CSS
CSS is a language that describes the style of an HTML document.
CSS describes how element should be displayed.
CSS stands for Cascading Style Sheets.
CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
CSS saves a lot of work. It can control the layout of multiple web pages all at once.
External style sheets are stored in CSS files.

Example
```
body {
   background-color: lightblue;
}

h1 {
   color: white;
   text-align: center;
}

p {
   font-family: verdana;
   font-size: 20px;
}
```

Why Use CSS?
CSS is used to define styles for your web pages, including the design, layout in display for different devices and screen sizes.
CSS Saves a Lot of Work!
The style definitions are normally saved in external **.css** files.
With an external style sheet file, you can change the look of an entire website by changing just one file.

CSS Syntax
A CSS rule-set consists of a selector and a declaration block.

```
H1 {color: blue; font-size: 12px;}
```

The selector points to the HTML element you want to style.
The declaration block contains one or more declarations separated by semicolons.
Each declaration includes a CSS property name and a value, separated by a colon.
A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.
In the following example all <p> elements will be center-aligned, with a red text color:

Example
```
p {
   color: red;
   text-align: center;
}
```

CSS Selectors
CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

The element Selector
The element selector selects elements based on the element name.
You can select all <p> elements on a page like this (in this case, all <p> elements will be center-aligned, with a red text color):

Example
```
p {
    text-align: center;
    color: red;
}T
```

The id Selector
The id selector uses the id attribute of an HTML element to select a specific element.
The id of an element should be unique within a page, so the id selector is used to select one unique element!
To select an element with a specific id, write a hash (#) character, followed by the id of the element.
The style rule below will be applied to the HTML element with id="para1":

Example
```
#para1 {
    text-align: center;
    color: red;
}
```

The class Selector
The class selector selects elements with a specific class attribute.
To select elements with a specific class, write a period (.) character, followed by the name of the class.
In the example below, all HTML elements with class="center" will be red and center-aligned:

Example
```
.center {
    text-align: center;
    color: red;
}
```
You can also specify that only specific HTML elements should be affected by a class.
In the example below, only <p> elements with class="center" will be center-aligned:

Example
```
p.center {
    text-align: center;
    color: red;
}
```

Grouping Selectors
If you have elements with the same style definitions, like this:

```
h1 {
    text-align: center;
    color: red;
}

h2 {
    text-align: center;
    color: red;
}

p {
    text-align: center;
    color: red;
}
```

It will be better to group the selectors, to minimize the code.
To group selectors, separate each selector with a comma.
In the example below we have grouped the selectors from the code above:

Example

```
h1, h2, p {
    text-align: center;
    color: red;
}
```

CSS Borders
The CSS border properties allow you to specify the style, width, and color of an element's border.
The following values are allowed:

- Dotted - Defines a dotted border
- Dashed - Defines a dashed border
- Solid - Defines a solid border
- Double - Defines a double border
- Groove - Defines a 3D grooved border. The effect depends on the border-color value
- Ridge - Defines a 3D ridged border. The effect depends on the border-color value
- Inset - Defines a 3D inset border. The effect depends on the border-color value
- Outset - Defines a 3D outset border. The effect depends on the border-color value
- None - Defines no border
- Hidden - Defines a hidden border

The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

Example

```
<!DOCTYPE html>
<html>
<head>
```

```
<style>
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
</style>
</head>
<body>

<h2>The border-style Property</h2>
<p>This property specifies what kind of border to display:</p>

<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
<p class="outset">An outset border.</p>
<p class="none">No border.</p>
<p class="hidden">A hidden border.</p>
<p class="mix">A mixed border.</p>

</body>
</html>
```

CSS Background

The CSS background properties are used to define the background effects for elements.

CSS background properties:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

CSS background-color

The CSS Background property specifies the background color of an element.

Example

```
body {
  background-color: lightblue;
}
```

CSS background-image

The background-image property specifies an image to use as the background of an element.

Example

```
body {
  background-image: url("paper.gif");
}
```

CSS background-repeat

By default, the background-image property repeats an image both horizontally and vertically.

Some images should be repeated only horizontally or vertically, or they will look strange, like this:

Example

```
body {
  background-image: url("gradient_bg.png");
  background-repeat: repeat-x;
}
```

CSS background-position

The background-position property is used to specify the position of the background image.

Example

```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
}
```

CSS background-attachment

The background-attachment property specifies whether the background image should scroll or be fixed

Example

```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
  background-attachment: fixed;
}
```

Text Decoration

The text-decoration property is used to set or remove decorations from text.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```
h1 {
  text-decoration: overline;
}

h2 {
  text-decoration: line-through;
}

h3 {
  text-decoration: underline;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>

</body>
</html>
```

Text Transformation

The text-transform property is used to specify uppercase and lowercase letters in a text. It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p.uppercase {
  text-transform: uppercase;
}

p.lowercase {
  text-transform: lowercase;
}

p.capitalize {
  text-transform: capitalize;
}
</style>
</head>
<body>

<p class="uppercase">This is some text.</p>
<p class="lowercase">This is some text.</p>
<p class="capitalize">This is some text.</p>
</body>
</html>
```

CSS Fonts
The font family of a text is set with the font-family property.

Example
```
<!DOCTYPE html>
<html>
<head>
<style>
p.serif {
  font-family: "Times New Roman", Times, serif;
}

p.sansserif {
  font-family: Arial, Helvetica, sans-serif;
}
</style>
</head>
<body>

<h1>CSS font-family</h1>
<p class="serif">This is a paragraph, shown in the Times New Roman font.</p>
<p class="sansserif">This is a paragraph, shown in the Arial font.</p>

</body>
</html>
```

CSS Font Style
The font-style property is mostly used to specify italic text.
This property has three values:

- normal - The text is shown normally
- italic - The text is shown in italics

Example
```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {
  font-style: normal;
}

p.italic {
  font-style: italic;
}

p.oblique {
  font-style: oblique;
}
</style>
</head>
<body>
```

```
<p class="normal">This is a paragraph in normal style.</p>
<p class="italic">This is a paragraph in italic style.</p>
<p class="oblique">This is a paragraph in oblique style.</p>

</body>
</html>
```

CSS Layout

A website is often divided into headers, menus, content and footer.
We will create a 3-column layout, and change it to a 1-column layout on smaller screens.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Website Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}

body {
  margin: 0;
}

/* Style the header */
.header {
  background-color: #f1f1f1;
  padding: 20px;
  text-align: center;
}

/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;
}

/* Style the topnav links */
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}
```

```
/* Change color on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}

/* Create three equal columns that floats next to each other */
.column {
  float: left;
  width: 33.33%;
  padding: 15px;
}

/* Clear floats after the columns */
.row:after {
  content: "";
  display: table;
  clear: both;
}

/* Responsive layout - makes the three columns stack on top of each other instead of
next to each other */
@media screen and (max-width:600px) {
  .column {
    width: 100%;
  }
}
</style>
</head>
<body>

<div class="header">
  <h1>Header</h1>
  <p>Resize the browser window to see the responsive effect.</p>
</div>

<div class="topnav">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
</div>

<div class="row">
  <div class="column">
    <h2>Column</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </div>
```

```
<div class="column">
   <h2>Column</h2>
   <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. </p>
  </div>

  <div class="column">
   <h2>Column</h2>
   <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </div>
</div>

</body>
</html>
```

CSS Transitions
CSS transitions allows you to change property values smoothly (from one value to another), over a given duration.

How to Use CSS Transitions?
To create a transition effect, you must specify two things:

- the CSS property you want to add an effect to
- the duration of the effect

Note: If the duration part is not specified, the transition will have no effect, because the default value is 0.

Specify the Speed Curve of the Transition
The transition-timing-function property specifies the speed curve of the transition effect.
The transition-timing-function property can have the following values:

- **Ease -** specifies a transition effect with a slow start, then fast, and then end slowly.
- **linear -** specifies a transition effect with the same speed from start to end
- **ease-in -** specifies a transition effect with a slow start
- **ease-out -** specifies a transition effect with a slow end
- **ease-in-out -** specifies a transition effect with a slow start and end

The following example shows the some of the different speed curves that can be used:

Example
```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
```

```
 /* Standard syntax */
  transition-property: width;
  transition-duration: 2s;
  transition-timing-function: linear;
  transition-delay: 1s;
}

div:hover {
  width: 300px;
}
</style>
</head>
<body>

<h1>The transition Properties Specified One by One</h1>

<p>Hover over the div element below, to see the transition effect:</p>

<div></div>

<p><b>Note:</b> The transition effect has a 1 second delay before starting.</p>

<p><b>Note:</b> This example does not work in Internet Explorer 9 and earlier
versions.</p>

</body>
</html>
```

CSS Animations

An animation lets an element gradually change from one style to another.
You can change as many CSS properties you want, as many times you want.
To use CSS animation, you must first specify some key frames for the animation.
.Key frames hold what styles the element will have at certain times.
.When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
 animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: 2;
  animation-direction: alternate;
}
```

```
/* Standard syntax */
@keyframes example {
  0%   {background-color:red; left:0px; top:0px;}
  25%  {background-color:yellow; left:200px; top:0px;}
  50%  {background-color:blue; left:200px; top:200px;}
  75%  {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>

<p><b>Note:</b> This example does not work in Internet Explorer 9 and earlier versions.</p>

<div></div>

</body>
</html>
```

CSS Transforms

CSS transforms allow you to move, rotate, scale, and skew elements.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 300px;
  height: 100px;
  background-color: yellow;
  border: 1px solid black;
}

div#myDiv1 {
  transform: matrix(1, -0.3, 0, 1, 0, 0); /* Standard syntax */
}

div#myDiv2 {
  transform: matrix(1, 0, 0.5, 1, 150, 0); /* Standard syntax */
}
</style>
</head>
<body>

<h1>The matrix() Method</h1>
<p>The matrix() method combines all the 2D transform methods into one.</p>

<div>
This a normal div element.
```

```
</div>
<div id="myDiv1">
Using the matrix() method.
</div>

<div id="myDiv2">
Another use of the matrix() method.
</div>

</body>
</html>
```

HTML Basics

Welcome to HTML Basics. This workshop leads you through the basics of Hyper Text Markup Language (HTML). HTML is the building block for web pages. You will learn to use HTML to author an HTML page to display in a web browser.

Objectives:

By the end of this workshop, you will be able to:

- Use a text editor to author an HTML document.
- Be able to use basic tags to denote paragraphs, emphasis or special type.
- Create hyperlinks to other documents.
- Create an email link.
- Add images to your document.
- Use a table for layout.
- Apply colors to your HTML document.

Prerequisites:

You will need a text editor, such as Notepad and an Internet browser, such as Internet Explorer or Netscape.

> **Q:** What is Notepad and where do I get it?
> **A:** Notepad is the default Windows text editor. On most Windows systems, click your Start button and choose Programs then Accessories. It should be a little blue notebook.
>
> **Mac Users:** SimpleText is the default text editor on the Mac. In OSX use TextEdit and change the following preferences: Select (in the preferences window) Plain text instead of Rich text and then select Ignore rich text commands in HTML files. This is very important because if you don't do this HTML codes probably won't work.

One thing you should avoid using is a word processor (like Microsoft Word) for authoring your HTML documents.

What is an html File?

HTML is a format that tells a computer how to display a web page. The documents themselves are plain text files with special "tags" or codes that a web browser uses to interpret and display information on your computer screen.

- HTML stands for Hyper Text Markup Language
- An HTML file is a text file containing small markup tags
- The markup tags tell the Web browser how to display the page
- An HTML file must have an htm or html file extension

Try It?

Open your text editor and type the following text:

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body>
This is my first homepage. <b>This text is bold</b>
</body>
</html>
```

Save the file as **mypage.html**. Start your Internet browser. Select **Open** (or Open Page) in the **File** menu of your browser. A dialog box will appear. Select **Browse** (or Choose File) and locate the html file you just created - **mypage.html** - select it and click **Open**. Now you should see an address in the

dialog box, for example **C:\MyDocuments\mypage.html**. Click **OK**, and the browser will display the page. To view how the page should look, visit this web page: **http://profdevtrain.austincc.edu/html/mypage.html**

Example Explained

What you just made is a skeleton html document. This is the minimum required information for a web document and all web documents should contain these basic components. The first tag in your html document is `<html>`. This tag tells your browser that this is the start of an html document. The last tag in your document is `</html>`. This tag tells your browser that this is the end of the html document.

The text between the `<head>` tag and the `</head>` tag is header information. Header information is not displayed in the browser window.

The text between the `<title>` tags is the title of your document. The `<title>` tag is used to uniquely identify each document and is also displayed in the title bar of the browser window.

The text between the `<body>` tags is the text that will be displayed in your browser.

The text between the `<b>` and `</b>` tags will be displayed in a bold font.

HTM or HTML Extension?

When you save an HTML file, you can use either the .htm or the .html extension. The .htm extension comes from the past when some of the commonly used software only allowed three letter extensions. It is perfectly safe to use either .html or .htm, but be consistent. **mypage.htm** and mypage.html are treated as different files by the browser.

How to View HTML Source

A good way to learn HTML is to look at how other people have coded their html pages. To find out, simply click on the View option in your browsers toolbar and select Source or Page Source. This will open a window that shows you the actual HTML of the page. Go ahead and view the source html for this page.

HTML Tags

What are HTML tags?

- HTML tags are used to mark-up HTML elements
- HTML tags are surrounded by the two characters < and >
- The surrounding characters are called angle brackets
- HTML tags normally come in pairs like and
- The first tag in a pair is the start tag, the second tag is the end tag
- The text between the start and end tags is the element content
- HTML tags are not case sensitive, means the same as <B>

Logical vs. Physical Tags

In HTML there are both logical tags and physical tags. Logical tags are designed to describe (to the browser) the enclosed text's meaning. An example of a logical tag is the `<strong> </strong>` tag. By

placing text in between these tags you are telling the browser that the text has some greater importance. By default all browsers make the text appear bold when in between the `<strong>` and `</strong>` tags.

Physical tags on the other hand provide specific instructions on how to display the text they enclose. Examples of physical tags include:

- `<b>`: Makes the text bold.
- `<big>`: Makes the text usually one size bigger than what's around it.
- `<i>`: Makes text italic.

Physical tags were invented to add style to HTML pages because style sheets were not around, though the original intention of HTML was to not have physical tags. Rather than use physical tags to style your HTML pages, you should use style sheets.

HTML Elements

Remember the HTML example from the previous page:

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body>
This is my first homepage. <b>This text is bold</b>
</body>
</html>
```

This is an HTML element:

```
<b>This text is bold</b>
```

The HTML element begins with a start tag: `<b>`
The content of the HTML element is: This text is bold
The HTML element ends with an end tag: `</b>`

The purpose of the `<b>` tag is to define an HTML element that should be displayed as bold.

This is also an HTML element:

```
<body>
This is my first homepage. <b>This text is bold</b>
</body>
```

This HTML element starts with the start tag `<body>`, and ends with the end tag `</body>`. The purpose of the `<body>` tag is to define the HTML element that contains the body of the HTML document.

Nested Tags

You may have noticed in the example above, the `<body>` tag also contains other tags, like the `<b>` tab. When you enclose an element in with multiple tags, the last tag opened should be the first tag closed. For example:

```
<p><b><em>This is NOT the proper way to close nested tags .</p></em></b>
```

```
<p><b><em>This is the proper way to close nested tags. </em></b></p>
```

> **Note:** It doesn't matter which tag is first, but they must be closed in the proper order.

Why Use Lowercase Tags?

You may notice we've used lowercase tags even though I said that HTML tags are not case sensitive. `<B>` means the same as `<b>`. The World Wide Web Consortium (W3C), the group responsible for developing web standards, recommends lowercase tags in their HTML 4 recommendation, and XHTML (the next generation HTML) requires lowercase tags.

Tag Attributes

Tags can have attributes. Attributes can provide additional information about the HTML elements on your page. The `<tag>`tells the browser to do something, while the attribute tells the browser how to do it. For instance, if we add the bgcolor attribute, we can tell the browser that the background color of your page should be blue, like this: `<body bgcolor="blue">`.

This tag defines an HTML table: `<table>`. With an added border attribute, you can tell the browser that the table should have no borders: `<table border="0">`. Attributes always come in name/value pairs like this: name="value". Attributes are always added to the start tag of an HTML element and the value is surrounded by quotes.

Quote Styles, "red" or 'red'?

Attribute values should always be enclosed in quotes. Double style quotes are the most common, but single style quotes are also allowed. In some rare situations, like when the attribute value itself contains quotes, it is necessary to use single quotes:

name='George "machine Gun" Kelly'

> **Note:** Some tags we will discuss are deprecated, meaning the World Wide Web Consortium (W3C) the governing body that sets HTML, XML, CSS, and other technical standards decided those tags and attributes are marked for deletion in future versions of HTML and XHTML. Browsers should continue to support deprecated tags and attributes, but eventually these tags are likely to become obsolete and so future support cannot be guaranteed.

For a complete list of tags, visit W3C.org.

Basic HTML Tags

The most important tags in HTML are tags that define headings, paragraphs and line breaks.

Basic HTML Tags

Tag	Description
<html>	Defines an HTML document
<body>	Defines the document's body
<h1> to <h6>	Defines header 1 to header 6
<p>	Defines a paragraph
 	Inserts a single line break
<hr>	Defines a horizontal rule
<!-->	Defines a comment

Headings

Headings are defined with the `<h1>` to `<h6>` tags. `<h1>` defines the largest heading while `<h6>` defines the smallest.

<h1>This is a heading</h1>

<h2>This is a heading</h2>

<h3>This is a heading</h3>

<h4>This is a heading</h4>

<h5>This is a heading</h5>

<h6> This is a heading</h6>

HTML automatically adds an extra blank line before and after a heading. A useful heading attribute is align.

```
<h5 align="left">I can align headings </h5>

<h5 align="center">This is a centered heading </h5>

<h5 align="right">This is a heading aligned to the right </h5>
```

Paragraphs

Paragraphs are defined with the <p> tag. Think of a paragraph as a block of text. You can use the align attribute with a paragraph tag as well.

```
<p align="left">This is a paragraph</p>

<p align="center">this is another paragraph</p>
```

Important: You must indicate paragraphs with <p> elements. A browser ignores any indentations or blank lines in the source text. Without <p> elements, the document becomes one large paragraph. HTML automatically adds an extra blank line before and after a paragraph.

Line Breaks

The <br> tag is used when you want to start a new line, but don't want to start a new paragraph. The <br> tag forces a line break wherever you place it. It is similar to single spacing in a document.

This Code	Would Display
`<p>This   is a para  graph with line breaks</p>`	This is a para graph with line breaks

The <br> tag has no closing tag.

Horizontal Rule

The <hr> element is used for horizontal rules that act as dividers between sections, like this:

The horizontal rule does not have a closing tag. It takes attributes such as align and width. For instance:

This Code	Would Display
`<hr width="50%" align="center">`	

Comments in HTML

The comment tag is used to insert a comment in the HTML source code. A comment can be placed anywhere in the document and the browser will ignore everything inside the brackets. You can use comments to write notes to yourself, or write a helpful message to someone looking at your source code.

This Code	Would Display
`<p> This html comment would <!-- This is a comment --> be displayed like this.</p>`	This HTML comment would be displayed like this.

Notice you don't see the text between the tags `<!--` and `-->`. If you look at the source code, you would see the comment. To view the source code for this page, in your browser window, select **View** and then select **Source**.

Note: You need an exclamation point after the opening bracket `<!--` but not before the closing bracket `-->`.

HTML automatically adds an extra blank line before and after some elements, like before and after a paragraph, and before and after a heading. If you want to insert blank lines into your document, use the `<br>` tag.

Try It Out!

Open your text editor and type the following text:

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body>
<h1 align="center">My First Webpage</h1>
<p>Welcome to my first web page. I am writing this page using a text editor and plain
old html.</p>
<p>By learning html, I'll be able to create web pages like a pro... <br>
which I am of course.</p>
</body>
</html>
```

Save the page as **mypage2.html**. Open the file in your Internet browser. To view how the page should look, visit this web page: **http://profdevtrain.austincc.edu/html/mypage2.html**

Other HTML Tags

As mentioned before, there are logical styles that describe what the text should be and physical styles which actually provide physical formatting. It is recommended to use the logical tags and use style sheets to style the text in those tags.

Logical Tags

Tag	Description
<abbr>	Defines an abbreviation
<acronym>	Defines an acronym
<address>	Defines an address element
<cite>	Defines a *citation*
<code>	Defines `computer code` text
<blockquote>	Defines a long quotation
<del>	Defines text
<dfn>	Defines a *definition* term
	Defines *emphasized* text
<ins>	Defines inserted text
<kbd>	Defines keyboard text
<pre>	Defines preformatted text
<q>	Defines a short quotation
<samp>	Defines sample computer code
	Defines **strong** text
<var>	Defines a *variable*

Physical Tags

Tag	Description
<b>	Defines **bold** text
<big>	Defines big text
<i>	Defines *italic* text
<small>	Defines small text
<sup>	Defines superscripted text
<sub>	Defines subscripted text
<tt>	Defines teletype text
<u>	Deprecated. Use styles instead

Character tags like and produce the same physical display as <b> and <i> but are more uniformly supported across different browsers.

Some Examples:

The following paragraph uses the <blockquote>tag. In the previous sentence, the blockquote tag is enclosed in the <samp> Sample tag.

> We the people of the United States, in order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common defense, promote the general Welfare, and secure the Blessings of Liberty to ourselves and our Posterity, do ordain and establish this Constitution for the United States of America.

Although most browsers render blockquoted text by indenting it, that's not specifically what it's designed to do. It's conceivable that some future browser may render blockquoted text in some other way. However, for the time being, it is perfectly safe to indent blocks of text with the <blockquote>.

This Code	Would Display
<abbr title="World Wide Web">WWW</abbr>	WWW

When you hold your mouse pointer over the WWW, text in the title attribute will appear in.

HTML Character Entities

Some characters have a special meaning in HTML, like the less than sign (<) that defines the start of an HTML tag. If we want the browser to actually display these characters we must insert character entities in place of the actual characters themselves.

The Most Common Character Entities:

Result	Description	Entity Name	Entity Number
	non-breaking space		
<	less than	&lt;	&#60;
>	greater than	&gt;	&#62;
&	ampersand	&amp;	&#38;
"	quotation mark	&quot;	&#34;
'	apostrophe	' (does not work in IE)	&#39;

A character entity has three parts: an ampersand (&), an entity name or an entity number, and finally a semicolon (;). The & means we are beginning a special character, the **;** means ending a special character and the letters in between are sort of an abbreviation for what it's for. To display a less than sign in an HTML document we must write: **<** or **<** The advantage of using a name instead of a number is that a name is easier to remember. The disadvantage is that not all browsers support the newest entity names, while the support for entity numbers is very good in almost all browsers.

> **Note:** Entities are case sensitive.

Non-breaking Space

The most common character entity in HTML is the non-breaking space ** **. Normally HTML will truncate spaces in your text. If you add 10 spaces in your text, HTML will remove 9 of them. To add spaces to your text, use the   character entity.

This Code	Would Display
`<p> This code would appear as this.</p>`	This code would appear as this.

This Code	Would Display
`<p> This code     would appear with three extra spaces.</p>`	This code would appear with three extra spaces.

To see a list of character entities, visit this page:
http://profdevtrain.austincc.edu/html/entities.htm

HTML Fonts

The `<font>` tag in HTML is deprecated. The World Wide Web Consortium (W3C) has removed the `<font>` tag from its recommendations. In future versions of HTML, style sheets (CSS) will be used to define the layout and display properties of HTML elements.

The `<font>` Tag Should **NOT** be used.

HTML Backgrounds

Backgrounds

The `<body>` tag has two attributes where you can specify backgrounds. The background can be a color or an image.

Bgcolor

The bgcolor attribute specifies a background-color for an HTML page. The value of this attribute can be a hexadecimal number, an RGB value, or a color name:

```
<body bgcolor="#000000">
<body bgcolor="rgb(0,0,0)">
<body bgcolor="black">
```

The lines above all set the background-color to black.

Background

The background attribute can also specify a background-image for an HTML page. The value of this attribute is the URL of the image you want to use. If the image is smaller than the browser window, the image will repeat itself until it fills the entire browser window.

```
<body background="clouds.gif">
<body background="http://profdevtrain.austincc.edu/html/graphics/clouds.gif">
```

The URL can be relative (as in the first line above) or absolute (as in the second line above).
If you want to use a background image, you should keep in mind:

- Will the background image increase the loading time too much?
- Will the background image look good with other images on the page?
- Will the background image look good with the text colors on the page?
- Will the background image look good when it is repeated on the page?
- Will the background image take away the focus from the text?

Note: The bgcolor, background, and the text attributes in the <body> tag are deprecated in the latest versions of HTML (HTML 4 and XHTML). The World Wide Web Consortium (W3C) has removed these attributes from its recommendations. Style sheets (CSS) should be used instead (to define the layout and display properties of HTML elements).

Try It Out!

Open your text editor and type the following text:

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body background="http://profdevtrain.austincc.edu/html/graphics/clouds.gif"
bgcolor="#EDDD9E">
<h1 align="center">My First Webpage</h1>
<p>Welcome to my <strong>first</strong> webpage. I am writing this page using a text
editor and plain old html.</p>
<p>By learning html, I'll be able to create webpages like a <del>beginner</del>
pro....<br>
which I am of course.</p>
</body>
</html>
```

Save your page as **mypage3.html** and view it in your browser. To view how the page should look, visit this web page: **http://profdevtrain.austincc.edu/html/mypage3.html**
Notice we gave our page a background color as well as a background image. If for some reason the web page is unable to find the picture, it will display our background color.

HTML Colors

Color Values

Colors are defined using a hexadecimal notation for the combination of red, green, and blue color values (RGB). The lowest value that can be given to one light source is 0 (hex #00). The highest value is 255 (hex #FF). This table shows the result of combining red, green, and blue:

Color	Color HEX	Color RGB
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#C0C0C0	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)

Color Names

A collection of color names is supported by most browsers. To view a table of color names that are supported by most browsers visit this web page: **http://profdevtrain.austincc.edu/html/color_names.htm**

> **Note:** Only 16 **color names** are supported by the W3C HTML 4.0 standard (aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow). For all other colors you should use the **Color HEX** value.

Color	Color HEX	Color Name
	#F0F8FF	AliceBlue
	#FAEBD7	AntiqueWhite
	#7FFFD4	Aquamarine
	#000000	Black

	#0000FF	Blue
	#8A2BE2	BlueViolet
	#A52A2A	Brown

Web Safe Colors

A few years ago, when most computers supported only 256 different colors, a list of 216 Web Safe Colors was suggested as a Web standard. The reason for this was that the Microsoft and Mac operating system used 40 different "reserved" fixed system colors (about 20 each). This 216 cross platform web safe color palette was originally created to ensure that all computers would display all colors correctly when running a 256 color palette. To view the 216 Cross Platform Colors visit this web page: **http://profdevtrain.austincc.edu/html/216.html**

16 Million Different Colors

The combination of Red, Green and Blue values from 0 to 255 gives a total of more than 16 million different colors to play with (256 x 256 x 256). Most modern monitors are capable of displaying at least 16,384 different colors. To assist you in using color schemes, check out **http://wellstyled.com/tools/colorscheme2/index-en.html**. This site lets you test different color schemes for page backgrounds, text and links.

HTML Lists

HTML provides a simple way to show unordered lists (bullet lists) or ordered lists (numbered lists).

Unordered Lists

An unordered list is a list of items marked with bullets (typically small black circles). An unordered list starts with the `<ul>` tag. Each list item starts with the `<li>` tag.

This Code	Would Display

`<ul>`
`<li>Coffee</li>`
`<li>Milk</li>`
`</ul>`
	▪ Coffee
	▪ Milk

Ordered Lists

An ordered list is also a list of items. The list items are marked with numbers. An ordered list starts with the `<ol>` tag. Each list item starts with the `<li>` tag.

This Code	Would Display

`<ol>`
`<li>Coffee</li>`
`<li>Milk</li>`
`</ol>`
	1. Coffee
	2. Milk

Inside a list item you can put paragraphs, line breaks, images, links, other lists, etc.

Definition Lists

Definition lists consist of two parts: a **term** and a **description**. To mark up a definition list, you need three HTML elements; a container <dl>, a definition term <dt>, and a definition description <dd>.

This Code	Would Display
<dl> <dt>Cascading Style Sheets</dt> <dd>Style sheets are used to provide presentational suggestions for documents marked up in HTML. </dd> </dl>	Cascading Style Sheets Style sheets are used to provide presentational suggestions for documents marked up in HTML.

Inside a definition-list definition (the <dd> tag) you can put paragraphs, line breaks, images, links, other lists, etc

Try It Out

Open your text editor and type the following:

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body bgcolor="#EDDD9E">
<h1 align="center">My First Webpage</h1>
<p>Welcome to my <strong>first</strong> webpage. I am writing this page using a text
editor and plain old html.</p>
<p>By learning html, I'll be able to create web pages like a pro... <br>
which I am of course.</p>
Here's what I've learned:
<ul>
<li>How to use HTML tags</li>
<li>How to use HTML colors</li>
<li>How to create Lists</li>
</ul>
</body>
</html>
```

Save your page as **mypage4.html** and view it in your browser. To see how your page should look visit this web page: **http://profdevtrain.austincc.edu/html/mypage4.html**

HTML Links

HTML uses the `<a>` anchor tag to create a link to another document or web page.

The Anchor Tag and the Href Attribute

An anchor can point to any resource on the Web: an HTML page, an image, a sound file, a movie, etc. The syntax of creating an anchor:

```
<a href="url">Text to be displayed</a>
```

The `<a>` tag is used to create an anchor to link from, the href attribute is used to tell the address of the document or page we are linking to, and the words between the open and close of the anchor tag will be displayed as a hyperlink.

This Code	Would Display
`<a href="http://www.austincc.edu/">Visit ACC!</a>`	Visit ACC!

The Target Attribute

With the target attribute, you can define **where** the linked document will be opened. By default, the link will open in the current window. The code below will open the document in a new browser window:

```
<a href=http://www.austincc.edu/ target="_blank">Visit ACC!</a>
```

Email Links

To create an email link, you will use mailto: plus your email address. Here is a link to ACC's Help Desk:

```
<a href="mailto:helpdesk@austincc.edu">Email Help Desk</a>
```

To add a subject for the email message, you would add `?subject=`after the email address. For example:

```
<a href="mailto:helpdesk@austincc.edu?subject=Email Assistance">Email Help Desk</a>
```

The Anchor Tag and the Name Attribute

The name attribute is used to create a named anchor. When using named anchors we can create links that can jump directly to a specific section on a page, instead of letting the user scroll around to find what he/she is looking for. Unlike an anchor that uses href, a named anchor doesn't change the appearance of the text (unless you set styles for that anchor) or indicate in any way that there is anything special about the text. Below is the syntax of a named anchor:

```
<a name="top">Text to be displayed</a>
```

To link directly to the top section, add a # sign and the name of the anchor to the end of a URL, like this:

This Code	Would Display
Back to top of page A hyperlink to the top of the page from within the file 10links.html will look like this: Back to top of page 	Back to top of page Back to top of page

Note: Always add a trailing slash to subfolder references. If you link like this: href="http://profdevtrain.austincc.edu/html", you will generate two HTTP requests to the server, because the server will add a slash to the address and create a new request like this: href="http://profdevtrain.austincc.edu/html/"

Named anchors are often used to create "table of contents" at the beginning of a large document. Each chapter within the document is given a named anchor, and links to each of these anchors are put at the top of the document. If a browser cannot find a named anchor that has been specified, it goes to the top of the document. No error occurs.

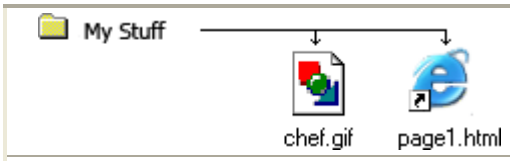
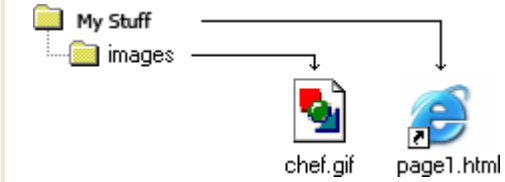
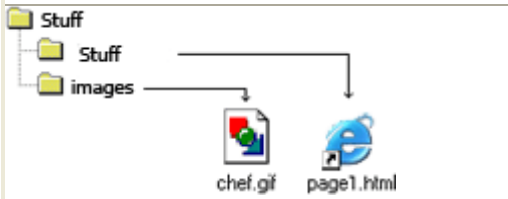
HTML Images

The Image Tag and the Src Attribute

The <img> tag is empty, which means that it contains attributes only and it has no closing tag. To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display on your page. The syntax of defining an image:

This Code	Would Display
<img src="graphics/chef.gif">	

Not only does the source attribute specify what image to use, but where the image is located. The above image, graphics/chef.gif, means that the browser will look for the image name **chef.gif** in a **graphics** folder in the same folder as the html document itself.

	`src="chef.gif"` means that the image is in the same folder as the html document calling for it.
	`src="images/chef.gif"` means that the image is one folder down from the html document that called for it. This can go on down as many layers as necessary.
	`src="../chef.gif"` means that the image is in one folder up from the html document that called for it.
	`src="../../chef.gif"` means that the image is two folders up from the html document that called for it.
	`src="../images/chef.gif"` means that the image is one folder up and then another folder down in the images directory.
	`src="../../../other/images/chef.gif"` means this goes multiple layers up.

The browser puts the image where the image tag occurs in the document. If you put an image tag between two paragraphs, the browser shows the first paragraph, then the image, and then the second paragraph.

The Alt Attribute

The alt attribute is used to define an alternate text for an image. The value of the alt attribute is author-defined text:

```
<img src="graphics/chef.gif" alt="Smiling Happy Chef ">
```

The alt attribute tells the reader what he or she is missing on a page if the browser can't load images. The browser will then display the alternate text instead of the image. It is a good practice to include the alt attribute for each image on a page, to improve the display and usefulness of your document for people who have text-only browsers or use screen readers.

Image Dimensions

When you have an image, the browser usually figures out how big the image is all by itself. If you put in the image dimensions in pixels however, the browser simply reserves a space for the image, then

loads the rest of the page. Once the entire page is loads it can go back and fill in the images. Without dimensions, when it runs into an image, the browser has to pause loading the page, load the image, then continue loading the page. The chef image would then be:

```
<img src="graphics/chef.gif" width="130" height="101" alt="Smiling Happy Chef">
```

Open the file **mypage2.html** in your text editor and add code highlighted in bold:

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body>
<h1 align="center">My First Web page</h1>
<p>Welcome to my first webpage. I am writing this page using a text editor and plain old html.</p>
<p>By learning html, I'll be able to create web pages like a pro....<br>
which I am of course.</p>
<!-- Who would have guessed how easy this would be :) -->
<p><img src="graphics/chef.gif" width="130" height="101" alt="Smiling Happy Chef"
align="center"></p>
<p align="center">This is my Chef</p>
</body>
</html>
```

Save your page as **mypage5.html** and view it in your browser. To see how your page should look visit this web page: **http://profdevtrain.austincc.edu/html/mypage5.html**

Tables

Tables are defined with the `<table>` tag. A table is divided into rows (with the `<tr>` tag), and each row is divided into data cells (with the `<td>` tag). The letters td stands for table data, which is the content of a data cell. A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.

This Code	Would Display

`<table>`
`<tr>`
`<td>row 1, cell 1</td>`
`<td>row 1, cell 2</td>`
`</tr>`
`<tr>`
`<td>row 2, cell 1</td>`
`<td>row 2, cell 2</td>`
`</tr>`
`</table>`
	row 1, cell 1 row 1, cell 2
	row 2, cell 1 row 2, cell 2

Tables and the Border Attribute

To display a table with borders, you will use the border attribute.

This Code	Would Display

`<table border="1">`
`<tr>`
`<td>Row 1, cell 1</td>`
`<td>Row 1, cell 2</td>`
`</tr>`
`</table>`
	row 1, cell 1 row 1, cell 2

and....

This Code	Would Display
<table border="5"> <tr> <td>Row 1, cell 1</td> <td>Row 1, cell 2</td> </tr> </table>	row 1, cell 1 row 1, cell 2

Open up your text editor. Type in your <html>, <head> and <body> tags. From here on I will only be writing what goes between the <body> tags. Type in the following:

```
<table border="1">
<tr>
<td>Tables can be used to layout information</td>
<td>  <img src="http://profdevtrain.austincc.edu/html/graphics/chef.gif">  
</td>
</tr>
</table>
```

Save your page as **mytable1.html** and view it in your browser. To see how your page should look visit this web page: **http://profdevtrain.austincc.edu/html/mytable1.html**

Headings in a Table

Headings in a table are defined with the <th> tag.

This code	Would Display
<table border="1"> <tr> <th>Heading</th> <th>Another Heading</th> </tr> <tr> <td>row 1, cell 1</td> <td>row 1, cell 2</td> </tr> <tr> <td>row 2, cell 1</td> <td>row 2, cell 2</td> </tr> </table>	Heading / Another Heading row 1, cell 1 / row 1, cell 2 row 2, cell 1 / row 2, cell 2

Cell Padding and Spacing

The <table> tag has two attributes known as cellspacing and cellpadding. Here is a table example without these properties. These properties may be used separately or together.

This Code	Would Display
	some text / some text some text / some text

```
<table border="1">
<tr>
<td>some text</td>
<td>some text</td>
</tr>
<tr>
<td>some text</td>
<td>some text</td>
</tr>
</table>
```

Cellspacing is the pixel width between the individual data cells in the table (The thickness of the lines making the table grid). The default is zero. If the border is set at 0, the cellspacing lines will be invisible.

This Code	Would Display

`<table border="1" cellspacing="5">`
`<tr>`
`<td>some text</td>`
`<td>some text</td>`
`</tr><tr>`
`<td>some text</td>`
`<td>some text</td>`
`</tr>`
`</table>`
	some text some text
	some text some text

Cellpadding is the pixel space between the cell contents and the cell border. The default for this property is also zero. This feature is not used often, but sometimes comes in handy when you have your borders turned on and you want the contents to be away from the border a bit for easy viewing. Cellpadding is invisible, even with the border property turned on. Cellpadding can be handled in a style sheet.

This Code	Would Display

`<table border="1" cellpadding="10">`
`<tr>`
`<td>some text</td>`
`<td>some text</td>`
`</tr><tr>`
`<td>some text</td>`
`<td>some text</td>`
`</tr>`
`</table>`
	some text some text
	some text some text

Table Tags

Tag	Description
<table>	Defines a table
<th>	Defines a table header
<tr>	Defines a table row
<td>	Defines a table cell
<caption>	Defines a table caption
<colgroup>	Defines groups of table columns
<col>	Defines the attribute values for one or more columns in a table

Table Size

Table Width

The width attribute can be used to define the width of your table. It can be defined as a fixed width or a relative width. A fixed table width is one where the width of the table is specified in pixels. For example, this code, `<table width="550">`, will produce a table that is 550 pixels wide. A relative table width is specified as a percentage of the width of the visitor's viewing window. Hence this code, `<table width="80%">`, will produce a table that occupies 80 percent of the screen.

This table width is 250 pixels

This table width is 50%

There are arguments in favor of giving your tables a relative width because such table widths yield pages that work regardless of the visitor's screen resolution. For example, a table width of 100% will always span the entire width of the browser window whether the visitor has a 800x600 display or a 1024x768 display (etc). Your visitor never needs to scroll horizontally to read your page, something that is regarded by most people as being very annoying.

HTML Layout - Using Tables

One very common practice with HTML, is to use HTML tables to format the layout of an HTML page.
A part of this page is formatted with two columns. As you can see on this page, there is a left column and a right column.
This text is displayed in the left column.

An HTML <table> is used to divide a part of this Web page into two columns.
The trick is to use a table without borders, and maybe a little extra cell-padding.
No matter how much text you add to this page, it will stay inside its column borders.

Try It Out!

Let's put everything you've learned together to create a simple page. Open your text editor and type the following text:

```
<html>
<head>
<title>My First Web Page </title>
</head>
<body>
<table width="90%" cellpadding="5" cellspacing="0" >
  <tr bgcolor="#EDDD9E">
    <td width="200" valign="top"><img src="graphics/contact.gif" width="100"
height="100"></td>
    <td valign="top"><h1 align="right">Janet Doeson</h1>
    <h3 align="right">Technical Specialist</h3></td>
  </tr>
  <tr>
    <td width="200">
      <h3>Menu</h3>
      <ul>
        <li><a href="home.html">Home</a></li>
        <li> <a href="faq.html">FAQ</a></li>
        <li> <a href="contact.html">Contact</a></li>
        <li> <a href="http://www.austincc.edu">Links</a> </li>
    </ul></td>
    <td valign="top"><h2 align="center">Welcome!</h2>
    <p>Welcome to my first webpage. I created this webpage without the assistance of a
webpage editor. Just my little text editor and a keen understanding of html.</p>
    <p>Look around. Notice I'm able to use paragraphs, lists and headings. You may not
be able to tell, but the layout is done with a table. I'm very clever. </p>
    <blockquote>
      <p>I always wanted to be somebody, but now I realize I should have been more
specific.</p>
      <cite>Lily Tomlin </cite> </blockquote>    </td>
  </tr>
</table>
<hr width="90%" align="left">
<address>
   Janet Doeson<br>
   Technical Specialist<br>
   512.555.5555
</address>
<p>Contact me at <a href="mailto:jdoeson@acme.com">jdoeson@acme.com</a> </p>
```

```
</body>
</html>
```

Save your page as **mytable2.html** and view it in your browser. To see how your page should look visit this web page: **http://profdevtrain.austincc.edu/html/mytable2.html**

I have indented some of the HTML code in the above example. Indenting the code can make your HTML document easier to read.

Create Your Own Page

It's time to create your own page. Use your text editor to create a page which contains the following:

- the required HTML page codes
- link to another web page
- an email link
- a picture/graphic
- a list of information

Save the file as **xyhtml_basics.html** where xy is your initials. Email the file to jmorales@austincc.edu.