When a web browser loads an HTML page into a browser window, it reads the HTML code, finds the image resources marked by the IMG tags, and downloads them; it then assembles the text and pictures together and displays them in the browser window. Wherever you place an IMG tag in the flow of HTML text, a picture will be inserted.

Tag: **IMG**  
Attribute: **SRC**  
Value: a URL (relative or absolute) which gives the location of the image resource.  
Description: The SRC attribute of the IMG tag states exactly where the desired picture resource is located on the Internet.

<img src="./graphics/capitalY.gif" /> (relative URL)

<img src="./capitalY.gif" /> (relative URL)

<img src="capitalY.gif" /> (relative URL)

<img src="http:/fog.ccsf.org/capitalA.gif" /> (absolute URL)

The IMG tag has two additional attributes which should ALWAYS be used: WIDTH and HEIGHT. These attributes state the width (in pixels) and the height (in pixels) of the digital image in question.

 If you forget to put WIDTH and HEIGHT into your IMG tag, the web browser may prevent the display of anything on the HTML page following that IMG tag until the entire picture is finished downloading.

You may resize an image in the browser window, on the fly, by setting the WIDTH and HEIGHT attributes of the IMG tag to values other than the actual width and height of the image. This is not recommended for most pictures, however, as browsers are extremely poor at resizing images on the fly, introducing ugly pixelations, distortions, empty gaps, or other visual flaws into the picture. The only time you'll resize pictures in this manner is when you are putting what are called "spacer" GIFs into an HTML page; because these "spacer" GIFs are just one color (white, black, some other color, or transparent),

In conclusion, the minimum complement of attributes for the IMG tag must include SRC, WIDTH, and HEIGHT.

A pixel is a square or rectangular dot of colored light. Computers use pixels to display visual content to a user; you put enough pixels together, and you have a picture. Pixels are always formed in a square grid, just like graph paper, in rows and columns.

For the Web, digital images must be created with a resolution of 72 pixels per inch. Remember this when making images in Photoshop for web-based delivery. For print, of course, 300 pixels per inch, 600 pixels per inch, and even 1200 pixels per inch resolutions are not uncommon, but images for the Web are always 72 pixels per inch.

The ALT attribute sets a text equivalent for a picture, so that users with their images turned off will still see something in the browser window where the picture should be. The ALT attribute should be always be included within the IMG tag to accomodate those with sight disabilities. Users with images turned on (normal) will also see the ALT text before the picture downloads completely. On some browsers, the ALT attribute text also pops-up as a "rollover" comment as the user mouses over the picture; this feature, however, is not universal among browsers and platforms and cannot be relied upon. Note that Firefox does not display the ALT attribute text as a pop-up "rollover" comment as the user mouses over the picture. Alternatively, in Firefox you can use the TITLE attribute to pop-up the "rollover" comment.

Tag: **IMG**  
Attribute: **ALIGN**  
Values: **left**, **right**, **top**, **middle**, **bottom**  
Description: sets the alignment of text in relation to an image.

The two most common values for ALIGN are "left" and "right", which allow content to flow around your picture very handily.

If you wish to center a graphic on a page, use the DIV tag with its ALIGN attribute set to CENTER, placed around the IMG tag.

VSPACE allows us to set the vertical space (space at the top and bottom of the IMG), in pixels, between the edge of the IMG and the beginning of text or other content. HSPACE allows us to set the horizontal space (space at the left and right of the IMG), in pixels, between the edge of the IMG and the beginning of other content.

Tag: **IMG**  
Attributes:  
**SRC** (states URL where the image resource is located)  
**WIDTH** (states the width of the image, in pixels)  
**HEIGHT** (states the height of the image, in pixels)  
**ALT** (states the alternative text for the image when images are disabled)  
**TITLE** (pops-up as a "rollover" comment as the user mouses over the picture)  
**BORDER** (states the border for the image, usually set to 0 (zero) in combination with the A (anchor) tag)  
**ALIGN** (allows text wrap around a picture when set to "left" or "right")  
**VSPACE** (creates space, in pixels, above and below the picture)  
**HSPACE** (creates space, in pixels, to the left and right of the picture)

There are two primary compression schemes for image files used on the Web today: GIF and JPEG.

GIF compression works best with images that have lots of flat color space (no gradations or continuous tones), such as drawings and flat-color graphics.

JPEG compression works best with pictures that are photographic in nature, without sharp edges, and containing lots of graduated, continuous tone color.

PNG is a bitmapped image format that employs lossless data compression. PNG was created to improve and replace the GIF format, as an image-file format not requiring a patent license. PNG is pronounced "ping".

There are two ways to express color in HTML: 1) named colors, and 2) hex codes.

 There are 16 "official" named colors: aqua, gray, navy, silver, black, green, olive, teal, blue, lime, purple, yellow, fuchsia, maroon, red, and white.

named colors have not traditionally been consistent in hue from platform to platform and browser to browser, despite official standards. Secondly, these named colors are mostly not "web-safe";

Hex codes express color precisely using the 24-bit RGB color palette, the standard for full-color reproduction on computers. Hex codes are expressed in seven digits, the first digit always being the # (pound) symbol, followed by six numbers representing the desired color.

Why are there LETTERS in some of these hex codes, and what do these numbers and letters represent?

Hex codes are written in Base 16, the hexadecimal counting system.

Base 16 uses:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10

Because Base 16, or hex, rolls over to "10" at 16 rather than 10, we need to add some letters into our numbers in order to make up the difference. This means that "A" is really 10, "B" is really 11, C is really 12, etc, until you get to "10" which is equivalent to our 16.

 24-bit color, there is one 8-bit palette with 256 shades of red, one 8-bit palette with 256 shades of green, and one 8-bit palette with 256 shades of blue; the three palettes are combined to make millions of possible colors.

Web-safe colors may only use the following digits (in hex): **00, 33, 66, 99, CC, FF**. The following randomly selected codes, then, WOULD be web-safe: **#3399CC, #6666FF, #9900FF**, etc. The following randomly selected hex codes would NOT be web-safe: **#3399A5, #1234AB, #DDDDDD**, etc.

Testing your designs in grayscale will give you a much clearer (although not completely accurate) idea of what your design will look like to the color-blind.

Don't overload your front page with content;

maintain a clear and obvious focal point for the page.

don't succumb to clutter

Maintain a clear navigational structure. There is nothing more annoying than losing your way in a website of any size, small or gargantuan, and having no clear idea of where you are or how to get anyplace else.

Use designs with strong focal areas which don't require pinpoint positioning.

complete design considers the entire user experience, from ease of access to key content and information, to arrangement of content, to navigational structures and site maps, to proper management of expectations regarding wait and delay (a primary component of most web experience)

img turned into hyper link

<a href="http://www.yahoo.com/"><img src="capitalA.gif" width="54" height="54" border="0" alt="Capital A" /></a>

an imagemap allows you to take an ordinary image and make multiple clickable hyper-references available to the user from that one image.

A shareware program can be downloaded from the Web called MapEdit, which can handle imagemap construction for you automatically. MapEdit is offered for Mac, PC, and UNIX platforms.

On the computer, locations on the coordinate plane are mapped in a slightly different manner. The origin point of the coordinate plane on the computer is always in the upper-left-hand corner of whatever is being scrutinized. If you are determining locations on the computer screen, the origin point is at the upper-left-hand corner of the screen. If you are determining locations on a web browser window, the origin point is at the upper-left-hand corner of the browser window. If you are determining locations on a digital image, the origin point is at the upper-left-hand-corner of the image itself.

On the computer, values on the y-axis increase as you move DOWN, not up; this is because the origin point is at the upper-left-hand corner of things.

When you are deriving the position and shape of clickable areas for an image, you will be doing so using these pixel-based x,y coordinates.

There are only three types of shapes which may be defined as clickable areas in an imagemap: **RECT**,**CIRCLE**, and **POLY**.

The **RECT**shape includes both rectangles and squares. It is defined by two points on the digital image: the position of the upper-left-hand corner of the rectangle, and the position of the lower-right-hand corner of the rectangle.

x,y,x1,y1

The **CIRCLE**shape includes perfect circles only; no ellipses, ovoids, or other irregular shapes are permitted. A CIRCLE shape is defined on an image by the position of the center point of the circle, followed by the radius value of the circle, in pixels;

x,y,r

The **POLY**shape includes all polygons, beginning with triangles and increasing in complexity to the most peculiar and irregular shape you can imagine. A POLY shape must have at least three x,y coordinate points defining it (a triangle), but may have as many x,y coordinate points as desired, in any shape, no matter how squiggly or peculiar.

 It doesn't matter whether you move clockwise or counter-clockwise around the polygon, just as long as you present the x,y coordinates in order. If you don't do this, you will not get the shape you want, but some ugly mess, as the computer draws the shape from first point to second point to third point, etc. You don't need to repeat the x,y coordinate for the starting point once you get to the last x,y coordinate of the polygon.

x,y,x1,y1,... xn,yn

All that you need for deriving x,y coordinates is image editing software, such as Adobe Photoshop or Paint Shop Pro.

the **IMG**tag which displays the image on the XHTML page, and the **MAP**and **AREA**tags (invisible to the user) which contain all of the imagemap information.

Because the actual imagemap information is coded only into **MAP**and **AREA**tags, the **IMG**tag needs to link to the **MAP**tag and its imagemap information using a special attribute, **USEMAP**

**MAP**opens and closes around the imagemap information, with one **AREA**tag for each clickable area on the imagemap. Here is an example of the basic structure (without attributes):

<map>

<area />

<area />

<area />

</map>

The **MAP**tag ALWAYS closes after the **AREA**statements.

The **MAP**tag has one attribute, **ID**, which is required for the imagemap to operate properly.

Tag: **MAP**  
Attribute: **NAME**  
Value: any valid name (based on the naming rules discussed in earlier modules)  
Description: the **NAME**attribute of the **MAP**tag provides an identifier which the **USEMAP**attribute of the **IMG**tag will use to connect itself to the imagemap information in the **MAP**.  
Example (abbreviated): **<map name="fred"></map>**

Note: In order for your XHTML pages to validate with the W3C standards, you need to insert an id attribute into your map statement. For example:  
  
<map name="fred" id="fred">

Tag: **AREA**  
Attribute: **SHAPE**  
Value: **rect**, **circle**, **poly**  
Description: sets the shape of the clickable area which the individual **AREA**tag will define. Each **AREA**tag in a**MAP**may be a different **SHAPE**, or they may all be the same **SHAPE**.

Tag: **AREA**  
Attribute: **COORDS**  
Value: integer **x,y**(and **r**) coordinates separated by commas; the number of integers required depends entirely on the value of the **SHAPE**attribute for that **AREA**tag.  
Description: the **COORDS**attribute of the **AREA**tag defines the **x,y**(and **r**) coordinates for the shape defined by the **SHAPE**attribute of that particular **AREA**tag.

Tag: **AREA**  
Attribute: **HREF**  
Value: any URL, relative or absolute  
Description: the **HREF**attribute of the **AREA**tag defines the hyper-reference for that clickable area in the imagemap. Note that HTML5 does not require the area tag to self-close.

Example:

<img src="demoMap.gif" width="200" height="100" alt="Demo Map" border="0" usemap="#ethel" />

<map name="ethel" id="ethel">

<area shape="rect" coords="7,7,71,71" href="destination1.html" />

<area shape="circle" coords="127,202,25" href="destination2.html" />

<area shape="poly" coords="80,247,41,187,39,131,45,111,84,124" href="destination3.html" />

</map>

**BORDER**attribute of the **IMG**tag must be set equal to **"0"**when that IMG is being used for an imagemap. Because an **IMG**tag used as an imagemap is considered to be a hyper-reference by a web browser, a blue hot-link border will appear around the image (in many browsers) unless the **BORDER**attribute is set equal to **"0"**.

 DO NOT have TWO **MAP**tags of the SAME NAME on a single XHTML page; this is not only pointless, it is forbidden.