HTML - Image Links

We have seen how to create hypertext link using text and we also learnt how to use images in our webpages. Now, we will learn how to use images to create hyperlinks.

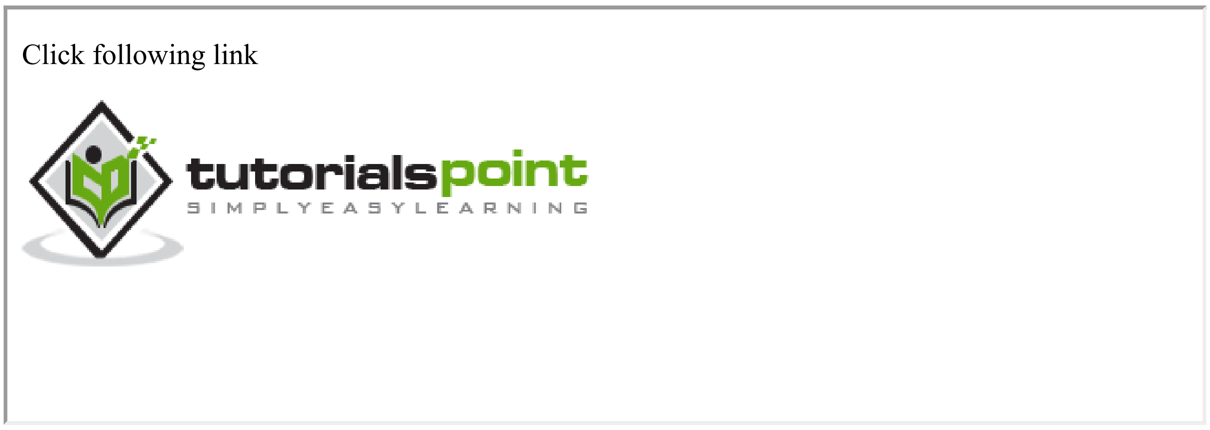
Example

It's simple to use an image as hyperlink. We just need to use an image inside hyperlink at the place of text as shown below −

Live Demo

```
<!DOCTYPE html>
<html>

   <head>
      <title>Image Hyperlink Example</title>
   </head>

   <body>
      <p>Click following link</p>
      <a href = "https://www.tutorialspoint.com" target = "_self">
         <img src = "/images/logo.png" alt = "Tutorials Point" border = "0"/>
      </a>
   </body>

</html>
```

This will produce the following result, where you can click on the images to reach to the home page of Tutorials Point.



Click following link

This was the simplest way of creating hyperlinks using images. Next we will see how we can create Mouse-Sensitive Image Links.

Mouse-Sensitive Images

The HTML and XHTML standards provides a feature that lets you embed many different links inside a single image. You can create different links on the single image based on different coordinates available on the image. Once different links are attached to different coordinates, we can click different parts of the image to open target documents. Such mouse-sensitive images are known as image maps.

There are two ways to create image maps −

- **Server-side image maps** − This is enabled by the **ismap** attribute of the <img> tag and requires access to a server and related image-map processing applications.
- **Client-side image maps** − This is created with the **usemap** attribute of the <img> tag, along with corresponding <map> and <area> tags.

Server-Side Image Maps

Here you simply put your image inside a hyper link and use **ismap** attribute which makes it special image and when the user clicks some place within the image, the browser passes the coordinates of the mouse pointer along with the URL specified in the <a> tag to the web server. The server uses the mouse-pointer coordinates to determine which document to deliver back to the browser.

When *ismap* is used, the href attribute of the containing <a> tag must contain the URL of a server application like a cgi or PHP script etc. to process the incoming request based on the passed coordinates.

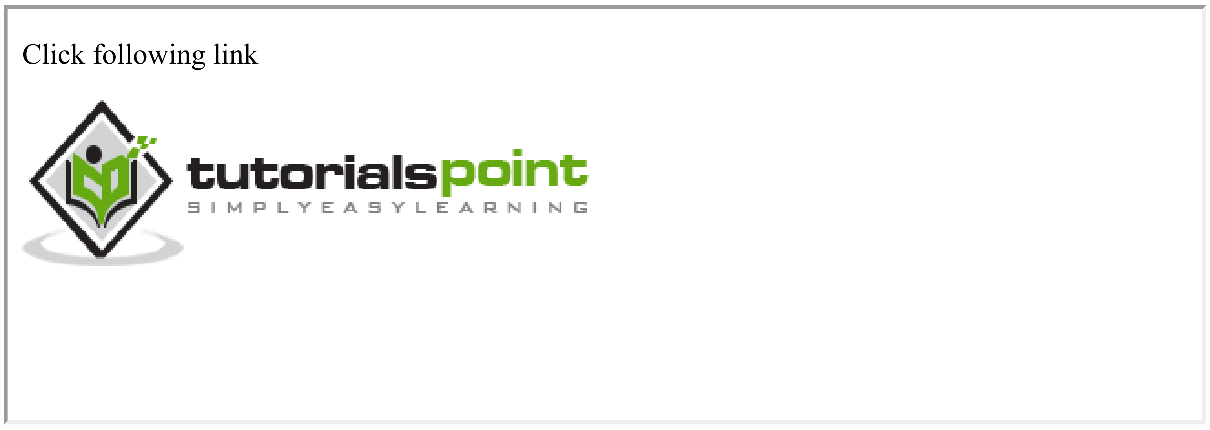
The coordinates of the mouse position are screen pixels counted from the upper-left corner of the image, beginning with (0,0). The coordinates, preceded by a question mark, are added to the end of the URL.

For example, if a user clicks 20 pixels over and 30 pixels down from the upper-left corner of the following image −

Which has been generated by the following code snippet −

Live Demo

```
<!DOCTYPE html>
<html>

   <head>
      <title>ISMAP Hyperlink Example</title>
   </head>

   <body>
      <p>Click following link</p>

      <a href = "/cgi-bin/ismap.cgi" target = "_self">
         <img ismap src = "/images/logo.png" alt = "Tutorials Point" border = "0"/>
      </a>
   </body>

</html>
```



Click following link

Then the browser sends the following search parameters to the web server which can be processed by **ismap.cgi** script or **map file** and you can link whatever documents you like to these coordinates −

```
/cgi-bin/ismap.cgi?20,30
```

This way you can assign different links to different coordinates of the image and when those coordinates are clicked, you can open corresponding linked document. To learn more about **ismap** attribute, you can check How to use Image ismap?

> **Note** − You will learn CGI programming when you will study Perl programming. You can write your script to process these passed coordinates using PHP or any other script as well. For now, let's concentrate on learning HTML and later you can revisit this section.

Client-Side Image Maps

Client side image maps are enabled by the **usemap** attribute of the <img /> tag and defined by special <map> and <area> extension tags.

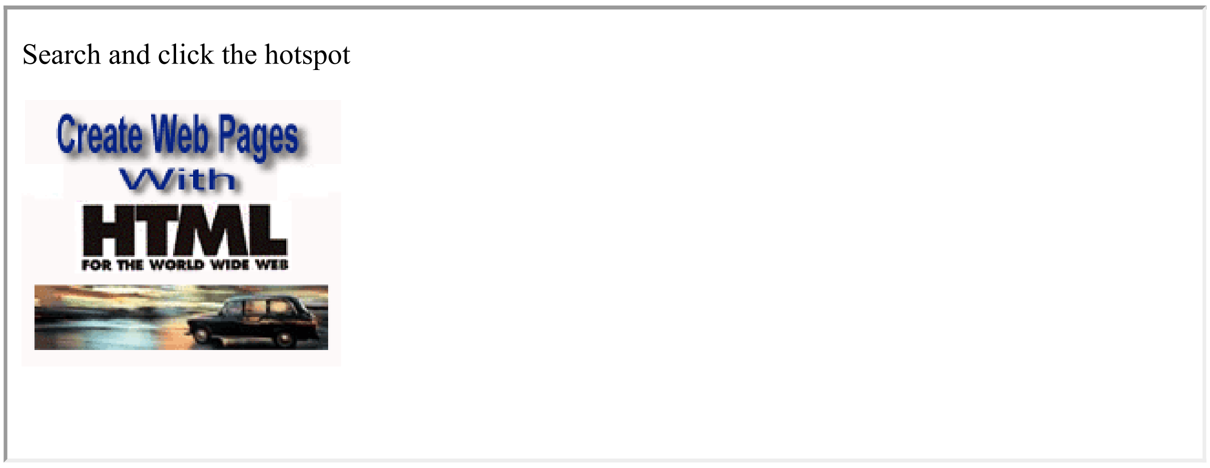
The image that is going to form the map is inserted into the page using the <img /> tag as a normal image, except it carries an extra attribute called **usemap**. The value of the usemap attribute is the value which will be used in a <map> tag to link map and image tags. The <map> along with <area> tags define all the image coordinates and corresponding links.

The <area> tag inside the map tag, specifies the shape and the coordinates to define the boundaries of each clickable hotspot available on the image. Here's an example from the image map −

Live Demo

```
<!DOCTYPE html>
<html>

   <head>
      <title>USEMAP Hyperlink Example</title>
   </head>

   <body>
      <p>Search and click the hotspot</p>
      <img src = /images/html.gif alt = "HTML Map" border = "0" usemap = "#html"/>
      <!-- Create  Mappings -->

      <map name = "html">
         <area shape = "circle" coords = "80,80,20"
            href = "/css/index.htm" alt = "CSS Link" target = "_self" />

         <area shape = "rect" coords = "5,5,40,40" alt = "jQuery Link"
            href = "/jquery/index.htm" target = "_self" />
      </map>
   </body>

</html>
```

This will produce the following result −



Search and click the hotspot

Coordinate System

The actual value of coords is totally dependent on the shape in question. Here is a summary, to be followed by detailed examples −

- **rect = x_1 , y_1 , x_2 , y_2**

  x_1 and y_1 are the coordinates of the upper left corner of the rectangle; x_2 and y_2 are the coordinates of the lower right corner.

- **circle = x_c , y_c , radius**

x_c and y_c are the coordinates of the center of the circle, and radius is the circle's radius. A circle centered at 200,50 with a radius of 25 would have the attribute *coords = "200,50,25"*

- **poly = x_1 , y_1 , x_2 , y_2 , x_3 , y_3 , ... x_n , y_n**

  The various x-y pairs define vertices (points) of the polygon, with a "line" being drawn from one point to the next point. A diamond-shaped polygon with its top point at 20,20 and 40 pixels across at its widest points would have the attribute *coords = "20,20,40,40,20,60,0,40"*.

All coordinates are relative to the upper-left corner of the image (0,0). Each shape has a related URL. You can use any image software to know the coordinates of different positions.

🖶 **Print Page**