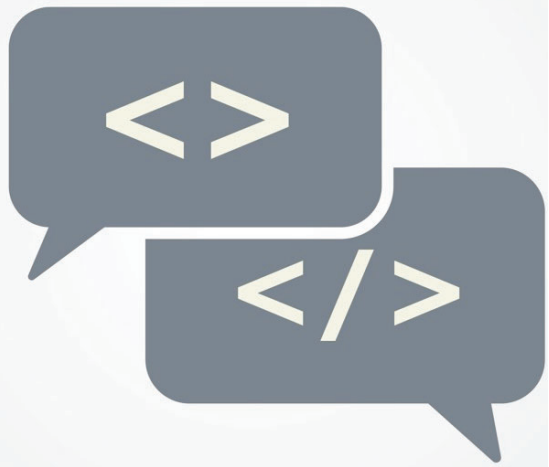
Essential HTML Tags

ESSENTIAL HTML tags are the basic structures required for every HTML file, and they control the overall structure of the webpage. The inclusion of images and links makes an HTML file more design rich. Additionally, CSS styles can be used to change color, to add font characteristics, and to size an HTML file to enhance a website.



Objective:

Apply appropriate HTML tags, anchor and img elements, inline styles, borders, and forms.

Key Terms:

alt attribute
anchor element
attribute
background property
body tag
border property
color property
CSS
element
essential HTML tags
font-family property
font property
font-size property
font-style property
font-weight property
form element
header tag
href attribute
HTML tag
img tag
inline style
letter-spacing property
line-height property
link
margins property
padding property
rgb function
span
src attribute
style attribute
SVG
tag
target
text-align property
text-decoration property
text-indent property
text-transformation property
title tag
word-spacing property

Understanding Essential HTML Tags

When working on webpages, HTML skills must be strong.

ESSENTIAL HTML TAGS

Essential HTML tags are the basic structures required for every HTML file that controls the overall structure of the webpage. AN HTML document is a text document with formatting tags to change the appearance of some parts of the pages. Through the lens of a web browser, an HTML document appears to be a word-processing document with some additional parts: pictures, audio, and/or movies. The essential tags found on all webpages to create a consistent and stable structure are HTML, header, title, and body tags.

HTML Tag

The **HTML tag** is a structure that defines the document as a webpage and identifies the beginning and the end of the HTML document. All other tags are arranged between the start and end HTML tags.

<HTML> (start tag) and </HTML> (end tag)

Header Tag

The **header tag** is a structure that contains the document information provided to search engines and browsers. It is information not viewable on the actual webpage: document title, author, keywords, style sheet type used, and any meta-tags. The header information is embedded in the HTML tags.

<HEAD> </HEAD>

Title Tag

The **title tag** is a structure that defines the name/label that appears in the title bar of the web browser (and in search engine results). A title tag does not appear on the webpage. It must be embedded between the HEAD tags.

<TITLE> </TITLE>

Body Tag

The **body tag** is a structure that includes all the visible content of the document and other visible content, including images, links, and plain text. All content must be embedded between the start and end BODY tags.

<BODY> </BODY>

TAGS, ELEMENTS, AND ATTRIBUTES

The terms "tag" and "element" refer to the markup used to write HTML. They are similar but not interchangeable terms. When you write HTML, you are writing an HTML **tag**, which is a structure that provides web browsers with instructions about the page: where to display images, how the document is structured, etc. A tag is always enclosed in angle brackets: < >.

Element

An **element** is an object on a page: a heading, paragraph, and/or an image.

Attribute

HTML elements may contain attributes. In HTML, an **attribute** is additional information or a quality (or qualities) that describe an element: width, height, etc. Working with attributes is important because some HTML elements cannot work properly until the correct attributes are set. Specifically, note the following:

- An attribute may not exist by itself. It must be part of an element, and it is always specified in the start tag of the element. [NOTE: In HTML syntax, an attribute is added to an HTML start tag.]
- End tags are not populated with attributes.
- A single element may have multiple attributes.
- Attributes are separated from one another by one or more spaces.

HTML Element	Attribute
`<h1 id = "blueHeading">`	`id="blueHeading"`
`<p title = "subject_matter">`	`title = "subject_matter"`
`<h3 align='left'>`	`align='left'`

FIGURE 1. Attribute values must be surrounded by matched double quotes or single quotes. Attribute names cannot be changed. The value placed in an attribute depends upon the application requirements.

An attribute is always specified with two parts: name and values. The first part is the name of the attribute, followed by a "=" symbol and the value to be placed in the attribute. See FIGURE 1 for details.

See FIGURE 2. It shows commonly used attributes in HTML. Note that not all elements support all attributes. For example, the src attribute is only used across the img element and a few other elements.

Attribute	Description
href	Specifies the URL (web address) for a link
hd	Specifies a unique id for an element
src	Specifies the URL for an image
style	Specifies an inline CSS style for an element
title	Specifies extra information about an element (displayed as a tool tip)
class	Specifies subtype of this element

FIGURE 2. These are examples of elements with attributes. An attribute may not exist by itself. It has to be part of an element and is always specified in the start tag of the element. End tags are not populated with attributes. A single element may have multiple attributes. Attributes are separated from one another by one or more spaces.

IMAGE FILE LOCATION AND img ELEMENT

Images enhance the look and feel of the webpage and convey information visually. Ask your teacher for a copy of VM–D. The **img tag** is a structure that places images in an HTML file. The element contains an **src attribute**, which is a structure that specifies the name of the file (along with the file extension) to be displayed. The img tag also needs an **alt attribute**, which is specific alternative text for an image that cannot be rendered. See FIGURE 3.

HTML Code	Browser View
<!DOCTYPE html> <html> <head> <meta charset="UTF-8" /> <title>Example 1</title> </head> <body> Images </body> </html>	

FIGURE 3. This is the rendering of the file in a browser window. No text is placed between the start tag and the end tag. The start and end tag can be collapsed in the start tag.

The HTML file and the image file it uses should be saved in the same folder on the computer. If this is not the case, the image file will not be found. If the image file and the HTML file are placed in different folders, relative path notation is needed in the HTML file to locate the image file. Also, the value specified in the src attribute must contain path information in addition to the file name.

ANCHOR ELEMENT

HTML files may contain links to other files or to a web URL. A key feature of the web is the ability to **link** (connect or hyperlink) resources: other webpages, locations in the current document, media files (e.g., images and video), and programs. By default, links appear in blue underlined text. When a link is clicked, the browser navigates to another HTML page. The **anchor element** is the link source and destination. An anchor is coded in HTML using the <a> element. This tag has a start and an end tag.

Hyperlink Parts

A hyperlink consists of two parts. The first part holds the target address or location. This is where the browser software goes when the hyperlink field is clicked. The **href attribute** is a part of the <a> element's start tag that maintains this target address or location. This part of the hyperlink is invisible to the user. The second part holds the text that appears in the webpage. This text

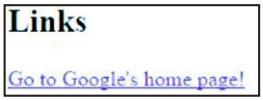
HTML Code	Browser Display
<!DOCTYPE html> <html> <head> <meta charset="UTF-8" /> <title>Example 1</title> </head> <body> Links Go to Google's home page! </body> </html>	

FIGURE 4. This is an anchor tag, along with the browser display. The text between the <a> and tags is displayed in the browser. When the text is linked, the URL specified in the href attribute will be loaded in the browser window.

is placed between the start and the end tag. The close tag is coded after the display text. See FIGURE 4.

href Attribute

The href attribute can point to an HTML file instead of a web URL. For example, the href tag can be specified as <a href = "myexample1.html">. If no path notation is provided in the href attribute, the HTML file specified in the href attribute must be in the same folder as the calling HTML file. If the target HTML file is not in the same folder as the HTML file, relative path notation must be used.

Target

The anchor element can contain an optional attribute called "target." **Target** is an attribute that specifies the location of the target URL. The default value is "_self". This loads the URL in the same browser window as the one calling it. When target is set to "_blank", the target URL is loaded in a new window or tab.

SCALABLE VECTOR GRAPHICS (SVG)

SVG is scalable vector graphics and allows for the creation of 2-D vector graphics. SVG uses XML principles, and SVG files are XML files stored with the svg extension. SVG files can be displayed in an img element, and the browser treats them in the same manner as image files. See FIGURE 5.

The img Element with svg File

HTML Code

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="UTF-8" />
      <title>Example 1</title>
   </head>
   <body>
      <h2>Images</h2>
      <img src = "caution.svg"
      alt = "Atom" height = "100" width = "100"/>
   </body>
</html>
```

View in Browser

FIGURE 5. This is an img element using an svg file in its src attribute. The image drawn by the svg file is large. To scale it, img element's width and height attributes are set.

INLINE STYLES IN HTML

It is important for you to know the options and when to use them appropriately.

CSS: Cascading Style Sheets

Cascading style sheets or **CSS** is a technology developed for HTML files to change properties—font face, size, font in tables, link color, and hover color as well as other display-related attributes. All browsers support CSS. The current version is CSS3, and it incorporates all the features of the earlier CSS2.1 version. CSS2.1 is supported in all browsers. Unfortunately, CSS3 is not fully supported in some browsers.

A collection of display-related attributes is called a style. CSS styles are composed of many individual style properties. Styles can be incorporated into an HTML file in one of three ways: within a file, in a separate file, or for a single element. Styles can be specified:

- Within a file—Styles can be set for use across the entire file.
- In a separate file—Multiple files can link to them and make use of their styles.
- Single element—Creating a style for a single element is an **inline style**. See FIGURE 6.

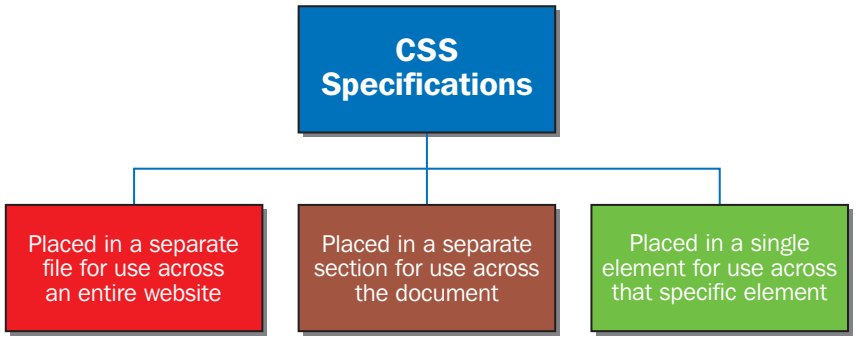


FIGURE 6. Styles can be incorporated into an HTML file in one of three ways: within a file, in a separate file, or created for a single element.

Inline styles are set up using a **style attribute**, which is a format that specifies an inline style for an element and overrides any globally set style. It is placed in the start tag of an element. See FIGURE 7.

HTML Code	Browser View
<!DOCTYPE html> <html> <head> <meta charset="UTF-8" /> <title>Example 1</title> </head> <body> Links This is a blue paragraph! </body> </html>	Links This is a blue paragraph!

CSS Declaration

CSS Property: Value

FIGURE 7. This is an HTML file with an inline CSS style applied across a element. The style is applied using the style attribute. Inside this attribute, css declarations are placed in this format: css property : property value [NOTE: In the code shown here, the color property is set to blue.]

CSS Formatting Properties

Wide arrays of styles that relate to formatting data using CSS are available.[NOTE: Use http://www.cssdesk.com/ to write CSS code and test its rendering in a browser.] When specifying multiple properties in the style attribute, "property:value " pairs are separated by the ";" symbol.

Color-Related Properties

CSS allows you to set the color for the text in an element, place a colored border around an element, and set the background color for an element. CSS supports by name all the colors supported by HTML. The 16 standard colors supported by CSS are aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow. A list of col-

ors supported by name in HTML is available online (e.g., http://www.w3schools.com/colors/colors_names.asp).

Colors can be created using the Red-Green-Blue (RGB) system. RGB notation creates colors by combining different proportions of red, green, and blue. Each of these colors can be used in the range 0 to 255. Thus, a color is specified as containing three numbers. The first number represents the proportion of red in the color. The second number represents the proportion of green, and the third number represents the proportion of blue. When all the numbers are 0 (zero), the color rendered is black. For example, rgb(0,0,0). When all the numbers are 255, the color rendered is white, such as, rgb(255,255,255). RGB numbers can be used to create a color using the rgb function. The **rgb function** is a format that sets the parameters and the color intensity. For example:

- The **color property** is a format that sets the color of the text in the element.
- The **background property** is a format that specifies the background color for the text. See FIGURE 8.

HTML Code

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="UTF-8" />
      <title>Example 1</title>
   </head>
   <body>
      <h2
         style = "color:blue;
                  background : rgb(220,220,250);"
      >
      Links</h2>
      <p>This is a paragraph!</p>
   </body>
</html>
```

Browser View

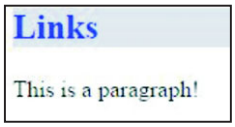


FIGURE 8. It shows code to set the text color of a heading tag to blue and the background color to light blue. The background color is specified using the rgb function. [NOTE: On http://www.colorschemer.com/online.html, numeric values are used in the rgb function and render the color that corresponds to the numbers.]

Font-Related Properties

CSS contains various font-related properties. The four main font-related properties are font family, font style, font weight, and font size. In most Windows applications, a specific font may be applied to selected text. However, in web applications, fonts are not specified by name. Instead, they are specified by families because it is hard to predict whether a specific font is present on a client computer. In CSS, the **font-family property** is a format that specifies the font for an element. When setting up font-family, generic font families or specific fonts may be used.

Generic families are:

- Serif (e.g., Times New Roman)
- Sans serif (e.g., Arial)
- Cursive (font used depends upon browser settings)
- Fantasy (font used depends upon browser settings)
- Mono-space (e.g., Courier)

Specific fonts may be specified using a family name, such as "Times New Roman."

When using the font-family property, you specify the required font followed by fallback fonts in case the required font is not found in the system. See FIGURE 9.

Font-Style Property

The **font-style property** is a format used to set up font styles for an element. It can have one of three values: normal, italic, or oblique. Visually, there is little difference between the oblique and italic settings. However, they are rendered using different techniques. The oblique setting uses the normal setting and skews it to the right. The italic setting uses the italic version of the font set. If the font does not provide an italic set, the system uses the oblique setting, even if italic is specified.

HTML Code

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="UTF-8" />
      <title>Example 1</title>
   </head>
   <body>
      <h2 style="font-family:'Caslon', serif;">
         Links
      </h2>
      <p style="font-family:monospace">
         This is a paragraph!
      </p>
      <p style="font-family:fantasy">
         This is a paragraph about
         fantasy football!
      </p
   </body>
</html>
```

Browser View in Chrome

Links

This is a paragraph!

This is a paragraph about fantasy football!

Browser View in Firefox

Links

This is a paragraph!

This is a paragraph about fantasy football!

FIGURE 9. For the h2 element, the font-family property specifies that the Caslon font be used. If the Caslon font is not found on the machine, the serif font specified in the browser's settings is to be used. The first paragraph uses a mono-space font (which defaults to the Consolas font in Chrome), and the second paragraph uses a fantasy font. The display in Firefox uses different fonts.

Font-Weight Property

The **font-weight property** is a structure used to change the thickness of the rendered font. It can hold the value normal, bold, bolder, or lighter in addition to a number that is a multiple of 100 in the range 100 to 900. The property values "lighter" and "bolder" are relative to the current property value. The property value "normal" corresponds to a numeric value of 400, while bold has a value of 700. Whether other numeric weights are supported depends upon the font.

Not all fonts support all weights. For example, the Times New Roman font only supports bold and normal weights. Even when using a font family that supports multiple weights, there is not much difference between the renderings of text in a browser. It is best to use only normal and bold as property values for the font-weight property.

Font-Size Property

The **font-size property** is a format that specifies the size of the font. The font-size property is specified by using named terms or relative terms. It may be specified in numeric terms, with an appropriate unit of measure.

- The named terms are small, medium, large, X-large, and XX-large.
- The relative terms are larger or smaller.
- When font size is specified using numeric values, it is usually specified using "px" (pixel) as the unit of measure. Size can also be specified as a percentage. When a percentage is specified, it is relative to the font size of the parent element.

Font Property

The **font property** is a structure that CSS allows for setting of all font-related properties. The font property value should be specified in this order: font-style font-weight font-size/line-height font-family. When the font property is used, font-size and font-family are required. If other values are not specified, default values are used.

Text-Related Properties

Text-related properties allow for the formatting of text characteristics. These properties are word spacing, letter spacing, text decoration, text transformation, text alignment, text indentation, and line height.

Word-Spacing Property

The **word-spacing property** is a format that specifies the amount of white space between words. The default value is 0.25em, which is 25 percent of the space taken up by the latter "m." This default value can be changed. The change may be specified in px, cm, or em.

Letter-Spacing Property

The **letter-spacing property** is a format that specifies the space between letters in a word. By default, no extra space is left between characters in a word. It may be specified as normal or in units of px, cm, or em.

Text-Decoration Property

The **text-decoration property** is a format that can be set to none, underline, over-line, or line-through.

Text-Transformation Property

The **text-transformation property** is a format that alters text in one of four ways.

- Capitalize: This capitalizes the first character of each word.

- Uppercase: This capitalizes all characters of each word.
- Lowercase: This uses small letters (non-capital letters) for all characters of each word.
- None: The initial value of the text is provided.

Text-Align Property

The **text-align property** is a format used to arrange text in an element. Possible values are left, right, center, and justify.

Text-Indent Property

The **text-indent property** is a format that defines the amount of distance or space (indentation) before the first line of the text in an element. The value is specified as a length or as a percentage and refers to the default width of the element.

Line-Height Property

The **line-height property** is a format that determines the space between lines of text when it is displayed in the browser. It is specified as a percentage or as a number with the units used in CSS for length, such as px or em.

FURTHER EXPLORATION...

ONLINE CONNECTION: Web Server Programs

The World Wide Web can be visualized as a vast network of web servers and client machines connected by communication protocols. Designing and developing HTML files is only part of the picture. It is essential to understand the inner workings of web servers that host HTML files.

Various types of web server programs exist. The type of web server program running on the web server needs to be taken into consideration when developing a website. It is also important to learn the web server environment when creating HTML forms to collect user information to send to a program on a web server.

This is a network engineer in a datacenter server room.

To find out more about web server software, visit the "Web-Server Types" link at:

http://www.tutorialspoint.com/web_developers_guide/web_server_types.htm

Border-Related Properties

CSS allows for the placement of borders around elements. The **border property** is a format that allows a person to specify the style, width, and color of an element's border. Width is specified in one of the units (in, cm, px, pt, mm). Color is specified by name or by using the rgb function. Various styles are valid for borders: solid, dashed, dotted, double, outset, inset, groove, and ridge. For example, a border is set in one declaration, such as:

border:width style color

Layout Properties

Element layout properties are specified when padding and margin properties are set. These two properties can use "in" for inches, "cm" for centimeters, "mm" for millimeters, "px" for pixels, and "pt" for points.

Margins Property

The **margins property** is a format that specifies the space to be left between elements, outside of the element border.

Padding Property

The **padding property** is a format that leaves space around an element. The size of padding is specified as a percentage of the size of the parent element or in absolute terms with units of measure. When padding is specified, it places spaces between the element and its border.

BLOCK AND INLINE ELEMENTS

Block and inline elements impact the way something is used.

Display Properties of Elements

Each HTML element has a display property. This can be a block display or an inline display. See FIGURE 10.

Inline Display

Inline are those elements that do not span a line. For example, the inline element span can be used to apply formatting for a small section of text within a paragraph. The **span** is an element with a start and an end tag.

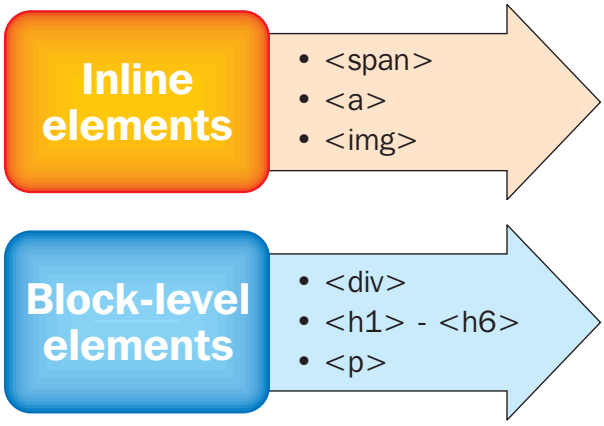


FIGURE 10. These are block and inline element categories.

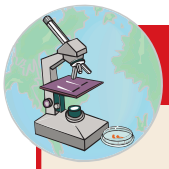
Block Display

A block element is a container element and may hold other elements inside it. Block elements take up entire lines and span the entire width of a line. Since heading elements are block level elements, the border spans the entire heading line, past the right edge of text.

- The div element is a special block-level element. It is always specified with the start and the end tag. While a div tag may be used to hold text, it is common practice to use a div tag as a container to hold other elements. The nested elements are referred to as the child elements of the div element.
- A div start tag may contain style parameters. When style properties are set for the div element, it is applied to all elements placed inside the div element. The third div element contains two child elements inside it: an h3 element and a paragraph element. The style specifications in the parent div element flow down to the child elements.

CSS Hierarchy

CSS styles can be specified for a div element and for the children element inside the div element. If this is encountered, the CSS specification closest to the element takes effect. Specifications for the paragraph element take precedence over specifications for the parent div element.



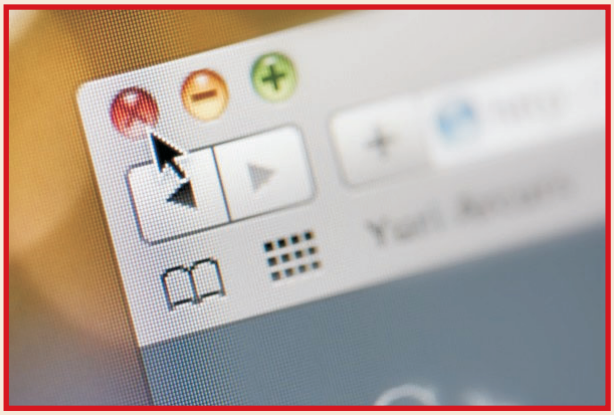
EXPLORING OUR WORLD...

UNCOVERING ADDITIONAL FACTS: Browser Wars

HTML and CSS can be used to populate and format information in an HTML file. However, it is the browser software that ultimately presents the information to the end user. Therefore, it is important to understand how browsers work.

Currently, the three main browser software applications on the market are Google Chrome, Mozilla Firefox, and Microsoft Edge/ Internet Explorer. Other browsers are Safari and Opera.

These three main browsers have been locked in a "browser war" for market share. The competition has impacted HTML standards by pushing new features and calling for greater standardization.



It is the browser software that ultimately presents the information to the end user. What is your favorite browser? What makes it your favorite?

To find out more about browser wars, visit the Elegant Themes Blog link at

https://www.elegantthemes.com/blog/editorial/the-current-state-of-the-internet-browser-wars

CREATE HTML FORMS

You must know how to create HTML forms correctly.

HTML Form Element

An HTML file displays static text and images using elements and tags. If an HTML file needs to collect information and process it, HTML forms need to be coded in the file. An HTML form simulates a paper form that collects end user information. Specialized form elements are placed in the file to collect user information. Once data entry has been completed, a button is clicked. Then the form data is sent to a program maintained on a web server. The web server program is written in PHP, VB.NET, Java, or any other web server language.

Form Element

The **form element** is a two-sided element to collect data and is always coded with a start and an end tag. Data collected in a form is sent to a single web server program. A single HTML file may contain multiple forms, with data from each form sent to a different server program.

The form start tag requires you to specify two attributes: method attribute and action attribute. The method attribute describes how user data is transferred to the server. It may have the value "get" or "post." The action specifies the name of the program on the server that receives form data. See FIGURE 11.

HTML Code

```
<!DOCTYPE html>
<html>
<head>
   <title>Trying out HTML</title>
</head>
<body>
<form method="post" action="http://www.mysite.com/checkForm.php">
.
.
.
</form>
</body>
</html>
```

FIGURE 11. The HTML code contains one logical form, represented within the <form> and </form> tags. The POST method is used to send the user data collected by this form back to the server. The data will be sent a program called checkForm.php on http://www.mysite.com.

get vs. post—Two values exist for the method attribute of the form element: "get" and "post." When "get" is used, the information from the form is appended to the URL of the web server program that receives the form data. When "post" is used, form data is sent via the HTTP request. It is not appended to the URL.

Form Fields

Form fields (form controls) can be added to an HTML file to collect information.

The input element with the type attribute set to "text" is used to collect single-line text input. Other input elements allow users to enter "yes/no" values. The select element displays a drop-down list box where one element can be selected. If no selection is made, the first element is automatically selected.

The last element placed is a form in the "submit" element. When this button is clicked, form data is collected and sent to the program whose name is specified via the action attribute of the form open tag.

CSS can be used across form elements.

Checking Form Fields

Form fields cannot be validated using HTML. They can be validated using JavaScript or by a program on a web server.

Summary:

Essential HTML tags are the basic structures required for every HTML file that controls the overall structure of the webpage. The HTML tag is a structure that defines the document as a webpage and identifies the beginning and the end of the HTML document. The header tag is a structure that contains the document information provided to search engines and browsers. In contrast, the title tag is a structure that defines the name/label that appears in the title bar of the web browser (and in search engine results). The body tag is a structure that includes all the visible content of the document and other visible content, including images, links, and plain text. An element is an object on a page (e.g., a heading, paragraph, and/or an image).

Checking Your Knowledge:

1. Why is the alt attribute used in an img start tag?
2. What happens if the URL specified in the href attribute of the anchor tag is invalid?
3. Explain the differences between font-related and text-related properties.
4. How are form elements used to collect user information?
5. Can the color of a single word in a multi-line paragraph be highlighted using CSS?

Expanding Your Knowledge:

Code an HTML table with names and pictures of your family and friends. Create a different color for family and a different color for friends.

Web Links:

Color Scheme Generator
http://www.colorschemer.com/online.html

CSSDesk
http://www.cssdesk.com/

HTML Attributes
http://www.w3schools.com/html/html_attributes.asp

HTML Color Names
http://www.w3schools.com/colors/colors_names.asp

HTML <a> Tag
http://www.w3schools.com/tags/tag_a.asp