In this tutorial you'll learn how to create client-side image maps in HTML, and we'll touch on server-side image maps too.

**What is an image map?**
An image map is a way of defining "hot spot" links within an image on a web page. This means that, rather than having the whole image behave as one link, you can have lots of different links within the one image.

For example, the single image below has an associated image map containing 3 hot spots that, when clicked on, bring up different JavaScript messages:

Try clicking on each of the shapes and you'll see that each shape has its own link, bringing up its own JavaScript message!

**The "usemap" attribute**
So how's it done? Well, to associate an image map with an image, we simply add the usemap attribute to the standard img tag for the image. In the above example, the image map is called shapes, so our img tag looks like this:

```html
<img src="images/shapes.jpg" width="375" height="102" border="0" usemap="#shapes">
```

Note the usemap="#shapes" attribute, that associates the image map with the image.

**The "map" tag**
The other half of the image map is the map definition itself. In this definition, we tell the browser where the hot spots are in the image, and what the hot spots need to link to.

The map is defined using the `<map></map>` tag. In our example above, the map tag looks like this:
You can see that we've defined 3 'hot spot' areas - a circle, a rectangle, and a polygon - and that we've linked each of these areas to a JavaScript function to display the appropriate shape name.

The above map is placed after the image in our HTML file. In fact, it can be placed anywhere within the HTML page body.

The general syntax for the map tag is:

```
<map name="map name">
  <area shape="area shape"
       coords="area coordinates"
       href="area hyperlink" or nohref
       target="hyperlink target"
       title="area title">
  <area shape="area shape..."/>
</map>
```

So, each image map is given a name (map name), and one or more area tags to specify the hot spots in the image.

The area tag has the following attributes:
shape="rect | circle | poly | default"
Specifies the shape of the area. Possible values are:

rect (a rectangular shape),
circle (a circular shape),
poly (an arbitrary polygon, with 3 or more points), or
default (which represents the remaining area of the image not defined by any
area tags).
coords="area coordinates"
Specifies the coordinates that define the corners of the shape. The
coordinates depend on the shape specified in the shape attribute:

<table>
<thead>
<tr>
<th>Shape</th>
<th>Coordinates</th>
<th>Description</th>
</tr>
</thead>
</table>
| rect  | coords="x1,y1,x2,y2"  
(The top left and bottom right corners of the rectangle) | |
| circle| coords="x,y,r"  
(The centre and radius of the circle) | |
| poly  | coords="x1,y1,x2,y2,x3,y3,..."  
(The corners of the polygon) | |

Note that all coordinate values are relative to the top left corner of the
image. In other words, the top left corner always has coordinates (0,0).

Note also that the default shape type does not need any coordinates.

href="area hyperlink"
This is the URL that you'd like to link the hot spot to. It works just like a
standard a href tag.

Note: You can specify a nohref attribute instead, in which case the hot spot
will not link to anything.

target="hyperlink target"
This is the optional target window or frame to open the linked URL in. Again,
it works just like the target attribute in a standard a href tag.

title="area title"
This attribute allows you to give the area a title. When the mouse is rolled
over this hot spot, the browser will pop up a tool tip displaying this title.

**Putting the image map in a separate file**
The name of the image map specified in the usemap attribute is really a URI,
which means it can reference a map in another file on your website, if
required.
For example, if you saved your map tag in a file called shapes.map in the same directory as your HTML file, you would reference the map using:

```html
<img src="images/shapes.jpg" width="375" height="102" border="0" usemap="shapes.map"/>
```

**Server-side image maps**

As an alternative to defining the whole image map in HTML for the browser to read, you can use server-side image maps. With this type of map, the browser simply sends the (x,y) coordinates of the point clicked on to a server-side script (such as a CGI script).

To define a server-side map, you simply include the ismap attribute, and place an `<a href>` tag around the image, specifying the server-side script to send the (x,y) information to:

```html
<a href="shapemap.cgi">
    <img src="images/shapes.jpg" width="375" height="102" border="0" ismap>
</a>
```

Then, when you click on the image, the browser sends the (x,y) coordinate of the point that you clicked on to the server-side script, which can then interpret these (x,y) values and take an appropriate action. The coordinates are appended as parameters to the end of the script URL:

For example, if you wanted the user to choose a country from a world map image, you could use the server-side script to calculate which country was clicked on, and then display information about that country.

Another way of creating a server-side image map is with the image input type in web forms:

```html
<form action="shapemap.cgi">
    <input type="image" name="shapes_image"
```
In this case, the (x,y) coordinates are sent as form fields named fieldname.x and fieldname.y. So in the above example, the coordinates would be contained in the fields shapes_image.x and shapes_image.y.

It's best to use a server-side map whenever the map has many areas, or where the areas are not easily defined by simple shapes such as circles, rectangles and polygons.

**Handy tips for working out image map coordinates**

If you're using a web page editor such as Macromedia's Dreamweaver you can draw image maps straight onto your images and let the editor work out the coordinates, but what if you're editing your page by hand?

One easy way to work out coordinates is to change your image map from client-side to server-side temporarily, by changing the usemap="mapname" attribute to ismap, and adding a dummy `<a href>` tag around the image, e.g.:

```html
<a href="#"><img src="images/shapes.jpg" width="375" height="102" border="0" ismap></a>
```

Then, as you roll the mouse over the image, you should see the coordinates appear after the ? in the status bar of your browser! Try moving your mouse over the image below to see if this works:

![Image Map](images/shapes.jpg)

If you can't get that working, another technique is to open your image in a graphics package such as Adobe's Photoshop. You can then move the mouse over the image and see the mouse coordinates in the info palette.