

Dreamweaver CC

2017

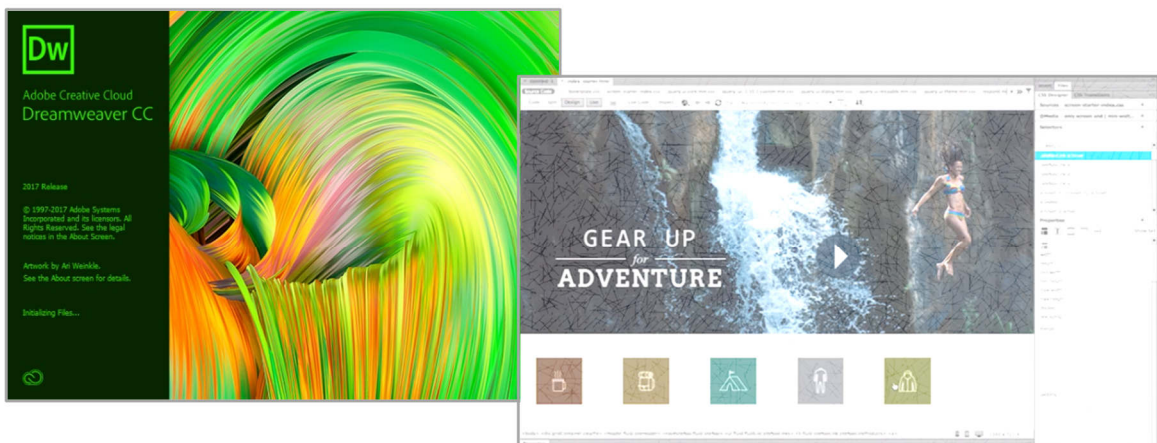
HTML & CSS

These advanced resources get stuck in to the HTML code and Cascading Style Sheets (CSS) that help designers build professional websites with a consistent feel across the pages.

If you'd prefer to avoid the coding, then perhaps have a look at our *Dreamweaver Basics* tutorial.

This version is for Dreamweaver CC 2017. There are also versions available for the earlier Dreamweaver CC 2015 and CS6.

- A. Designing the Layout
- B. Setting Up the Website
- C. Introduction to HTML
- D. Inline Styles
- E. Internal Style Sheets
- F. External Style Sheets
- G. Planning Divs
- H. Creating the Container Div
- I. Creating the Other Divs
- J. Page and Website Properties
- K. Adding a List of Links
- L. Inserting Images
- M. HTML for Hyperlinks
- N. Finishing Your Homepage
- O. Tables HTML
- P. Tables CSS
- Q. Background Images
- R. Summary of HTML and CSS





HTML stands for *HyperText Markup Language*. This is the language that tells your computer how to construct a webpage so that it can be viewed in your browser. The page itself is delivered as a simple text file. Your web browser has to interpret the meaning of this text so that it can display the page as it was designed to be seen. Pictures, movies and sounds are not part of the HTML, but separate files called upon by the webpage. The HTML simply shows how the browser should display these objects.

HTML makes use of tags. Most HTML tags tell the web browser when to start formatting in a certain style and then when to stop using that style. For example, to produce the strong, emphasised text '***This is HTML***' on your page, you could use the following code:

HTML Code	Tag name	Instruction to browser
<code></code>	Strong	- start using strong, or important, type (it appears bold)
<code></code>	Emphasised	- start using emphasised type (it appears as italics)
This is HTML		- write the text
<code></code>	Slash emphasised	- stop using emphasised type
<code></code>	Slash strong	- stop using strong type

The tags are the things in brackets. `` is the tag that instructs the web browser to use strong text. `` is the tag that tells it to stop using emphasised text. The code for this text may all be written on one line:

```
<strong><em>This is HTML</em></strong>
```

Note: There are actually tags called bold (``) and italic (`<i>`) which were once commonly used in HTML. These have a similar appearance to strong and emphasised. The difference is that with the strong and emphasised tags, the browser knows that the actual words between the tags are important. These days, this can make a difference in many situations, for example when the text is being read by the screen readers used by the visually impaired or on small mobile screens. Bold and italic effects can still be used, but these should be put into a style sheet (as shown later).

Some points about HTML

1. Dreamweaver and other web design applications will allow you to build much of your site in WYSIWYG view (what you see is what you get – or *Design view*). However, to build professional, efficient websites based on a specific design, you need to get stuck into the HTML.
2. HTML can be written in *UPPER* or *lower*-case text (or a mixture). Generally, lower case is preferred.
3. Web browsers will ignore tags that they do not understand.
4. HTML is not a programming language. Mistakes will not cause your computer to crash – they will just cause the page to be displayed incorrectly in a browser.
5. Users can set their own preferences regarding text sizes, link colours and whether or not to display pictures. They also have different browsers, platforms and screen sizes. The task of a web designer is usually to design pages that are acceptable to as many people as possible. For example, you should not spend time trying to write text that exactly fits across your screen – it is highly unlikely to appear as you planned on different devices.
6. Tags should not be interlaced. i.e. for the ***strong, emphasised type*** above, we used the code:

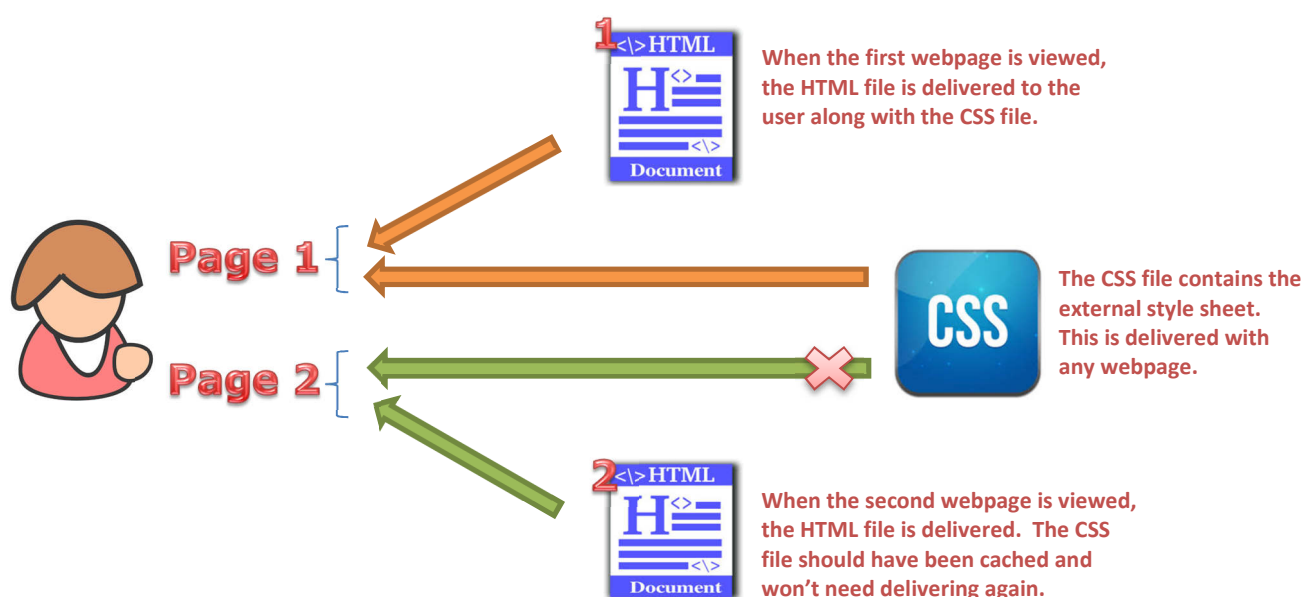
```
<strong>
<em>
This is HTML
</em>
</strong>
```

rather than...

```
<strong>
<em>
This is HTML
</strong>
</em>
```

Internal style sheets are great, but what if we want to use all those styles again in another page? We'd have to create the class selectors again and have all the same code in the head section. This is a waste of time and results in duplicate code being passed to the user each time a page loads.

The perfect solution will involve putting all your styling information into an external style sheet. This sheet is in a CSS file, named something like 'style.css'. The CSS file can then be delivered to the user with each webpage that is viewed. The beauty of this solution is that you only have to edit the CSS file in order to change the appearance of every webpage. Some large websites may have thousands of pages, each of which can be redesigned simply by editing the CSS file.



Caching

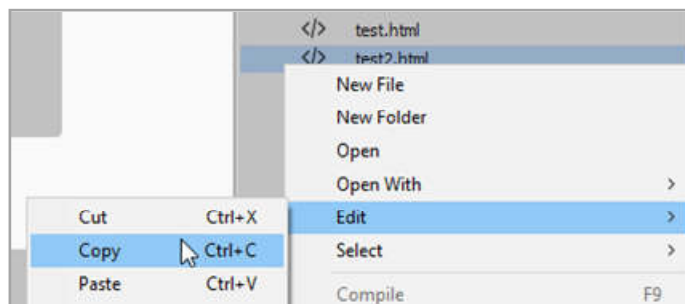
Another benefit of this method is that the CSS file doesn't actually have to be delivered through the internet each time a page loads. The user's computer will store a copy in a local folder called a cache. When another page loads using the same CSS file, the user's computer will check to see if the file on the website is still the same as the cached copy. If it is the same, the computer will just use the local copy. If it is different, the new version will be downloaded.

Task 1 – Creating a CSS File

- a. Start by making a copy of your 2nd test page, partly as a record of your work and partly as a backup in case things go wrong. You may copy and paste the file in the folder outside Dreamweaver, or use the *Files* panel on the right.

If using the *Files* panel, right click on the file name and use 'Edit / Copy' and 'Edit / Paste' to duplicate the file.

Name the file 'test3.html'.



We will now create our divs. You may have learned a method of creating divs in our basic tutorial that used the menus and windows to avoid complex CSS processes. However, if you have understood everything so far, then creating divs using CSS is easy. What's more, with the div styles being placed in an external CSS file, we can use them again quickly in our other pages.

Task 1 – Creating your Container Div

We are going to create the 6 divs shown on the right in order to control the layout of the pages. For now, we'll colour each one so that we can see what it looks like. We can remove the colour later.

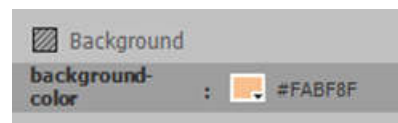
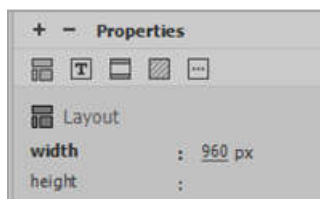
The first div is the container.

- Open your *index.html* page (the homepage) and delete any text that you entered previously.
- We will place all the div styles in a CSS file so that we can use them on each page. Using the CSS Designer, create a new source called 'style.css'. Place this in your css folder.
- Select the 'style.css' file in the Sources section of the CSS Designer, then click the *Add Selector* button. Name this selector '#container'.

Why the #?

This time, our selector starts with a hash (#) rather than a dot. This is because our container is an ID rather than a class. An ID is an element that will appear only once on a page. A class is a style that can be used repeatedly. There is a reason for the difference, but for now, just remember that an ID is unique on a page and starts with a hash; a class can be used multiple times and starts with a dot.

- Select the new *container* selector and set the properties as shown below. The width should be 960px, the left and right margins set to 'auto' (resulting in the div being aligned centre) and a temporary background colour chosen from further down the list. Check that it is the background colour that you are setting and not the text or border colour.

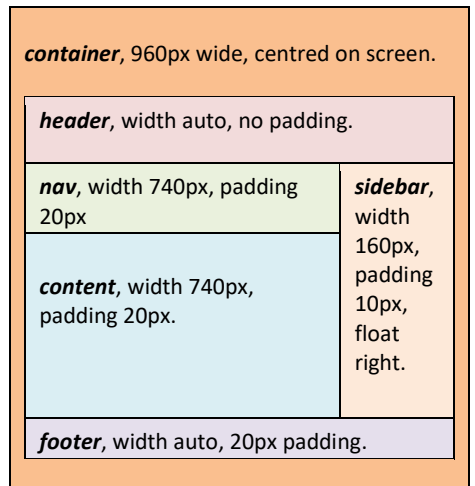


Have a quick look at the CSS file. It should now look like the one on the right (with the properties in any order and probably a different colour in hexadecimal).

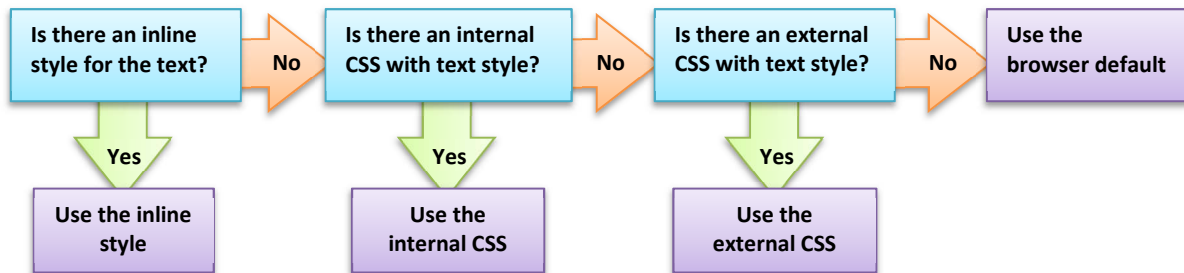
```

1 @charset "utf-8";
2 #container {
3     width: 960px;
4     margin-left: auto;
5     margin-right: auto;
6     background-color: #FABF8F;
7 }
    
```

- With your selector now ready, place your cursor back in the empty Design window and click 'Insert / Div'. Select 'container' from the ID box and click OK. The div should appear in the top middle of the screen. Remember it will get taller as we place other content inside.



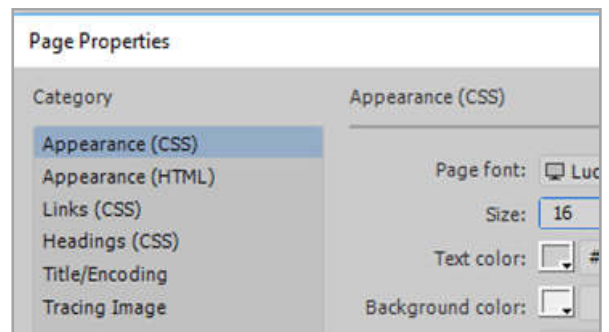
In our basic tutorial, we looked at how to use the *Page Properties* window to set out formatting and layout for the whole page. These settings are actually stored in an internal style sheet. We'll now look at how to set these properties for the whole website using our external CSS file. The diagram below shows how the styling decision is made by a browser when displaying some text on a web page.



Note: The actual process takes place in reverse. The browser will at first adopt the most general rule and then replace it with any more specific rules it finds (e.g. an inline style will replace an internal style sheet). In this way, styles will cascade down to the most specific available, replacing the general rules on their way. This is why they are called Cascading Style Sheets.

Task 1 - Setting Page Properties

- Still in your homepage, click on 'File / Page Properties'.
- In the *Appearance (CSS)* tab, set any text properties you would like to apply to the whole page. Remember that you can override these settings on individual bits of text later, so go with what you would like to use most of the time.
- Click *Apply* to see the effect of your changes and *OK* when you have made your decision.



Task 2 - Cutting and Pasting the Internal CSS

Have a look at your code. You should notice that setting the Page Properties has created an internal style sheet at the top of your webpage code (in the head section).

Any declaration block with the selector 'body' will apply the styles to the **body** section. Any declaration block using the selector 'body,td,th' will apply your font selections to anything in the **body** section, or in **table data / table header** cells.

Cut these declaration blocks out and paste them into your CSS file (at the top, under the '@charset "utf-8";' line). They will now be applied to every page that uses your external style sheet. You can delete the leftover '<style type="text/css">' and '</style>' tags from your webpage.

```

<style type="text/css">
body,td,th {
    font-family: "Lucida Grande",
    font-size: 16px;
    color: #CECECE;
}
body {
    background-color: #F5CACB;
}
</style>
  
```

Note: You could paste them further down your style sheet, but it's normal to put the major declaration blocks first.

We will now add some images to your homepage. We'll have a go at creating an image of a specific size using Photoshop (or another image editor). Images can actually be cropped in Dreamweaver but the results are often unpredictable. Incorporating Photoshop is a good skill to learn.

It is very common to find pictures on the internet to use on websites. Please remember that every image is owned by someone and you can't just take another person's property without their permission (especially if you are going to publish your website on the internet). Thankfully, there are a large number of files on the web that you have been given permission to use. Images from Wikimedia commons, Flickr and many other sources can often be legally used. An easy way to find these images is to use Google.

Note: *The images do not become part of the webpage file; they remain a separate file which is 'collected' by the page as it is viewed. You will need to save a copy of the images to your computer before adding them to your pages.*

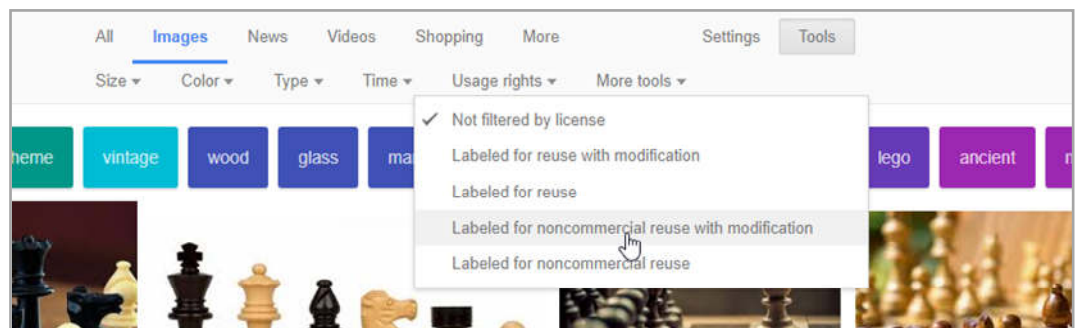
Task 1 - Finding Legal Images

- Perform a Google images search as usual. We are looking for an image, or part of an image, that will make a good banner across the top of the page. The part of the image you will use should be short and wide.

A possible target section of image.



- Click on 'Tools' in the top menu and then on 'Usage rights' in the submenu that appears. Select 'Labeled for non-commercial reuse with modification' so that you are free to change the appearance of the images you find.



- Mouse-over some of the results and find one that is at least 960px wide, or considerably more if you are not going to use the full width (Google gives you the dimensions). Remember that we are looking for a short, wide section of image.

Why should you find images of the right size?

Images that are smaller than the space you want to fill will need to be stretched. This lowers the quality of the display and might result in the image appearing blurred or fuzzy.

Images that are larger than the space you want to fill will need to be shrunk in size. This means that you are transferring more data than necessary over the web. Although internet speeds are increasing, you should still try and keep file sizes to a minimum.

Basically, if you know what size space you have, try and use an image of the same size.

So far, you have used the menu options to create hyperlinks which navigate between your pages. We will now look at the HTML behind some of the other types of hyperlink. These include:

- **Internal hyperlinks** - link to a page on the same website
- **External hyperlinks** - link to a page on a different website
- **Image hyperlinks** - an image made into either of the hyperlinks above
- **File hyperlinks** - link to a file which is downloaded

Have a look at the HTML code for your list of links. The first link should look something like the code below:

```

<div id="nav">
  <ul>
    <li><a href="index.html" title="Homepage">Homepage</a></li>
  </ul>
</div>

```

Open the anchor tag and specify that this will be a hyperlink 'a' = anchor, or link. 'href' = hypertext reference.

The destination. This one is internal. External links usually start with 'http...'.

The title. Displayed as a tool tip when the mouse is hovering over the link.

The actual text that becomes the link.

Close the anchor tag.

Internal (or Relative) Hyperlinks

Internal hyperlinks point to another page in the same website. All we need as a destination is the path to another file (this is why they are also known as *relative* hyperlinks). This example link points to a file called 'members.html'.

`` - links to a page in the same folder

If you have a lot of pages, you may organise these into folders. For example, you may have created a page for each of the members in a club. These pages may be placed in a folder called 'personal'. In this case, the link from a webpage in your root folder to one of your member pages will look like this:

`` - links to the page 'paul.html' in a folder named 'personal'

If you want to link from your page 'paul.html' back to your 'members.html' page, you will need to use the code '../'. This code effectively means 'move up one folder'.

`` - links from the page "paul.html" back to the members page.

Note: In reality, a website designer wouldn't create lots of static member pages. They would store the information in a database and generate each page dynamically. This is a lot more complicated and won't be covered here.

You should by now have enough skills to improve the quality of your homepage. You may do any of the following:

- Change or remove the coloured backgrounds from your divs. Perhaps change the container div to white (#FFF) so that your text is on a white background where it can be read easily. You could use the colour dropper to pick colours from your images.
- Improve the colours used for your navigation buttons, headings and text so that they fit with your header image colours.
- Add a margin to the left and right of your navigation buttons to separate them a little more.
- Add further images and text to the content and sidebar divs.


Note: We have used the heading style 'h3' in our sidebar, with a negative value used for the bottom margin. This has the effect of pulling up the paragraph below. This can be a useful trick.

- Add links to other websites, perhaps in the footer. We have used the declaration 'text-align: center;' to align this list of links in the centre of the div.
- Make any other changes that improve your page.

```
#container {
    width: 960px;
    margin: auto;
    background-color: #FFF;
}
#header {
}
#nav {
    width: 740px;
    padding: 20px;
}
```

```
h3 {
    font-size: 15px;
    margin: 30px 0px -15px 0px;
}
```

```
#footer {
    padding: 20px;
    background-color: #E6E0EC;
    clear: both;
    text-align: center;
}
```



Homepage

Data

Choice


Test

Chess

A Quick Overview

Chess is a two-player strategy board game played on a chessboard, a checkered gameboard with 64 squares arranged in an eight-by-eight grid. It is one of the world's most popular games, played by millions of people worldwide in homes, urban parks, clubs, online, correspondence, and in tournaments.

[Top Players](#)



Years active
c. 6th-century India to present

Genre
Board game
Abstract strategy game

Players
2

Skills required
Strategy, tactics

Rules
Notation
History
Chess Computers



HTML

Header Tags	
<html> </html>	- use HTML code
<head> </head>	- begin and end header section
<body> </body>	- begin and end body section
<title> </title>	- begin and end title
<meta ...>	- include meta-tags

Text Tags	
 	- strong text
 	- emphasised text
<h1> </h1>	- use largest heading
<h6> </h6>	- use smallest heading
<i><u>	should all be set in CSS

Layout	
 	- insert single line break
<p> </p>	- create a paragraph
<blockquote> </b..>	- create a block quote
<div> </div>	- create a div

Lists	
 	- create a numbered list
 	- create a bulleted list
 	- create a list item
<dl>, <dt> & <dd>	also used for definition lists

Hyperlinks	
	- create a relative hyperlink
	- create an absolute hyperlink
	- link to target on page
	- email hyperlink (now dated)

Images	
	- include an image
	- set image width
	- set image height

Table Layout	
<table> </table>	- create a table
<tr> </tr>	- start and end a table row
<td> </td>	- start and end a table data cell
<td colspan="2">	- set columns to span
<td rowspan="2">	- set rows to span
<table width... height...>	- fix size of table
 	- space placed in empty cells
<td width... height...>	set in CSS

Selectors	
<div style="...>	- apply inline styles to a div
<div class="...>	- apply a class from a CSS to a div
<div id="...>	- name a div and use the styles set
Or eg. <p style="...", <td class="...> etc	

There are lots more HTML tags, but try and place your styles in the CSS wherever possible.

CSS

Size and Position of Elements	
width	- fixed width in pixels or as a percentage
height	- fixed height in pixels or as a percentage
max-height	- max height if not fixed (or min/width)
position	- fixed on the page or after the last element
left	- a set distance from the left of its container...
right, top, bottom	... or from the right, top or bottom
float	- eg. float right and let other elements flow
margin-left	- the left margin, or right, top, bottom
margin	- margin on all sides, top, right, bottom, left
padding-left	- the left padding, or right, top, bottom
padding	- padding on all sides, top, right, bottom, left
clear	- stop elements left, right or both overlapping
visibility	- set to 'hidden' for invisible elements
z-index	- stack order of overlapping elements
overflow	- allow contents to overflow
display	- display option eg. for lists.

Text	
text-align	- alignment of text, left, right or center
text-indent	- indentation of the first line
line-height	- line height in pixels or percentage of font size
text-transform	- uppercase, lowercase or capitilize (first letter)
font-family	- font family specific / generic eg. "arial", serif;
font-size	- size of the text in pixels
font-style	- normal, italic or oblique
font-weight	- normal, bold or e.g. 700 (where 400 is normal)
font	- set font properties in one declaration N.B. order.
text-decoration	- underline, overline, line-through
text-shadow	- position h, position v, colour e.g. 2px 2px #aaa

Backgrounds	
background-color	- background colour of an element
background-image	- location of background image
background-position	- position of a background image eg. left top
background-repeat	- how a background image repeats
background-size	- background image size in pixels or percentage
background	- set all the background properties N.B. order
box-shadow	- attach a drop-shadow to the box

Borders	
border-width	- width of all borders e.g. 3px
border-style	- style of all borders e.g. solid, dotted
border-color	- set the colour of all four borders
border	- set all widths, styles and colours
border-top-width	- width e.g. 3px (or left, right, bottom)
border-top-style	- style e.g. solid, dotted (or left, right, bottom)
border-top-color	- colour of the top border (or left, right, bottom)
border-top	- set width, style and colour in one declaration
border-image	- use an image to create a border

Tables	
border-collapse	- collapse, to create a single border
border-spacing	- space between cells eg. 10px

Look online for more CSS properties