**Imagemaps: Mapping Shapes on the Coordinate Plane**

Before we can get into details of mapping shapes, we must be certain that we are clear regarding basic graphing skills, mapping coordinates on the two-dimensional Cartesian coordinate plane.

Here is an example of what is known as the Cartesian coordinate plane:

To define coordinates on the Cartesian coordinate plane, we use the x-axis and the y-axis to determine how far from the origin point we are (the origin point being 0,0 on the diagram, where the x-axis and y-axis meet); points on the x-axis increase in value as we move to the right, while points on the y-axis increase in value as we move up. A point on the coordinate plain is always defined as "x,y", which defines the location of the point in relation to the x-axis and the y-axis. In the diagram, there are two points mapped: "2,3", which is two over on the x-axis and three up on the y-axis; and "4,1", which is four over on the x-axis and one up on the y-axis.

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.

Here is a diagram to make this point clearer (note the very large pixels that I've drawn):

As you can see, the origin (0,0) is in the upper-left-hand corner of this representation of a digital image, and marks the first pixel in the image. I have labeled three other points: "4,1" and "3,4" and "0,2" which all mark the "x,y" coordinates of the various pixels on the image.

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**. Eventually, you will be deriving x,y coordinates for these various shapes and coding those coordinates into HTML tags to create clickable imagemaps. Before you can do that, however, we need to look at the type of coordinates required to define each type of shape.

**RECT**

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. Once you've derived the positions of the upper-left-hand corner and lower-right-hand corner of the rectangle, you put the two sets of x,y coordinates together into one unit, separated by commas, like this:

x,y,x1,y1

On a theoretical image, the coordinates for a RECT with an upper-left-hand corner at "7,7" and a lower-right-hand corner at "71,71" would be:

7,7,71,71

**CIRCLE**

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; remember that the radius of a circle is 1/2 of its diameter, the diameter being the total width of the circle at its widest point (in pixels). For instance, if a circle is 40 pixels wide (in other words, if its diameter is 40 pixels), the radius value for the circle would be 20 (pixels).

Once you've figured out the radius value for the circle, the center point x,y coordinate position and the radius value would be combined, separated by commas, like this:

x,y,r

On a theoretical image, the coordinates for a CIRCLE with a center point position of "100,50" and a radius value of 25 pixels would be:

100,50,25

**POLY**

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. POLYs are used to outline irregular and bizarre shapes, such as county lines on state maps, or the picture of the edge of a book on a bookshelf, or cat's eyes, or crumpled tissue paper, or whatever.

The **POLY**shape must have its coordinates defined, in order, around the circumference of the shape. 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.

As with the **RECT**, you take all of the x,y coordinates for the **POLY**and combine them, separated by commas, like this:

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

For a theoretical triangle shape with three x,y coordinates at its corners ("5,10" and "20,30" and "10,0"), the final coordinates would be:

5,10,20,30,10,0