

5. Introduction to HTML: Part 2

Tables

Attributes of the `table` element:

`border`: width of the table border in pixels (0 makes all lines invisible)

`align`: horizontal alignment (`left`, `center`, or `right`)

`width`: just as in the `hr` element (given in pixels or percentage of screen)

A `caption` element is inserted directly above the table.

A table has

- a head: `<thead>`
- a body: `<tbody>`

Each contains zero or more `tr` (table row) elements.

Each `tr` element in the head (resp., body) contains zero or more `th` (resp., `td`) elements (i.e., cells).

Each of these kinds of elements can have the usual attributes.

Example

```
<table border = "1" align = "center" width = "40%">
<caption>Example Table</caption>
<thead>
  <tr><th>Column 1</th><th>Column 2</th><tr>
</thead>
<tbody>
  <tr align = "center"><td>Cell 11</td>
                                <td>Cell 12</td></tr>
  <tr><td>Cell 21</td>
      <td align = "center">Cell 22</td></tr>
</tbody>
</table>
```

Example Table

Column 1	Column 2
Cell 11	Cell 12
Cell 21	Cell 22

The `colgroup` element (before the `thead` element) lets us format groups of columns.

It contains `col` elements, each a single tag with attributes affecting an adjacent group of columns.

`span`: the number of columns in the group.

Example:

```
<colgroup>
  <col span = "2" align = "left">
  <col span = "3" align = "center">
</colgroup>
```

The `valign` (vertical alignment) attribute (of any element in a table) accepts values `top`, `middle` (default), `bottom`, and `baseline`.

All cells in a row whose `valign` attribute is `baseline` have their first text line on a common baseline.

For a cell:

`rowspan`: number of rows it occupies (default = 1)

`colspan`: number of columns it occupies (default = 1)

Example

```
<td rowspan = "2" colspan = "2" valign = "top">
...
</td>
```

Example:

```

<table border = "1" align = "center" width = "100%">
<caption>A Complex Table</caption>
<colgroup>
  <col span = "2" align = "left">
  <col span = "3" align = "center">
</colgroup>
<thead>
<tr><th rowspan = "2" colspan = "2">
  *** Head A ***</th>
  <th colspan = "3">Head B</th></tr>
<tr><th>Head C1</th>
  <th>Head C2</th>
  <th>Head C3</th></tr>
</thead>
<tbody>
<tr><td>D11</td><td>D12</td><td>C11</td>
  <td>C12</td><td>C13</td></tr>
<tr><td>D21</td><td>D22</td>
  <td rowspan = "2" colspan = "2"
    valign = "top">CCC</td>
  <td>C23</td>
</tr>
<tr><td>D31</td><td>D32</td><td>C33</td>
<tr><td>D41</td><td>D42</td><td>C41</td>
  <td>C42</td><td>C43</td></tr>
</tbody>
</table>

```

A Complex Table

*** Head A ***		Head B		
		Head C1	Head C2	Head C3
D11	D12	C11	C12	C13
D21	D22	CCC		C23
D31	D32			C33
D41	D42	C41	C42	C43

Internal Linking

Internal linking lets you assign a location name to a point within an XHTML file.

Adding the location name to the page's URL gives an external reference to exactly that point in the page (as opposed to its top).

The location name on its own can be used within the page as a reference to that point.

To establish an internal hyperlink at a point, place there an element

```
<a name = "name" ></a>
```

where *name* becomes the location name for that point.

If *filename* is the name of the page, then the URL of this point is

```
filename.html#name
```

Insert a regular hyperlink within the same page to this point with

```
<a href = "#name" > ... </a>
```

Clicking on this hyperlink scrolls the browser window to the given point on the page.

Example:

```
<p><a href = "#commonTrees">
    Go to common trees</a></p>
<p><a href = "#commonShrubs">
    Go to common shrubs</a></p>

<a name = "commonTrees"></a>
<h2>Some Common Trees</h2>
<ul>
    <li>Elm</li>
    <li>Oak</li>
    <li>Ash</li>
</ul>

<a name = "commonShrubs"></a>
<h2>Some Common Shrubs</h2>
<ul>
    <li>Azalea</li>
    <li>Box</li>
    <li>Forsythea</li>
</ul>
```

[Go to common trees](#)

[Go to common shrubs](#)

Some Common Trees

- Elm
- Oak
- Ash

Some Common Shrubs

- Azalea
- Box
- Forsythea

Image Maps

An *image map* allows certain image sections to be designated *hotspots*, used as anchors for linking.

Image map elements (the `area` elements) are inside a `map` element with required attribute name.

To use the image map, use the

```
usemap = "#name"
```

attribute value assignment in the `img` element, where

name

is the value of the `name` attribute in the `map` element.

A hotspot is designated with the `area` element, with attributes

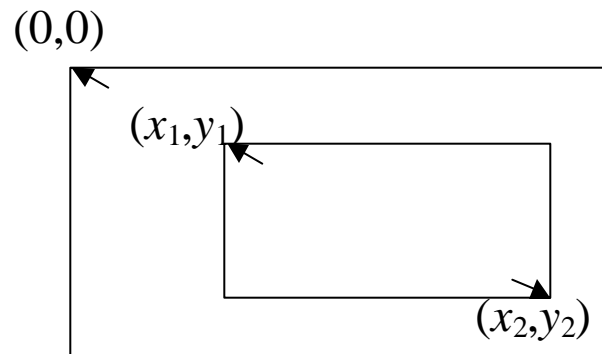
`href`: the target for the link

`shape, coords`: the characteristics of the area

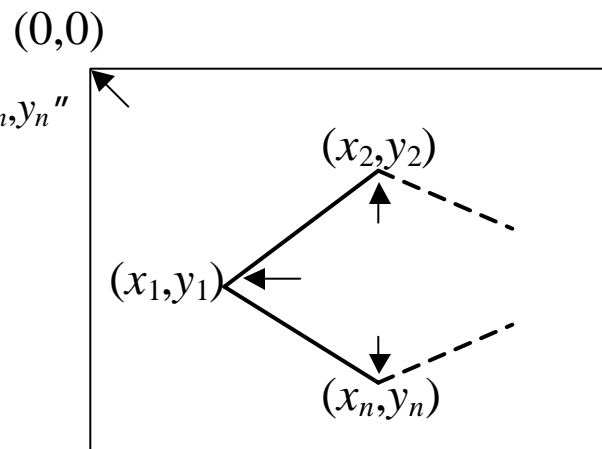
`alt`: text to display if the image isn't displayed (as with `img`)

Possible Shapes

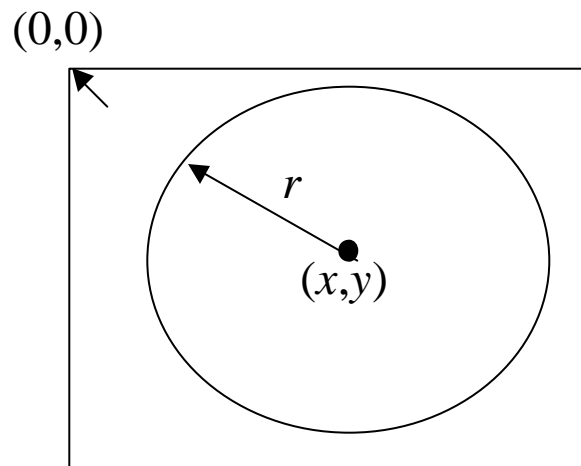
If shape = "rect" then
coords = "x₁,y₁,x₂,y₂"



If shape = "poly" then
coords = "x₁,y₁,x₂,y₂,...,x_n,y_n"



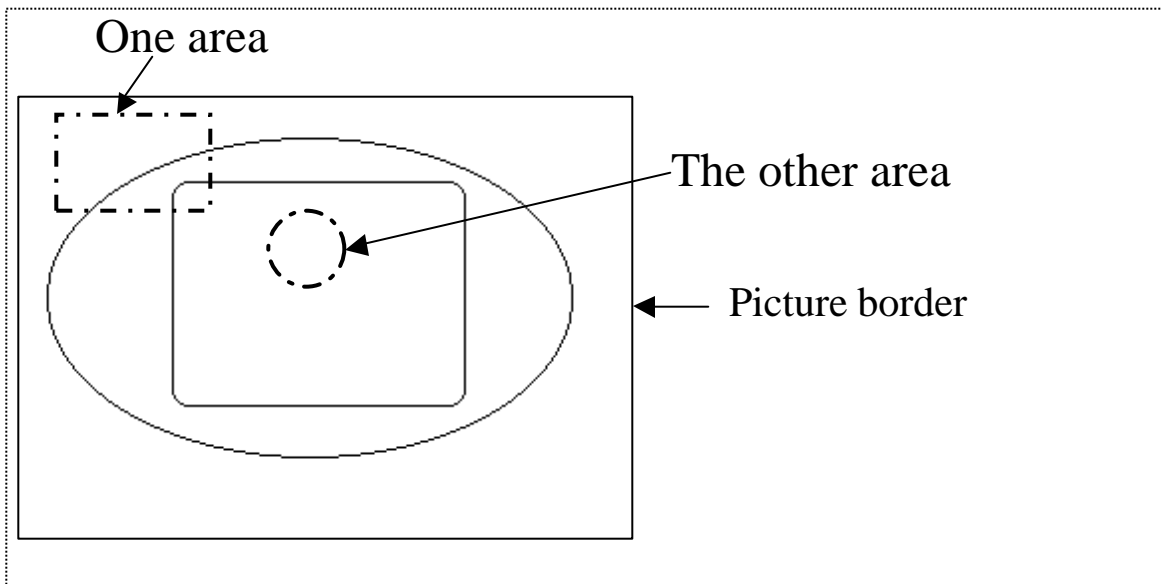
If shape = "circle" then
coords = "x, y, r"



Example:

```
<map name = "picture">
<area href = "f1.html" shape = "rect"
      coords = "10, 10, 100, 50"
      alt = "Go to f1">
<area href = "f2.html" shape = "circle"
      coords = "145, 80, 20" alt = "go to f2">
</map>

<img src = "bmex.bmp" width = "300" height = "200"
      border = "1" alt = "map Example"
      usemap = "#picture">
```



meta Tags

The main HTML element that interacts with search engines is the meta tag (in the header) with attributes

name: keywords or description
content

Example:

```
<head>
<meta name = "keywords"
      content = "Web page, XHTML, examples">
<meta name = "description"
      content = "This Web site contains simple
                Examples of using XHTML.">
<title>A Simple Example of XHTML</title>
</head>
```

The frameset Element

Frames let you display more than one XHTML file at a time.

Use a `frameset` element instead of a `body`.

The `frameset` element has two alternative attributes:

`cols`: width of each frame, left to right (with vertical frames)

`rows`: height of each frame, top to bottom (with horizontal frames)

The format for the values of these attributes is

`size1, size2, ..., sizen,`

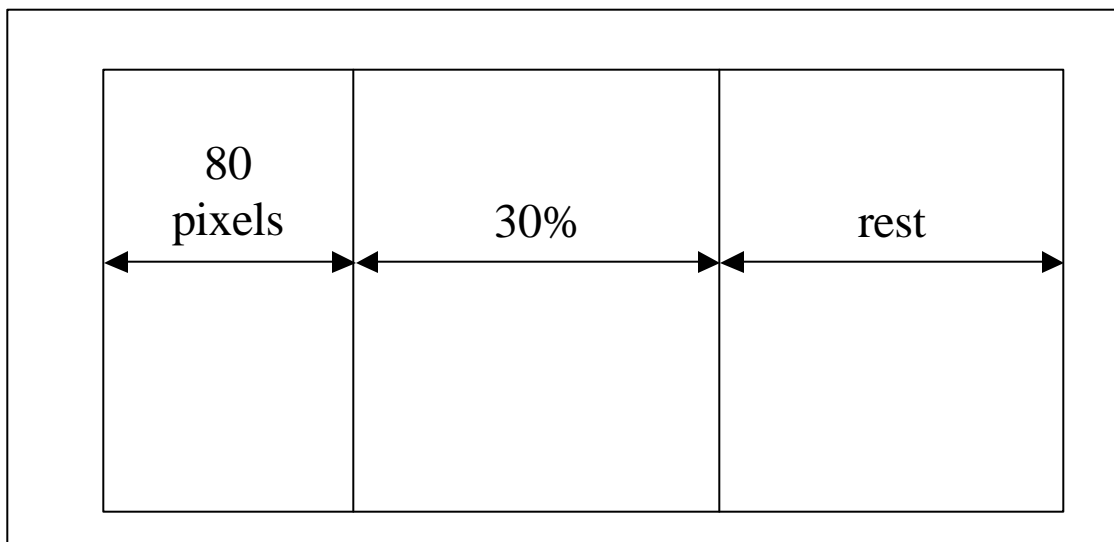
where

`sizei` is in pixels or percentage of the screen, and

`sizen` can be `*`, indicating “the rest of the screen”.

Example:

```
<frameset cols = "80, 30%, *">
```



The individual frames in a frameset are `frame` elements (no closing tag) with attributes

`name`: identifies the frame into which a hyperlink is to be loaded

`src`: the URL of the page displayed in the frame

The `noframes` element displays text in browsers not supporting frames.

In the file that is loaded into a frame, an anchor

```
<a href = "filename" target = "name" >
```

loads the file with name *filename* into the frame whose name attribute is *name*.

There are predefined values for the `target` attribute:

`_blank`: load the page into a new browser window (leaving the current window unchanged)

`_self`: load the page into the same frame where the page containing the anchor has been loaded

`_top`: load the page into the full browser (overwriting the frameset)

Example:

```
<frameset rows = "80, 30%, *">  
  <frame name = "f1" src = "f1.html">  
  <frame name = "f2" src = "f2.html">  
  <frame name = "f3" src = "f3.html">  
  <noframes>  
    <p>Your browser cannot render frames.</p>  
  </noframes>  
</frameset>
```

The bodies of f1.html – f4.html are, respectively:

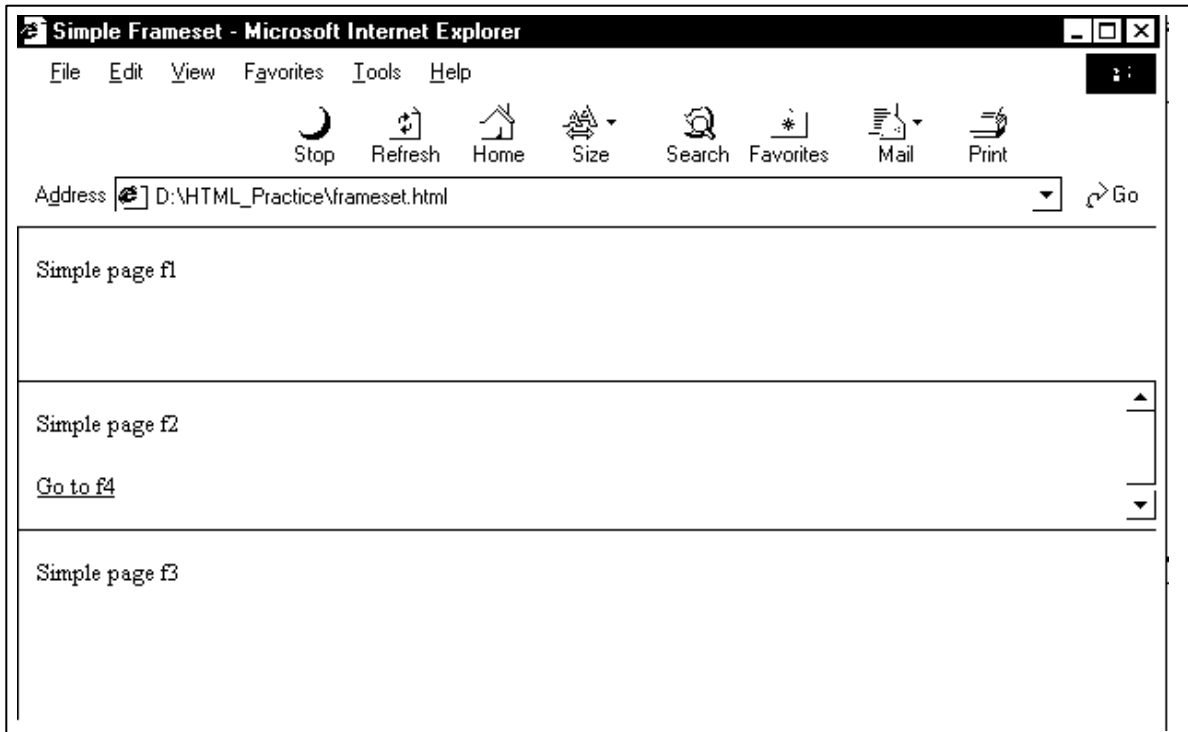
```
<p>Simple page f1</p>
```

```
<p>Simple page f2</p>  
<p><a href = "f4.html" target = "f1">  
  Go to f4</a>  
</p>
```

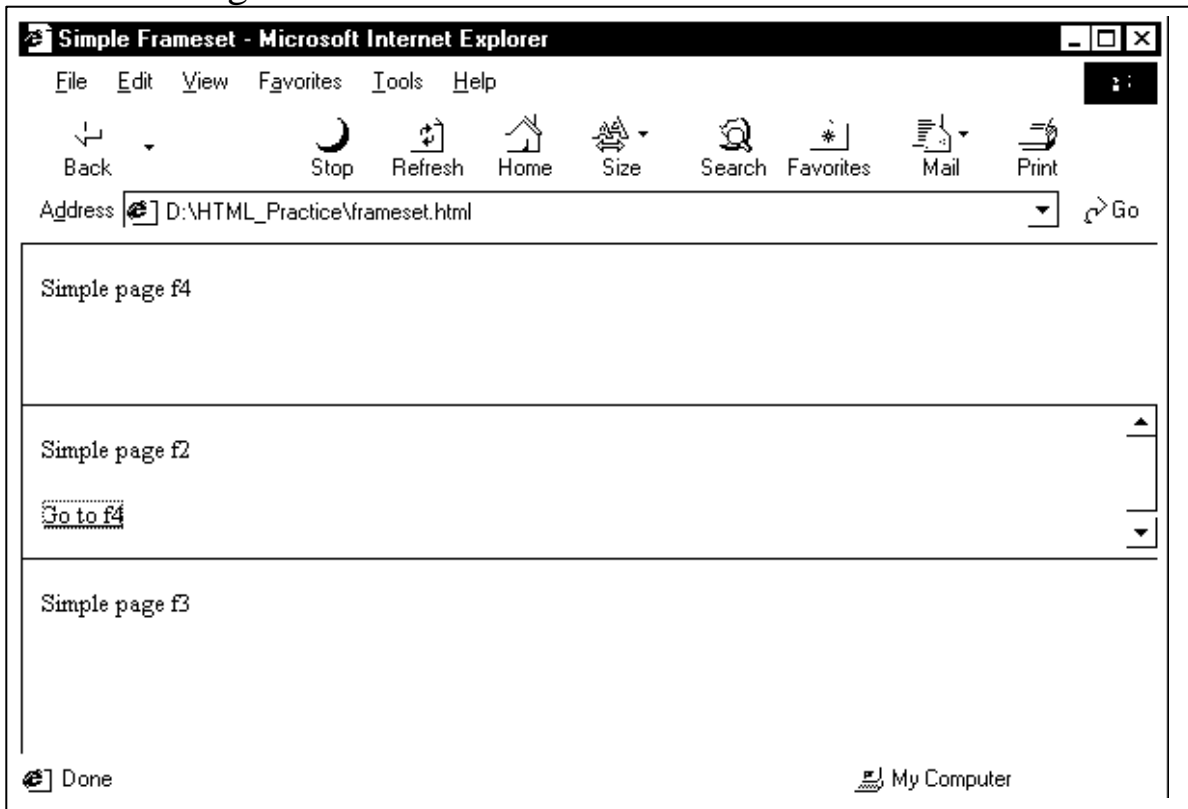
```
<p>Simple page f3</p>
```

```
<p>Simple page f4</p>
```

When the page is loaded, we get:



After clicking on "Go to f4":



Framesets can be nested.

Another predefined value for the `target` attribute on an anchor in a loaded page that makes sense here is

`_parent`: load the page into the parent frameset (enclosing the current frameset).

Example:

```
<frameset rows = "80, *">
  <frameset cols = "80, *">
    <frame name = "f1" src = "f1.html">
    <frame name = "f2" src = "f2.html">
  </frameset>
  <frameset cols = "80, *">
    <frame name = "f3" src = "f3.html">
    <frame name = "f5" src = "f5.html">
  </frameset>
</frameset>
<noframes>
  <p> Your browser cannot render frames.</p>
</noframes>
</frameset>
```

Simple page f1	Simple page f2 Go to f4
Simple page f3	Simple page f5

Other frame attributes:

`scrolling`: when "no", suppresses scroll bars on the frame

`noresize` (no value): prevents the user from resizing the frame with the mouse.

Example:

```
<frame name = "f1" scrolling = "no"  
        noresize src = "f1.html">
```