CSS Beginner's Guide

htmldog.com

Like the HTML Beginner's Guide, the **CSS Beginner's Guide** assumes that you know as much about CSS as you do about the cumulative effects of sea squirt discharge on the brain chemistry of Germanic ammonites. The purpose of this guide is to teach the **bare essentials** - just enough to get started. The CSS Intermediate Guide and CSS Advanced Guide go into more depth about CSS.

CSS, or '**Cascading Styles Sheets**' are a way to style HTML. Whereas the HTML is the **content**, the style sheet is the **presentation** of that document.

Styles don't smell or taste anything like HTML, they have a format of '**property: value**' and most properties can be applied to most HTML tags.

Applying CSS

There are three ways of applying CSS to HTML.

In-line

In-line styles are plonked straight into the HTML tags using the `style` attribute.

They look something like this:

```
<p style="color: red">text</p>
```

This will make that specific paragraph red.

The approach of HTML Dog is that the HTML should be a stand-alone, **presentation free** document, and so in-line styles should be avoided wherever possible.

Internal

Internal styles are used for the whole page. Inside the `head` tags, the `style` tag surrounds all of the styles for the page.

This would look something like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>CSS Example</title>
<style type="text/css">p {color: red;}a {color: blue;}</style>
...
```

This will make all of the paragraphs in the page red and all of the links blue.

Similarly to the in-line styles, you should keep the HTML and the CSS files separate, and so we are left with our saviour...

External

External styles are used for the whole, multiple-page website. There is a **separate CSS file**, which will simply look something like:

```
p {color: red;}a {color: blue;}
```

If this file is saved as 'web.css' then it can be linked to in the HTML like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>CSS Example</title>
<link rel="stylesheet" type="text/css" href="web.css" />
...
```

In the CSS Advanced Guide, we will see that there are other ways of linking external style sheets, but this will suffice for now.

To get the most from this guide, it would be a good idea to try out the code as we go along, so start a fresh new file with your text-editor and save the blank document as 'web.css' in the same directory as your HTML file.

Now change your HTML file so that it starts something like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>My first web page</title>
<link rel="stylesheet" type="text/css" href="web.css" />
</head>
...
```

Save the HTML file. This now links to the CSS file, which is empty at the moment, so won't change a thing. As you work your way through the CSS Beginner's Guide, you will be able to add to and change the CSS file and see the results by simply refreshing the browser window that has the HTML file in it, as we did before.

Selectors, Properties and Values

Whereas HTML has **tags**, CSS has '**selectors**'. Selectors are the names given to styles in internal and external style sheets. In this CSS Beginner's Guide we will be concentrating on **HTML selectors**, which are simply the names of HTML tags and are used to change the style of a specific tag.

For each selector there are '**properties**' inside **curly brackets**, which simply take the form of words such as `color`, `font-weight` or `background-color`.

A **value** is given to the property following a **colon** (NOT an 'equals' sign) and **semi-colons** separate the properties.

```
body {font-size: 0.8em;color: navy;}
```

This will apply the given values to the `font-size` and `color` properties to the `body` selector.

So basically, when this is applied to an HTML document, text between the `body` tags (which is the content of the whole window) will be 0.8 em's in size and navy in colour.

Lengths and Percentages

There are many property-specific units for values used in CSS, but there are some general units that are used in a number of properties and it is worth familiarising yourself with these before continuing.

'**em**' (such as `font-size: 2em`) is an element approximately equal to the **height of a character**.

'**px**' (such as `font-size: 12px`) is the unit for **pixels**.

'**pt**' (such as `font-size: 12pt`) is the unit for **points**.

'**%**' (such as `font-size: 80%`) is the unit for... wait for it... **percentages**.

Other units include '**pc**' (picas), '**cm**' (centimetres), '**mm**' (millimetres) and '**in**' (inches).

When a value is **zero**, you do not need to state a unit. For example, if you wanted to specify no border, it would be **`border: 0`**.

> ### *Note*
>
> A web page is not a static, absolute medium. It is meant to be flexible and the user should be allowed to view the web page how the hell they like, which includes the font size AND the size of the screen.
>
> Because of this, it is generally accepted that 'em' or '%' are the best units to use for font-sizes (and possibly even heights and widths, which we shall come across in the CSS Advanced Guide), rather than 'px', which leads to non-resizable text in most browsers, and should be used sparingly, for border sizes for example.

Colours

CSS brings **16,777,216** colours to your disposal. They can take the form of a **name**, an '**rgb**' (red/green/blue) value or a '**hex**' code.

`red`
Is the same as
`rgb(255,0,0)`
Which is the same as
`rgb(100%,0%,0%)`
Which is the same as
`#ff0000`
Which is the same as
`#f00`

There are 16 valid predefined colour names. These are **`aqua`**, **`black`**, **`blue`**, **`fuchsia`**, **`gray`**, **`green`**, **`lime`**, **`maroon`**, **`navy`**, **`olive`**, **`purple`**, **`red`**, **`silver`**, **`teal`**, **`white`**, and **`yellow`**.

`transparent` is also a valid value.

The three values in the rbg value are from 0 to 255, 0 being the lowest level (for example no red), 255 being the highest level (for example full red). These values can also be a percentage.

Hexadecimal (previously and more accurately known as '**sexadecimal**') is a **base-16** number system. We are generally used to the **decimal** number system (**base-10**, from 0 to 9), but hexadecimal has 16 digits, from 0 to f.

The hex number is defined by a hash character (**#**) and can be three or six digits in length. Basically, the three-digit version is a compressed version of the six-digit (`#f00` becomes `#ff0000`, `#c96` becomes `#cc9966` etc.). The three-digit version is easier to decipher (the first digit, like the first value in rgb, is red, the second green and the third blue) but the six-digit version gives you more control over the exact colour.

'color' and 'background-color'

Colours can be applied by using **color** and **background-color** (note that this must be the American English 'color' and not 'colour').

A blue background and yellow text could look like this:
h1 {**color: yellow;background-color: blue;**}

These colours might be a little too harsh, so you could change the code of your CSS file for slightly different shades:
body {font-size: 0.8em;color: navy;}h1 {**color: #ffc;background-color: #009;**}

Save the CSS file and refresh your browser. You will see the colours of the first heading (the `h1` element) have changed to yellow and blue.

You can apply the `color` and `background-color` properties to most HTML elements, including `body`, which will change the colours of the page and everything in it.

Text

You can alter the size and shape of the text on a web page with a range of properties, outlined below:

'font-family'

This is the font itself, such as 'Times New Roman', 'Arial' or 'Verdana'.

The font you specify must be on the user's computer, so there is little point in using obscure fonts. There are a select few '**safe**' fonts (the most commonly used are arial, verdana and times new roman), but you can specify more than one font, separated by **commas**. The purpose of this is that if the user does not have the first font you specified, the browser will go through the list until it finds one it does have. This is useful because different computers sometimes have different fonts installed. So `font-family: arial, helvetica`, for example, is used so that similar fonts are used on PC (which usually has arial, but not helvetica) and Apple Mac (which does not usually have arial, and so helvetica, which it does normally have, will be used).

Note: if the name of a font is more than one word, it should be put in quotation marks, such as `font-family: "Times New Roman"`.

'font-size'

The size of the font. Be careful with this - text such as headings should not just be a paragraph in a large font; you should still use headings (`h1`, `h2` etc.) even though, in practice, you could make the font-size of a paragraph larger than that of a heading (not recommended for sensible people).

'font-weight'

This states whether the text is **bold** or not. In practice this usually only works as `font-weight: bold` or `font-weight: normal`. In theory it can also be `bolder`, `lighter`, `100`, `200`, `300`, `400`, `500`, `600`, `700`, `800` or `900`, but seeing as many browsers shake their heads and say 'I don't think so', it's safer to stick with `bold` and `normal`.

'font-style'

This states whether the text is **italic** or not. It can be `font-style: italic` or `font-style: normal`.

'text-decoration'

This states whether the text is underlined or not. This can be;
_ `text-decoration: overline`, which places a line above the text.
_ `text-decoration: line-through`, strike-through, which puts a line through the text.
_ `text-decoration: underline` **SHOULD ONLY BE USED FOR LINKS** because users generally expect underlined text to be links.
This property is usually used to decorate links, such as specifying no underline with `text-decoration: none`.

'text-transform'

This will change the case of the text.

`text-transform: capitalize` turns the first letter of every word into uppercase.
`text-transform: uppercase` turns everything into uppercase.
`text-transform: lowercase` turns everything into lowercase.
`text-transform: none` I'll leave for you to work out.

```
body {
    font-family: arial, helvetica, sans-serif;font-size: 0.8em;
}
h1 {
    font-size: 2em;
}
h2 {
    font-size: 1.5em;
}
a {
    text-decoration: none;
}
strong {
    font-style: italic;
    text-transform: uppercase;
}
```

Text spacing

The **letter-spacing** and **word-spacing** properties are for spacing between letters or words. The value can be a length or **normal**.

The **line-height** property sets the height of the lines in an element, such as a paragraph, without adjusting the size of the font. It can be a length, a percentage or **normal**.

The **text-align** property will align the text inside an element to **left**, **right**, **center** or **justify**.

The **text-indent** property will **indent** the first line of a paragraph to a given length or percentage. This is a format usually used in print, and rarely in digital media such as the web.

```
p {
    letter-spacing: 0.5em;
    word-spacing: 2em;
    line-height: 1.5em;
    text-align: center;
}
```

Margins and Padding

margin and **padding** are the two most commonly used properties for spacing-out elements. A margin is the space **outside** of the element, whereas padding is the space **inside** the element.

Change the code for **h2** to the following:

```
h2 {
    font-size: 1.5em;
    background-color: #ccc;
    margin: 1em;padding: 3em;
}
```

You will see that this leaves one-character width space around the secondary header and the header itself is fat from the three-character width padding.

The four sides of an element can also be set individually. **margin-top**, **margin-right**, **margin-bottom**, **margin-left**, **padding-top**, **padding-right**, **padding-bottom** and **padding-left** are the self-explanatory properties you can use.

The Box Model

Margins, padding and borders (see next page) are all part of what's known as the **Box Model**. The Box Model works like this: in the middle you have the element box (let's say an image), surrounding that you have the padding box, surrounding that you have the border box and surrounding that you have the margin box. It can be visually represented like this:
Margin box
Border box
Padding box
Element box

You don't have to use all three surrounding 'boxes', but it can be helpful to remember that the box model can be applied to every element on the page.

Borders

Borders can be applied to most HTML elements within the body.

To make a border around an element, all you need is **border-style**. The values can be **solid**, **dotted**, **dashed**, **double**, **groove**, **ridge**, **inset** and **outset**.

border-width sets the **width** of the border, which is usually in pixels. There are also properties for **border-top-width**, **border-right-width**, **border-bottom-width** and **border-left-width**.

Finally, **border-color** sets the colour.

Add the following code to the CSS file:

```
h2 {
    border-style: dashed;
    border-width: 3px;
    border-left-width: 10px;
    border-right-width: 10px;
    border-color: red;
}
```

This will make a red dashed border around all HTML secondary headers (the **h2** element) that is 3 pixels wide on the top and bottom and 10 pixels wide on the left and right (these having over-ridden the 3 pixel wide width of the entire border).

Putting It All Together

You should already have an HTML file like the one made at the end of the HTML Beginner's Guide, with a line that we added at the start of this CSS Beginner's Guide, linking the HTML file to the CSS file.

The code below covers all of the CSS methods in this section. If you save this as your CSS file and look at the HTML file then you should now understand what each CSS property does and how to apply them. The best way to fully understand all of this is to mess around with the HTML and the CSS files and see what happens when you change things.

```
body {
      font-family: arial, helvetica, sans-serif;
      font-size: 80%;
      color: black;
      background-color: #ffc;
      margin: 1em;padding: 0;
}
p {
      line-height: 1.5em;
}
h1 {
      color: #ffc;
      background-color: #900;
      font-size: 2em;
      margin: 0;
      margin-bottom: 0.5em;
      padding: 0.25em;
      font-style: italic;
      text-align: center;
      letter-spacing: 0.5em;
      border-bottom-style: solid;
      border-bottom-width: 0.5em;
      border-bottom-color: #c00;
}
```

```
h2 {
      color: white;
      background-color: #090;
      font-size: 1.5em;
      margin: 0;
      padding: 0.1em;
      padding-left: 1em;
}
h3 {
      color: #999;
      font-size: 1.25em;
}
img {
      border-style: dashed;
      border-width: 2px;
      border-color: #ccc;
}
a {
      text-decoration: none;
}
strong {
      font-style: italic;
      text-transform: uppercase;
}
li {
      color: #900;
      font-style: italic;
}
table {
      background-color: #ccc;
}
```