue, and Moog Source.

ARP

ARP Instruments Inc. was set up by Alan Peralman, and was the main competitor for Moog during the 1970's. Popular models include the Arp 2600 and the ARP Odyssey.

Sequential Circuits

Sequential Circuits Inc was founded by Dave Smith, and the company was pivotal in the creation of MIDI. Famous models include the Prophet 5, Prophet 600, and Pro-One.

The padding and margin properties are very helpful in adding space between various items on the page.

Designers refer to the space between items on a page as **white space**. Imagine you had a border around a box. You would not want the text to touch this border or it would become harder to read.

Or, imagine you had two boxes sitting side by side (each with a black border). You would not necessarily want the boxes to touch edges as this would make the line look twice as thick on the facing sides.

If the bottom margin of any box touches the top margin of another, the browser will render it differently than you might expect. It will only show the larger of the two margins. If both margins are the same size, it will only show one.

BORDER WIDTH

border-width

The `border-width` property is used to control the width of a border. The value of this property can either be given in pixels or using one of the following values:

`thin`
`medium`
`thick`

(You cannot use percentages with this property.)

You can control the individual size of borders using four separate properties:

`border-top-width`
`border-right-width`
`border-bottom-width`
`border-left-width`

You can also specify different widths for the four border values in one property, like so:

```
border-width: 2px 1px 1px  
2px;
```

The values here appear in clockwise order: top, right, bottom, left.

chapter-13/border-width.html

HTML

```
<p class="one">Hohner's "Clavinet" is essentially an  
electric clavichord.</p>  
<p class="two">Hohner's "Clavinet" is essentially an  
electric clavichord.</p>  
<p class="three">Hohner's "Clavinet" is essentially  
an electric clavichord.</p>
```

CSS

```
p.one {  
border-width: 2px;}  
p.two {  
border-width: thick;}  
p.three {  
border-width: 1px 4px 12px 4px;}
```

RESULT

Hohner's "Clavinet" is
essentially an electric
clavichord.

Hohner's "Clavinet" is
essentially an electric
clavichord.

Hohner's "Clavinet" is
essentially an electric
clavichord.

BORDER STYLE

border-style

HTML

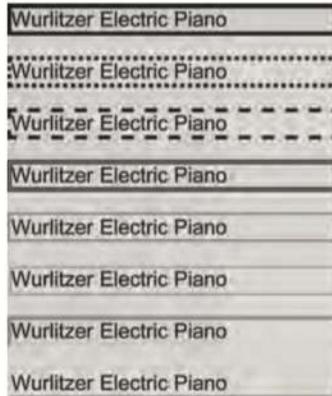
chapter-13/border-style.html

```
<p class="one">Wurlitzer Electric Piano</p>
<p class="two">Wurlitzer Electric Piano</p>
<p class="three">Wurlitzer Electric Piano</p>
<p class="four">Wurlitzer Electric Piano</p>
<p class="five">Wurlitzer Electric Piano</p>
<p class="six">Wurlitzer Electric Piano</p>
<p class="seven">Wurlitzer Electric Piano</p>
<p class="eight">Wurlitzer Electric Piano</p>
```

CSS

```
p.one {border-style: solid;}
p.two {border-style: dotted;}
p.three {border-style: dashed;}
p.four {border-style: double;}
p.five {border-style: groove;}
p.six {border-style: ridge;}
p.seven {border-style: inset;}
p.eight {border-style: outset;}
```

RESULT



You can control the style of a border using the `border-style` property. This property can take the following values:

solid a single solid line

dotted a series of square dots
(if your border is 2px wide, then the dots are 2px squared with a 2px gap between each dot)

dashed a series of short lines

double two solid lines (the value of the `border-width` property creates the sum of the two lines)

groove appears to be carved into the page

ridge appears to stick out from the page

inset appears embedded into the page

outset looks like it is coming out of the screen

hidden / none no border is shown

You can individually change the styles of different borders using:
`border-top-style`
`border-left-style`
`border-right-style`
`border-bottom-style`

BORDER COLOR

border-color

You can specify the color of a border using either RGB values, hex codes or CSS color names (as you saw on pages 251-252).

It is possible to individually control the colors of the borders on different sides of a box using:

```
border-top-color  
border-right-color  
border-bottom-color  
border-left-color
```

It is also possible to use a shorthand to control all four border colors in the one property:

```
border-color: darkcyan  
deeppink darkcyan  
deeppink;
```

The values here appear in clockwise order: top, right, bottom, left.

You could also use HSL values to specify the color as shown on pages 255-256. However, these were only introduced in CSS3 and will not work in older browsers.

chapter-13/border-color.html

HTML

```
<p class="one">The ARP Odyssey was introduced in  
1972.</p>  
<p class="two">The ARP Odyssey was introduced in  
1972.</p>
```

CSS

```
p.one {  
    border-color: #0088dd;}  
p.two {  
    border-color: #bbbbbaa #111111 #ee3e80 #0088dd;}
```

RESULT

The ARP Odyssey was
introduced in 1972.

The ARP Odyssey was
introduced in 1972.

PADDING

padding

The padding property allows you to specify how much space should appear between the content of an element and its border.

The value of this property is most often specified in pixels (although it is also possible to use percentages or ems). If a percentage is used, the padding is a percentage of the browser window (or of the containing box if it is inside another box).

Please note: If a width is specified for a box, padding is added onto the width of the box.

As you can see, the second paragraph here is much easier to read because there is a space between the text and the border of the box. The box is also wider because it has padding.

You can specify different values for each side of a box using:

`padding-top`
`padding-right`
`padding-bottom`
`padding-left`

Or you can use a shorthand (where the values are in clockwise order: top, right, bottom, left):

`padding: 10px 5px 3px 1px;`

chapter-13/padding.html

HTML

```
<p>Analog synths produce a wave sound, whereas the  
sounds stored on a digital synth have been  
sampled and then turned into numbers.</p>  
<p class="example">Analog synths produce a wave  
sound, whereas the sounds stored on a digital  
synth have been sampled and then ... </p>
```

CSS

```
p {  
width: 275px;  
border: 2px solid #0088dd;}  
p.example {  
padding: 10px;}
```

RESULT

Analog synths produce a wave sound,
whereas the sounds stored on a digital
synth have been sampled and then
turned into numbers.

Analog synths produce a wave sound,
whereas the sounds stored on a digital
synth have been sampled and then
turned into numbers.

The value of the padding property is not inherited by child elements in the same way that the color value of the font-family property is; so you need to specify the padding for every element that needs to use it.

Up until Internet Explorer 6, the width of the box would include the padding and margins. You can see more about this on page 316.

MARGIN

margin

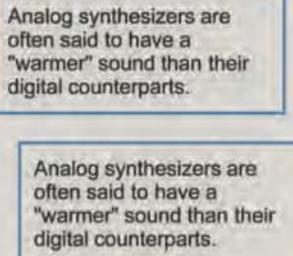
HTML

```
<p>Analog synthesizers are often said to have a  
"warmer" sound than their digital counterparts.  
</p>  
<p class="example">Analog synthesizers are often  
said to have a "warmer" sound than their digital  
counterparts.</p>
```

CSS

```
p {  
    width: 200px;  
    border: 2px solid #0088dd;  
    padding: 10px;}  
.example {  
    margin: 20px;}
```

RESULT



The value of the `margin` property is not inherited by child elements in the same way that the `color` value of the `font-family` property is, so you need to specify the `margin` for every element that needs to use it.

Up until Internet Explorer 6, the width of the box would include the padding and margins. You can see more about this on page 316.

The `margin` property controls the gap between boxes. Its value is commonly given in pixels, although you may also use percentages or ems.

If one box sits on top of another, margins are collapsed, which means the larger of the two margins will be used and the smaller will be disregarded.

Please note: If the width of a box is specified then the margin is added to the width of the box.

You can specify values for each side of a box using:

`margin-top`
`margin-right`
`margin-bottom`
`margin-left`

You can also use the shorthand (where the values are in clockwise order: top, right, bottom, left):

`margin: 1px 2px 3px 4px;`

Sometimes you might see the following, which means that the left and right margins should be 10 pixels and the top and bottom margins should be 20 pixels:

`margin: 10px 20px;`

(This same shorthand shown above can also be applied to `padding`.)

CENTERING CONTENT

If you want to center a box on the page (or center it inside the element that it sits in), you can set the `left-margin` and `right-margin` to `auto`.

In order to center a box on the page, you need to set a `width` for the box (otherwise it will take up the full width of the page).

Once you have specified the width of the box, setting the left and right margins to `auto` will make the browser put an equal gap on each side of the box. This centers the box on the page (or within the element that the box sits inside).

In order for this to work in older browsers (particularly IE6), the element that the box sits inside should have a `text-align` property with its value set to `center`.

The `text-align` property is inherited by child elements. You therefore also need to specify the `text-align` property on the centered box if you do not want the text inside it to be centered.

chapter-13/centering-content.html

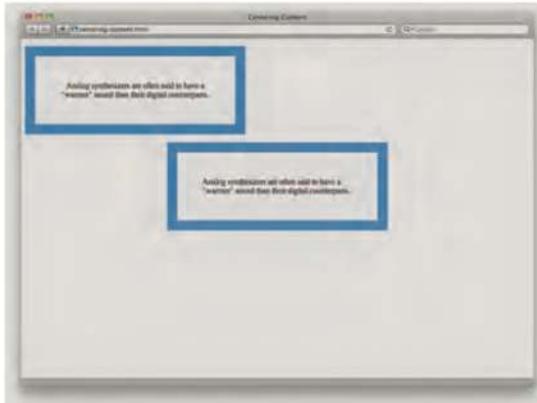
HTML

```
<body>
  <p>Analog synthesizers are often said to have a
     "warmer" sound than their digital
     counterparts.</p>
  <p class="example">Analog synthesizers are often
     said to have a "warmer" sound than their
     digital counterparts.</p>
</body>
```

CSS

```
body {
  text-align: center;
}
p {
  width: 300px;
  padding: 50px;
  border: 20px solid #0088dd;
}
.example {
  margin: 10px auto 10px auto;
  text-align: left;
```

RESULT



KEY CONCEPTS IN POSITIONING ELEMENTS

BUILDING BLOCKS

CSS treats each HTML element as if it is in its own box. This box will either be a **block-level** box or an **inline** box.

Block-level boxes start on a new line and act as the main building blocks of any layout, while inline boxes flow between surrounding text. You can control how much space each box takes up by setting the width of the boxes (and sometimes the height, too). To separate boxes, you can use borders, margins, padding, and background colors.

BLOCK-LEVEL ELEMENTS START ON A NEW LINE

Examples include:

<h1> <p>

Section 2

Lore ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit.

- Lorem ipsum dolor sit
 - Consectetur adipisicing
 - Elit, sed do eiusmod

INLINE ELEMENTS FLOW IN BETWEEN SURROUNDING TEXT

Examples include:

 <i>

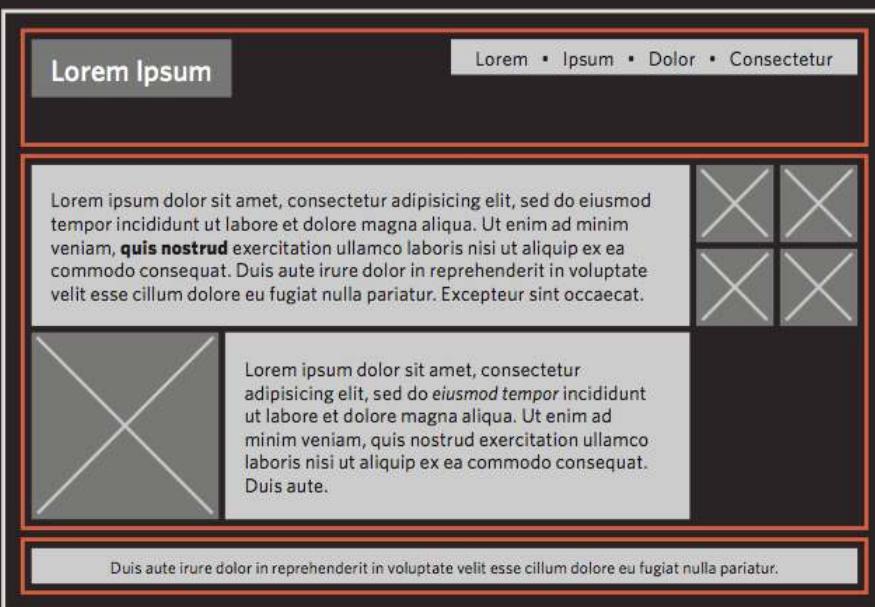


Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed ut eiusmod tempor incididunt ut labore et dolore magna aliqua.

CONTAINING ELEMENTS

If one block-level element sits inside another block-level element then the outer box is known as the **containing** or **parent** element.

It is common to group a number of elements together inside a `<div>` (or other block-level) element. For example, you might group together all of the elements that form the header of a site (such as the logo and the main navigation). The `<div>` element that contains this group of elements is then referred to as the **containing element**.



A box may be nested inside several other block-level elements. The containing element is always the **direct parent** of that element.

The orange lines in this diagram represent `<div>` elements. The header (containing the logo and navigation) are in one `<div>` element, the main content of the page is in another, and the footer is in a third. The `<body>` element is the containing element for these three `<div>` elements. The second `<div>` element is the containing element for two paragraphs of Latin text and images (represented by crossed squares).

CONTROLLING THE POSITION OF ELEMENTS

CSS has the following **positioning schemes** that allow you to control the layout of a page: **normal flow**, **relative positioning**, and **absolute positioning**. You specify the positioning scheme using the **position** property in CSS. You can also float elements using the **float** property.

NORMAL FLOW

Every block-level element appears on a new line, causing each item to appear lower down the page than the previous one. Even if you specify the width of the boxes and there is space for two elements to sit side-by-side, they will not appear next to each other. This is the default behavior (unless you tell the browser to do something else).

The paragraphs appear one after the other, vertically down the page.

RELATIVE POSITIONING

This moves an element from the position it would be in normal flow, shifting it to the top, right, bottom, or left of where it would have been placed. This does not affect the position of surrounding elements; they stay in the position they would be in in normal flow.

The second paragraph has been pushed down and right from where it would otherwise have been in normal flow.

ABSOLUTE POSITIONING

This positions the element in relation to its containing element. It is taken out of normal flow, meaning that it does not affect the position of any surrounding elements (as they simply ignore the space it would have taken up). Absolutely positioned elements move as users scroll up and down the page.

The heading is positioned to the top right, and the paragraphs start at the top of the screen (as if the heading were not there).

NORMAL FLOW

position:static

In normal flow, each block-level element sits on top of the next one. Since this is the default way in which browsers treat HTML elements, you do not need a CSS property to indicate that elements should appear in normal flow, but the syntax would be:

```
position: static;
```

I have not specified a width property for the heading element, so you can see how it stretches the width of the entire browser window by default.

The paragraphs are restricted to 450 pixels wide. This shows how the elements in normal flow start on a new line even if they do not take up the full width of the browser window.

All of the examples that demonstrate positioning will use a similar HTML structure.

chapter-15/normal-flow.html

HTML

```
<body>
  <h1>The Evolution of the Bicycle</h1>
  <p>In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster...</p>
</body>
```

CSS

```
body {
  width: 750px;
  font-family: Arial, Verdana, sans-serif;
  color: #665544;}
h1 {
  background-color: #efefef;
  padding: 10px;}
p {
  width: 450px;}
```

RESULT

The Evolution of the Bicycle

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster: two same-size in-line wheels, the front one steerable, mounted in a frame upon which you straddled. The device was propelled by pushing your feet against the ground, thus rolling yourself and the device forward in a sort of gliding walk.

The machine became known as the Draisienne (or "hobby horse"). It was made entirely of wood. This enjoyed a short lived popularity as a fad, not being practical for transportation in any other place than a well maintained pathway such as in a park or garden.

The next appearance of a two-wheeled riding machine was in 1865, when pedals were applied directly to the front wheel. This machine was known as the velocipede (meaning "fast foot") as well as the "bone shaker," since its wooden structure combined with the cobblestone roads of the day made for an extremely uncomfortable ride. They also became a fad and indoor riding academies, similar to roller rinks, could be found in large cities.

RELATIVE POSITIONING

position: relative

HTML

chapter-15/position-relative.html

```
<body>
  <h1>The Evolution of the Bicycle</h1>
  <p>In 1817 Baron von Drais invented a walking
    machine that would help him get around the
    royal gardens faster...</p>
</body>
```

CSS

```
p.example {
  position: relative;
  top: 10px;
  left: 100px;}
```

RESULT

The Evolution of the Bicycle

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster: two same-size in-line wheels, the front one steerable, mounted in a frame upon which you straddled. The device was propelled by pushing your feet against the ground, thus rolling yourself and the device forward in a sort of gliding walk.

The machine became known as the Draisienne (or "hobby horse"). It was made entirely of wood. This enjoyed a short lived popularity as a fad, not being practical for transportation in any other place than a well maintained pathway such as in a park or garden.

The next appearance of a two-wheeled riding machine was in 1865, when pedals were applied directly to the front wheel. This machine was known as the velocipede (meaning "fast foot") as well as the "bone shaker," since its wooden structure combined with the cobblestone roads of the day made for an extremely uncomfortable ride. They also became a fad and indoor riding academies, similar to roller rinks, could be found in large cities.

Relative positioning moves an element in relation to where it would have been in normal flow.

For example, you can move it 10 pixels lower than it would have been in normal flow or 20% to the right.

You can indicate that an element should be relatively positioned using the `position` property with a value of `relative`.

You then use the offset properties (`top` or `bottom` and `left` or `right`) to indicate how far to move the element from where it would have been in normal flow.

To move the box up or down, you can use either the `top` or `bottom` properties.

To move the box horizontally, you can use either the `left` or `right` properties.

The values of the box offset properties are usually given in pixels, percentages or ems.

ABSOLUTE POSITIONING

position:absolute

When the position property is given a value of absolute, the box is taken out of normal flow and no longer affects the position of other elements on the page. (They act like it is not there.)

The box offset properties (top or bottom and left or right) specify where the element should appear in relation to its containing element.

In this example, the heading has been positioned at the top of the page and 500 pixels from its left edge. The width of the heading is set to be 250 pixels wide.

The width property has also been applied to the <p> elements in this example to prevent the text from overlapping and becoming unreadable.

By default, most browsers add a margin to the top of the <h1> element. This is why there is a gap between the top of the browser and the box containing the <h1> element. If you wanted to remove this margin, you could add the following code to the <h1> element's style rules:

```
margin: 0px;
```

chapter-15/position-absolute.html

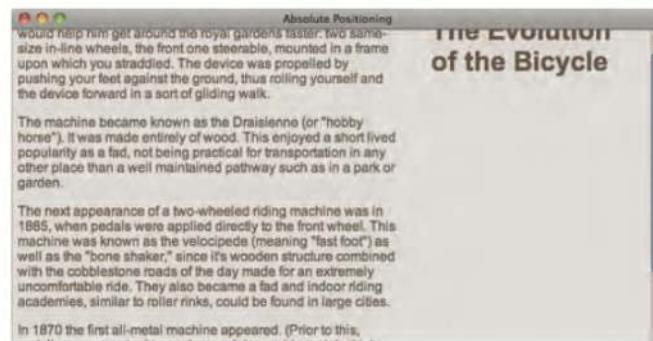
HTML

```
<body>
  <h1>The Evolution of the Bicycle</h1>
  <p>In 1817 Baron von Drais invented a walking
    machine that would help him get around the
    royal gardens faster...</p>
</body>
```

CSS

```
h1 {
  position: absolute;
  top: 0px;
  left: 500px;
  width: 250px;}
p {
  width: 450px;}
```

RESULT



FIXED POSITIONING

position:fixed

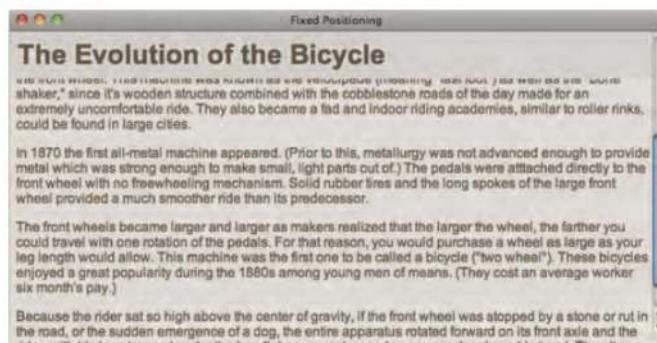
HTML

```
<body>
  <h1>The Evolution of the Bicycle</h1>
  <p class="example">In 1817 Baron von Drais
    invented a walking machine that would help him
    get around the royal gardens faster...</p>
</body>
```

CSS

```
h1 {
  position: fixed;
  top: 0px;
  left: 50px;
  padding: 10px;
  margin: 0px;
  width: 100%;
  background-color: #efefef;}
p.example {
  margin-top: 100px;}
```

RESULT



Fixed positioning is a type of absolute positioning that requires the position property to have a value of fixed.

It positions the element in relation to the browser window. Therefore, when a user scrolls down the page, it stays in the exact same place. It is a good idea to try this example in your browser to see the effect.

To control where the fixed position box appears in relation to the browser window, the box offset properties are used.

In this example, the heading has been positioned to the top left hand corner of the browser window. When the user scrolls down the page, the paragraphs disappear behind the heading.

The `<p>` elements are in normal flow and ignore the space that the `<h1>` element would have taken up. Therefore, the `margin-top` property has been used to push the first `<p>` element below where the fixed position `<h1>` element is sitting.

FLOATING ELEMENTS

float

HTML

```
<h1>The Evolution of the Bicycle</h1>
<blockquote>"Life is like riding a bicycle.
To keep your balance you must keep moving." -
Albert Einstein</blockquote>
<p>In 1817 Baron von Drais invented a walking
machine that would help him get around the royal
gardens faster: two same-size in-line wheels, the
front one steerable, mounted in a frame ... </p>
```

CSS

```
blockquote {
    float: right;
    width: 275px;
    font-size: 130%;
    font-style: italic;
    font-family: Georgia, Times, serif;
    margin: 0px 0px 10px 10px;
    padding: 10px;
    border-top: 1px solid #665544;
    border-bottom: 1px solid #665544;}
```

RESULT

The Evolution of the Bicycle

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster: two same-size in-line wheels, the front one steerable, mounted in a frame upon which you straddled. The device was propelled by pushing your feet against the ground, thus rolling yourself and the device forward in a sort of gliding walk.

The machine became known as the Draisienne (or "hobby horse"). It was made entirely of wood. This enjoyed a short lived popularity as a fad, not being practical for transportation in any other place than a well maintained pathway such as in a park or garden.

The next appearance of a two-wheeled riding machine was in 1865, when pedals were applied directly to the front wheel. This machine was known as the velocipede (meaning "fast foot") as well as the "bone shaker," since its wooden structure combined with the cobblestone roads of the day made for an extremely uncomfortable ride. They also became a fad and indoor riding academies, similar to roller rinks, could be found in large cities.

chapter-15/float.html

The `float` property allows you to take an element in normal flow and place it as far to the left or right of the containing element as possible.

Anything else that sits inside the containing element will flow around the element that is floated.

When you use the `float` property, you should also use the `width` property to indicate how wide the floated element should be. If you do not, results can be inconsistent but the box is likely to take up the full width of the containing element (just like it would in normal flow).

In this example, a `<blockquote>` element is used to hold a quotation. Its containing element is the `<body>` element.

The `<blockquote>` element is floated to the right, and the paragraphs that follow the quote flow around the floated element.



Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.

[Download Bootstrap](#)

Currently v3.3.7