
INTRODUCTION

CSS (Cascading Style Sheets)

The groups who help guide the internet standards ([C.E.R.N.](#), [W3C](#), [Internic](#), to name a few) are ever vigilant in their efforts at making web pages not only standardized but also flexible. Early in the development of HTML, these groups introduced HTML commands in which we could display information in eye-pleasing formats. Some of these capabilities came in the form of '', '' (bolding), '<TABLE>', etc..

Each of these commands gave the web developer some control in the color, size, and position of the information. But, as the web developed and the developers (you and I) grew in experience, calls went out to these organizations for more flexibility. Soon more standards were drafted where more and more attributes could be applied to commands. For instance, the '<TABLE>' command grew in flexibility by allowing you to designate a color '<TABLE BGCOLOR="blue">' and border sizes '<TABLE BORDER="3" BGCOLOR="blue">'.

As more and more attributes were added to each individual command, the groups who were responsible for these standards realized they needed a whole new approach and thus came the invention of CSS (cascading style sheets).

The Assignment

At the end of this assignment you should be able to...

- [INTRODUCTION](#)
- (1) [How to define a style](#)
- (2) [How to apply a style](#)
- (3) [On-the-fly styles](#)

(1) How to define a style

A style is simply a label that you place on parts of your information. This label instructs the browser to display information/items a certain way depending on how you defined the label. Before thinking about how this would work in a web page, let's watch the cartoon "As The Styles Turn" now in progress...

As The Styles Turn

1



Head 1



Head 2



Head 3

Ms. Arnold, I need you to use these three heads in some future commercials, but DON'T make the changes permanent!



2

Commercial One:

Need two heads with sunglasses and a red clown nose.

p.s. remember, DON'T permanently change MY heads!

-- Arnold

3

I will create a shield labeled 'glassandnose' and place it in front of a head when it needs to look that way.



4



Head 1



Head 2



Head 3

Commercial One Shoot:

Heads 1 and 3 used in shot, glassandnose shield applied. Shields removed after use.

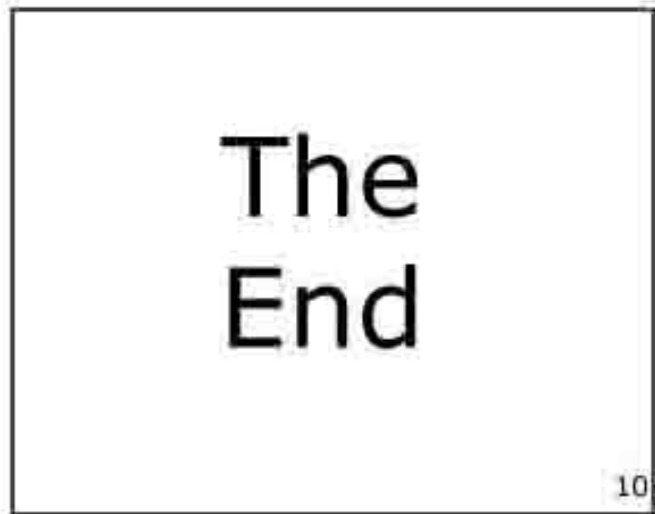
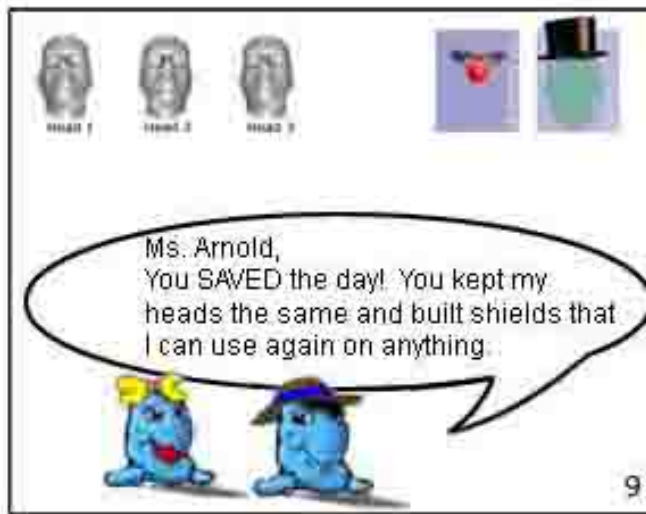
5

Commercial Two:
Needs two heads with green skin and hat. Use heads 2 and 3.

p.s. remember, DON'T permanently change MY heads!

-- Arnold

6



Style sheets work in the same manner as Ms. Arnold's shields. The basic steps to a style sheet are:

1. define a style and give it a name
2. use that name on any part of your page that you want that style

As with all web code, the browser reads and understands your code from top to bottom. The same holds true for style sheets. You can not use a style sheet unless you have already defined it somewhere above where you want to use it. For example, if you defined a style called 'bluedog' half way down your code but tried to use that style at the top of your code, it would be totally useless and ignored.

Ready to start coding? Then let's get typing...

There are two ways to define a style sheet. By the way, you can define as many as you need on a page. You will need to give each style its own name. The two ways are (1) `<STYLE>` and (2) on-the-fly. We will discuss the first and most used way here and then dive into the on-the-fly method in part #3 below.

To create styles you first need to have a section where you define your styles for later use. To do this, start a `STYLE` section and then have a matching ending `STYLE` section command:

```
<STYLE>  
</STYLE>
```



TIP! You should place your style definition right after your `</TITLE>` (ending Title command) and before your `</HEAD>` (ending Head command) at the top of your code. This way you can use the styles anywhere in your page and you will know that they will have already been defined.

Between the `<STYLE>` and `</STYLE>` commands you will place your self defined styles.



Rule #1: Each defined style has to have its own name preceded by a period '!'.

Here I will make style called **daveStyleOne** (note: style names ARE case sensitive. This means that if you use any capitalization in your name, you **MUST** use the same when using that style name later in your page).

```
<HTML>  
<HEAD>  
<TITLE>Dave's Page On Styles</TITLE>
```

```
<STYLE>  
.daveStyleOne  
</STYLE>
```

```
</HEAD>  
<BODY>  
</BODY>  
</HTML>
```



Rule #2: Each style you defined has its attributes contained within a set of brackets '{' and '}'.

I will now add the an attribute section to my **daveStyleOne** style by putting some brackets.

```
<HTML>
<HEAD>
<TITLE>Dave's Page On Styles</TITLE>

<STYLE>
.daveStyleOne
{
}
</STYLE>

</HEAD>
<BODY>
</BODY>
</HTML>
```

There ya go, it's that simple. Of course the style **daveStyleOne** doesn't have any attributes and so would not do anything yet. Before we add some attributes to **daveStyleOne**, let's add a second style so you can see how easy it is to have more. In the following I have added another style called **bobsCrayons**.

```
<HTML>
<HEAD>
<TITLE>Dave's Page On Styles</TITLE>

<STYLE>
.daveStyleOne
{
}
.bobsCrayons
{
}
</STYLE>

</HEAD>
<BODY>
```

```
</BODY>
</HTML>
```

Notice how each style is preceded with a period '.' and has its own set of brackets immediately following the name?

You can keep adding as many styles (each with their own name) as you wish.

Okay, okay, I know...you want to know how to define the attributes (i.e. how the style will change how something looks). You have waited long enough, here goes...

To add an attribute, go to a style that you have defined (see above) and enter that attribute between the brackets for that style. Each attribute should be on its own line and end in a semicolon ';'.

On **daveStyleOne**, I have decided that I want anything that uses that style to have text (font) at a size of 28 pixels (px). So I would add the line:

```
font-size: 28px;
```

This attribute line would be placed inside the brackets right after **daveStyleOne** as follows.

```
<HTML>
<HEAD>
<TITLE>Dave's Page On Styles</TITLE>
```

```
<STYLE>
.daveStyleOne
{
font-size: 28px;
}
.bobsCrayons
{
}
</STYLE>
```

```
</HEAD>
<BODY>
</BODY>
</HTML>
```

Remember, nothing has been applied to the page yet, we are simply defining our styles for use later. In section #2 below, we will apply the styles we defined here to actual information.

Now that I have decided that **daveStyleOne** has a font size of 28px, I also want it to be in the font lettering of Comic Sans, so I could add the following line.

```
font-family: Comic Sans MS;
```

And once again place this attribute between the brackets. (note: it does not matter what order the attributes are in).

```
<HTML>
<HEAD>
<TITLE>Dave's Page On Styles</TITLE>
```

```
<STYLE>
.daveStyleOne
{
font-size: 28px;
font-family: Comic Sans MS;
}
.bobsCrayons
{
}
</STYLE>
```

```
</HEAD>
<BODY>
</BODY>
</HTML>
```

The **bobsCrayons** should have something to do. We will make the style **bobsCrayons** display any text in a 14px size font, in the Arial type lettering and with a border on the top and bottom that is red at 3px width and finally a background color of #DDDDFF (yellowish). These are lines that you would add.

```
font-size: 14px;
font-family: Arial;
border-top: #FF0000 3px solid;
border-bottom: #FF0000 3px dashed;
```

```
background-color: #FFFFD9;
```

Further definition: The **#FF0000 1px solid** instructs that style to put a 3 pixel wide line in a solid or dashed format in the color red (#FF0000 - see [Assignment 7: About Colors](#) for help on Hex codes)

Place the previous 5 lines inside of the **bobsCrayons** and that style will now affect text in that way, as follows:

```
<HTML>
<HEAD>
<TITLE>Dave's Page On Styles</TITLE>

<STYLE>
.daveStyleOne
{
font-size: 28px;
font-family: Comic Sans MS;
}
.bobsCrayons
{
font-size: 14px;
font-family: Arial;
border-top: #FF0000 3px solid;
border-bottom: #FF0000 3px dashed;
background-color: #FFFFD9;
}
</STYLE>

</HEAD>
<BODY>
</BODY>
</HTML>
```

(2) How to apply a style

Now you get to see how **easy** it is to apply these styles to anything on your page.

There are three ways to apply a style to a part (or parts) of your page:

1. by pointing to it from another HTML command
2. by using the **** command
3. by using the **<DIV>** command

For now we will concentrate on the first method 'point to it from another HTML command'. To use one of your styles on an HTML command, you just place the attribute `class=""` inside of that command and supply the name of the style. (note: the period preceding the style name is only used when defining, you do not use it when applying it).

For example, let's say you have three paragraph commands `<P>`, each containing some text and finally an ending paragraph command `</P>`, like:

Code Used	Result
<code><P></code> 1. Welcome to the new Year. <code></P></code>	1. Welcome to the new Year.
<code><P></code> 2. Hope you have a nice day. <code></P></code>	2. Hope you have a nice day.
<code><P></code> 3. Where did I put that remote? <code></P></code>	3. Where did I put that remote?

In this case you wanted to apply the style `daveStyleOne` to paragraphs 1 and 3, and apply style `bobsCrayons` to paragraph 2, you could put the attribute `class="daveStyleOne"` inside paragraphs 1 and 3 command and the `class="bobsCrayons"` inside paragraph 2 command, like:

Code Used	Result
<code><P class="daveStyleOne"></code> 1. Welcome to the new Year. <code></P></code>	1. Welcome to the new Year.
<code><P class="bobsCrayons"></code> 2. Hope you have a nice day. <code></P></code>	
<code><P class="daveStyleOne"></code> 3. Where did I put that remote? <code></P></code>	2. Hope you have a nice day.
	3. Where did I put that remote?

Are you excited now? Do you see how you can use a style over and over again as needed?

You can apply a style to most other commands, like:

- Table command: `<TABLE class="bobsCrayons">`
- Row command: `<TR class="bobsCrayons">`
- Cell command: `<TD class="bobsCrayons">`
- Paragraph command: `<P class="bobsCrayons">`
- Italics command: `<I class="bobsCrayons">`
- etc..

So what about those times when you want to apply a style but don't have the information inside of a convenient command? That is where the `` and `<DIV>` commands come in. These two commands were created specifically for you to use when you just want to apply styles.

First we will discuss the `` command. This command can be placed around information that you want a style applied to anywhere that information is. For instance, you might have information in a cell and don't want the style applied to the entire cell but just a portion of the information, or maybe you have a phrase in the middle of a paragraph that you want affected but not the other parts of the paragraph, well just use the `` command!

In the following, I have a regular paragraph, but in the middle of the paragraph I want to apply the `bobsCrayons` style to a few words, like:

Code Used	Result
<pre><P> Hello all you worm racing facing! Tonight we are pleased to announce the arrival of the Humungous, the Super worm from the Northwest... Whirlwind Worm! </P></pre>	<p>Hello all you worm racing facing! Tonight we are pleased to announce the arrival of the Humungous, the Super worm from the Northwest... Whirlwind Worm!</p>

See how easy that was?

Now to move on to **<DIV>**. The only difference between **SPAN** and **DIV** is that the **DIV** command is considered its own section and will get automatic line breaks before and after the command. Later, you can learn how to **move** the **DIV** sections on requests (like flying around the screen), which the **SPAN** sections can't do.

Here is the previous example redone with the same code, except that I have replaced the **SPAN** commands with **DIV**.

Code Used	Result
<pre><P> Hello all you worm racing facing! Tonight we are pleased to announce the arrival of the <div class="bobsCrayons">Humungous</div>, the <div class="bobsCrayons">Super</div> worm from the Northwest... <div class="bobsCrayons">Whirlwind Worm</div>! </P></pre>	<p>Hello all you worm racing facing! Tonight we are pleased to announce the arrival of the Humungous , the Super worm from the Northwest... Whirlwind Worm !</p>

See how each line with a **DIV** was automatically move to the next line?

Most of the time you will use the **SPAN** command, but later as you advance you may want to jump into using the **DIV** and all the possibilities it offers.

(3) On-the-fly styles

There are going to be times that you want to apply a style to something on your page but you are only going to affect that one thing and so don't need a 'named' style that you can use over and over again.

In these cases you can apply specific styles to a HTML command by using the attribute **style=""**. The following information needs paragraph 2 to be a different color and larger, but I only needed to do this once, so I apply the **style=""** attribute to the **<P>** command as follows:

Code Used	Result
-----------	--------

<P>

**A page about absolutely nothing at all.
We are glad you visited.**

</P>

<P style="font-size:24px;color:#91C3E1">

**Did you really want to be here, or is
there just nothing on T.V. that grabbed
your attention?**

</P>

<P>

Well, thanks for visiting anyway!

</P>

A page about absolutely nothing at all. We are glad you visited.

Did you really want to be here, or is there just nothing on T.V. that grabbed your attention?

Well, thanks for visiting anyway!

It's that simple!

CONGRATULATIONS!!...You've now learned CSS.
