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.

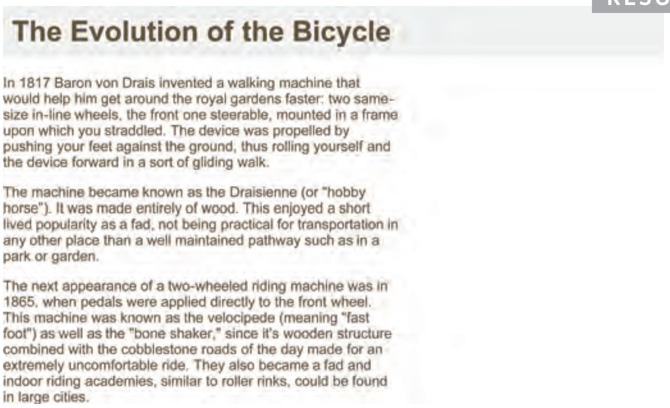
chapter-15/normal-flow.html HTML

```
<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 it's 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

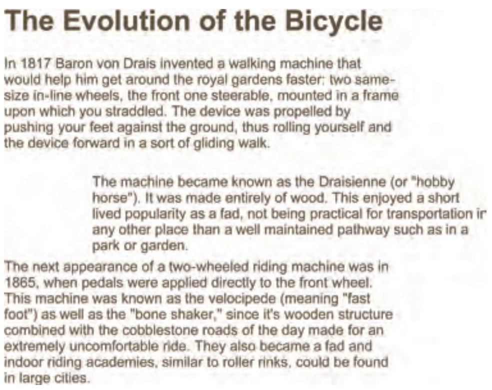
HTML chapter-15/position-relative.html

```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

```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 ir 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 it's 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

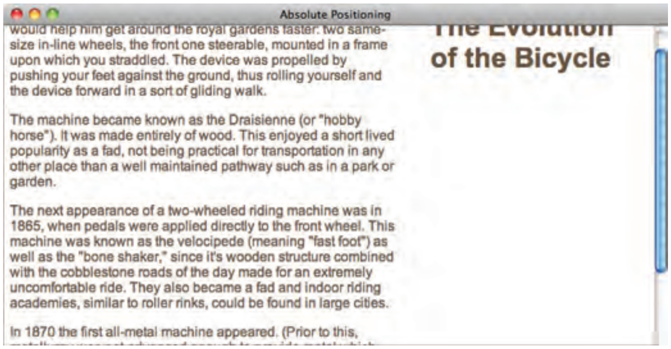
HTML

```
<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



The Evolution of the Bicycle

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 it's 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.

In 1870 the first all-metal machine appeared. (Prior to this,

FIXED POSITIONING
position:fixed

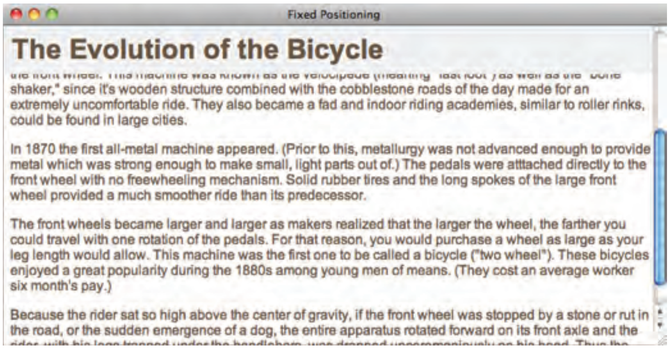
HTML chapter-15/position-fixed.html

```
<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



The Evolution of the Bicycle

the front wheel. This machine was known as the velocipede (meaning "fast foot") as well as the "bone shaker," since it's 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.

In 1870 the first all-metal machine appeared. (Prior to this, metallurgy was not advanced enough to provide metal which was strong enough to make small, light parts out of.) The pedals were atttached directly to the front wheel with no freewheeling mechanism. Solid rubber tires and the long spokes of the large front wheel provided a much smoother ride than its predecessor.

The front wheels became larger and larger as makers realized that the larger the wheel, the farther you could travel with one rotation of the pedals. For that reason, you would purchase a wheel as large as your leg length would allow. This machine was the first one to be called a bicycle ("two wheel"). These bicycles enjoyed a great popularity during the 1880s among young men of means. (They cost an average worker six month's pay.)

Because the rider sat so high above the center of gravity, if the front wheel was stopped by a stone or rut in the road, or the sudden emergence of a dog, the entire apparatus rotated forward on its front axle and the rider, with his legs trapped under the handlebars, was dropped unceremoniously on his head. Thus the

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.

OVERLAPPING ELEMENTS

z-index

When you use relative, fixed, or absolute positioning, boxes can overlap. If boxes do overlap, the elements that appear later in the HTML code sit on top of those that are earlier in the page.

If you want to control which element sits on top, you can use the z-index property. Its value is a number, and the higher the number the closer that element is to the front. For example, an element with a z-index of 10 will appear over the top of one with a z-index of 5.

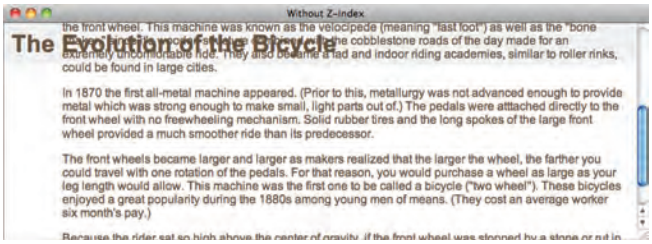
This example looks similar to the one on page 368, but it uses relative positioning for the <p> elements. Because the paragraphs are relatively positioned, by default they would appear over the top of the heading as the user scrolls down the page. To ensure that the <h1> element stays on top, we use the z-index property on the rule for the <h1> element.

The z-index is sometimes referred to as the **stacking context** (as if the blocks have been stacked on top of each other on a z axis). If you are familiar with desktop publishing packages, it is the equivalent of using the 'bring to front' and 'send to back' features.

chapter-15/z-index.html CSS

```
h1 {
  position: fixed;
  top: 0px;
  left: 0px;
  margin: 0px;
  padding: 10px;
  width: 100%;
  background-color: #efefef;
  z-index: 10;}
p {
  position: relative;
  top: 70px;
  left: 70px;}
```

RESULT WITHOUT Z-INDEX



the front wheel. This machine was known as the velocipede (meaning "fast foot") as well as the "bone

The Evolution of the Bicycle

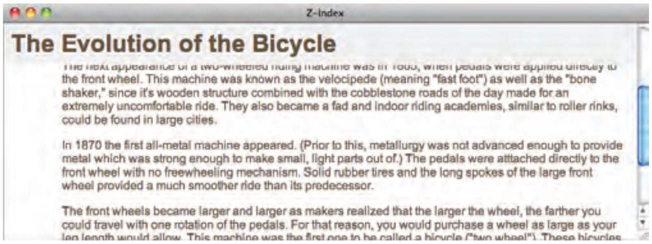
the cobblestone roads of the day made for an extremely uncomfortable ride. They also became a fad and indoor riding academies, similar to roller rinks, could be found in large cities.

In 1870 the first all-metal machine appeared. (Prior to this, metallurgy was not advanced enough to provide metal which was strong enough to make small, light parts out of.) The pedals were atttached directly to the front wheel with no freewheeling mechanism. Solid rubber tires and the long spokes of the large front wheel provided a much smoother ride than its predecessor.

The front wheels became larger and larger as makers realized that the larger the wheel, the farther you could travel with one rotation of the pedals. For that reason, you would purchase a wheel as large as your leg length would allow. This machine was the first one to be called a bicycle ("two wheel"). These bicycles enjoyed a great popularity during the 1880s among young men of means. (They cost an average worker six month's pay.)

Because the rider sat so high above the center of gravity, if the front wheel was stopped by a stone or rut in

RESULT WITH Z-INDEX



The Evolution of the Bicycle

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 it's 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.

In 1870 the first all-metal machine appeared. (Prior to this, metallurgy was not advanced enough to provide metal which was strong enough to make small, light parts out of.) The pedals were atttached directly to the front wheel with no freewheeling mechanism. Solid rubber tires and the long spokes of the large front wheel provided a much smoother ride than its predecessor.

The front wheels became larger and larger as makers realized that the larger the wheel, the farther you could travel with one rotation of the pedals. For that reason, you would purchase a wheel as large as your leg length would allow. This machine was the first one to be called a bicycle ("two wheel"). These bicycles

FLOATING ELEMENTS

float

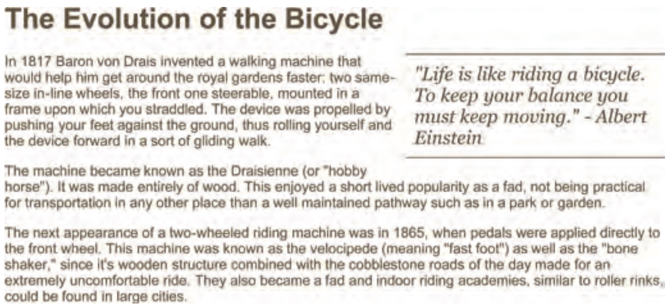
HTML chapter-15/float.html

```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

```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.

"Life is like riding a bicycle. To keep your balance you must keep moving." - Albert Einstein

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 it's 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.

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. It's 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.

USING FLOAT TO PLACE ELEMENTS SIDE-BY-SIDE

A lot of layouts place boxes next to each other. The `float` property is commonly used to achieve this.

When elements are floated, the height of the boxes can affect where the following elements sit.

In this example, you can see six paragraphs, each of which has a `width` and a `float` property set.

The fourth paragraph does not go across to the left hand edge of the page as one might expect. Rather it sits right under the third paragraph.

The reason for this is that the fourth paragraph has space to start under the third paragraph, but it cannot go any further to the left because the second paragraph is in the way.

Setting the height of the paragraphs to be the same height as the tallest paragraph would solve this issue, but it is rarely suited to real world designs where the amount of text in a paragraph or column may vary. It is more common to use the `clear` property (discussed on the next page) to solve this issue.

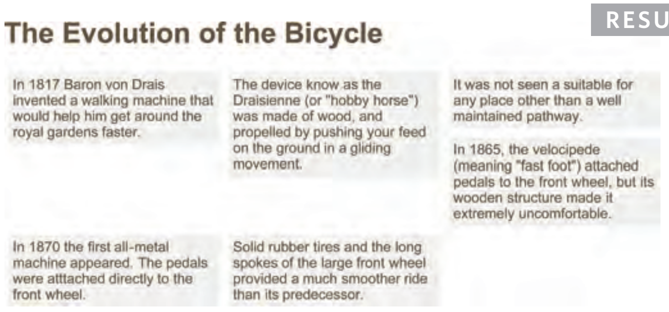
chapter-15/using-float.html HTML

```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...</p>
</body>
```

CSS

```css
body {
  width: 750px;
  font-family: Arial, Verdana, sans-serif;
  color: #665544;}
p {
  width: 230px;
  float: left;
  margin: 5px;
  padding: 5px;
  background-color: #efefef;}
```

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.

The device know as the Draisienne (or "hobby horse") was made of wood, and propelled by pushing your feed on the ground in a gliding movement.

It was not seen a suitable for any place other than a well maintained pathway.

In 1865, the velocipede (meaning "fast foot") attached pedals to the front wheel, but its wooden structure made it extremely uncomfortable.

In 1870 the first all-metal machine appeared. The pedals were atttached directly to the front wheel.

Solid rubber tires and the long spokes of the large front wheel provided a much smoother ride than its predecessor.

CLEARING FLOATS

clear

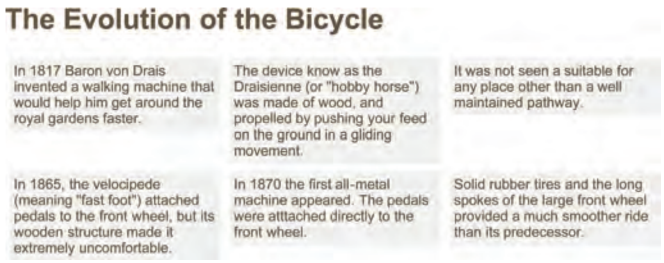
HTML chapter-15/clear.html

```
<p class="clear">In 1865, the velocipede (meaning
   "fast foot") attached pedals to the front wheel,
   but its wooden structure made it extremely
   uncomfortable.</p>
```

CSS

```
body {
  width: 750px;
  font-family: Arial, Verdana, sans-serif;
  color: #665544;}
p {
  width: 230px;
  float: left;
  margin: 5px;
  padding: 5px;
  background-color: #efefef;}
.clear {
  clear: left;}
```

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.

The device know as the Draisienne (or "hobby horse") was made of wood, and propelled by pushing your feed on the ground in a gliding movement.

It was not seen a suitable for any place other than a well maintained pathway.

In 1865, the velocipede (meaning "fast foot") attached pedals to the front wheel, but its wooden structure made it extremely uncomfortable.

In 1870 the first all-metal machine appeared. The pedals were atttached directly to the front wheel.

Solid rubber tires and the long spokes of the large front wheel provided a much smoother ride than its predecessor.

The `clear` property allows you to say that no element (within the same containing element) should touch the left or right-hand sides of a box. It can take the following values:

left

The left-hand side of the box should not touch any other elements appearing in the same containing element.

right

The right-hand side of the box will not touch elements appearing in the same containing element.

both

Neither the left nor right-hand sides of the box will touch elements appearing in the same containing element.

none

Elements can touch either side.

In this example, the fourth paragraph has a class called `clear`. The CSS rule for this class uses the `clear` property to indicate that nothing should touch the left-hand side of it. The fourth paragraph is therefore moved further down the page so no other element touches its left-hand side.

PARENTS OF FLOATED ELEMENTS: PROBLEM

If a containing element *only* contains floated elements, some browsers will treat it as if it is zero pixels tall.

As you can see in this example, the one pixel border assigned to the containing element has collapsed, so the box looks like a two pixel line.

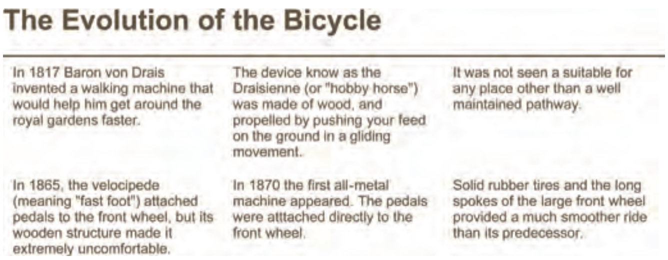
chapter-15/float-problem.html **HTML**

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

CSS

```
div {
  border: 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.

The device know as the Draisienne (or "hobby horse") was made of wood, and propelled by pushing your feed on the ground in a gliding movement.

It was not seen a suitable for any place other than a well maintained pathway.

In 1865, the velocipede (meaning "fast foot") attached pedals to the front wheel, but its wooden structure made it extremely uncomfortable.

In 1870 the first all-metal machine appeared. The pedals were atttached directly to the front wheel.

Solid rubber tires and the long spokes of the large front wheel provided a much smoother ride than its predecessor.

PARENTS OF FLOATED ELEMENTS: SOLUTION

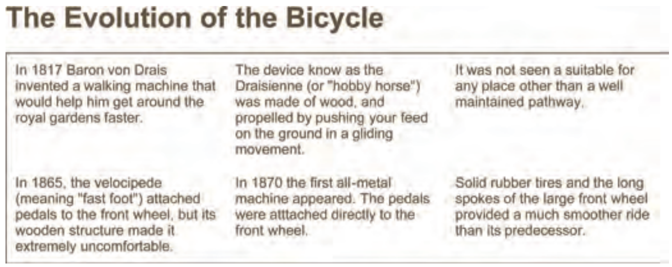
HTML chapter-15/float-solution.html

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

CSS

```
div {
  border: 1px solid #665544;
  overflow: auto;
  width: 100%;}
```

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.

The device know as the Draisienne (or "hobby horse") was made of wood, and propelled by pushing your feed on the ground in a gliding movement.

It was not seen a suitable for any place other than a well maintained pathway.

In 1865, the velocipede (meaning "fast foot") attached pedals to the front wheel, but its wooden structure made it extremely uncomfortable.

In 1870 the first all-metal machine appeared. The pedals were atttached directly to the front wheel.

Solid rubber tires and the long spokes of the large front wheel provided a much smoother ride than its predecessor.

Traditionally, developers got around this problem by adding an extra element after the last floated box (inside the containing element). A CSS rule would be applied to this additional element setting the `clear` property to have a value of `both`. But this meant that an extra element was added to the HTML just to fix the height of the containing element.

More recently, developers have opted for a purely CSS-based solution because it means that there is no need to add an extra element to the HTML page after the floated elements. The pure CSS solution adds two CSS rules to the containing element (in this example the `<div>` element):

- The `overflow` property is given a value `auto`.
- The `width` property is set to `100%`.

CREATING MULTI-COLUMN LAYOUTS WITH FLOATS

Many web pages use multiple columns in their design. This is achieved by using a `<div>` element to represent each column. The following three CSS properties are used to position the columns next to each other:

`width`

This sets the width of the columns.

`float`

This positions the columns next to each other.

`margin`

This creates a gap between the columns.

A two-column layout like the one shown on this page would need two `<div>` elements, one for the main content of the page and one for the sidebar.

Inside each of the `<div>` elements there can be headings, paragraphs, images, and even other `<div>` elements.

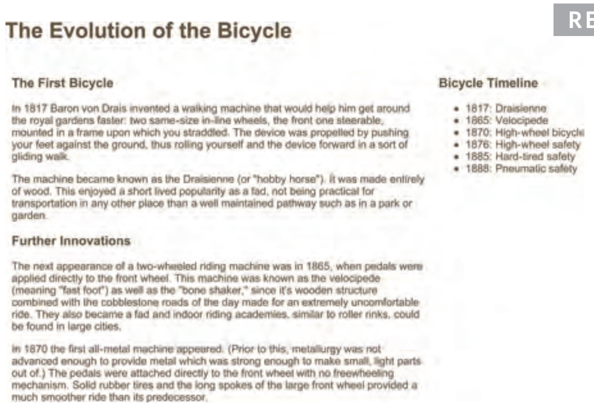
chapter-15/columns-two.html **HTML**

```html
<h1>The Evolution of the Bicycle</h1>
<div class="column1of2">
  <h3>The First Bicycle</h3>
  <p>In 1817 Baron von Drais invented a walking
     machine that would help him get around the
     royal gardens faster: two same-size ...</p>
</div>
<div class="column2of2">
  <h3>Bicycle Timeline</h3> ...
</div>
```

CSS

```css
.column1of2 {
  float: left;
  width: 620px;
  margin: 10px;}
.column2of2 {
  float: left;
  width: 300px;
  margin: 10px;}
```

RESULT



The Evolution of the Bicycle

The First 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.

Further Innovations

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 it's 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.

In 1870 the first all-metal machine appeared. (Prior to this, metallurgy was not advanced enough to provide metal which was strong enough to make small, light parts out of.) The pedals were attached directly to the front wheel with no freewheeling mechanism. Solid rubber tires and the long spokes of the large front wheel provided a much smoother ride than its predecessor.

Bicycle Timeline

- 1817: Draisienne
- 1865: Velocipede
- 1870: High-wheel bicycl
- 1876: High-wheel safety
- 1885: Hard-tired safety
- 1888: Pneumatic safety

Similarly, a three column layout could be created by floating three `<div>` elements next to each other, as shown on this page.

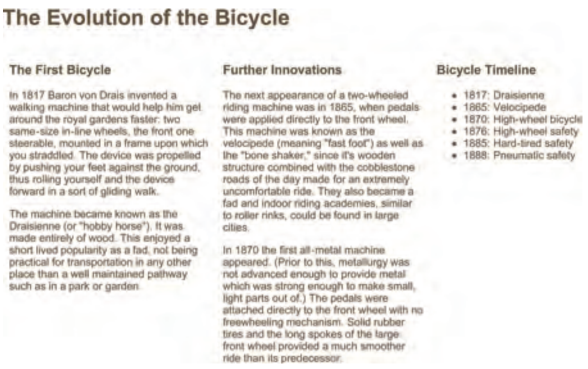
HTML chapter-15/columns-three.html

```
<h1>The Evolution of the Bicycle</h1>
<div class="column1of3">
  <h3>The First Bicycle</h3> ...
</div>
<div class="column2of3">
  <h3>Further Innovations</h3> ...
</div>
<div class="column3of3">
  <h3>Bicycle Timeline</h3> ...
</div>
```

CSS

```
.column1of3, .column2of3, .column3of3 {
  width: 300px;
  float: left;
  margin: 10px;}
```

RESULT



The Evolution of the Bicycle

The First 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.

Further Innovations

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 it's 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.

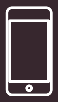
In 1870 the first all-metal machine appeared. (Prior to this, metallurgy was not advanced enough to provide metal which was strong enough to make small, light parts out of.) The pedals were attached directly to the front wheel with no freewheeling mechanism. Solid rubber tires and the long spokes of the large front wheel provided a much smoother ride than its predecessor.

Bicycle Timeline

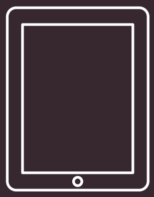
- 1817: Draisienne
- 1865: Velocipede
- 1870: High-wheel bicycle
- 1876: High-wheel safety
- 1885: Hard-tired safety
- 1888: Pneumatic safety

SCREEN SIZES

Different visitors to your site will have different sized screens that show different amounts of information, so your design needs to be able to work on a range of different sized screens.



iPhone 4
Size: 3.5 inches
Resolution: 960 x 640 pixels



iPad 2
Size: 9.7 inches
Resolution: 1024 x 768 pixels

When designing for print, you always know the size of the piece of paper that your design will be printed on. However, when it comes to designing for the web, you are faced with the unique challenge that different users will have different sized screens.

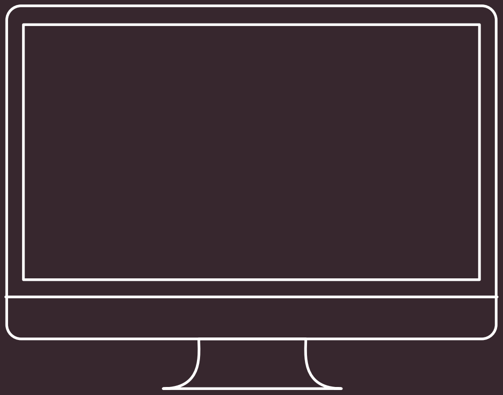
Since computers have been sold to the public, the size of screens has been steadily increasing. This means that some people viewing your site might have 13 inch monitors while others may have 27+ inch monitors.

The size of a user's screen affects how big they can open their windows and how much of the page they will see. There are also an increasing number of handheld devices (mobile phones and tablets) that have smaller screens.

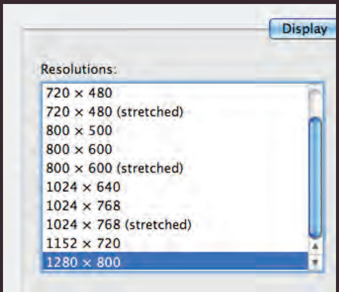
SCREEN RESOLUTION

Resolution refers to the number of dots a screen shows per inch. Some devices have a higher resolution than desktop computers and most operating systems allow users to adjust the resolution of their screens.

13" MacBook
Size: 13.3 inches
Resolution: 1280 x 800 pixels



27" iMac
Size: 27 inches
Resolution: 2560 x 1440 pixels



Resolutions:
720 × 480
720 × 480 (stretched)
800 × 500
800 × 600
800 × 600 (stretched)
1024 × 640
1024 × 768
1024 × 768 (stretched)
1152 × 720
1280 × 800

Most computers will allow owners to adjust the resolution of the display or the number of pixels that are shown on the screen. For example, here you can see the options to change the screen size from 720 x 480 pixels up to 1280 x 800 pixels.

It is interesting to note that the higher the resolution, the smaller the text appears. Many mobile devices have screens that are higher resolution than their desktop counterparts.

PAGE SIZES

Because screen sizes and display resolutions vary so much, web designers often try to create pages of around 960-1000 pixels wide (since most users will be able to see designs this wide on their screens).

Judging the height that people are likely to see on the screen without scrolling down the page is much harder. For several years, designers assumed that users would see the top 570-600 pixels of a page without having to scroll and some tried to fit all of the key messages in this area (fearing that people would not scroll down the page).

As screen sizes have increased and handheld devices have become more popular, the area users will see is far more variable.

The area of the page that users would see without scrolling was often referred as being "above the fold" (a term newspapers had originally coined to describe the area of the front page you would see if the paper were folded in half).

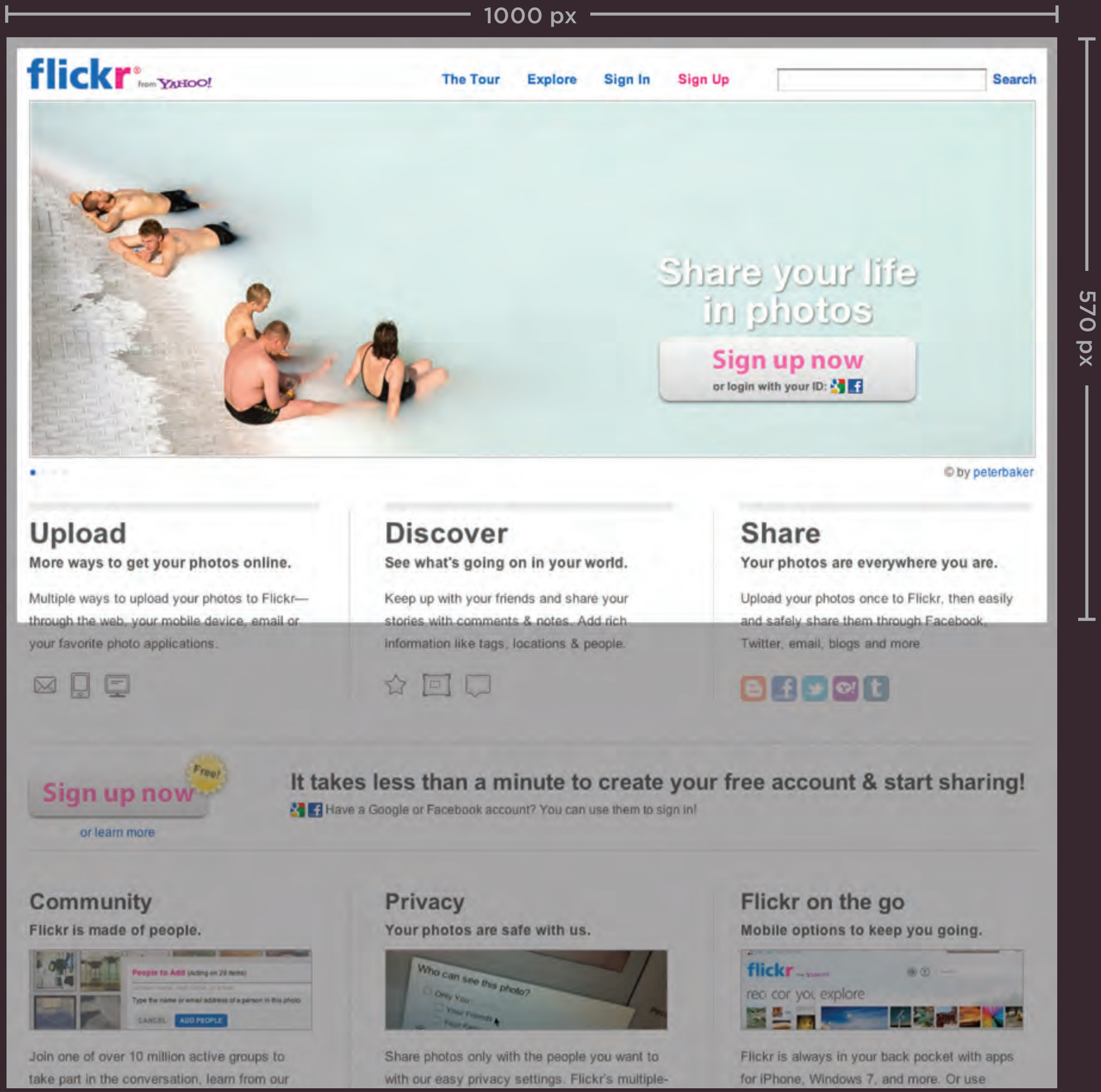
It is now recognized that if someone is interested in the content of the page, they are likely to scroll down to see more. Having said which, usability studies have shown that visitors can judge a page in under a second so it is still important to let new visitors know that the site is relevant to them and their interests.

As a result, many designs still try to let the user know what the site is about within the top 570-600 pixels, as well as hint at more content below this point. But do not try to cram too much into that top area.

At the time of writing, there was a growing trend for people to create adaptive or responsive designs that could change depending on the size of the screen.

The shaded area is hidden by the constraints of the browser window, so the user must scroll in order to view the lower region.

However, the user gets a taste for what is lower on the page and can tell that there will be more to see if they scroll down.



Upload

More ways to get your photos online.

Multiple ways to upload your photos to Flickr—through the web, your mobile device, email or your favorite photo applications.

Discover

See what's going on in your world.

Keep up with your friends and share your stories with comments & notes. Add rich information like tags, locations & people.

Share

Your photos are everywhere you are.

Upload your photos once to Flickr, then easily and safely share them through Facebook, Twitter, email, blogs and more.

It takes less than a minute to create your free account & start sharing!

Have a Google or Facebook account? You can use them to sign in!

Community

Flickr is made of people.

Join one of over 10 million active groups to take part in the conversation, learn from our

Privacy

Your photos are safe with us.

Share photos only with the people you want to with our easy privacy settings. Flickr's multiple-

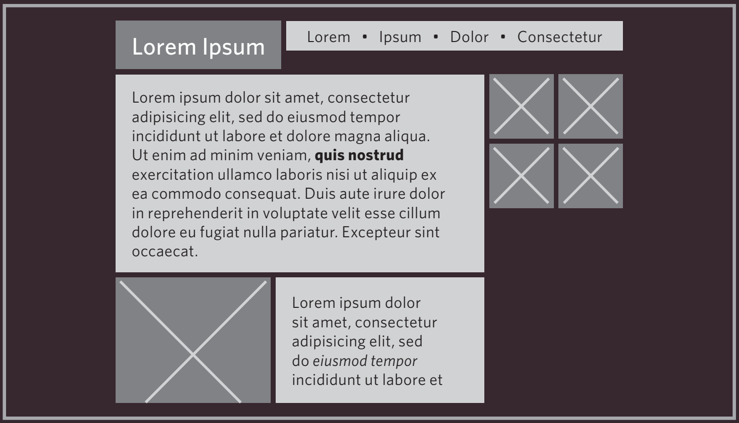
Flickr on the go

Mobile options to keep you going.

Flickr is always in your back pocket with apps for iPhone, Windows 7, and more. Or use

FIXED WIDTH LAYOUTS

Fixed width layout designs do not change size as the user increases or decreases the size of their browser window. Measurements tend to be given in pixels.



Lorem 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 esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do *eiusmod tempor* incididunt ut labore et

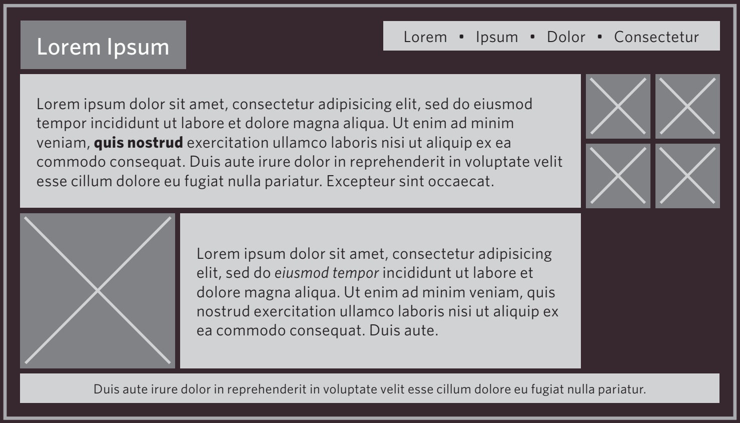
ADVANTAGES

- Pixel values are accurate at controlling size and positioning of elements.
- The designer has far greater control over the appearance and position of items on the page than with liquid layouts.
- You can control the lengths of lines of text regardless of the size of the user's window.
- The size of an image will always remain the same relative to the rest of the page.

DISADVANTAGES

- You can end up with big gaps around the edge of a page.
- If the user's screen is a much higher resolution than the designer's screen, the page can look smaller and text can be harder to read.
- If a user increases font sizes, text might not fit into the allotted spaces.
- The design works best on devices that have a site or resolution similar to that of desktop or laptop computers.
- The page will often take up more vertical space than a liquid layout with the same content.

LIQUID LAYOUTS



Lorem 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 esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat.

Lorem 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.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Liquid layout designs stretch and contract as the user increases or decreases the size of their browser window. They tend to use percentages.

ADVANTAGES

- Pages expand to fill the entire browser window so there are no spaces around the page on a large screen.
- If the user has a small window, the page can contract to fit it without the user having to scroll to the side.
- The design is tolerant of users setting font sizes larger than the designer intended (because the page can stretch).

DISADVANTAGES

- If you do not control the width of sections of the page then the design can look very different than you intended, with unexpected gaps around certain elements or items squashed together.
- If the user has a wide window, lines of text can become very long, which makes them harder to read.
- If the user has a very narrow window, words may be squashed and you can end up with few words on each line.
- If a fixed width item (such as an image) is in a box that is too small to hold it (because the user has made the window smaller) the image can overflow over the text.

Because liquid layouts can stretch the entire width of the browser, resulting in long lines of text that are hard to read, some liquid layouts only let part of the page expand and contract. Other parts of the page have minimum and maximum widths.

A FIXED WIDTH LAYOUT

To create a fixed width layout, the width of the main boxes on a page will usually be specified in pixels (and sometimes their height, too).

Here you can see several `<div>` elements, each of which uses an `id` or `class` attribute to indicate its purpose on the page.

In a book like this, the result of both the fixed and liquid layouts look similar. To get a real feel for them, you need to view them in your browser and see how they react when you adjust the size of the browser window.

The fixed width layout will stay the same width no matter what size the browser window is, whereas the liquid layout will stretch (or shrink) to fill the screen.

The HTML is the same for both the fixed width layout example on this page and the liquid layout example you see next.

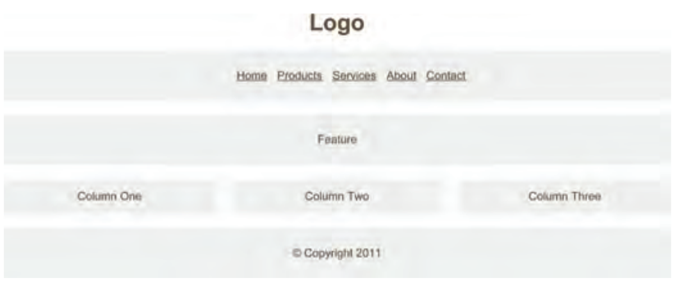
chapter-15/fixed-width-layout.html **HTML**

```
<body>
  <div id="header">
    <h1>Logo</h1>
    <div id="nav">
      <ul>
        <li><a href="">Home</a></li>
        <li><a href="">Products</a></li>
        <li><a href="">Services</a></li>
        <li><a href="">About</a></li>
        <li><a href="">Contact</a></li>
      </ul>
    </div>
  </div>
  <div id="content">
    <div id="feature">
      <p>Feature</p>
    </div>
    <div class="article column1">
      <p>Column One</p>
    </div>
    <div class="article column2">
      <p>Column Two</p>
    </div>
    <div class="article column3">
      <p>Column Three</p>
    </div>
  </div>
  <div id="footer">
    <p>&copy; Copyright 2011</p>
  </div>
</body>
```

CSS

```
body {
  width: 960px;
  margin: 0 auto;}
#content {
  overflow: auto;
  height: 100%;}
#nav, #feature, #footer {
  background-color: #efefef;
  padding: 10px;
  margin: 10px;}
.column1, .column2, .column3 {
  background-color: #efefef;
  width: 300px;
  float: left;
  margin: 10px;}
li {
  display: inline;
  padding: 5px;}
```

RESULT



The rule for the `<body>` element is used to fix the width of the page at 960 pixels, and it is centered by setting the left and right margins to `auto`.

The main boxes on the page have a margin of 10 pixels to create a gap between them.

The navigation, feature, and footer panels stretch to the width of the containing element (which in this instance is the `<body>` element), so we do not need to specify a width for them.

The three columns are each 300 pixels wide and use the `float` property, which allows them to sit next to each other.

Sometimes an extra HTML element is used to contain the page, rather than fixing the width of the `<body>`. This allows the background of the browser window to have a different color than the background of the content.

A LIQUID LAYOUT

The liquid layout uses percentages to specify the width of each box so that the design will stretch to fit the size of the screen.

When trying this in your browser, remember to make the window smaller and larger.

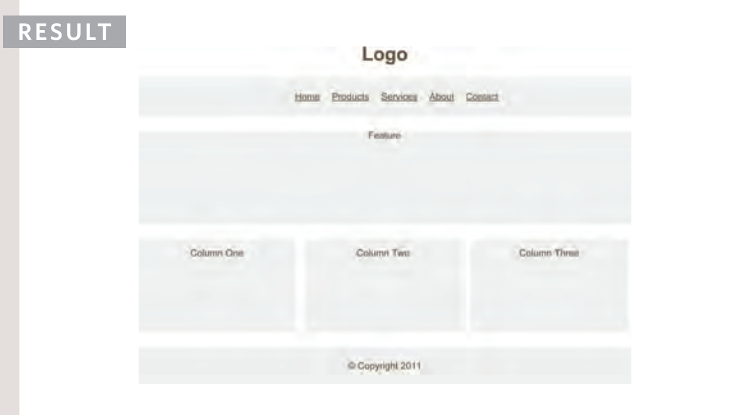
chapter-15/liquid-layout.html **HTML**

```html
<body>
  <div id="header">
    <h1>Logo</h1>
    <div id="nav">
      <ul>
        <li><a href="">Home</a></li>
        <li><a href="">Products</a></li>
        <li><a href="">Services</a></li>
        <li><a href="">About</a></li>
        <li><a href="">Contact</a></li>
      </ul>
    </div>
  </div>
  <div id="content">
    <div id="feature">
      <p>Feature</p>
    </div>
    <div class="article column1">
      <p>Column One</p>
    </div>
    <div class="article column2">
      <p>Column Two</p>
    </div>
    <div class="article column3">
      <p>Column Three</p>
    </div>
  </div>
  <div id="footer">
    <p>&copy; Copyright 2011</p>
  </div>
</body>
```

CSS

```
body {
  width: 90%;
  margin: 0 auto;}
#content {overflow: auto;}
#nav, #feature, #footer {
  margin: 1%;}
.column1, .column2, .column3 {
  width: 31.3%;
  float: left;
  margin: 1%;}
.column3 {margin-right: 0%;}
li {
  display: inline;
  padding: 0.5em;}
#nav, #footer {
  background-color: #efefef;
  padding: 0.5em 0;}
#feature, .article {
  height: 10em;
  margin-bottom: 1em;
  background-color: #efefef;}
```

RESULT



There is a rule on the `<body>` element to set the width of the page to 90% so that there is a small gap between the left and right-hand sides of the browser window and the main content.

The three columns are all given a margin of 1% and a width of 31.3%. This adds up to 99.9% of the width of the `<body>` element, so some browsers might not perfectly align the right-hand side of the third column with other elements on the page.

The `<div>` elements that hold the navigation, feature, and footer will stretch to fill the width of the containing `<body>` element. They are given a 1% margin to help them align with the columns.

If you imagine the browser window to be very wide or very narrow, you can see how lines of text could become very long or very short.

This is where the `min-width` and `max-width` properties help create boundaries within which the layout can stretch (although Internet Explorer 7 was the first version of IE to support these properties).

LAYOUT GRIDS

Composition in any visual art (such as design, painting, or photography) is the placement or arrangement of visual elements — how they are organized on a page. Many designers use a grid structure to help them position items on a page, and the same is true for web designers.

On the right, you can see a set of thick vertical lines superimposed over the top of a newspaper website to show you how the page was designed according to a grid. This grid is called the **960 pixel grid** and is widely used by web designers.

Grids set consistent proportions and spaces between items which helps to create a professional looking design.

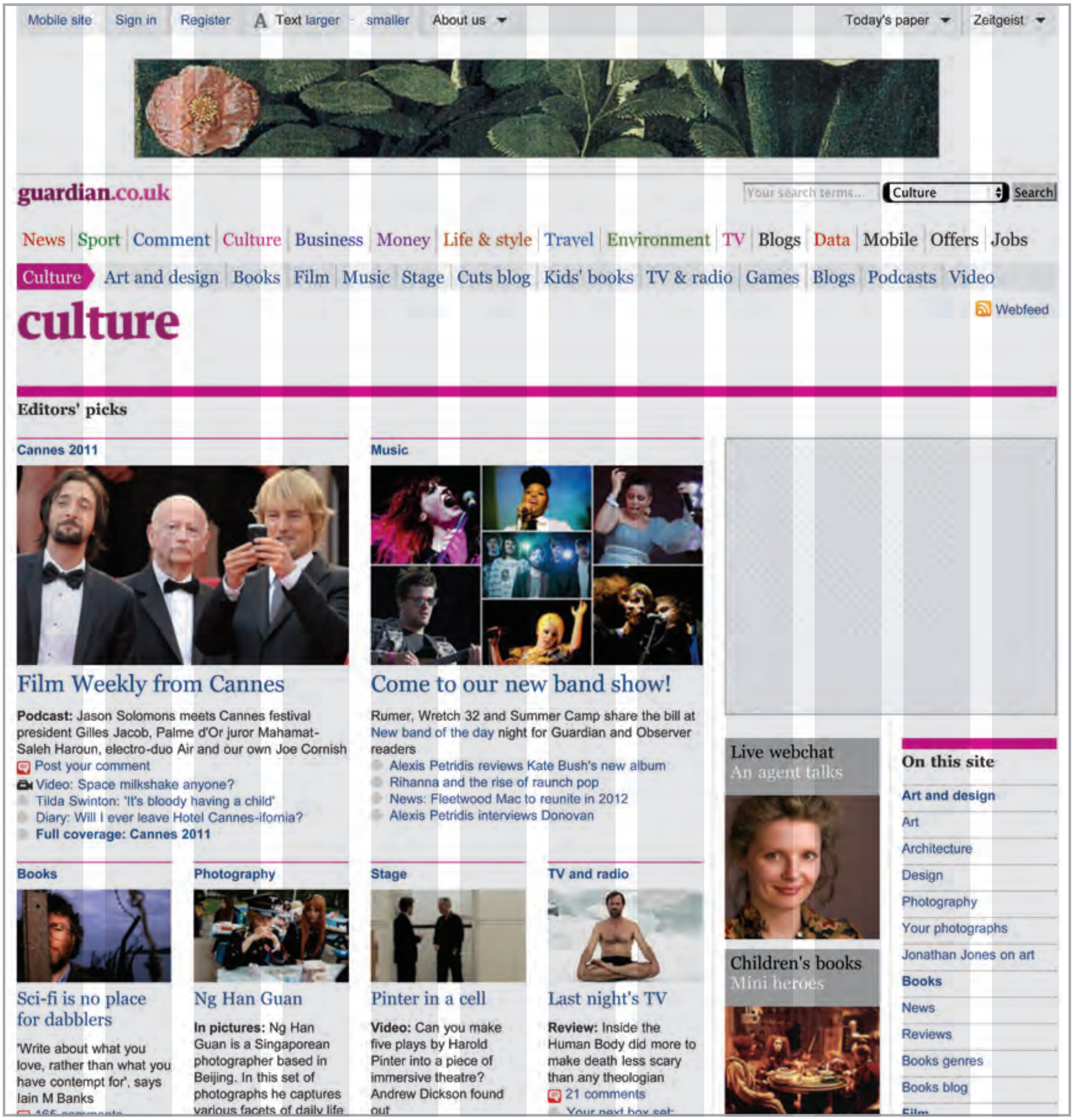
If you flick back through the pages of this book you will see that it, too, has been constructed according to a grid (comprising three columns).

As you will see on pages 389-390, it is possible to create many different layouts using this one versatile grid.

While a grid might seem like a restriction, in actual fact it:

- Creates a continuity between different pages which may use different designs
- Helps users predict where to find information on various pages
- Makes it easier to add new content to the site in a consistent way
- Helps people collaborate on the design of a site in a consistent way

EXAMPLE GRID



Mobile site | Sign in | Register | A Text larger · smaller | About us | Today's paper | Zeitgeist

guardian.co.uk

Your search terms... | Culture | Search

News | Sport | Comment | Culture | Business | Money | Life & style | Travel | Environment | TV | Blogs | Data | Mobile | Offers | Jobs

Culture | Art and design | Books | Film | Music | Stage | Cuts blog | Kids' books | TV & radio | Games | Blogs | Podcasts | Video

culture

Webfeed

Editors' picks

Cannes 2011

Film Weekly from Cannes

Podcast: Jason Solomons meets Cannes festival president Gilles Jacob, Palme d'Or juror Mahamat-Saleh Haroun, electro-duo Air and our own Joe Cornish

Post your comment

Video: Space milkshake anyone?

Tilda Swinton: 'It's bloody having a child'

Diary: Will I ever leave Hotel Cannes-ifornia?

Full coverage: Cannes 2011

Music

Come to our new band show!

Rumer, Wretch 32 and Summer Camp share the bill at New band of the day night for Guardian and Observer readers

Alexis Petridis reviews Kate Bush's new album

Rihanna and the rise of raunch pop

News: Fleetwood Mac to reunite in 2012

Alexis Petridis interviews Donovan

Books

Sci-fi is no place for dabblers

'Write about what you love, rather than what you have contempt for', says Iain M Banks

165 comments

Photography

Ng Han Guan

In pictures: Ng Han Guan is a Singaporean photographer based in Beijing. In this set of photographs he captures various facets of daily life

Stage

Pinter in a cell

Video: Can you make five plays by Harold Pinter into a piece of immersive theatre? Andrew Dickson found out

TV and radio

Last night's TV

Review: Inside the Human Body did more to make death less scary than any theologian

21 comments

Your next box set:

Live webchat

An agent talks

Children's books

Mini heroes

On this site

Art and design

Art

Architecture

Design

Photography

Your photographs

Jonathan Jones on art

Books

News

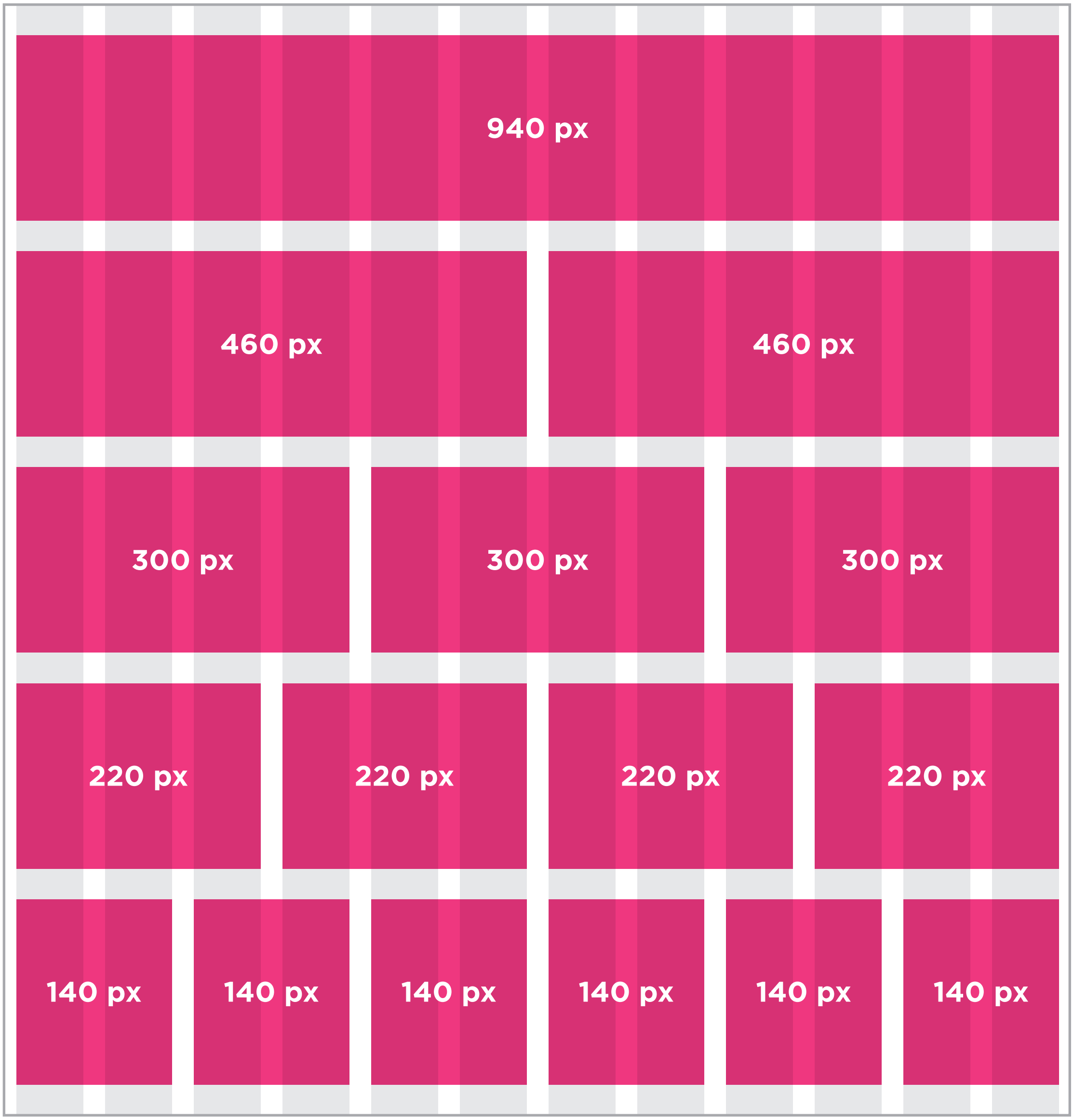
Reviews

Books genres

Books blog

Film

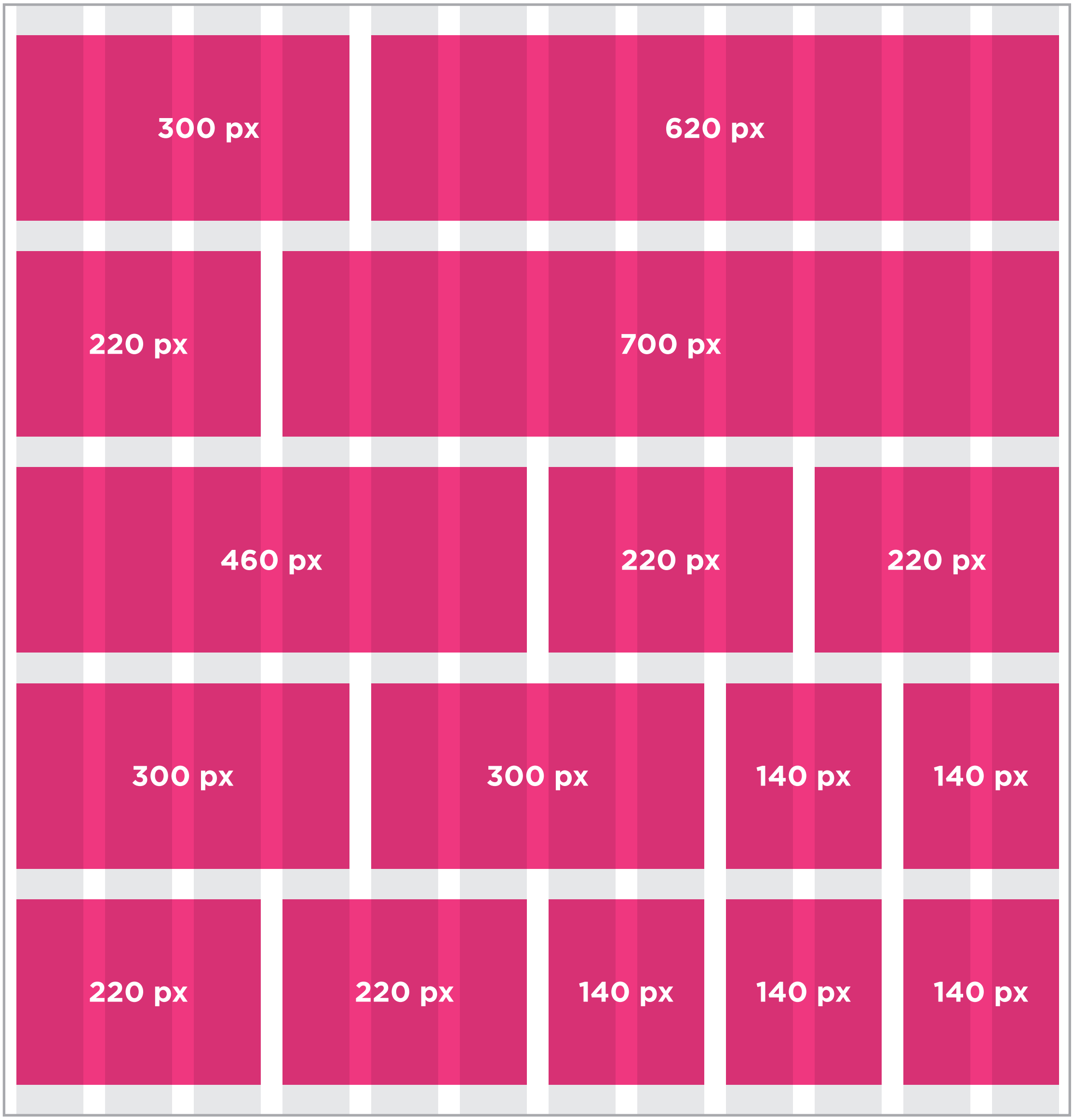
POSSIBLE LAYOUTS: 960 PIXEL WIDE 12 COLUMN GRID



These two pages illustrate a 960 pixel wide, 12 column grid. They demonstrate how it is possible to create a wide range of column layouts using this one grid.

The page is 960 pixels wide and there are 12 equal sized columns (shown in gray), each of which is is 60 pixels wide.

Each column has a margin set to 10 pixels, which creates a a gap of 20 pixels between each column and 10 pixels to the left and right-hand sides of the page.



CSS FRAMEWORKS

CSS frameworks aim to make your life easier by providing the code for common tasks, such as creating layout grids, styling forms, creating printer-friendly versions of pages and so on. You can include the CSS framework code in your projects rather than writing the CSS from scratch.

ADVANTAGES

- They save you from repeatedly writing code for the same tasks.
- They will have been tested across different browser versions (which helps avoid browser bugs).

DISADVANTAGES

- They often require that you use class names in your HTML code that only control the presentation of the page (rather than describe its content).
- In order to satisfy a wide variety of needs, they often contain more code than you need for your particular web page (commonly referred to as code "bloat").

INTRODUCING THE 960.GS CSS FRAMEWORK

One of the most popular uses of CSS frameworks is in creating grids to layout pages. There are several grid frameworks out there, but the one we will be looking at over the next few pages is the 960 Grid System (available at `www.960.gs`).

960.gs provides a style sheet that you can include in your HTML pages. Once our page links to this style sheet, you can provide the appropriate classes to your HTML code and it will create multiple column layouts for you. The 960.gs website also provides templates you can download to help design your pages using a 12 column grid. (In addition, there is a variation on the grid that uses 16 columns.)

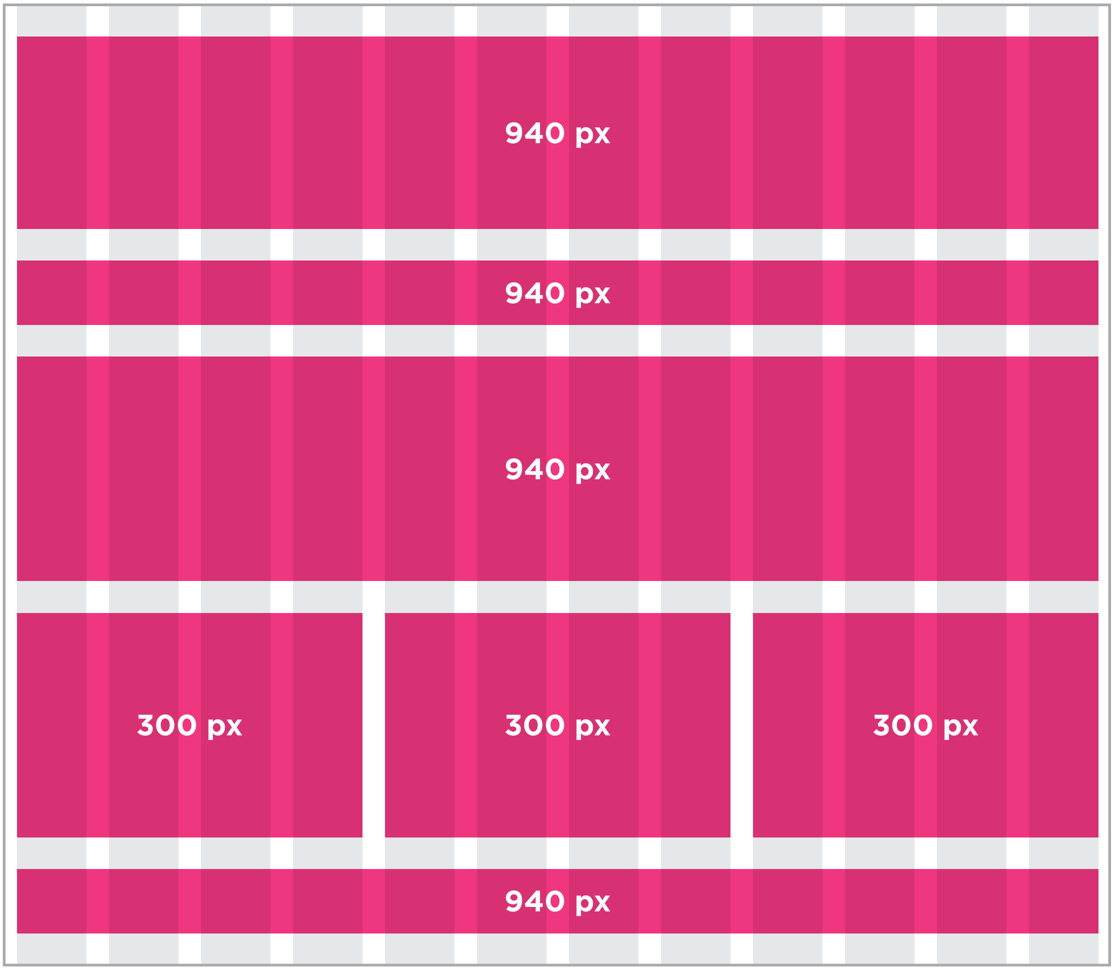
To create a 12 column grid, an element that contains the entire page is given a `class` attibute whose value is `container_12`. This sets the content of the page to be 960 pixels wide and indicates that we are using a 12 column grid.

There are different classes for blocks that take up 1, 2, 3, 4, and up to 12 columns of the grid. Each block uses class names such as `grid_3` (for a block that stretches over three columns), `grid_4` (for a block that stretches over 4 columns) and and so on through to `grid_12` (for a box that is the full width of the page). These columns all float to the left, and there is a 10 pixel margin to the left and the right of each one.

There are several other grid-based CSS frameworks available online, such as those at:
`blueprintcss.org`
`lessframework.com`
`developer.yahoo.com/yui/grids/`

USING THE 960.GS GRID

Below you can see a sample layout of a page just like the fixed width page example. On the next page, we will recreate this using the 960.gs stylesheet. Instead of writing our own CSS to control layout, we will need to add classes to the HTML indicating how wide each section should be.



A GRID-BASED LAYOUT USING 960.GS

Let's take a look at an HTML page and how it has been marked up to use the 960.gs grid system.

You can see that we include the CSS for the grid using the `<link>` element inside the `<head>` of the page.

The styles we are writing ourselves are shown on the right hand page.

The `960_12_col.css` stylesheet contains all of the rules we need to control the grid layout. The HTML uses the class names:

`container_12` to act as a container for the whole page and indicate that we are using a 12 column grid

`clearfix` to ensure that browsers know the height of the containing box, because it only contains floated elements (this addresses the issue you met on pages 371-372)

`grid_12` to create a block that is twelve columns wide

`grid_4` to create a block that is four columns wide

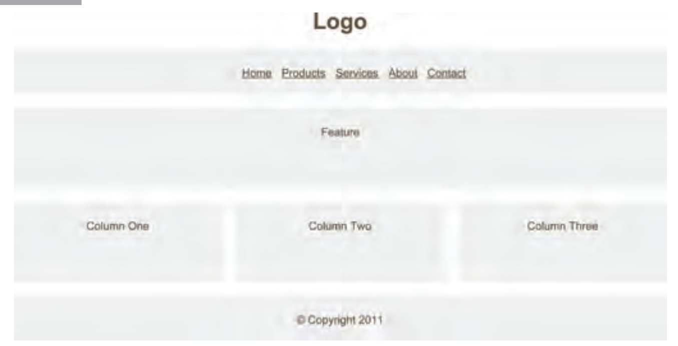
chapter-15/grid-layout.html **HTML**

```
<head>
  <title>Grid Layout</title>
  <link rel="stylesheet" type="text/css"
        href="css/960_12_col.css" />
  <style>See the right hand page</style>
</head>
<body>
  <div class="container_12 clearfix">
    <div id="header" class="grid_12">
      <h1>Logo</h1>
        <div id="nav">
          <ul>
            <li><a href="">Home</a></li>
            <li><a href="">Products</a></li>
            <li><a href="">Services</a></li>
            <li><a href="">About</a></li>
            <li><a href="">Contact</a></li>
          </ul>
        </div>
    </div>
    <div id="feature" class="grid_12">
      <p>Feature</p>
    </div>
    <div class="article grid_4">
      <p>Column One</p>
    </div>
    <div class="article grid_4">
      <p>Column Two</p>
    </div>
    <div class="article grid_4">
      <p>Column Three</p>
    </div>
    <div id="footer" class="grid_12">
      <p>&copy; Copyright 2011</p>
    </div>
  </div><!-- .container_12 -->
</body>
```

CSS

```
* {
  font-family: Arial, Verdana, sans-serif;
  color: #665544;
  text-align: center;}
#nav, #feature, .article, #footer {
  background-color: #efefef;
  margin-top: 20px;
  padding: 10px 0px 5px 0px;}
#feature, .article {
  height: 100px;}
li {
  display: inline;
  padding: 5px;}
```

RESULT



The 960.gs style sheet has taken care of the layout, creating the correct width for the columns and setting the spaces between them. Therefore, the only rules we needed to add are shown on this page. These rules:

- Control the font and the position of the text in the boxes
- Set the background colors for the boxes
- Set the height of the feature and article boxes
- Add a margin to the top and bottom of each box

MULTIPLE STYLE SHEETS
@import

Some web page authors split up their CSS style rules into separate style sheets. For example, they might use one style sheet to control the layout and another to control fonts, colors and so on.

Some authors take an even more **modular** approach to stylesheets, creating separate stylesheets to control typography, layout, forms, tables, even different styles for each sub-section of a site.

There are two ways to add multiple style sheets to a page:

1: Your HTML page can link to one style sheet and that stylesheet can use the `@import` rule to import other style sheets.

2: In the HTML you can use a separate `<link>` element for each style sheet.

The example on this page uses one `<link>` element in the HTML to link to a style sheet called `styles.css`. This stylesheet then uses the `@import` rule to import the `typography.css` and `tables.css` files.

If a styesheet uses the `@import` rule, it should appear before the other rules.

chapter-15/multiple-style-sheets-import.html HTML

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Multiple Style Sheets - Import</title>
    <link rel="stylesheet" type="text/css"
      href="css/styles.css" />
  </head>
  <body>
    <!-- HTML page content here -->
  </body>
</html>
```

chapter-15/styles.css CSS

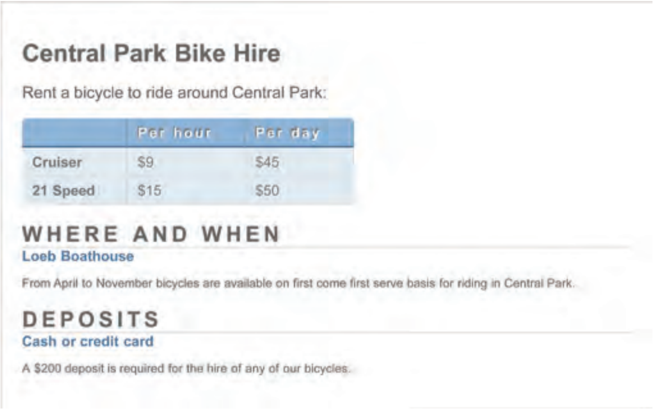
```css
@import url("tables.css");
@import url("typography.css");
body {
  color: #666666;
  background-color: #f8f8f8;
  text-align: center;}
#page {
  width: 600px;
  text-align: left;
  margin-left: auto;
  margin-right: auto;
  border: 1px solid #d6d6d6;
  padding: 20px;}
h3 {
  color: #547ca0;}
```

MULTIPLE STYLE SHEETS
link

HTML chapter-15/multiple-style-sheets-link.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Multiple Style Sheets - Link</title>
    <link rel="stylesheet" type="text/css"
      href="css/site.css" />
    <link rel="stylesheet" type="text/css"
      href="css/tables.css" />
    <link rel="stylesheet" type="text/css"
        href="css/typography.css" />
  </head>
  <body>
    <!-- HTML page content here -->
  </body>
</html>
```

RESULT



Central Park Bike Hire

Rent a bicycle to ride around Central Park:

	Per hour	Per day
Cruiser	$9	$45
21 Speed	$15	$50

WHERE AND WHEN

Loeb Boathouse

From April to November bicycles are available on first come first serve basis for riding in Central Park.

DEPOSITS

Cash or credit card

A $200 deposit is required for the hire of any of our bicycles.

On this page you can see the other technique for including multiple style sheets. Inside the `<head>` element is a separate `<link>` element for each style sheet.

The contents of `site.css` are identical to `styles.css` on the left hand page, except the code does not contain `@import` rules.

As with all style sheets, if two rules apply to the same element then rules that appear later in a document will take precedence over previous rules.

In the example on this page, any rules in `typography.css` would take precedence over rules in `site.css` (because the typography rules are included after the other rules).

In the example on the previous page, the rules in `styles.css` would take precedence over the rules in `typography.css`. This is because when the `@import` rule is used, that is the point at which the browser considers the rules live.

EXAMPLE
LAYOUT

This example demonstrates a modern magazine-style layout using the 960.gs grid. Using this style sheet saves us from having to create all of the CSS code ourselves.

Several classes from the 960.gs style sheet have been added to the code to indicate how many columns of the grid each element should stretch across. As you saw in this chapter, the 960.gs stylesheet uses the `float` property to position the blocks next to each other.

At the start of the page, the header uses fixed positioning (meaning that it does not move when the user scrolls down the page). The `z-index` property is added to the header to keep it on top of the remaining content as the user scrolls down the page.

Both the header and footer are contained within `<div>` elements which stretch the entire width of the page. Inside those container elements sit other elements that use classes from the 960.gs style sheet to ensure that the items in the header and footer align with the rest of the content.

The feature article takes up the entire width of the page. The `push_7` and `push_9` classes are part of the 960.gs style sheet and are used in the feature article to move the header and the content for this article over to the right.

Under the main article you can see four blocks, each of which is 3 columns wide. These contain images followed by links to more articles.

This example also uses background images to create a textured background for the page and header, and also to contain the images for the feature article. You will learn more about these in the next chapter.

Please note: If you view this example in Internet Explorer 6, the transparent PNGs used in this design may have a gray background. To learn more about this issue, visit the website accompanying this book where you can find a simple JavaScript that fixes this problem.

EXAMPLE

LAYOUT

```
<!DOCTYPE html
<html>
  <head>
    <title>Layout</title>
    <link rel="stylesheet" type="text/css" href="css/960_12_col.css" />
    <style type="text/css">
      @font-face {
        font-family: 'QuicksandBook';
        src: url('fonts/Quicksand_Book-webfont.eot');
        src: url('fonts/Quicksand_Book-webfont.eot?#iefix') format('embedded-opentype'),
          url('fonts/Quicksand_Book-webfont.woff') format('woff'),
          url('fonts/Quicksand_Book-webfont.ttf') format('truetype'),
          url('fonts/Quicksand_Book-webfont.svg#QuicksandBook') format('svg');
        font-weight: normal;
        font-style: normal;}
      body {
        color: #ffffff;
        background: #413f3b url("images/bg.jpg");
        font-family: Georgia, "Times New Roman", Times, serif;
        font-size: 90%;
        margin: 0px;
        text-align: center;}
      a {
        color: #b5c1ad;
        text-decoration: none;}
      a:hover {
        color: #ffffff;}
      .header {
        background-image: url("images/bg-header.jpg");
        padding: 0px 0px 0px 0px;
        height: 100px;
        position: fixed;
        top: 0px;
        width: 100%;
        z-index: 50;}
      .nav {
        float: right;
        font-family: QuicksandBook, Helvetica, Arial, sans-serif;
```

EXAMPLE
LAYOUT

```
  padding: 45px 0px 0px 0px;
  text-align: right;}
.wrapper {
  width: 960px;
  margin: 0px auto;
  background-image: url("images/bg-triangle.png");
  background-repeat: no-repeat;
  background-position: 0px 0px;
  text-align: left;}
.logo {
  margin-bottom: 20px;}
h1, h2 {
  font-family: QuicksandBook, Helvetica, Arial, sans-serif;
  font-weight: normal;
  text-transform: uppercase;}
h1 {
  font-size: 240%;
  margin-top: 140px;}
.date {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 75%;
  color: #b5c1ad;}
.intro {
  clear: left;
  font-size: 90%;
  line-height: 1.4em;}
.main-story {
  background-image: url("images/triangles.png");
  background-repeat: no-repeat;
  background-position: 122px 142px;
  height: 570px;}
.more-articles {
  border-top: 1px solid #ffffff;
  padding: 10px;}
.more-articles p {
  border-bottom: 1px solid #807c72;
  padding: 5px 0px 15px 0px;
  font-size: 80%;}
```

EXAMPLE

LAYOUT

```
    .more-articles p:last-child {
      border-bottom: none;}
    .footer {
      clear: both;
      background: rgba(0, 0, 0, 0.2);
      padding: 5px 10px;}
    .footer p {
      font-family: Helvetica, Arial, sans-serif;
      font-size: 75%;
      text-align: right;}
    .footer a {
      color: #807c72;}
  </style>
</head>
<body>
  <div class="header">
    <div class="container_12">
      <div class="grid_5">
        <img src="images/logo.png" alt="Pedal Faster - The modern bicycle magazine"
          width="216" height="37" class="logo" />
        <img src="images/header-triangle.png" alt="" width="116" height="100" />
      </div>
      <div class="nav grid_7">
        <a href="">home</a> / <a href="">news</a> / <a href="">archives</a> /
          <a href="">about</a> / <a href="">contact</a>
      </div>
    </div>
  </div>
  <div class="wrapper">
    <div class="main-story container_12">
      <div class="grid_6 push_6">
        <h1><a href="">Fixed Gear Forever</a></h1>
      </div>
      <div class="intro grid_3 push_9">
        <p class="date">16 APRIL 2011</p>
        <p>The veloheld combines minimalist design with superb quality. Devoid of
           excessive graphics and gear shift components, the veloheld product range
           delights us with its beauty and simplicity ... </p>
```

EXAMPLE
LAYOUT

```
        </div>
      </div><!-- .main-story -->
      <div class="more-articles container_12">
        <h2 class="grid_12"><a href="">More Articles</a></h2>
        <div class="grid_3">
          <img src="images/more1.jpg" alt="The road ahead" width="220" height="125" />
          <p><a href="">On the Road: from the fixed gear fanatic's point of view</a></p>
          <p><a href="">Brand History: Pashley Cycles - hand-built in England</a></p>
          <p><a href="">Frame Wars: Innovations in cycle manufacture and repair</a></p>
        </div>
        <div class="grid_3">
          <img src="images/more2.jpg" alt="Sketchbook" width="220" height="125" />
          <p><a href="">Touring Diary: A sketchbook in your basket</a></p>
          <p><a href="">Top Ten Newcomers for 2012: A peek at what's to come</a></p>
          <p><a href="">InnerTube: The best cycling videos on the web</a></p>
        </div>
        <div class="grid_3">
          <img src="images/more3.jpg" alt="Repair shop sign" width="220" height="125" />
          <p><a href="">Product Review: All baskets were not created equal</a></p>
          <p><a href="">Going Public: Out &amp; About with the founder of Public</a></p>
          <p><a href="">Cycle Lane Defence: Know your rights</a></p>
        </div>
        <div class="grid_3">
          <img src="images/more4.jpg" alt="Schwinn Spitfire" width="220" height="125" />
          <p><a href="">Bicyle Hall of Fame: The 1958 Schwinn Spitfire</a></p>
          <p><a href="">Reader Survey: Share your thoughts with us!</a></p>
          <p><a href="">Chain Gang: The evolution of the humble bike chain</a></p>
        </div>
      </div><!-- .more-articles -->
    </div><!-- .wrapper -->
    <div class="footer clearfix">
      <div class="container_12">
        <p class="grid_12"><a href="">Legal Information</a> | <a href="">Privacy
           Policy</a> | <a href="">Copyright &copy; Pedal Faster 2011</a></p>
      </div>
    </div>
  </body>
</html>
```

SUMMARY
LAYOUT

- `<div>` elements are often used as containing elements to group together sections of a page.
- Browsers display pages in normal flow unless you specify relative, absolute, or fixed positioning.
- The `float` property moves content to the left or right of the page and can be used to create multi-column layouts. (Floated items require a defined width.)
- Pages can be fixed width or liquid (stretchy) layouts.
- Designers keep pages within 960-1000 pixels wide, and indicate what the site is about within the top 600 pixels (to demonstrate its relevance without scrolling).
- Grids help create professional and flexible designs.
- CSS Frameworks provide rules for common tasks.
- You can include multiple CSS files in one page.

16

IMAGES

- Controlling size of images in CSS
- Aligning images in CSS
- Adding background images

Controlling the size and alignment of your images using CSS keeps rules that affect the presentation of your page in the CSS and out of the HTML markup.

You can also achieve several interesting effects using background images. In this chapter you will learn how to:

- Specify the size and alignment of an image using
- Add background images to boxes
- Create image rollovers in CSS

CONTROLLING SIZES OF IMAGES IN CSS

You can control the size of an image using the `width` and `height` properties in CSS, just like you can for any other box.

Specifying image sizes helps pages to load more smoothly because the HTML and CSS code will often load before the images, and telling the browser how much space to leave for an image allows it to render the rest of the page without waiting for the image to download.

You might think that your site is likely to have images of all different sizes, but a lot of sites use the same sized image across many of their pages.

For example, an e-commerce site tends to show product photos at the same size. Or, if your site is designed on a grid, then you might have a selection of image sizes that are commonly used on all pages, such as:

Small portrait: 220 x 360
Small landscape: 330 x 210
Feature photo: 620 x 400

Whenever you use consistently sized images across a site, you can use CSS to control the dimensions of the images, instead of putting the dimensions in the HTML.

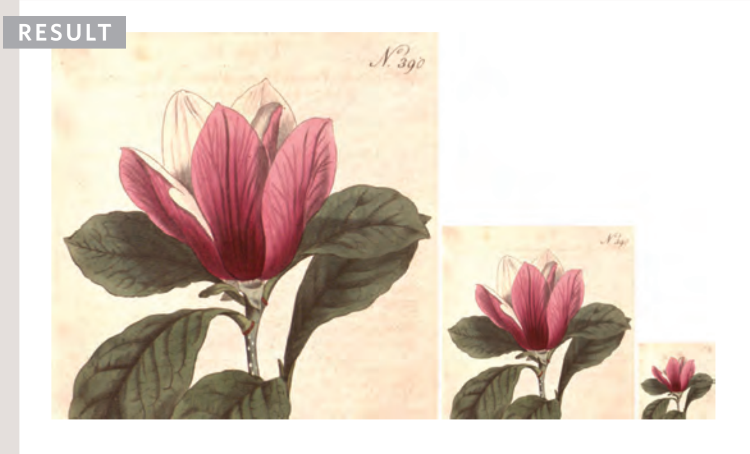
chapter-16/image-sizes.html HTML

```html
<img src="images/magnolia-large.jpg"
     class="large" alt="Magnolia" />
<img src="images/magnolia-medium.jpg"
     class="medium" alt="Magnolia" />
<img src="images/magnolia-small.jpg"
     class="small" alt="Magnolia" />
```

CSS

```css
img.large {
  width: 500px;
  height: 500px;}
img.medium {
  width: 250px;
  height: 250px;}
img.small {
  width: 100px;
  height: 100px;}
```

RESULT



First you need to determine the sizes of images that will be used commonly throughout the site, then give each size a name.

For example:
`small`
`medium`
`large`

Where the `<img>` elements appear in the HTML, rather than using `width` and `height` attributes you can use these names as values for the `class` attribute.

In the CSS, you add selectors for each of the class names, then use the CSS `width` and `height` properties to control the image dimensions.

ALIGNING IMAGES USING CSS

In the last chapter, you saw how the `float` property can be used to move an element to the left or the right of its containing block, allowing text to flow around it.

Rather than using the `<img>` element's `align` attribute, web page authors are increasingly using the `float` property to align images. There are two ways that this is commonly achieved:

1: The `float` property is added to the class that was created to represent the size of the image (such as the `small` class in our example).

2: New classes are created with names such as `align-left` or `align-right` to align the images to the left or right of the page. These class names are used in addition to classes that indicate the size of the image.

In this example you can see the `align-left` and `align-right` classes used to align the image.

It is also common to add a margin to the image to ensure that the text does not touch their edges.

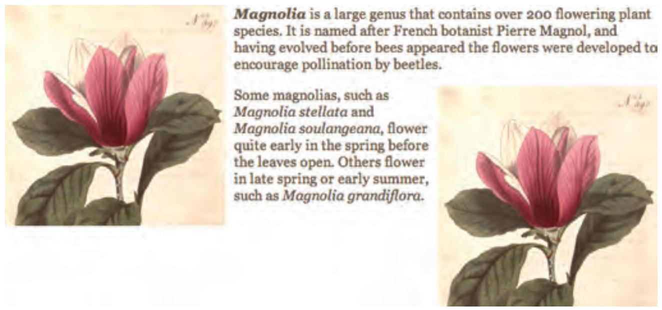
chapter-16/aligning-images.html **HTML**

```html
<p><img src="images/magnolia-medium.jpg"
   alt="Magnolia" class="align-left medium" />
   <b><i>Magnolia</i></b> is a large genus that
   contains over 200 flowering plant species...</p>
<p><img src="images/magnolia-medium.jpg"
   alt="Magnolia" class="align-right medium" />
   Some magnolias, such as <i>Magnolia stellata</i>
   and <i>Magnolia soulangeana</i>, flower quite
   early in the spring before the leaves open...</p>
```

CSS

```css
img.align-left {
  float: left;
  margin-right: 10px;}
img.align-right {
  float: right;
  margin-left: 10px;}
img.medium {
  width: 250px;
  height: 250px;}
```

RESULT



Magnolia is a large genus that contains over 200 flowering plant species. It is named after French botanist Pierre Magnol, and having evolved before bees appeared the flowers were developed to encourage pollination by beetles.

Some magnolias, such as *Magnolia stellata* and *Magnolia soulangeana*, flower quite early in the spring before the leaves open. Others flower in late spring or early summer, such as *Magnolia grandiflora.*

CENTERING IMAGES USING CSS

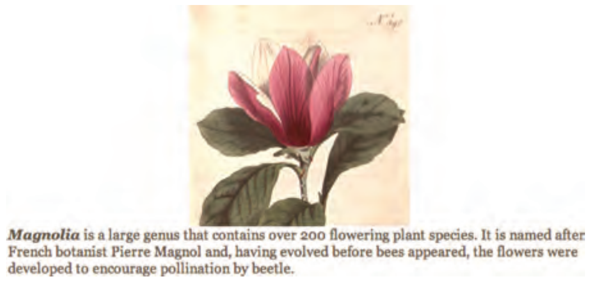
HTML chapter-16/centering-images.html

```
<p><img src="images/magnolia-medium.jpg"
    alt="Magnolia" class="align-center medium" />
   <b><i>Magnolia</i></b> is a large genus that
   contains over 200 flowering plant species. It
   is named after French botanist Pierre Magnol and,
   having evolved before bees appeared, the
   flowers were developed to encourage pollination
   by beetle.</p>
```

CSS

```
img.align-center {
  display: block;
  margin: 0px auto;}
img.medium {
  width: 250px;
  height: 250px;}
```

RESULT



Magnolia is a large genus that contains over 200 flowering plant species. It is named after French botanist Pierre Magnol and, having evolved before bees appeared, the flowers were developed to encourage pollination by beetle.

By default, images are inline elements. This means that they flow within the surrounding text. In order to center an image, it should be turned into a block-level element using the `display` property with a value of `block`.

Once it has been made into a block-level element, there are two common ways in which you can horizontally center an image:

1: On the containing element, you can use the `text-align` property with a value of `center`.

2: On the image itself, you can use the use the `margin` property and set the values of the left and right margins to `auto`.

You can specify class names that allow any element to be centered, in the same way that you can for the dimensions or alignment of images.

The techniques for specifying image size and alignment of images can also be used with the HTML5 `<figure>` element, which you met on page 119.

BACKGROUND IMAGES
background-image

The background-image property allows you to place an image behind any HTML element. This could be the entire page or just part of the page. By default, a background image will repeat to fill the entire box.

The path to the image follows the letters url, and it is put inside parentheses and quotes.

Here is the image tile used in this example.

In the first example, you can see a background image being applied to an entire page (because the CSS selector applies to the <body> element). In the second example, the background image just applies to a paragraph.

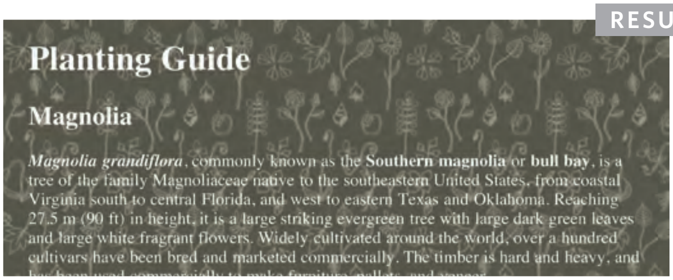
If you search online, you will find lots of resources that offer background textures that you can use on your pages.

Background images are often the last thing on the page to load (which can make a website seem slow to load). As with any images you use online, if the size of the file is large it will take longer to download.

chapter-16/background-image-body.html CSS

```
body {
  background-image: url("images/pattern.gif");}
```

RESULT



Planting Guide

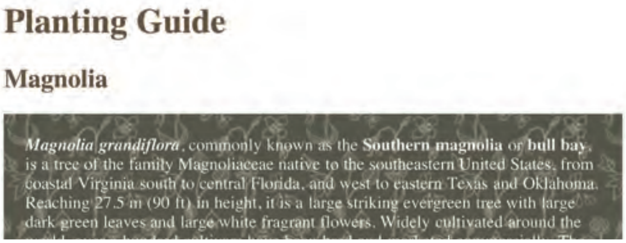
Magnolia

Magnolia grandiflora, commonly known as the **Southern magnolia** or **bull bay**, is a tree of the family Magnoliaceae native to the southeastern United States, from coastal Virginia south to central Florida, and west to eastern Texas and Oklahoma. Reaching 27.5 m (90 ft) in height, it is a large striking evergreen tree with large dark green leaves and large white fragrant flowers. Widely cultivated around the world, over a hundred cultivars have been bred and marketed commercially. The timber is hard and heavy, and

chapter-16/background-image-element.html CSS

```
p {
  background-image: url("images/pattern.gif");}
```

RESULT



Planting Guide

Magnolia

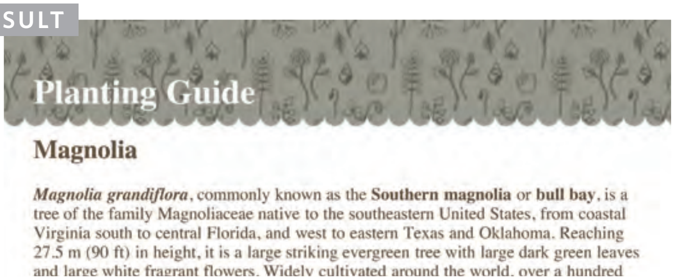
Magnolia grandiflora, commonly known as the **Southern magnolia** or **bull bay**, is a tree of the family Magnoliaceae native to the southeastern United States, from coastal Virginia south to central Florida, and west to eastern Texas and Oklahoma. Reaching 27.5 m (90 ft) in height, it is a large striking evergreen tree with large dark green leaves and large white fragrant flowers. Widely cultivated around the

REPEATING IMAGES
background-repeat
background-attachment

CSS chapter-16/background-repeat.html

```
body {
  background-image: url("images/header.gif");
  background-repeat: repeat-x;}
```

RESULT



Planting Guide

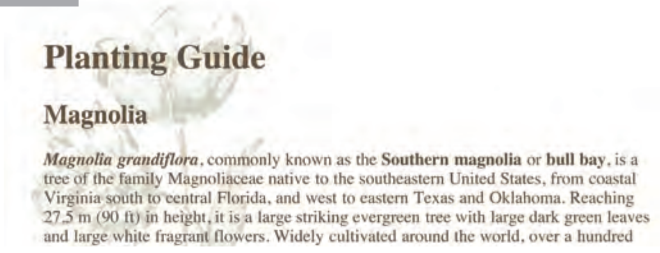
Magnolia

Magnolia grandiflora, commonly known as the **Southern magnolia** or **bull bay**, is a tree of the family Magnoliaceae native to the southeastern United States, from coastal Virginia south to central Florida, and west to eastern Texas and Oklahoma. Reaching 27.5 m (90 ft) in height, it is a large striking evergreen tree with large dark green leaves and large white fragrant flowers. Widely cultivated around the world, over a hundred

CSS chapter-16/background-attachment.html

```
body {
  background-image: url("images/tulip.gif");
  background-repeat: no-repeat;
  background-attachment: fixed;}
```

RESULT



Planting Guide

Magnolia

Magnolia grandiflora, commonly known as the **Southern magnolia** or **bull bay**, is a tree of the family Magnoliaceae native to the southeastern United States, from coastal Virginia south to central Florida, and west to eastern Texas and Oklahoma. Reaching 27.5 m (90 ft) in height, it is a large striking evergreen tree with large dark green leaves and large white fragrant flowers. Widely cultivated around the world, over a hundred

The `background-repeat` property can have four values:

repeat

The background image is repeated both horizontally and vertically (the default way it is shown if the `background-repeat` property isn't used).

repeat-x

The image is repeated horizontally only (as shown in the first example on the left).

repeat-y

The image is repeated vertically only.

no-repeat

The image is only shown once.

The `background-attachment` property specifies whether a background image should stay in one position or move as the user scrolls up and down the page. It can have one of two values:

fixed

The background image stays in the same position on the page.

scroll

The background image moves up and down as the user scrolls up and down the page.

BACKGROUND POSITION
background-position

When an image is not being repeated, you can use the `background-position` property to specify where in the browser window the background image should be placed.

This property usually has a pair of values. The first represents the horizontal position and the second represents the vertical.

- `left top`
- `left center`
- `left bottom`
- `center top`
- `center center`
- `center bottom`
- `right top`
- `right center`
- `right bottom`

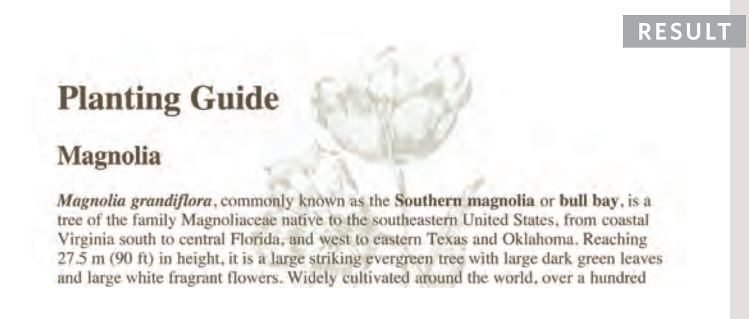
If you only specify one value, the second value will default to `center`.

You can also use a pair of pixels or percentages. These represent the distance from the top left corner of the browser window (or containing box). The top left corner is equal to `0% 0%`. The example shown, with values of `50% 50%`, centers the image horizontally and vertically.

chapter-16/background-position.html CSS

```
body {
  background-image: url("images/tulip.gif");
  background-repeat: no-repeat;
  background-position: center top;}
```

RESULT



Planting Guide

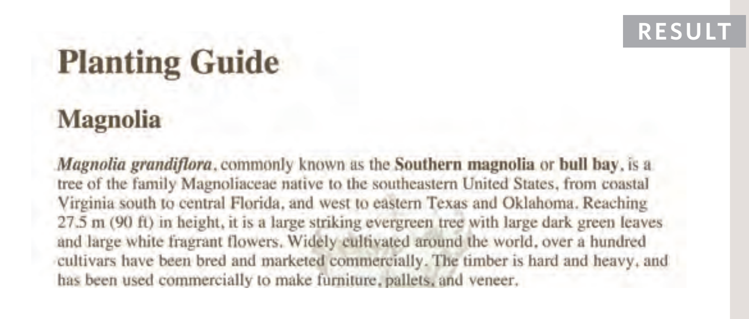
Magnolia

Magnolia grandiflora, commonly known as the **Southern magnolia** or **bull bay**, is a tree of the family Magnoliaceae native to the southeastern United States, from coastal Virginia south to central Florida, and west to eastern Texas and Oklahoma. Reaching 27.5 m (90 ft) in height, it is a large striking evergreen tree with large dark green leaves and large white fragrant flowers. Widely cultivated around the world, over a hundred

chapter-16/background-position-percentage.html CSS

```
body {
  background-image: url("images/tulip.gif");
  background-repeat: no-repeat;
  background-position: 50% 50%;}
```

RESULT



Planting Guide

Magnolia

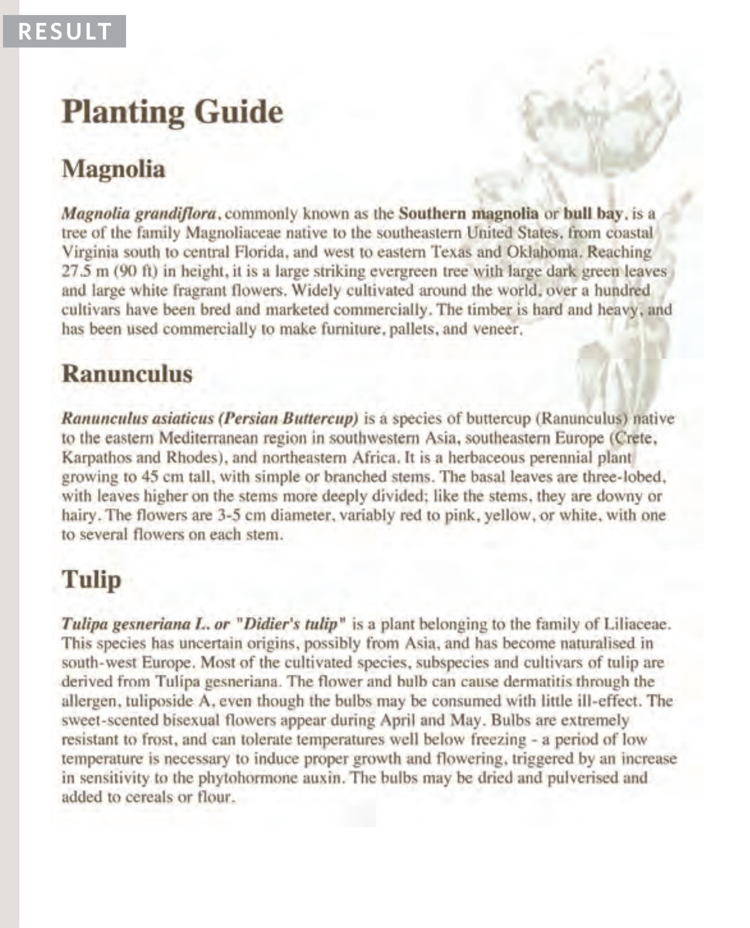
Magnolia grandiflora, commonly known as the **Southern magnolia** or **bull bay**, is a tree of the family Magnoliaceae native to the southeastern United States, from coastal Virginia south to central Florida, and west to eastern Texas and Oklahoma. Reaching 27.5 m (90 ft) in height, it is a large striking evergreen tree with large dark green leaves and large white fragrant flowers. Widely cultivated around the world, over a hundred cultivars have been bred and marketed commercially. The timber is hard and heavy, and has been used commercially to make furniture, pallets, and veneer.

SHORTHAND
background

CSS chapter-16/background-shorthand.html

```
body {
  background: #ffffff url("images/tulip.gif")
    no-repeat top right;}
```

RESULT



Planting Guide

Magnolia

Magnolia grandiflora, commonly known as the **Southern magnolia** or **bull bay**, is a tree of the family Magnoliaceae native to the southeastern United States, from coastal Virginia south to central Florida, and west to eastern Texas and Oklahoma. Reaching 27.5 m (90 ft) in height, it is a large striking evergreen tree with large dark green leaves and large white fragrant flowers. Widely cultivated around the world, over a hundred cultivars have been bred and marketed commercially. The timber is hard and heavy, and has been used commercially to make furniture, pallets, and veneer.

Ranunculus

Ranunculus asiaticus (Persian Buttercup) is a species of buttercup (Ranunculus) native to the eastern Mediterranean region in southwestern Asia, southeastern Europe (Crete, Karpathos and Rhodes), and northeastern Africa. It is a herbaceous perennial plant growing to 45 cm tall, with simple or branched stems. The basal leaves are three-lobed, with leaves higher on the stems more deeply divided; like the stems, they are downy or hairy. The flowers are 3-5 cm diameter, variably red to pink, yellow, or white, with one to several flowers on each stem.

Tulip

Tulipa gesneriana L. or "Didier's tulip" is a plant belonging to the family of Liliaceae. This species has uncertain origins, possibly from Asia, and has become naturalised in south-west Europe. Most of the cultivated species, subspecies and cultivars of tulip are derived from Tulipa gesneriana. The flower and bulb can cause dermatitis through the allergen, tuliposide A, even though the bulbs may be consumed with little ill-effect. The sweet-scented bisexual flowers appear during April and May. Bulbs are extremely resistant to frost, and can tolerate temperatures well below freezing - a period of low temperature is necessary to induce proper growth and flowering, triggered by an increase in sensitivity to the phytohormone auxin. The bulbs may be dried and pulverised and added to cereals or flour.

The `background` property acts like a shorthand for all of the other background properties you have just seen, and also the `background-color` property.

The properties must be specified in the following order, but you can miss any value if you do not want to specify it.

1: `background-color`
2: `background-image`
3: `background-repeat`
4: `background-attachment`
5: `background-position`

CSS3 will also support the use of multiple background images by repeating the `background` shorthand. Because few browsers supported this property at the time of writing, it was not commonly used.

```
div {
  background:
    url(example-1.jpg)
    top left no-repeat,
    url(example-2.jpg)
    bottom left no-repeat,
    url(example-3.jpg)
    centre top repeat-x;}
```

The first image is shown on top, with the last one on the bottom.

IMAGE ROLLOVERS & SPRITES

Using CSS, it is possible to create a link or button that changes to a second style when a user moves their mouse over it (known as a **rollover**) and a third style when they click on it.

This is achieved by setting a background image for the link or button that has three different styles of the same button (but only allows enough space to show one of them at a time). You can see the image we are using in this example on the right. It actually features two buttons on the one image.

When the user moves their mouse over the element, or clicks on it, the position of the background image is moved to show the relevant image.

When a single image is used for several different parts of an interface, it is known as a **sprite**. You can add the logo and other interface elements, as well as buttons to the image.

The advantage of using sprites is that the web browser only needs to request one image rather than many images, which can make the web page load faster.

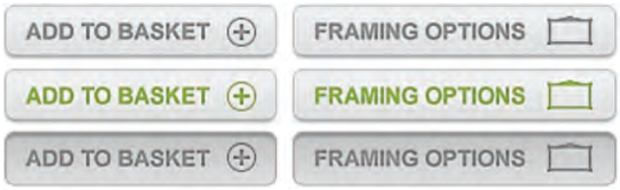
chapter-16/image-rollovers-and-sprites.html **HTML**

```html
<a class="button" id="add-to-basket">
   Add to basket</a>
<a class="button" id="framing-options">
   Framing options</a>
```

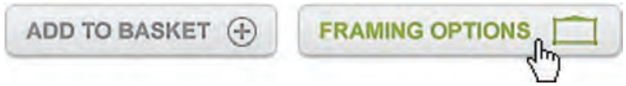
CSS

```css
a.button {
  height: 36px;
  background-image: url("images/button-sprite.jpg");
  text-indent: -9999px;
  display: inline-block;}
a#add-to-basket {
  width: 174px;
  background-position: 0px 0px;}
a#framing-options {
  width: 210px;
  background-position: -175px 0px;}
a#add-to-basket:hover {
  background-position: 0px -40px;}
a#framing-options:hover {
  background-position: -175px -40px;}
a#add-to-basket:active {
  background-position: 0px -80px;}
a#framing-options:active {
  background-position: -175px -80px;}
```

COMPLETE IMAGE SPRITE



RESULT



In this example, you can see two links that look like buttons. Each of the buttons has three different states. These are all represented in a single image.

Because the `<a>` element is an inline element, we set the `display` property of these links to indicate that they should be `inline-block` elements. This allows us to specify the width and height of each `<a>` element so that it matches the size of its corresponding button.

The `background-position` property is used to move the image in order to show the button in the right state.

When the user hovers over a link, the `:hover` pseudo-class has a rule that moves the `background-position` of the image to show a different state for that button.

Similarly, when the user clicks on the link, the `:active` pseudo-class has a rule to show the third state for that button.

Touch screen devices will not change a link's state when the user hovers over it, as the screens do not yet have a way to tell when the user is hovering. However, they will change their appearance when the user activates them.

CSS3: GRADIENTS
background-image

CSS3 is going to introduce the ability to specify a gradient for the background of a box. The gradient is created using the `background-image` property and, at the time of writing, different browsers required a different syntax.

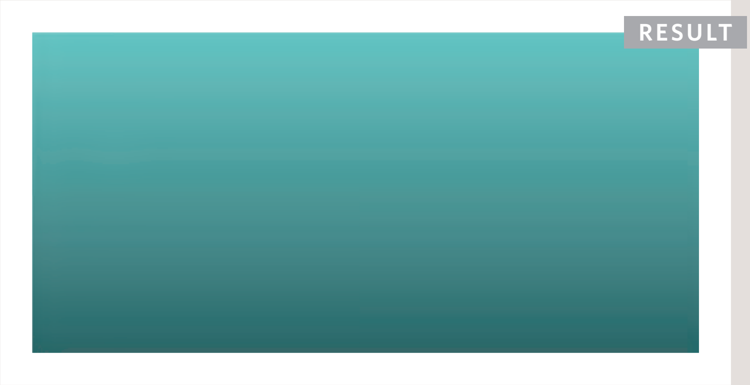
Since it is not supported by all browsers, it is possible to specify a background image for the box first (which would represent the gradient) and then provide the CSS alternatives for browsers that support gradients.

On this page, we are focusing on linear gradients. You can see that in order to create a linear gradient, we need to specify two colors (that the gradient is between).

Some browsers allow you to specify the angle of the gradient, or even different types of gradients (such as radial gradients), but support is not as widespread as that for linear gradients.

chapter-16/gradient.html CSS

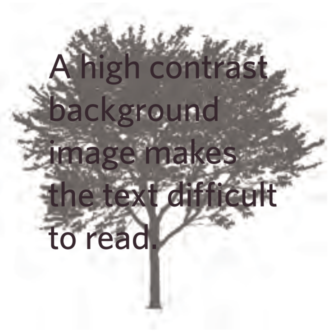
```
#gradient {
  /* fallback color */
  background-color: #66cccc;
  /* fallback image */
  background-image: url(images/fallback-image.png);
  /* Firefox 3.6+ */
  background-image: -moz-linear-gradient(#336666,
    #66cccc);
  /* Safari 4+, Chrome 1+ */
  background-image: -webkit-gradient(linear, 0% 0%,
    0% 100%, from(#66cccc), to(#336666));
  /* Safari 5.1+, Chrome 10+ */
  background-image: -webkit-linear-gradient(#336666,
    #66cccc);
  /* Opera 11.10+ */
  background-image: -o-linear-gradient(#336666,
    #66cccc);
  height: 150px;
  width: 300px;}
```



CONTRAST OF BACKGROUND IMAGES

If you want to overlay text on a background image, the image must be low contrast in order for the text to be legible.

HIGH CONTRAST


A high contrast background image makes the text difficult to read.

The majority of photographs have quite a high contrast, which means that they are not ideal for use as a background image.

LOW CONTRAST

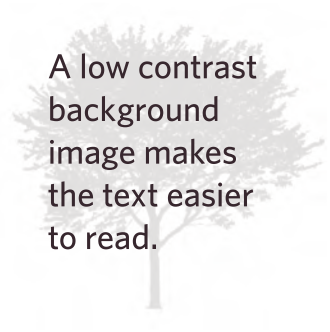
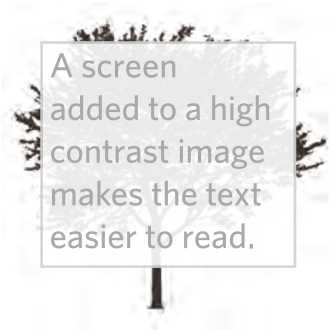

A low contrast background image makes the text easier to read.

Image editing applications such as Photoshop and GIMP have tools that allow you to manually adjust your images to have lower contrast.

SCREEN


A screen added to a high contrast image makes the text easier to read.

To overlay text on an image with high contrast, you can place a semi-transparent background color (or "screen") behind the text to improve legibility.

EXAMPLE
IMAGES

This example demonstrates how to use CSS to create a simple image gallery layout.

A background texture is applied to the whole page by repeating an image with the texture behind the <body> element. A repeating background image is sometimes referred to as **wallpaper**.

The content of the page is put inside a <div> element whose class is wrapper. This is used to fix the width of the page to 720 pixels. Its left and right margins are set to auto to center it in the middle of the screen.

The images sit inside an HTML5 <figure> element, and their captions are provided in the <figcaption> element. CSS is used to set the dimensions and background color for each <figure> element. The dimensions of the images themselves are also set using CSS, and they are given a single pixel gray border.

For the captions, a background image is used to the left of the text. We do not want this image to fill the background so we specify that it should not repeat. Padding is used to the left of the text so that the words do not go over the background image.

Each of the <figure> elements is contained within a <div>, which has two purposes. Firstly, it is used to create the three-column layout by specifying a width and margins for the element and then floating it to the left. Secondly, it adds a subtle shadow underneath each image. This creates a three-dimensional appearance making it look like a piece of card. To ensure that this sits underneath the image, the background-position property is used.

EXAMPLE

IMAGES

```
<!DOCTYPE html>
<html>
  <head>
    <title>Images</title>
    <style type="text/css">
      body {
        color: #665544;
        background-color: #d4d0c6;
        background-image: url("images/backdrop.gif");
        font-family: Georgia, "Times New Roman", serif;
        text-align: center;}
      .wrapper {
        width: 720px;
        margin: 0px auto;}
      .header {
        margin: 40px 0px 20px 0px;}
      .entry {
        width: 220px;
        float: left;
        margin: 10px;
        height: 198px;
        background-image: url("images/shadow.png");
        background-repeat: no-repeat;
        background-position: bottom;}
      figure {
        display: block;
        width: 202px;
        height: 170px;
        background-color: #e7e3d8;
        padding: 9px;
        text-align: left;}
      figure img {
        width: 200px;
        height: 150px;
        border: 1px solid #d6d6d6;}
      figcaption {
        background-image: url("images/icon.png");
        padding-left: 20px;
        background-repeat: no-repeat;}
    </style>
```

EXAMPLE

IMAGES

```
  </head>
  <body>
    <div class="wrapper">
      <div class="header">
        <img src="images/title.gif" alt="Galerie Botanique" width="456" height="122" />
        <p>Here is a selection of antique botanical prints held in our collection.</p>
      </div>
      <div class="entry">
        <figure><img src="images/print-01.jpg" alt="Helianthus" />
          <figcaption>Helianthus</figcaption>
        </figure>
      </div>
      <div class="entry">
        <figure><img src="images/print-02.jpg" alt="Passiflora" />
          <figcaption>Passiflora</figcaption>
        </figure>
      </div>
      <div class="entry">
        <figure><img src="images/print-03.jpg" alt="Nyctocalos" />
          <figcaption>Nyctocalos</figcaption>
        </figure>
      </div>
      <div class="entry">
        <figure><img src="images/print-04.jpg" alt="Polianthes" />
          <figcaption>Polianthes</figcaption>
        </figure>
      </div>
      <div class="entry">
        <figure><img src="images/print-05.jpg" alt="Ficus" />
          <figcaption>Ficus</figcaption>
        </figure>
      </div>
      <div class="entry">
        <figure><img src="images/print-06.jpg" alt="Dendrobium" />
          <figcaption>Dendrobium</figcaption>
        </figure>
      </div>
    </div>
  </body>
</html>
```

SUMMARY
IMAGES

- You can specify the dimensions of images using CSS. This is very helpful when you use the same sized images on several pages of your site.
- Images can be aligned both horizontally and vertically using CSS.
- You can use a background image behind the box created by any element on a page.
- Background images can appear just once or be repeated across the background of the box.
- You can create image rollover effects by moving the background position of an image.
- To reduce the number of images your browser has to load, you can create image sprites.

17

HTML5 LAYOUT

- HTML5 layout elements
- How old browsers understand new elements
- Styling HTML5 layout elements with CSS

HTML5 is introducing a new set of elements that help define the structure of a page.

They are covered here (rather than with the other HTML elements you met earlier in the book) because you'll find it easier to understand how they can be used now that you have seen how CSS can control the layout a page. These new elements are going to play an important part in creating layouts going forward. In this chapter you will see:

- The new HTML5 layout elements and their uses
- How they offer helpful alternatives to the `<div>` element
- How to ensure older browsers recognize these elements

As with all HTML5 and CSS3 content, its usage is still subject to change but it is already widely being used by web developers and it is likely that you will want to use them.

TRADITIONAL HTML LAYOUTS

For a long time, web page authors used `<div>` elements to group together related elements on the page (such as the elements that form a header, an article, footer or sidebar). Authors used `class` or `id` attributes to indicate the role of the `<div>` element in the structure of the page.

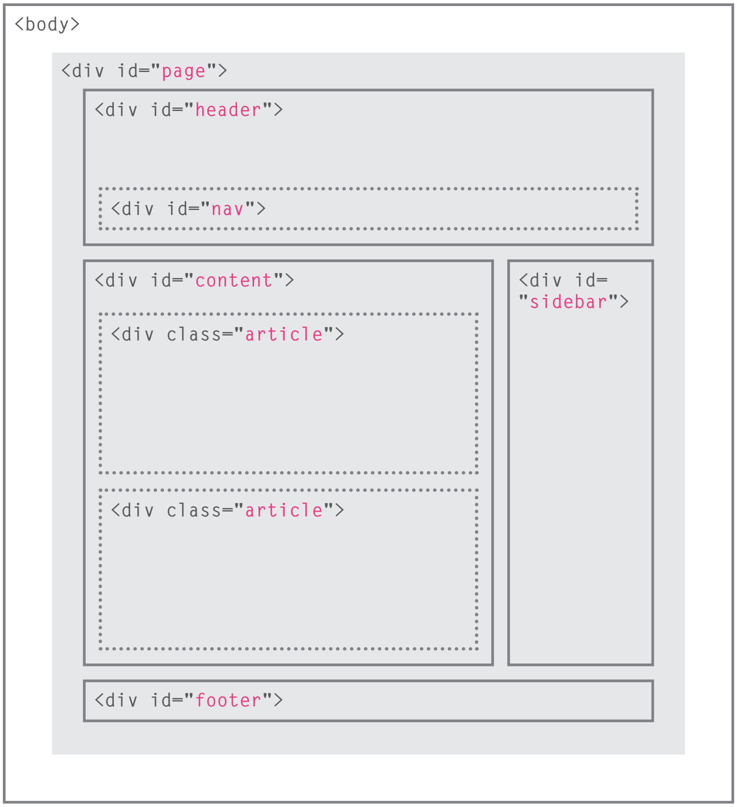
On the right you can see a layout that is quite common (particularly on blog sites).

At the top of the page is the header, containing a logo and the primary navigation.

Under this are one or more articles or posts. Sometimes these are summaries that link to individual posts.

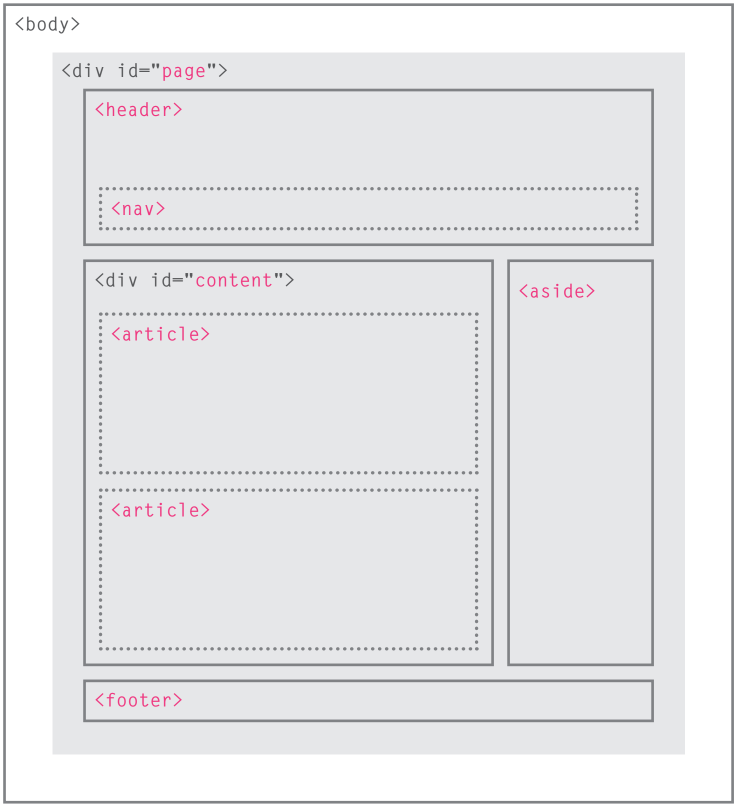
There is a side bar on the right-hand side (perhaps featuring a search option, links to other recent articles, other sections of the site, or even ads).

When coding a site like this, developers would usually put these main sections of the page inside `<div>` elements and use the `class` or `id` attributes to indicate the purpose of that part of the page.



NEW HTML5 LAYOUT ELEMENTS

HTML5 introduces a new set of elements that allow you to divide up the parts of a page. The names of these elements indicate the kind of content you will find in them. They are still subject to change, but that has not stopped many web page authors using them already.



This example has exactly the same structure as seen on the previous page. However, many of the `<div>` elements have been replaced by new HTML5 layout elements.

For example, the header sits inside a new `<header>` element, the navigation in a `<nav>` element, and the articles are in individual `<article>` elements.

The point of creating these new elements is so that web page authors can use them to help describe the structure of the page. For example, screen reader software might allow users to ignore headers and footers and get straight to the content. Similarly, search engines might place more weight on the content in an `<article>` element than that in the `<header>` or `<footer>` elements. I think you will agree that it also makes the code easier to follow.

HEADERS & FOOTERS

<header> <footer>

The <header> and <footer> elements can be used for:

- The main header or footer that appears at the top or bottom of every page on the site.
- A header or footer for an individual <article> or <section> within the page.

In this example, the <header> element used to contain the site name and the main navigation. The <footer> element contains copyright information, along with links to the privacy policy and terms and conditions. Each individual <article> and <section> element can also have its own <header> and <footer> elements to hold the header or footer information for that section within the page.

For example, on a page with several blog posts, each individual post can be thought of as a separate section. The <header> element can therefore be used to contain the title and date of each individual post, and the <footer> might contain links to share the article on social networking sites.

Please note that all of the code shown in this chapter is referenced in one HTML document which is called: html5-layout.html

chapter-17/example.html HTML

```
<header>
  <h1>Yoko's Kitchen</h1>
  <nav>
    <ul>
      <li><a href="" class="current">home</a></li>
      <li><a href="">classes</a></li>
      <li><a href="">catering</a></li>
      <li><a href="">about</a></li>
      <li><a href="">contact</a></li>
    </ul>
  </nav>
</header>
```

chapter-17/example.html HTML

```
<footer>
  &copy; 2011 Yoko's Kitchen
</footer>
```

NAVIGATION

<nav>

HTML chapter-17/example.html

```html
<nav>
  <ul>
    <li><a href="" class="current">home</a></li>
    <li><a href="">classes</a></li>
    <li><a href="">catering</a></li>
    <li><a href="">about</a></li>
    <li><a href="">contact</a></li>
  </ul>
</nav>
```

The `<nav>` element is used to contain the major navigational blocks on the site such as the primary site navigation.

Going back to our blog example, if you wanted to finish an article with links to related blog posts, these would not be counted as major navigational blocks and therefore should not sit inside a `<nav>` element.

At the time of writing, some of the developers that were already using HTML5 decided to use the `<nav>` element for the links that appear at the bottom of every page (links to things like privacy policy, terms and conditions and accessibility information). Whether this will be widely adopted is still yet to be seen.

ARTICLES
<article>

The <article> element acts as a container for any section of a page that could stand alone and potentially be syndicated.

This could be an individual article or blog entry, a comment or forum post, or any other independent piece of content.

If a page contains several articles (or even summaries of several articles), then each individual article would live inside its own <article> element.

The <article> elements can even be nested inside each other. For example, a blog post might live inside one <article> element and each comment on the article could live inside its own child <article> element.

chapter-17/example.html **HTML**

```html
<article>
  <figure>
    <img src="images/bok-choi.jpg" alt="Bok Choi" />
    <figcaption>Bok Choi</figcaption>
  </figure>
  <hgroup>
    <h2>Japanese Vegetarian</h2>
    <h3>Five week course in London</h3>
  </hgroup>
  <p>A five week introduction to traditional
    Japanese vegetarian meals, teaching you a
    selection of rice and noodle dishes.</p>
</article>
<article>
  <figure>
    <img src="images/teriyaki.jpg"
         alt="Teriyaki sauce" />
    <figcaption>Teriyaki Sauce</figcaption>
  </figure>
  <hgroup>
    <h2>Sauces Masterclass</h2>
    <h3>One day workshop</h3>
  </hgroup>
  <p>An intensive one-day course looking at how to
    create the most delicious sauces for use in a
    range of Japanese cookery.</p>
</article>
```

ASIDES
<aside>

HTML chapter-17/example.html

```
<aside>
  <section class="popular-recipes">
    <h2>Popular Recipes</h2>
    <a href="">Yakitori (grilled chicken)</a>
    <a href="">Tsukune (minced chicken patties)</a>
    <a href="">Okonomiyaki (savory pancakes)</a>
    <a href="">Mizutaki (chicken stew)</a>
  </section>
  <section class="contact-details">
    <h2>Contact</h2>
    <p>Yoko's Kitchen<br />
       27 Redchurch Street<br />
       Shoreditch<br />
       London E2 7DP</p>
  </section>
</aside>
```

The <aside> element has two purposes, depending on whether it is inside an <article> element or not.

When the <aside> element is used inside an <article> element, it should contain information that is related to the article but not essential to its overall meaning. For example, a pullquote or glossary might be considered as an aside to the article it relates to.

When the <aside> element is used outside of an <article> element, it acts as a container for content that is related to the entire page. For example, it might contain links to other sections of the site, a list of recent posts, a search box, or recent tweets by the author.

SECTIONS
<section>

The <section> element groups related content together, and typically each section would have its own heading.

For example, on a homepage there may be several <section> elements to contain different sections of the page, such as latest news, top products, and newsletter signup.

Because the <section> element groups related items together, it may contain several distinct <article> elements that have a common theme or purpose.

Alternatively, if you have a page with a long article, the <section> element can be used to split the article up into separate sections.

The <section> element should not be used as a wrapper for the entire page (unless the page only contains one distinct piece of content). If you want a containing element for the entire page, that job is still best left to the <div> element.

chapter-17/example.html **HTML**

```html
<section class="popular-recipes">
  <h2>Popular Recipes</h2>
  <a href="">Yakitori (grilled chicken)</a>
  <a href="">Tsukune (minced chicken patties)</a>
  <a href="">Okonomiyaki (savory pancakes)</a>
  <a href="">Mizutaki (chicken stew)</a>
</section>
<section class="contact-details">
  <h2>Contact</h2>
  <p>Yoko's Kitchen<br />
     27 Redchurch Street<br />
     Shoreditch<br />
     London E2 7DP</p>
</section>
```

HEADING GROUPS
<hgroup>

HTML chapter-17/example.html

```
<hgroup>
  <h2>Japanese Vegetarian</h2>
  <h3>Five week course in London</h3>
</hgroup>
```

The purpose of the <hgroup> element is to group together a set of one or more <h1> through <h6> elements so that they are treated as one single heading.

For example, the <hgroup> element could be used to contain both a title inside an <h2> element and a subtitle within an <h3> element.

This element has had a mixed reception. When it was first proposed by the people developing HTML5, there were some complaints and it was withdrawn from the HTML5 proposals. However, some people changed their minds and it has been added it back into the language. Some developers do not like the use of the <hgroup> element, and prefer to place a subtitle inside a <p> element (using an attribute to indicate that it is a subheading).

Some suggest that it is of little use other than as a styling hook. It has, however, been popular with those developers who believe that it is useful to group together the primary heading and the subheading (as both can be integral parts of a heading).

FIGURES

<figure> <figcaption>

You already met the <figure> element in Chapter 5 when we looked at images. It can be used to contain any content that is referenced from the main flow of an article (not just images).

It is important to note that the article should still make sense if the content of the <figure> element were moved (to another part of the page, or even to a different page altogether).

For this reason, it should only be used when the content simply references the element (and not for something that is absolutely integral to the flow of a page).

Examples of usage include:

- Images
- Videos
- Graphs
- Diagrams
- Code samples
- Text that supports the main body of an article

The <figure> element should also contain a <figcaption> element which provides a text decription for the content of the <figure> element. In this example, you can see a <figure> has been added inside the <article> element.

chapter-17/example.html **HTML**

```
<figure>
  <img src="images/bok-choi.jpg" alt="Bok Choi" />
  <figcaption>Bok Choi</figcaption>
</figure>
```

SECTIONING ELEMENTS
<div>

HTML chapter-17/example.html

```
<div class="wrapper">
  <header>
    <h1>Yoko's Kitchen</h1>
    <nav>
      <!-- nav content here -->
    </nav>
  </header>
  <section class="courses">
    <!-- section content here -->
  </section>
  <aside>
    <!-- aside content here -->
  </aside>
  <footer>
    <!-- footer content here -->
  </footer>
</div><!-- .wrapper -->
```

It may seem strange to follow these new elements by revisiting the `<div>` element again. (After all, the new elements are often going to be used in its place.)

However, the `<div>` element will remain an important way to group together related elements, because you should not be using these new elements that you have just met for purposes other than those explicitly stated.

Where there is no suitable element to group a set of elements, the `<div>` element will still be used. In this example, it is used as a wrapper for the entire page.

Some people have asked why there is no `<content>` element to contain the main part of a page. The reason is that anything that lies outside of the `<header>`, `<footer>` or `<aside>` elements can be considered as the main content.

LINKING AROUND BLOCK-LEVEL ELEMENTS

HTML5 allows web page authors to place an <a> element around a block level element that contains child elements. This allows you to turn an entire block into a link.

This is not a new element in HTML5, but it was not seen as a correct usage of the <a> element in earlier versions of HTML.

chapter-17/example.html HTML

```html
<a href="introduction.html">
  <article>
    <figure>
      <img src="images/bok-choi.jpg"
           alt="Bok Choi" />
      <figcaption>Bok Choi</figcaption>
    </figure>
    <hgroup>
      <h2>Japanese Vegetarian</h2>
      <h3>Five week course in London</h3>
    </hgroup>
    <p>A five week introduction to traditional
     Japanese vegetarian meals, teaching you a
     selection of rice and noodle dishes.</p>
  </article>
</a>
```

HELPING OLDER BROWSERS UNDERSTAND

CSS chapter-17/example.html

```
header, section, footer, aside, nav, article, figure
{
  display: block;}
```

HTML chapter-17/example.html

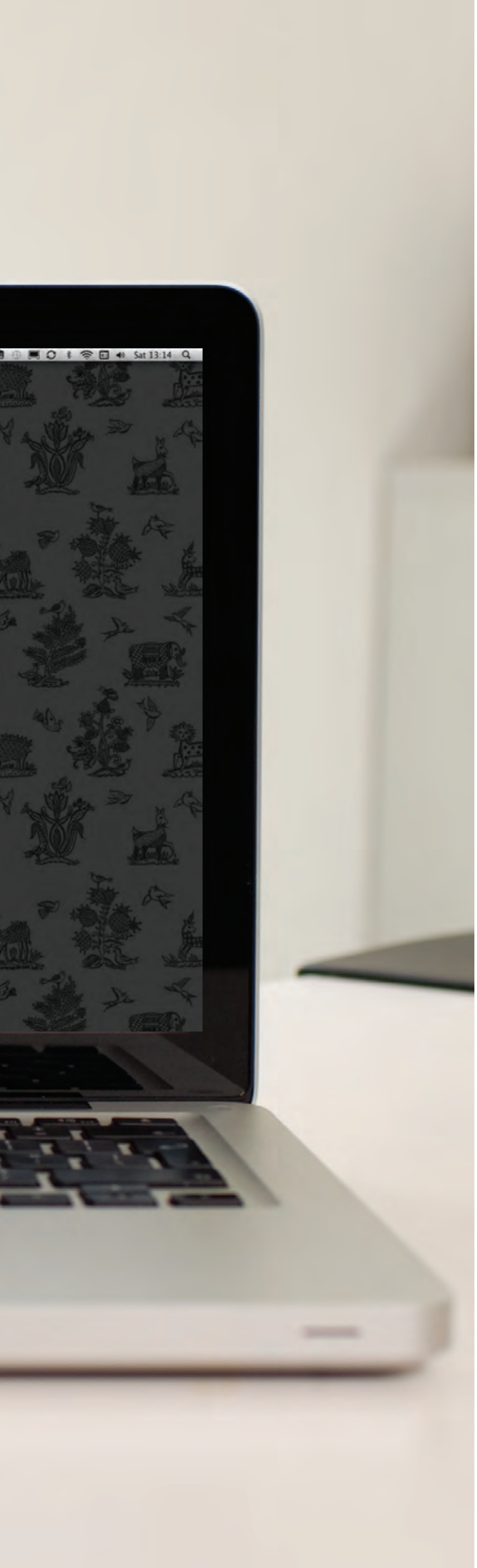
```
<!--[if lt IE 9]>
  <script src="http://html5shiv.googlecode.com/svn/
    trunk/html5.js"></script>
<![endif]-->
```

Older browsers that do not know the new HTML5 elements will automatically treat them as inline elements. Therefore, to help older browsers, you should include the line of CSS on the left which states which new elements should be rendered as block-level elements.

Also, IE9 was the first version of Internet Explorer to allow CSS rules to be associated with these new HTML5 layout elements. In order to style these elements using earlier versions of IE, you need to use a simple JavaScript known as the **HTML5 shiv** or **HTML5 shim**.

You do not need to understand JavaScript to use it. You can just link to a copy that Google hosts on its servers. It should be placed inside a **conditional comment** which checks if the browser version is less than (hence the `lt`) IE9.

Unfortunately, this workaround does require that anyone using IE8 or earlier versions of IE has JavaScript enabled in their browser. If they do not have JavaScript enabled then they will not be able to see the content of these HTML5 elements.



EXAMPLE
HTML5 LAYOUT

This example shows a cooking site built using new HTML5 elements to describe the structure of the page (rather than just grouping items using `<div>` elements).

The header and footer of the page sit inside `<header>` and `<footer>` elements. The courses are grouped together inside a `<section>` element that has a `class` attribute whose value is `courses` (to distinguish it from other `<section>` elements on the page). The sidebar sits inside an `<aside>` element.

Each of the courses lives inside an `<article>` element, and use the `<figure>` and `<figcaption>` elements to contain an image. The headings for the courses have subheadings, so these are grouped inside an `<hgroup>` element. In the sidebar, the recipes and contact details are placed inside separate `<section>` elements.

The page is styled using CSS. The only difference is that our selectors are using the new HTML5 elements to allow us to create rules that target those elements. In order for the CSS to work in versions of IE before Internet Explorer 9, the HTML5 page contains a link to the HTML5 shiv JavaScript (hosted on Google's servers) inside a conditional comment.

EXAMPLE

HTML5 LAYOUT

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML5 Layout</title>
    <style type="text/css">
      header, section, footer, aside, nav, article, figure, figcaption {
        display: block;}
      body {
        color: #666666;
        background-color: #f9f8f6;
        background-image: url("images/dark-wood.jpg");
        background-position: center;
        font-family: Georgia, times, serif;
        line-height: 1.4em;
        margin: 0px;}
      .wrapper {
        width: 940px;
        margin: 20px auto 20px auto;
        border: 2px solid #000000;
        background-color: #ffffff;}
      header {
        height: 160px;
        background-image: url(images/header.jpg);}
      h1 {
        text-indent: -9999px;
        width: 940px;
        height: 130px;
        margin: 0px;}
      nav, footer {
        clear: both;
        color: #ffffff;
        background-color: #aeaca8;
        height: 30px;}
      nav ul {
        margin: 0px;
        padding: 5px 0px 5px 30px;}
      nav li {
        display: inline;
        margin-right: 40px;}
      nav li a {
```

```
  color: #ffffff;}
nav li a:hover, nav li a.current {
  color: #000000;}
section.courses {
  float: left;
  width: 659px;
  border-right: 1px solid #eeeeee;}
article {
  clear: both;
  overflow: auto;
  width: 100%;}
hgroup {
  margin-top:40px;}
figure {
  float: left;
  width: 290px;
  height: 220px;
  padding: 5px;
  margin: 20px;
  border: 1px solid #eeeeee;}
figcaption {
  font-size: 90%;
  text-align: left;}
aside {
  width: 230px;
  float: left;
  padding: 0px 0px 0px 20px;}
aside section a {
  display: block;
  padding: 10px;
  border-bottom: 1px solid #eeeeee;}
aside section a:hover {
  color: #985d6a;
  background-color: #efefef;}
a {
  color: #de6581;
  text-decoration: none;}
h1, h2, h3 {
  font-weight: normal;}
h2 {
```

EXAMPLE

HTML5 LAYOUT

```
          margin: 10px 0px 5px 0px;
          padding: 0px;}
        h3 {
          margin: 0px 0px 10px 0px;
          color: #de6581;}
        aside h2 {
          padding: 30px 0px 10px 0px;
          color: #de6581;}
        footer {
          font-size: 80%;
          padding: 7px 0px 0px 20px;}
      </style>
      <!--[if lt IE 9]>
      <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
      <![endif]-->
    </head>
    <body>
      <div class="wrapper">
        <header>
          <h1>Yoko's Kitchen</h1>
          <nav>
            <ul>
              <li><a href="" class="current">home</a></li>
              <li><a href="">classes</a></li>
              <li><a href="">catering</a></li>
              <li><a href="">about</a></li>
              <li><a href="">contact</a></li>
            </ul>
          </nav>
        </header>
        <section class="courses">
          <article>
            <figure>
              <img src="images/bok-choi.jpg" alt="Bok Choi" />
              <figcaption>Bok Choi</figcaption>
            </figure>
            <hgroup>
              <h2>Japanese Vegetarian</h2>
              <h3>Five week course in London</h3>
            </hgroup>
```

EXAMPLE

HTML5 LAYOUT

```
          <p>A five week introduction to traditional Japanese vegetarian meals,
             teaching you a selection of rice and noodle dishes.</p>
        </article>
        <article>
          <figure>
            <img src="images/teriyaki.jpg" alt="Teriyaki sauce" />
            <figcaption>Teriyaki Sauce</figcaption>
          </figure>
          <hgroup>
            <h2>Sauces Masterclass</h2>
            <h3>One day workshop</h3>
          </hgroup>
          <p>An intensive one-day course looking at how to create the most delicious
             sauces for use in a range of Japanese cookery.</p>
        </article>
      </section>
      <aside>
        <section class="popular-recipes">
          <h2>Popular Recipes</h2>
          <a href="">Yakitori (grilled chicken)</a>
          <a href="">Tsukune (minced chicken patties)</a>
          <a href="">Okonomiyaki (savory pancakes)</a>
          <a href="">Mizutaki (chicken stew)</a>
        </section>
        <section class="contact-details">
          <h2>Contact</h2>
          <p>Yoko's Kitchen<br />
             27 Redchurch Street<br />
             Shoreditch<br />
             London E2 7DP</p>
        </section>
      </aside>
      <footer>
        &copy; 2011 Yoko's Kitchen
      </footer>
    </div><!-- .wrapper -->
  </body>
</html>
```

SUMMARY
HTML5 LAYOUT

- The new HTML5 elements indicate the purpose of different parts of a web page and help to describe its structure.
- The new elements provide clearer code (compared with using multiple `<div>` elements).
- Older browsers that do not understand HTML5 elements need to be told which elements are block-level elements.
- To make HTML5 elements work in Internet Explorer 8 (and older versions of IE), extra JavaScript is needed, which is available free from Google.

18

PROCESS & DESIGN

- How to approach building a site
- Understanding your audience and their needs
- How to present information visitors want to see

This section discusses a process that you can use when you are creating a new website.

It looks at who might be visiting your site and how to ensure the pages feature the information those visitors need. It also covers some key aspects of design theory to help you create professional looking sites. In this chapter, we will look at:

- How to understand the audience your site may attract and what information they will expect to find on it
- How to organize information so that visitors can find what they are looking for
- Design theory for presenting information in a way that helps visitors achieve their goals
- Design tips to help you create more attractive and professional sites

WHO IS THE SITE FOR?

Every website should be designed for the target audience—not just for yourself or the site owner. It is therefore very important to understand who your target audience is.

It can be helpful to ask some questions about the people you would expect to be interested in the subject of your site.

If you ask a client who a site is for, it is not uncommon for them to answer "the entire world."

Realistically, it is unlikely to be relevant to everyone. If your site sells light bulbs, even though most people using a computer probably use light bulbs, they are not likely to order them from someone in a different country.

Even if the site has a wide appeal, you can still think about the demographics of a sample of the target audience.

TARGET AUDIENCE: INDIVIDUALS

- What is the age range of your target audience?
- Will your site appeal to more women or men? What is the mix?
- Which country do your visitors live in?
- Do they live in urban or rural areas?
- What is the average income of visitors?
- What level of education do they have?
- What is their marital or family status?
- What is their occupation?
- How many hours do they work per week?
- How often do they use the web?
- What kind of device do they use to access the web?

TARGET AUDIENCE: COMPANIES

- What is the size of the company or relevant department?
- What is the position of people in the company who visit your site?
- Will visitors be using the site for themselves or for someone else?
- How large is the budget they control?

Invent some fictional visitors from your typical target audience. They will become your friends. They can influence design decisions from color palettes to level of detail in descriptions.

NAME	GORDON	MOLLY	JASPER	AYO	IVY
Gender	M	F	M	M	F
Age	28	47	19	32	35
Location	Chicago	San Francisco	New York	Miami	Boston
Occupation	Teacher	Attorney	Student	Retail	Journalist
Income	$62k	$180k	$24k	$160k	$75k
Web Use	2-3 days/wk	Daily	Daily	4-5 days/wk	Daily

If you have a question about how the site is going to be used, or what its priorities should be, you will be able to think back and ask yourself, "What would Gordon or Molly want in this situation?"

WHY PEOPLE VISIT YOUR WEBSITE

Now that you know who your visitors are, you need to consider **why** they are coming. While some people will simply chance across your website, most will visit for a specific reason.

Your content and design should be influenced by the goals of your users.

To help determine why people are coming to your website, there are two basic categories of questions you can ask:

1: The first attempts to discover the underlying **motivations** for why visitors come to the site.

2: The second examines the specific **goals** of the visitors. These are the triggers making them come to the site *now*.

KEY MOTIVATIONS

- Are they looking for general entertainment or do they need to achieve a specific goal?
- If there is a specific goal, is it a personal or professional one?
- Do they see spending time on this activity as essential or a luxury?

SPECIFIC GOALS

- Do they want general information / research (such as background on a topic / company), or are they after something specific (such as a particular fact or information on a product)?
- Are they already familiar with the service or product that you offer or do they need to be introduced to it?
- Are they looking for time sensitive information, such as the latest news or updates on a particular topic?
- Do they want to discover information about a specific product or service to help them decide whether to buy it or not?
- Do they need to contact you? If so, can they visit in person (which might require opening hours and a map)? Or might they need email or telephone contact details?

WHAT YOUR VISITORS ARE TRYING TO ACHIEVE

It is unlikely that you will be able to list every reason why someone visits your site but you are looking for key tasks and motivations. This information can help guide your site designs.

First you want to create a list of reasons why people would be coming to your site. You can then assign the list of tasks to the fictional visitors you created in the step described on the previous page.

GORDON bought a tennis racquet several years ago; now he wants to purchase one from your site for his girlfriend.

MOLLY has read about your new doggy daycare service in the press and wants to find out whether it would be suitable for her to use.

JASPER had a bad experience staying in a hotel when visiting Sydney, Australia, and wants to make a complaint.

AYO hopes to study architecture and wants to learn more about a new course that is being offered.

IVY is a picture editor and wants to look at a photographer's site to see examples of his work before deciding whether to commission him.

WHAT INFORMATION YOUR VISITORS NEED

You know who is coming to your site and why they are coming, so now you need to work out what information they need in order to achieve their goals quickly and effectively.

You may want to offer additional supporting information that you think they might find helpful.

Look at each of the reasons why people will be visiting your site and determine what they need to achieve their goals.

You can prioritize levels of information from key points down to non-essential or background information.

By ensuring that you provide the information that your visitors are looking for, they will consider your site more relevant to them.

Therefore, you will have more opportunity to tell them any extra information that you think would be helpful to them (or to expose them to other products and services you want to promote).

If you do not appear relevant to them by answering their needs, however, they are likely to go elsewhere.

Here are some questions to help you decide what information to provide for visitors to your site...

KEY INFORMATION

- Will visitors be familiar with your subject area / brand or do you need to introduce yourself?
- Will they be familiar with the product / service / information you are covering or do they need background information on it?
- What are the most important features of what you are offering?
- What is special about what you offer that differentiates you from other sites that offer something similar?
- Once people have achieved the goal that sent them to your site, are there common questions people ask about this subject area?

HOW OFTEN PEOPLE WILL VISIT YOUR SITE

Some sites benefit from being updated more frequently than others. Some information (such as news) may be constantly changing, while other content remains relatively static.

A website about fashion trends will need to change a lot more frequently than one that is promoting a service that people do not buy regularly (such as domestic plumbing or double glazing).

Once a site has been built, it can take a lot of time and resources to update it frequently.

Working out how often people are likely to revisit your site gives you an indication for how often you should update the site.

It can often be helpful to set a schedule for when a site will be updated (rather than doing it on an ad hoc basis).

You will often find that some parts of a site will benefit from being updated more frequently than others.

Here are some questions to help you decide how often to update your website content...

GOODS / SERVICES

- How often do the same people return to purchase from you?
- How often is your stock updated or your service changed?

INFORMATION

- How often is the subject updated?
- What percentage of your visitors would return for regular updates on the subject, compared with those who will just need the information once?

SITE MAPS

Now that you know what needs to appear on your site, you can start to organize the information into sections or pages.

The aim is to create a diagram of the pages that will be used to structure the site. This is known as a **site map** and it will show how those pages can be grouped.

To help you decide what information should go on each page, you can use a technique called **card sorting**.

This involves placing each piece of information that a visitor might need to know on a separate piece of paper and then organizing the related information into groups.

Each group relates to a page and, on larger sites the, pages can in turn can be grouped together to create different sections of the website.

The groups of information are then turned into the diagram that is known as the site map.

Sometimes it can be helpful to ask people who are the target audience to help you group related information together.

A site map will usually begin with the homepage. Additionally, if the site is large and is compartmentalized into sections, each section might require its own section homepage to link to all of the information within it.

For example, most online shops have section homepages for each type of product, which then in turn link to individual product pages.

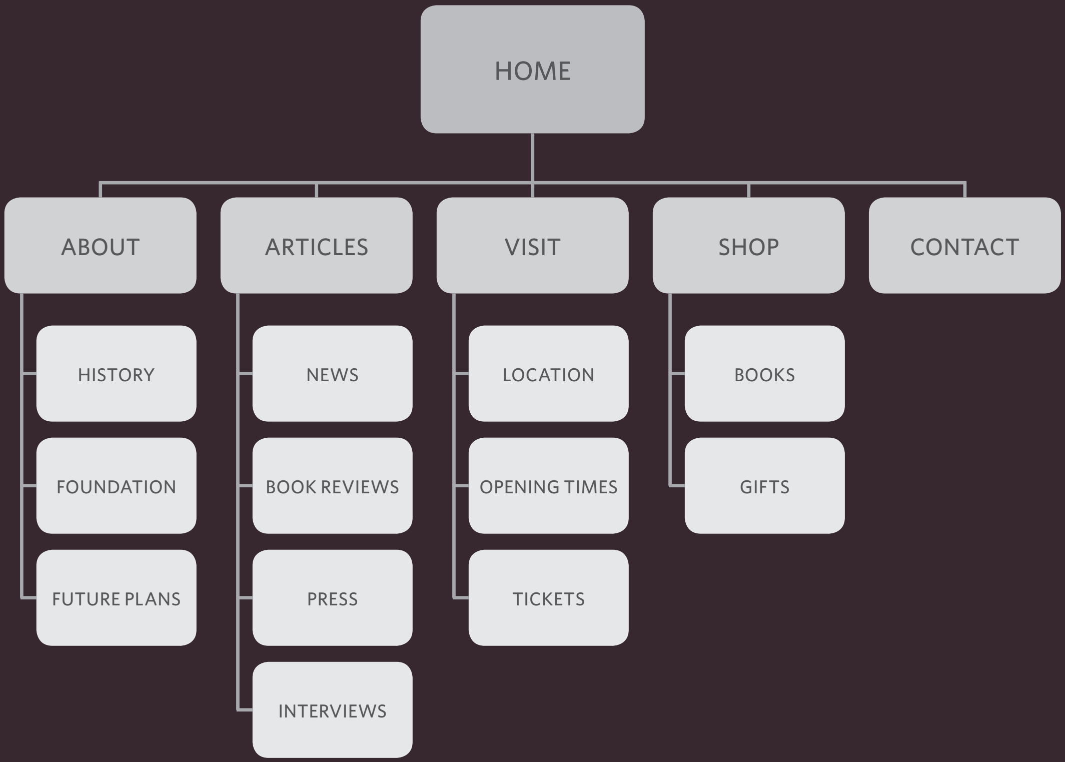
You may need to duplicate some information if it needs to appear on more than one page.

The pages (or groups of pages) will inform how users navigate through the site.

Remember to focus on the goals that your visitors want to achieve.

It is worth noting that the site owner might organize information in a way that is different to what the public expects. It is important to reflect the public's understanding of the subject (rather than just the site owner's understanding of it).

EXAMPLE SITE MAP



WIREFRAMES

A wireframe is a simple sketch of the key information that needs to go on each page of a site. It shows the hierarchy of the information and how much space it might require.

A lot of designers will take the elements that need to appear on each page and start by creating wireframes. This involves sketching or shading areas where each element of the page will go (such as the logo, primary navigation, headings and main bodies of text, user logins etc).

By creating a wireframe you can ensure that all of the information that needs to be on a page is included.

You should not include the color scheme, font choices, backgrounds or images for the website in the wireframe. It should focus on what information needs to be on each page and create a visual hierarchy to indicate the most important parts of each page.

The wireframes make design easier because you know what information needs to appear on which page before considering how the the page should look. It can be very helpful to show the wireframes of a site to a client before showing them a design. It enables the client to ensure the site has all the functions and information it needs to offer.

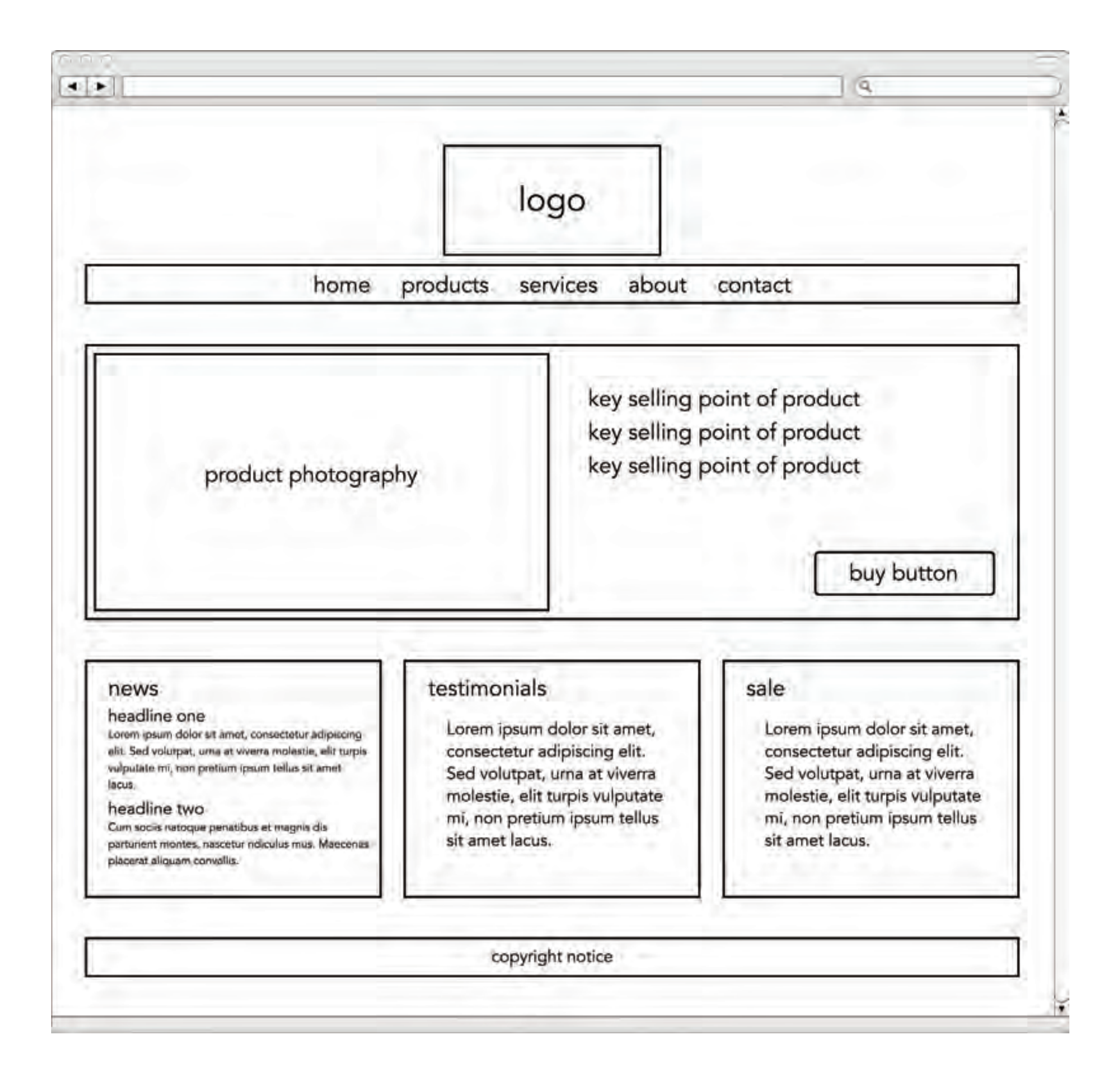
If you just present a site design to a client, it is common for them to focus on how the site looks, which means they may not raise issues about its function after the site has been built.

The example on the right was created in Photoshop using the templates that come with the 960.gs grid system.

You can sketch wireframes on paper or use a graphics application on your computer (such as Illustrator or InDesign).

There are also online wireframe tools such as those at:
`http://gomockingbird.com`
`http://lovelycharts.com`

EXAMPLE WIREFRAME


logo
home
products
services
about
contact
product photography
key selling point of product
key selling point of product
key selling point of product
buy button
news
headline one

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed volutpat, urna at viverra molestie, elit turpis vulputate mi, non pretium ipsum tellus sit amet lacus.

Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Maecenas placerat aliquam convallis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed volutpat, urna at viverra molestie, elit turpis vulputate mi, non pretium ipsum tellus sit amet lacus.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed volutpat, urna at viverra molestie, elit turpis vulputate mi, non pretium ipsum tellus sit amet lacus.

GETTING YOUR MESSAGE ACROSS USING DESIGN

The primary aim of any kind of visual design is to communicate. Organizing and prioritizing information on a page helps users understand its importance and what order to read it in.

CONTENT

Web pages often have a lot of information to communicate. For example, the pages of online newspapers will have information that does not appear on every page of the print equivalent:

- A masthead or logo
- Links to navigate the site
- Links to related content and other popular articles
- Login or membership options
- Ability for users to comment
- Copyright information
- Links to privacy policies, terms and conditions, advertising information, RSS feeds, subscription options

With so much on the page, the designer needs to **organize** and **prioritize** the information to communicate their message and help users find what they're looking for.

PRIORITIZING

If everything on a page appeared in the same style, it would be much harder to understand. (Key messages would not stand out.)

By making parts of the page look **distinct** from surrounding content, designers draw attention to (or away from) those items.

Designers create something known as a **visual hierarchy** to help users focus on the key messages that will draw people's attention, and then guide them to subsequent messages.

We look at visual hierarchy on pages 467-468.

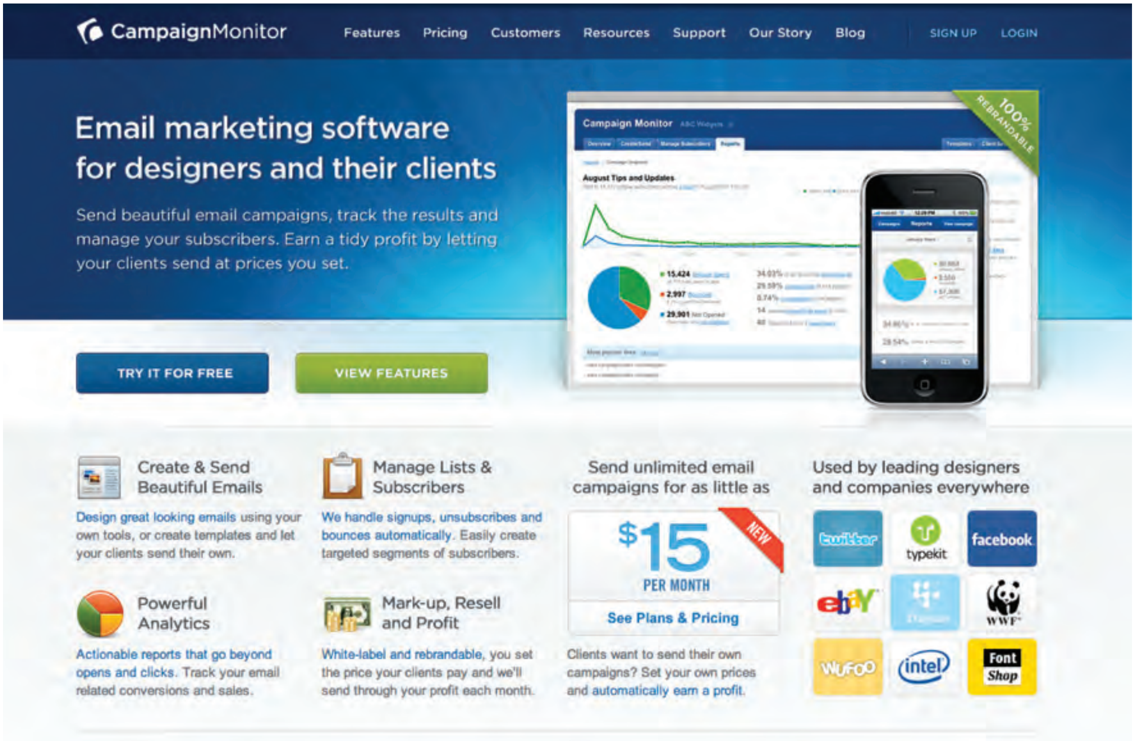
ORGANIZING

Grouping together related content into **blocks** or **chunks** makes the page look simpler (and easier to understand).

Users should be able to identify the purpose of each block without processing each individual item.

By presenting certain types of information in a **similar** visual style (such as using the same style for all buttons or all links), users will learn to associate that style with a particular type of content.

We look at grouping and similarity on pages 469-470.



Email marketing software for designers and their clients

Send beautiful email campaigns, track the results and manage your subscribers. Earn a tidy profit by letting your clients send at prices you set.

Create & Send Beautiful Emails

Design great looking emails using your own tools, or create templates and let your clients send their own.

Powerful Analytics

Actionable reports that go beyond opens and clicks. Track your email related conversions and sales.

Manage Lists & Subscribers

We handle signups, unsubscribes and bounces automatically. Easily create targeted segments of subscribers.

Mark-up, Resell and Profit

White-label and rebrandable, you set the price your clients pay and we'll send through your profit each month.

Send unlimited email campaigns for as little as

$15 PER MONTH

See Plans & Pricing

Clients want to send their own campaigns? Set your own prices and automatically earn a profit.

Used by leading designers and companies everywhere

Let's look at an example of how design can be used to effectively communicate the services of a company.

VISUAL HIERARCHY

Attention is immediately drawn to a picture that shows the service this company offers and a headline to explain it. The size and colored background reinforce that this is the primary message on the page.

Should this service appeal to the user, below they can see more detail about what it does, how much it costs, and who uses it.

GROUPING

There are several chunks of information on this page.

At the top you can see the logo and navigation. Under this is the information that introduces the company's services.

Further down are three distinct groups showing you what the services do, the costs involved and some of the services' users.

SIMILARITY

There are several examples of similarity within this page.

The four points (at the bottom left of the screenshot) are all presented in a similar manner with consistent headings and icons.

All of the links in the body text are in blue so it is clear what text is clickable.

VISUAL HIERARCHY

Most web users do not read entire pages. Rather, they skim to find information. You can use contrast to create a visual hierarchy that gets across your key message and helps users find what they are looking for.

SIZE

Larger elements will grab users' attention first. For this reason it is a good idea to make headings and key points relatively large.

Lorem ipsum
dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipscing elit.

COLOR

Foreground and background color can draw attention to key messages. Brighter sections tend to draw users' attention first.

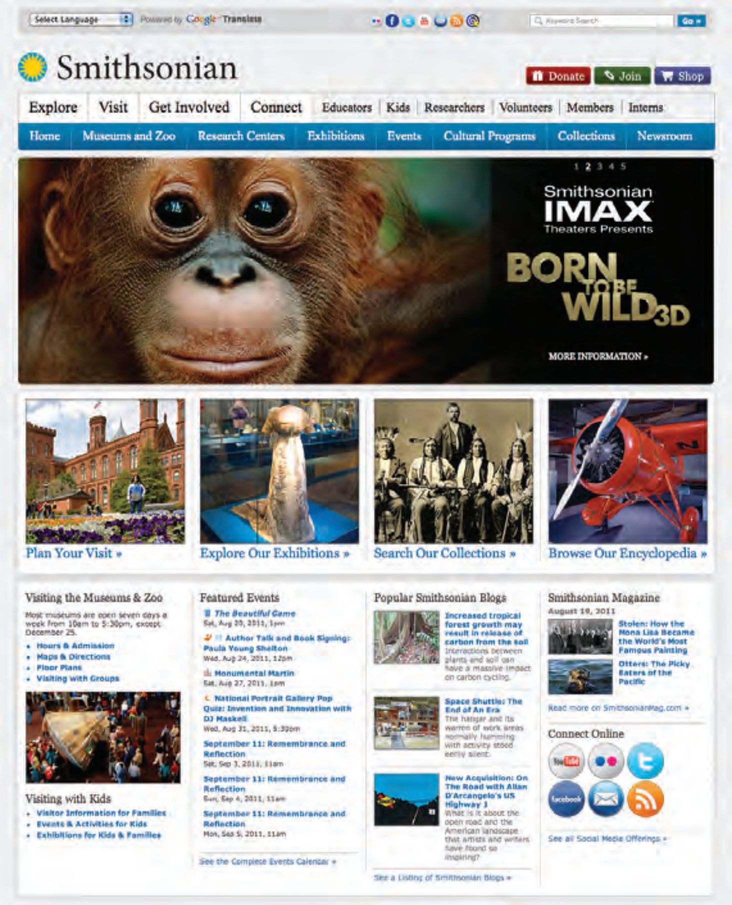
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Lorem ipsum dolor sit amet, consectetur adipscing elit.

STYLE

An element may be the same size and color as surrounding content but have a different style applied to it to make it stand out.

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Lorem ipsum dolor sit amet, consectetur adipscing elit.

Visual hierarchy refers to the order in which your eyes perceive what they see. It is created by adding **visual contrast** between the items being displayed. Items with higher contrast are recognized and processed first.



Visiting the Museums & Zoo

Most museums are open seven days a week from 10am to 5:30pm, except December 25.

- Hours & Admission
- Maps & Directions
- Floor Plans
- Visiting with Groups

Visiting with Kids

- Visitor Information for Families
- Events & Activities for Kids
- Exhibitions for Kids & Families

Featured Events

The Beautiful Game
Sat, Aug 20, 2011, 1pm

Author Talk and Book Signing: Paula Young Shelton
Wed, Aug 24, 2011, 12pm

Monumental Martin
Sat, Aug 27, 2011, 1pm

National Portrait Gallery Pop Quiz: Invention and Innovation with DJ Maskell
Wed, Aug 31, 2011, 5:30pm

September 11: Remembrance and Reflection
Sat, Sep 3, 2011, 11am

September 11: Remembrance and Reflection
Sun, Sep 4, 2011, 11am

September 11: Remembrance and Reflection
Mon, Sep 5, 2011, 11am

See the Complete Events Calendar »

Popular Smithsonian Blogs

Increased tropical forest growth may result in release of carbon from the soil
Interactions between plants and soil can have a massive impact on carbon cycling.

Space Shuttle: The End of An Era
The hangar and its warren of work areas normally humming with activity stood eerily silent.

New Acquisition: On The Road with Allan D'Arcangelo's US Highway 1
What is it about the open road and the American landscape that artists and writers have found so inspiring?

See a Listing of Smithsonian Blogs »

Smithsonian Magazine
August 10, 2011

Stolen: How the Mona Lisa Became the World's Most Famous Painting

Otters: The Picky Eaters of the Pacific

Read more on SmithsonianMag.com »

Connect Online

See all Social Media Offerings »

IMAGES

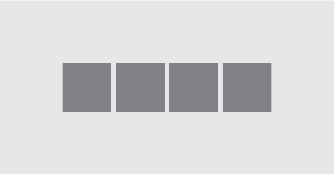
Images create a high visual contrast and often attract the eye first. They can be used to draw attention to a specific message within the page. In some cases, the right image can succinctly reveal more than an entire page of text.

The effect of a well-designed visual hierarchy is largely subliminal. Achieving a good hierarchy requires balance; if nothing stands out a site can be rather uninteresting, and if too many aspects are competing for your attention it can be hard to find the key messages. This example has a clear hierarchy which addresses the needs of visitors to the site.

GROUPING AND SIMILARITY

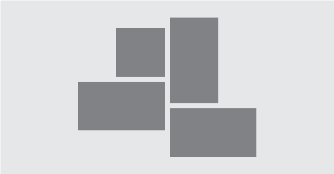
When making sense of a design, we tend to organize visual elements into groups. Grouping related pieces of information together can make a design easier to comprehend. Here are some ways this can be achieved.

PROXIMITY



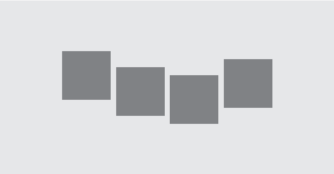
When several items are placed close together, they are perceived as more related than items that are placed further apart. (You can also nest groups of information within larger groups of information.)

CLOSURE



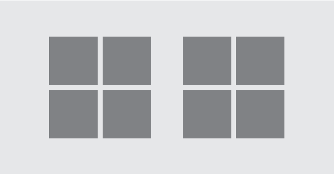
When faced with a complicated arrangement of items, we will often look for a single or recognisable pattern or form. A real or imaginary box can be formed around elements due to their proximity and alignment.

CONTINUANCE



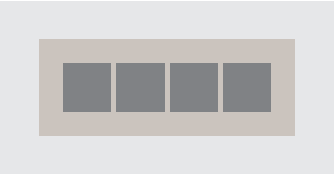
When elements are placed in a line or a curve then they are perceived to be more related than those that are not following the same direction. This can be used to direct a reader from one part of a page to the next.

WHITE SPACE



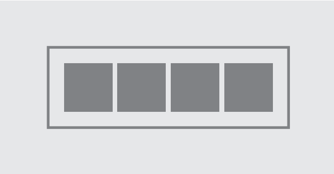
Placing related items closer together and leaving a bigger gap between unrelated items.

COLOR



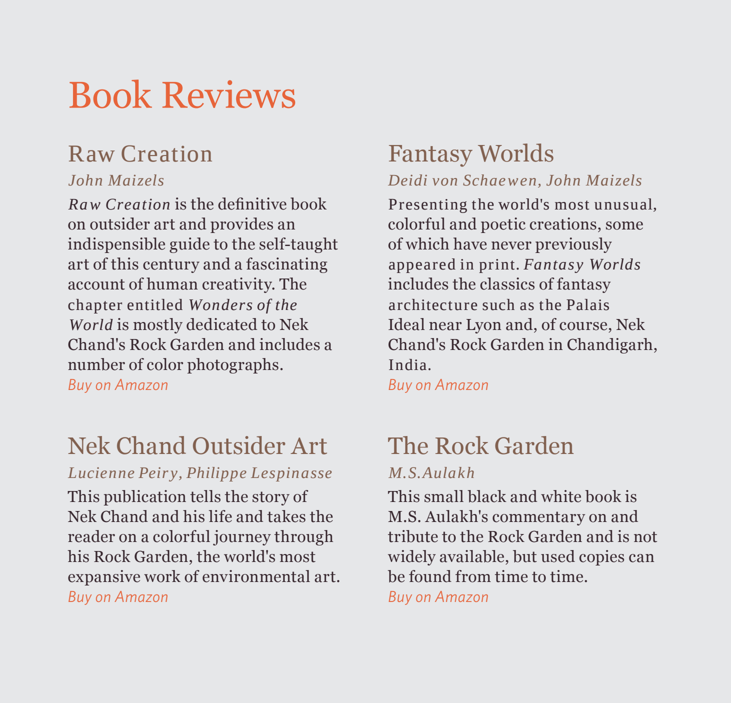
A background color placed behind related items to emphasize their connection.

BORDERS



A line can be drawn around the border of the group or between it and its neighbors.

We naturally observe similarities in design, and things that are similar are perceived to be more related than things that are dissimilar. Repetition of similar color, size, orientation, texture, font, or shape, suggests that matching elements have similar importance or meaning.



Book Reviews

Raw Creation

John Maizels

Raw Creation is the definitive book on outsider art and provides an indispensible guide to the self-taught art of this century and a fascinating account of human creativity. The chapter entitled *Wonders of the World* is mostly dedicated to Nek Chand's Rock Garden and includes a number of color photographs.

Buy on Amazon

Fantasy Worlds

Deidi von Schaewen, John Maizels

Presenting the world's most unusual, colorful and poetic creations, some of which have never previously appeared in print. *Fantasy Worlds* includes the classics of fantasy architecture such as the Palais Ideal near Lyon and, of course, Nek Chand's Rock Garden in Chandigarh, India.

Buy on Amazon

Nek Chand Outsider Art

Lucienne Peiry, Philippe Lespinasse

This publication tells the story of Nek Chand and his life and takes the reader on a colorful journey through his Rock Garden, the world's most expansive work of environmental art.

Buy on Amazon

The Rock Garden

M.S.Aulakh

This small black and white book is M.S. Aulakh's commentary on and tribute to the Rock Garden and is not widely available, but used copies can be found from time to time.

Buy on Amazon

Each visual chunk can contain its own hierarchy as shown in this example, where the individual books have their own subsections of title, author, text and link.

CONSISTENCY

In this example each book review has a consistent style for the book titles, author names, and purchasing links. Having read just one of the blocks it is possible to infer the meaning of the other items in this box that follow the same style.

HEADINGS

Giving a chunk of information a heading clearly tells the user whether or not the content of the grouping is relevant to them. If not, they can ignore all of the elements within it. It also helps users of screen readers, as users often have the option to hear the headings on the page.

DESIGNING NAVIGATION

Site navigation not only helps people find where they want to go, but also helps them understand what your site is about and how it is organized. Good navigation tends to follow these principles...

CONCISE

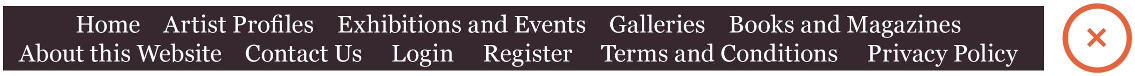
Ideally, the navigation should be quick and easy to read. It is a good idea to try to limit the number of options in a menu to no more than eight links. These can link to section homepages which in turn link to other pages.

CLEAR

Users should be able to predict the kind of information that they will find on the page before clicking on the link. Where possible, choose single descriptive words for each link rather than phrases.

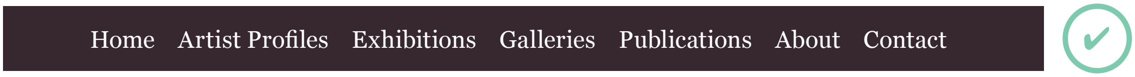
SELECTIVE

The primary navigation should only reflect the sections or content of the site. Functions like logins and search, and legal information like terms and conditions and so on are best placed elsewhere on the page.



Home Artist Profiles Exhibitions and Events Galleries Books and Magazines
About this Website Contact Us Login Register Terms and Conditions Privacy Policy

✕



Home Artist Profiles Exhibitions Galleries Publications About Contact

✓

A large site may have primary, secondary and even tertiary navigation. Primary navigation often appears across the top of the site from left to right, or down the left hand side of the page. Secondary navigation could be under the primary navigation or down the side of the page. Tertiary navigation often sits in the footer of the page. The menu will not be the only way users navigate the site. They will also use links within each page. Some sites also offer a search function.

ONLINE EXTRA

Go to the website accompanying this book for information on how to implement search functionality for your site using Google Search.

CONTEXT

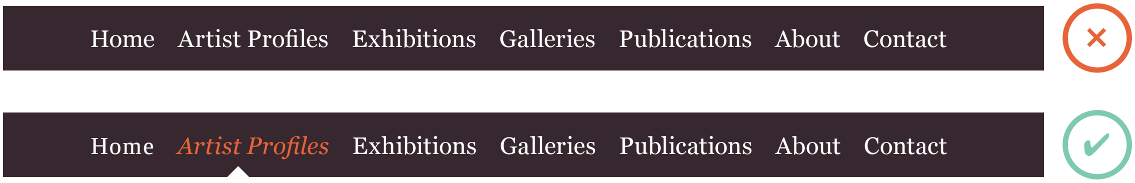
Good navigation provides context. It lets the user know where they are in the website at that moment. Using a different color or some kind of visual marker to indicate the current page is a good way to do this.

INTERACTIVE

Each link should be big enough to click on and the appearance of the link should change when the user hovers over each item or clicks on it. It should also be visually distinct from other content on the page.

CONSISTENT

The more pages a site contains, the larger the number of navigation items there will be. Although secondary navigation will change from page to page, it is best to keep the primary navigation exactly the same.


Home Artist Profiles Exhibitions Galleries Publications About Contact
✕
Home *Artist Profiles* Exhibitions Galleries Publications About Contact
✓

SUMMARY
PROCESS & DESIGN

- It's important to understand who your target audience is, why they would come to your site, what information they want to find and when they are likely to return.
- Site maps allow you to plan the structure of a site.
- Wireframes allow you to organize the information that will need to go on each page.
- Design is about communication. Visual hierarchy helps visitors understand what you are trying to tell them.
- You can differentiate between pieces of information using size, color, and style.
- You can use grouping and similarity to help simplify the information you present.

19

PRACTICAL INFORMATION

- Search engine optimization
- Using analytics to understand visitors
- Putting your site on the web

To wrap up the book we are going to look at some practical information that will help you launch a successful site.

There are entire books written about each of the topics covered in this chapter but I will introduce you to the key themes that each subject deals with and give you pointers for what you need to be considering. You will see:

- The basics of search engine optimization
- Using analytics to understand how people are using your site after it has launched
- Putting your site on the web

SEARCH ENGINE OPTIMIZATION (SEO)

SEO is a huge topic and several books have been written on the subject. The following pages will help you understand the key concepts so you can improve your website's visibility on search engines.

THE BASICS

Search engine optimization (or SEO) is the practice of trying to help your site appear nearer the top of search engine results when people look for the topics that your website covers.

At the heart of SEO is the idea of working out which terms people are likely to enter into a search engine to find your site and then using these terms in the right places on your site to increase the chances that search engines will show a link to your site in their results.

In order to determine who comes first in the search results, search engines do not only look at what appears on your site. They also consider how many sites link to you (and how relevant those links are). For this reason, SEO is often split into two areas: on-page techniques and off-page techniques.

ON-PAGE TECHNIQUES

On-page techniques are the methods you can use on your web pages to improve their rating in search engines.

The main component of this is looking at keywords that people are likely to enter into a search engine if they wanted to find your site, and then including these in the text and HTML code for your site in order to help the search engines know that your site covers these topics.

Search engines rely very heavily on the text that is in your pages so it is important that the terms people are going to search for are in text. There are seven essential places where you want your keywords to appear.

Ensuring that any images have appropriate text in the value of their `alt` attribute also helps search engines understand the content of images.

OFF-PAGE TECHNIQUES

Getting other sites to link to you is just as important as on-page techniques. Search engines help determine how to rank your site by looking at the number of other sites that link to yours.

They are particularly interested in sites whose content is related to yours. For example, if you were running a website that sold fish bait, then a link from a hairdresser is not likely to be considered as relevant as one from an angling community.

Search engines also look at the words between the opening `<a>` tag and closing `</a>` tag in the link. If the text in the link contains keywords (rather than just *click here* or your website address) it may be considered more relevant.

The words that appear in links to your site should also appear in the text of the page that the site links to.

ON-PAGE SEO

In every page of your website there are seven key places where keywords (the words people might search on to find your site) can appear in order to improve its findability.

1: PAGE TITLE

The page title appears at the top of the browser window or on the tab of a browser. It is specified in the `<title>` element which lives inside the `<head>` element.

2: URL / WEB ADDRESS

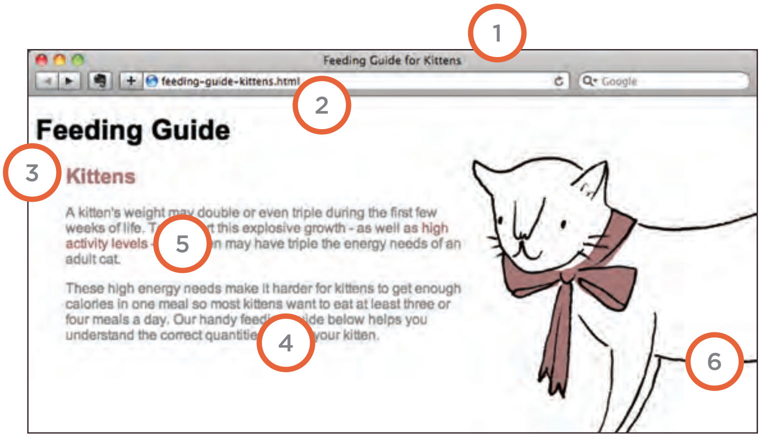
The name of the file is part of the URL. Where possible, use keywords in the file name.

3: HEADINGS

If the keywords are in a heading `<h`*n*`>` element then a search engine will know that this page is all about that subject and give it greater weight than other text.

4: TEXT

Where possible, it helps to repeat the keywords in the main body of the text at least 2-3 times. Do not, however, over-use these terms, because the text must be easy for a human to read.



Feeding Guide for Kittens

feeding-guide-kittens.html

Google

Feeding Guide

Kittens

A kitten's weight may double or even triple during the first few weeks of life. T rt this explosive growth - as well as high activity levels - n may have triple the energy needs of an adult cat.

These high energy needs make it harder for kittens to get enough calories in one meal so most kittens want to eat at least three or four meals a day. Our handy feedi ide below helps you understand the correct quantitie your kitten.

5: LINK TEXT

Use keywords in the text that create links between pages (rather than using generic expressions such as "click here").

6: IMAGE ALT TEXT

Search engines rely on you providing accurate descriptions of images in the alt text. This will also help your images show up in the results of image-based searches.

7: PAGE DESCRIPTIONS

The description also lives inside the `<head>` element and is specified using a `<meta>` tag. It should be a sentence that describes the content of the page. (These are not shown in the browser window but they may be displayed in the results pages of search engines.)

Never try to fool search engines! They will penalize you for it. For example, never add text in the same color as the background of the page as they can detect this.

HOW TO IDENTIFY KEYWORDS AND PHRASES

Determining which keywords to use on your site can be one of the hardest tasks when you start to think about SEO. Here are six steps that will help you identify the right keywords and phrases for your site.

1: BRAINSTORM

List down the words that someone might type into Google to find your site. Be sure to include the various topics, products or services your site is about.

It often helps to ask other people what words they would use to find your site because people less familiar with a topic might use different terms than you. (In particular, they are less likely to use industry-specific jargon.)

Your list may include some keyword phrases (not just individual words) if you have topics which are described by more than one word.

2: ORGANIZE

Group the keywords into separate lists for the different sections or categories of your website.

For example, if your website was a pet shop you might have different categories for different animals (such as dogs, cats and rabbits).

On a large site you may break this up further into sub-categories (for example, separate groups for different pet food brands).

3: RESEARCH

There are several tools that let you enter your keywords and then they will suggest additional keywords you might like to consider, such as:

`adwords.google.co.uk/select/KeywordToolExternal`
(When using this tool, select the "exact match" option rather than "broad match.")
`www.wordtracker.com`
`www.keyworddiscovery.com`

Once these tools have suggested additional keywords, add the relevant options to your lists. (Keyword tools will most likely suggest some terms that are irrelevant so do omit any that do not seem appropriate).

4: COMPARE

It is very unlikely that your site will appear at the top of the search results for every keyword. This is especially true for topics where there is a lot of competition. The more sites out there that have already been optimized for a given keyword, the harder it will be for you to rise up the search results when people search on that term.

Some of the keyword research sites can tell you how many people have searched for a specific keyword to help you know how much competition those terms have.

You can also use Google's advanced search feature to just search the titles of web pages. This will help you to determine how many sites have that keyword in the title of their pages. (The more pages with the term in the title, the more competition there is.)

5: REFINE

Now you need to pick which keywords you will focus on. These should always be the ones that are most relevant to each section of your site.

If there is a phrase that is very relevant but you find there is a lot of competition, you should still use it. To improve the chances of your site being found you can look at whether there are other words that could be incorporated into a phrase. For example, if the information or service you offer on your website is location specific, then you will often find that incorporating your location into your keyword list will help people find you.

If your site is promoting a slate roofing company in Australia then it is better to get 100 people from Australia who are looking for a slate roof than 10,000 from the USA who are looking for other kinds of rooves.

6: MAP

Now that you have a refined list of keywords, you know which have the most competition, and which ones are most relevant, it is time to start picking which keywords you will use for each page.

Pick 3-5 keywords or phrases that map to each page of your website and use these as the keywords for each page.

You should not need to repeat the same keywords on all of the pages. It is also likely that, as you move further away from the homepage into the sections of the site, the keywords will become more specific to the individual topic dealt with on each page.

ANALYTICS: LEARNING ABOUT YOUR VISITORS

As soon as people start coming to your site, you can start analyzing how they found it, what they were looking at and at what point they are leaving. One of the best tools for doing this is a free service offered by Google called Google Analytics.

SIGNING UP

The Google Analytics service relies on you signing up for an account at: `www.google.com/analytics` The site will give you a piece of tracking code which you need to put on every page of your site.

HOW IT WORKS

Every time someone loads a page of your site, the tracking code sends data to the Google servers where it is stored. Google then provides a web-based interface that allows you to see how visitors use your site.

THE TRACKING CODE

A tracking code is provided by Google Analytics for each website you are tracking. It should appear just before the closing `</head>` tag. The code does not alter the appearance of your web pages.

At the time of writing, Google was updating their Analytics interface to Version 5. If you log into your account and see a different interface, look for a link in the top right-hand corner of the page that says 'New Version.'

HOW MANY PEOPLE ARE COMING TO YOUR SITE?

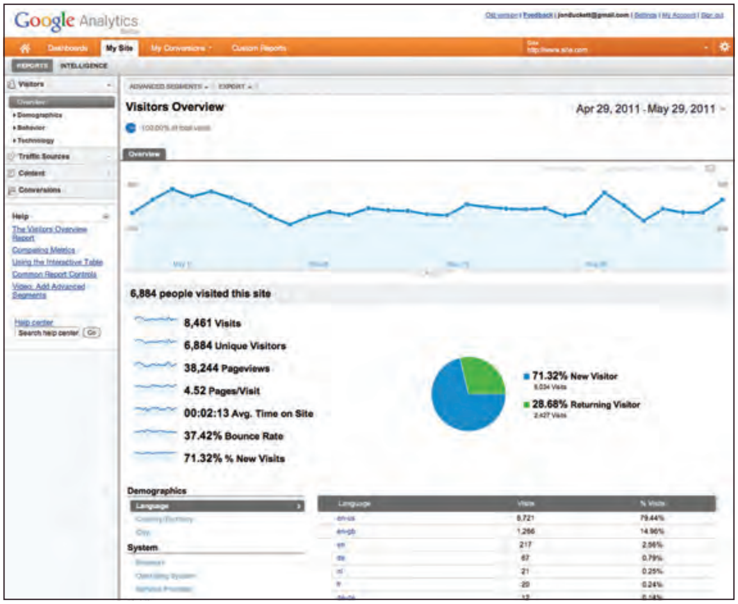
The overview page gives you a snapshot of the key information you are likely to want to know. In particular, it tells you how many people are coming to your site.

VISITS

This is the number of times people have come to your site. If someone is inactive on your site for 30 minutes and then looks at another page on your site, it will be counted as a new visit.

UNIQUE VISITS

This is the total number of people who have visited your site over the specified period. The number of unique visits will be lower than the number of visits if people have been returning to your site more than once in the defined period.



Visitors Overview

Apr 29, 2011 - May 29, 2011

6,884 people visited this site

8,461 Visits

6,884 Unique Visitors

38,244 Pageviews

4.52 Pages/Visit

00:02:13 Avg. Time on Site

37.42% Bounce Rate

71.32% % New Visits

71.32% New Visitor
6,034 Visits

28.68% Returning Visitor
2,427 Visits

Language	Visits	% Visits
en-us	6,721	79.44%
en-gb	1,266	14.96%
en	217	2.56%
de	67	0.79%
nl	21	0.25%
fr	20	0.24%
[illegible]	12	0.14%

PAGE VIEWS

The total number of pages all visitors have viewed on your site.

PAGES PER VISIT

The average number of pages each visitor has looked at on your site per visit.

AVERAGE TIME ON SITE

The average amount of time each user has spent on the site per visit.

DATE SELECTOR

Using the date selector in the top right hand corner of the site, you can change the period of time the reports display. When you log in, this is usually set to the last month, but you can change it to report on a specific time period.

EXPORT

The export link just above the title that says "visitors overview" allows you to export the statistics on this page for other applications such as Excel.

WHAT ARE YOUR VISITORS LOOKING AT?

The content link on the left-hand side allows you to learn more about what the visitors are looking at when they come to your site.

PAGES

This tells you which pages your visitors are looking at the most and also which pages they are spending the most time on.

LANDING PAGES

These are the pages that people arrive on when first visiting your site. This can be particularly helpful because you may find people are not always coming into your site via the homepage.

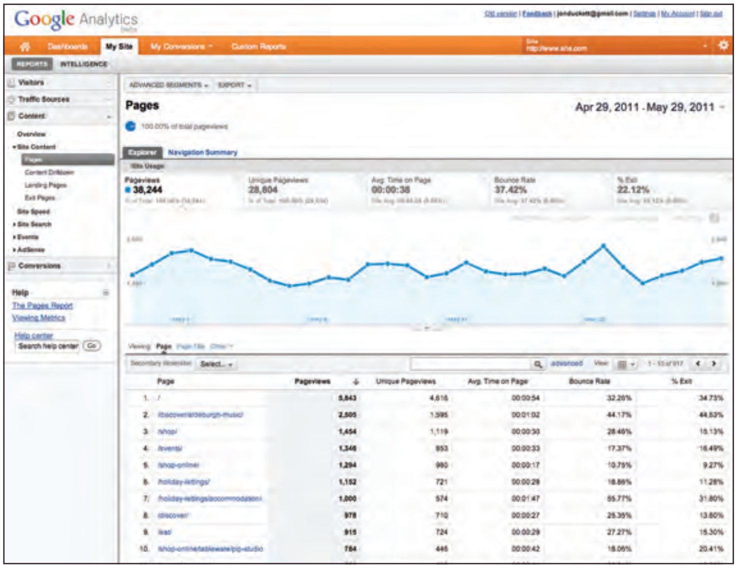
TOP EXIT PAGES

This shows which pages people most commonly leave from. If a lot of people are leaving from the same page then you might consider changing that page or improving it.

BOUNCE RATE

This shows the number of people who left on the same page that they arrived on. A high bounce rate suggests that the content is not what they were looking for or that the page did not sufficiently encourage them to look around the rest of the site. What counts as a bounce:

- Clicked a link to another site
- Clicked on an advertisement
- Entered a new URL
- Used the "back" button
- Closed the browser



Pages

Apr 29, 2011 - May 29, 2011

100.00% of total pageviews

Pageviews	Unique Pageviews	Avg. Time on Page	Bounce Rate	% Exit
38,244	28,804	00:00:38	37.42%	22.12%

	Page	Pageviews	Unique Pageviews	Avg. Time on Page	Bounce Rate	% Exit
1.	/	5,843	4,616	00:00:54	32.26%	34.73%
2.	/discover/aldeburgh-music/	2,505	1,595	00:01:02	44.17%	44.83%
3.	/shop/	1,454	1,119	00:00:30	28.46%	15.13%
4.	/events/	1,346	853	00:00:33	17.37%	16.49%
5.	/shop-online/	1,294	980	00:00:17	10.75%	9.27%
6.	/holiday-lettings/	1,152	721	00:00:28	18.86%	11.28%
7.	/holiday-lettings/accommodation/	1,000	574	00:01:47	55.77%	31.80%
8.	/discover/	978	710	00:00:27	25.35%	13.80%
9.	/eat/	915	724	00:00:29	27.27%	15.30%
10.	/shop-online/tableware/pip-studio	784	446	00:00:42	18.06%	20.41%

WHERE ARE YOUR VISITORS COMING FROM?

The traffic sources link on the left hand side allows you to learn where your visitors are coming from.

REFERRERS

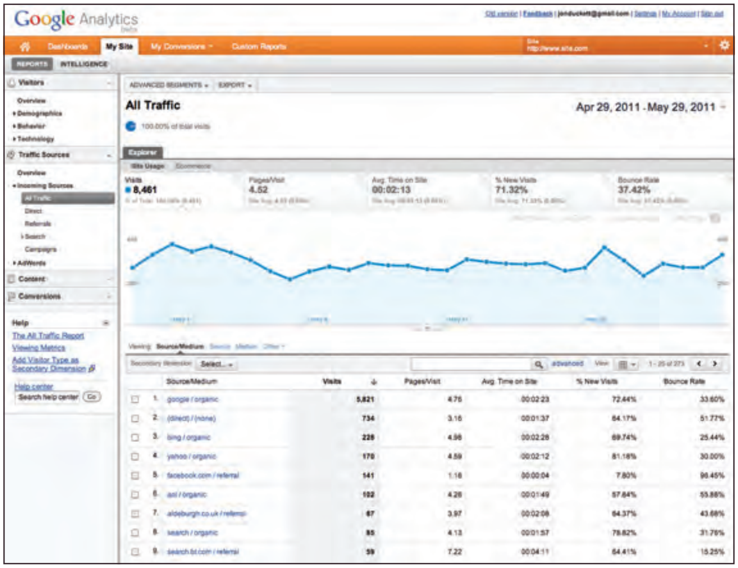
This shows the sites that have linked to you and the number of people who have come via those sites. If a site sends a lot of traffic to you, get in touch and try to work together to ensure that traffic keeps flowing. You could also try to find similar sites and ask them to link to you.

DIRECT

This shows which page a user arrived on if they did not come via a link on another site. They might have typed the URL into their browser, used a browser bookmark, or clicked a link in an email, PDF, or Word document.

SEARCH TERMS

This shows the terms users entered into a search engine to find your site. This can help you learn how visitors describe what they're looking for (which is often different to how someone might describe their own site). This can help you fine-tune your content and your SEO keywords.



All Traffic

Apr 29, 2011 - May 29, 2011

100.00% of total visits

Visits	Pages/Visit	Avg. Time on Site	% New Visits	Bounce Rate
8,461	4.52	00:02:13	71.32%	37.42%

		Source/Medium	Visits	Pages/Visit	Avg. Time on Site	% New Visits	Bounce Rate
□	1.	google / organic	5,821	4.76	00:02:23	72.44%	33.60%
□	2.	(direct) / (none)	734	3.16	00:01:37	64.17%	51.77%
□	3.	bing / organic	228	4.96	00:02:28	69.74%	25.44%
□	4.	yahoo / organic	170	4.59	00:02:12	81.18%	30.00%
□	5.	facebook.com / referral	141	1.16	00:00:04	7.80%	96.45%
□	6.	aol / organic	102	4.26	00:01:49	57.84%	55.88%
□	7.	aldeburgh.co.uk / referral	87	3.97	00:02:08	64.37%	43.68%
□	8.	search / organic	85	4.13	00:01:57	78.82%	31.76%
□	9.	search.bt.com / referral	59	7.22	00:04:11	64.41%	15.25%

ADVANCED FEATURES

We have only scratched the surface of what you can find out about your visitors from Google Analytics. Their help files tell you many more of the advanced features. If you run an online shop, it is well worth looking at their e-commerce tracking, which adds information about products sold, average basket size and much more. You can also set up goals where you specify the paths you want people to take, and then see how far they get through those paths, which is especially useful when gathering data from users.

DOMAIN NAMES & HOSTING

In order to put your site on the web you will need a domain name and web hosting.

DOMAIN NAMES

Your domain name is your web address (e.g. google.com or bbc.co.uk). There are many websites that allow you to register domain names. Usually you will have to pay an annual fee to keep that domain name.

These sites usually have a form that allows you to check whether your preferred domain name is available, and because millions of domain names have already been registered, it might take you a while to find the one that is right for your site.

A lot of sites that offer domain name registration also offer web hosting.

WEB HOSTING

So that other people can see your site, you will need to upload it to a web server. Web servers are special computers that are constantly connected to the Internet. They are specially set up to serve web pages when they are requested.

With the exception of some very large sites, most websites live on web servers run by web hosting companies. This is usually far cheaper and more reliable than trying to run your own web servers.

There are lots of different types of hosting on offer. We will now take a look at some of the key things that will help you choose which hosting company to use.

DISK SPACE

This refers to the total size of all of the files that make up your site (all of the HTML and CSS files, images and scripts).

BANDWIDTH

This is the amount of data the hosting company will send to your site's visitors. If you imagine 10 people looked at every page on your site, then it would be the equivalent to 10 times the amount of disk space you use.

BACKUPS

Check whether the hosting company performs backups on your site (and how often). Some only create backups so that they can restore your website in the event of a server breaking. Others allow access to backups (which can be helpful if you accidentally break the site when updating it).

EMAIL ACCOUNTS

Most hosting companies will provide email servers with their web hosting packages. You will want to check the size of mailbox you are allowed and the number of mailboxes you can use.

SERVER-SIDE LANGUAGES AND DATABASES

If you are using a content management system, it will likely use a server-side programming language and a database (such as PHP with a MySQL database, or ASP.Net with a SQL Server database). Be sure to check that your hosting company supports the technologies your software needs to run.

It is often worth searching for reviews of a hosting company to see what other people's experience has been with a hosting company. Unfortunately, you often can only tell how good a hosting company is when something goes wrong, at which point you find out how they are able to help you (so you can expect to see a few negative reviews for any company).

HOSTED SERVICES

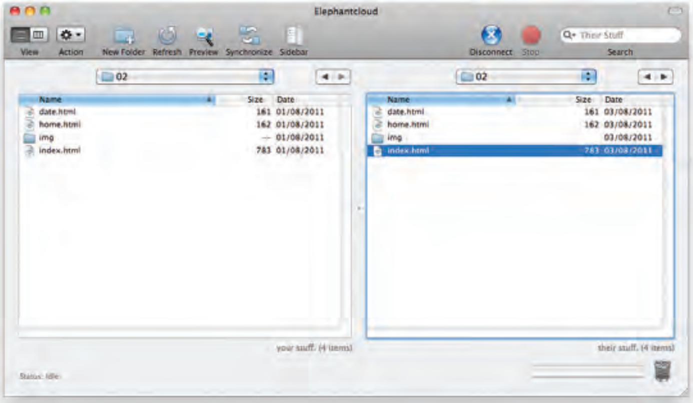
There are a number of online services that allow you to point your domain name to their servers. Blogging platforms such as WordPress.com, Tumblr, and Posterous, or e-commerce platforms such as Big Cartel and Shopify provide the servers that your site is hosted on. If you are using a platform like this you will not need your own hosting for the website, although you often still need hosting for your email. If this is the case, some web hosting companies offer packages that will just offer email services.

FTP & THIRD PARTY TOOLS

To transfer your code and images from your computer to your hosting company, you use something known as File Transfer Protocol.

As the name suggests, File Transfer Protocol (or FTP) allows you to transfer files across the Internet from your computer to the web server hosting your site.

There are many FTP programs that offer a simple interface that shows you the files on your computer alongside the files that are on your web server. These allow you to drag and drop files from your computer to the server or vice versa.



Name	Size	Date
date.html	161	01/08/2011
home.html	162	01/08/2011
img	--	01/08/2011
index.html	783	01/08/2011

your stuff. (4 items)

Name	Size	Date
date.html	161	03/08/2011
home.html	162	03/08/2011
img		03/08/2011
index.html	783	03/08/2011

their stuff. (4 items)

Status: Idle

There are a wide variety of sites that offer services commonly created by web developers (to save you having to build them yourself).

Some hosting companies offer tools to upload files to their servers using a web browser, but it is more common to use an FTP program as they are faster at transmitting files.

When you purchase your web hosting, you will be given FTP details that you enter into your FTP program in order to connect to the server. Usually this will be an address (such as `ftp://mydomain.com`), a username, and a password. It is important to keep this information secure in order to prevent strangers from gaining access to your server.

Here is a list of some popular FTP applications:

FileZilla
`filezilla-project.org`
Windows, Mac, Linux

FireFTP
`fireftp.mozdev.org`
Windows, Mac, Linux

CuteFTP
`cuteftp.com`
Windows, Mac

SmartFTP
`smartftp.com`
WIndows

Transmit
`panic.com/transmit`
Mac

Here is a list of some popular third party tools:

BLOGS
`wordpress.com`
`tumblr.com`
`posterous.com`

E-COMMERCE
`shopify.com`
`bigcartel.com`
`go.magento.com`

EMAIL NEWSLETTERS
`campaignmonitor.com`
`mailchimp.com`

SOCIAL NETWORKING SHARING BUTTONS
`addthis.com`
`addtoany.com`

SUMMARY
PRACTICAL INFORMATION

- Search engine optimization helps visitors find your sites when using search engines.
- Analytics tools such as Google Analytics allow you to see how many people visit your site, how they find it, and what they do when they get there.
- To put your site on the web, you will need to obtain a domain name and web hosting.
- FTP programs allow you to transfer files from your local computer to your web server.
- Many companies provide platforms for blogging, email newsletters, e-commerce and other popular website tools (to save you writing them from scratch).