

USING HTML ISMAP

http://www.tutorialspoint.com/html/using_html_ismap.htm

Copyright © tutorialspoint.com

To use an image with ismap attribute, 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 50 pixels over and 30 pixels down from the upper-left corner of the following image:

Click following link



Which has been generated by the following code snippet:

```
<!DOCTYPE html>
<html>
<head>
<title>ISMAP Hyperlink Example</title>
</head>
<body>
<p>Click following link</p>
<a href="/cgi-bin/ismap.cgi" target="_self">
  
</a>
</body>
</html>
```

Then the browser sends the following search parameters to the web server:

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

Now these passed coordinates can be processed by one of the following two ways:

- Using a CGI or PHP file if you are not using CGI file
- Using a map file

A CGI or PHP file

Following is the perl code for **ismap.cgi** script which is being used in the above example:

```
#!/usr/bin/perl

local ($buffer, $x, $y);
# Read in text
$ENV{'REQUEST_METHOD'} =~ tr/a-z/A-Z/;
if ($ENV{'REQUEST_METHOD'} eq "GET")
```

```

{
    $buffer = $ENV{'QUERY_STRING'};
}
# Split information into name/value pairs
($x, $y) = split(/,/, $buffer);

print "Content-type:text/html\r\n\r\n";
print "<html>";
print "<head>";
print "<title>ISMAP</title>";
print "</head>";
print "<body>";
print "<h2>Passed Parameters are : X = $x, Y = $y </h2>";
print "</body>";
print "</html>";

1;

```

Because you are able to parse passed coordinates, you can put a list of **if conditions** to check passed coordinates and send appropriate linked document back to the browser.

A map file

A map file can be used to store the location of HTML files that you want the reader to be taken to when the area between the identified coordinates is "clicked."

You keep default file at the first location and other files are put corresponding to various coordinates as shown below in **ismap.map** file.

```

default http://www.tutorialspoint.com
rect    http://www.tutorialspoint.com/html 5,5 64,141
rect    http://www.tutorialspoint.com/css 91,5 127,196
circle  http://www.tutorialspoint.com/javascript 154,150,59

```

This way you can assign different links to different part of the image and when those coordinates are clicked, you can open linked document. So let's rewrite above example using **ismap.map** file:

```

<!DOCTYPE html>
<html>
<head>
<title>ISMAP Hyperlink Example</title>
</head>
<body>
<p>Click following link</p>
<a href="/html/ismap.map" target="_self">
    
</a>
</body>
</html>

```

Before trying above example, you need to make sure that your webserver has required configuration to support image map file.

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. You can use any available tool like Adobe Photoshop or MS Paint to detect various coordinates available on the image to be used for ISMAP.

A lines beginning with # are comments. Every other non-blank line consists of the following:

rect = x₁ , y₁ , x₂ , y₂

x₁ and y₁ are the coordinates of the upper left corner of the rectangle; x₂ and y₂ 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.

Loading [MathJax]/jax/output/HTML-CSS/jax.js