HTML

Dr. L.M. Jenila Livingston

VIT Chennai

HTML

- Short for ***H**yper **T**ext **M**arkup **L**anguage,* the language used to create documents on the World Wide Web.

- HTML was invented by Tim Berners-Lee while at CERN, the European Laboratory for Particle Physics in Geneva.

HTML Versions

Version	Year
HTML	1991
HTML+	1993
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2012

HTML

- Hyper Text Markup Language
  - Hypertext – Dealing with links
  - Markup – Designing documents with some tags, styles or lay outs

Tags

- A tag is a unit of markup, start with a less-than sign (<) followed by a keyword and concluded or ended with a greate-then sign (>).
- The symbol great-than and less-than are called angle brackets.

HTML Creation:
What do you need?

- Editor (Notepad)
- Type contents and save with a .htm or .html extension
  - Eg. MyPage.html
- Browser to view results
- Deployment: Place to put your web site on the internet (store the files that contain the HTML code, images, video, music, etc.)
- URL for your website

HTML Elements

An HTML element usually consists of a **start** tag and **end** tag, with the content inserted in between:

<tagname>Content**</tagname>**

The HTML document begins with <html> and ends with </html>.

HTML Tags

- Paired Tags
- Unpaired Tags

Paired Tags

- A tag is said to be a **paired tag** if the text is placed between a tag and its companion tag.
- In paired tags, the first tag is referred to as *Opening Tag* and the second tag is referred to as *Closing Tag.*
- Example of Paired Tags in HTML:

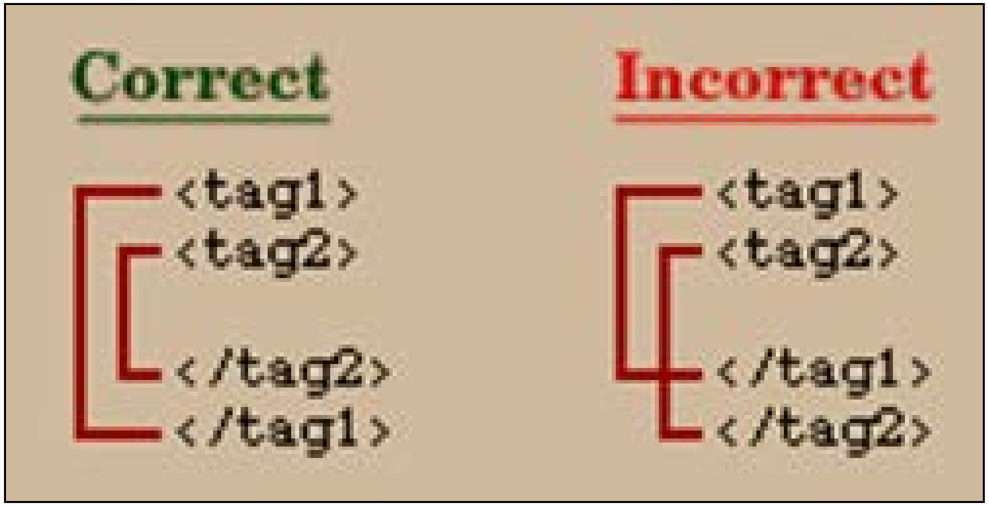
<p>Paragraph</p>

Unpaired Tags

- An **unpaired tag** does not have a companion tag. Unpaired tags are also known as *Singular tags* or *Stand-Alone* Tags.
- Example of Unpaired Tags in HTML:
- <br> <img src='test.png' >

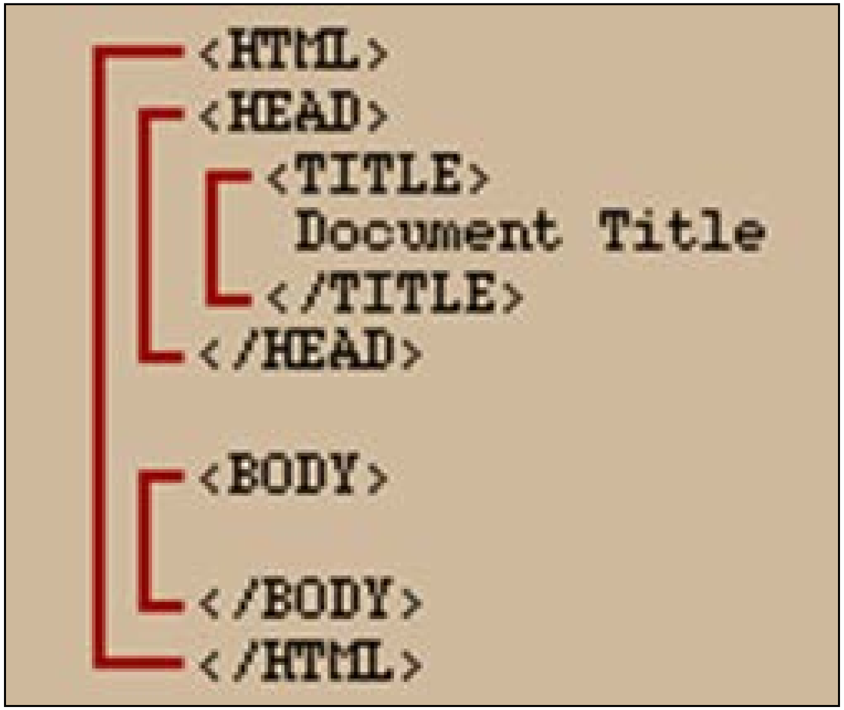
HTML tags

- Case insensitive
- Unrecognised tags are simply ignored by browser!!
- Container **tags must be nested**!!



Structure of HTML document

- Basic structure:



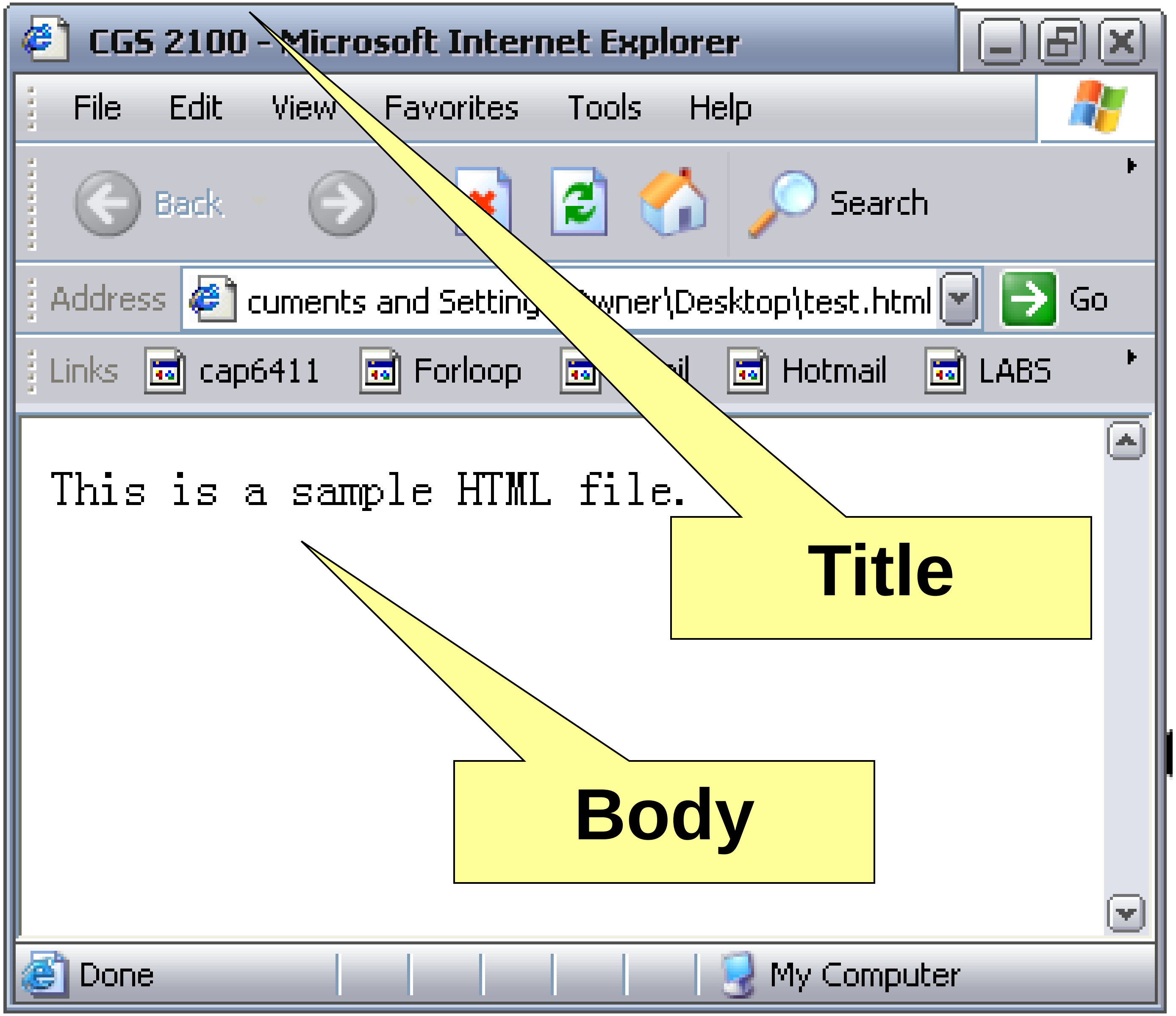
```
<HTML>
<HEAD>
  <TITLE>
   Document Title
  </TITLE>
</HEAD>

<BODY>

</BODY>
</HTML>
```

Structure of the web page

- Inside the <html></html> tag
  - Each web page has a **body** part described in <body></body> tag:

```
<!-- This is a comment -->
  <!DOCTYPE html>              Declares the document as HTML5
<html>
    <head>
       <title> CGS 2100 </title>
    </head>
     <body>
           This is a sample HTML file.
     </body>
</html>
```

The content of the whole web page should be put here



This is a sample HTML file.

Title

Body

HTML <!--Type your comment-->

- <!--This is a comment. Comments are not displayed in the browser-->

- <!-- Do not view the picture at this moment
<img border=“2" src=“orange.jpg"
alt=“Orange">
-->

Body Tag - Attributes

Attribute	Value	Description
background	*URL*	Specifies a background image for a document
bgcolor	*color*	Specifies the background color of a document
text	*color*	Specifies the color of the text in a document
link	*color*	Specifies the color of unvisited links in a document
vlink	*color*	Specifies the color of visited links in a document
alink	*color*	Specifies the color of an active link in a document

Obsolete in HTML5, use CSS instead

Body tag attributes - events

onafterprint	Function to call user has printed document
onbeforeprint	Function to call when user requests document to be printed
onbeforeunload	Funtion to call when document is to be unloaded
onblur	Function to call when document has lost focus
onerror	Function to call when document fails
onfocus	Function to call when document has focus
onhaschange	Function to call when fragment identifier portion of document's address has changed
onload	Function to call when document has loaded
onmessage	Function to call when the document received a message
onoffline	Function to call when Network communication fails
ononline	Function to call when Network communication is restored
onpopstate	Function to call when user navigated session history
onredo	Function to call when user moved forward in undo history
onresize	Function to call when document was resized
onstorage	Function to call when storage area changed
onundo	Function to call when user moved backward in undo history
onunload	Function to call when document is being unloaded

etc.).

Coding Conventions

- Declare Document Type
  - Start with the <!DOCTYPE html> declaration to specify the HTML document
- Include the required elements: <html>, <head>, and <body>.
  - Omitting `<html>` and `<body>` can also crash DOM and XML software
  - Omitting `<body>` can produce errors in older browsers.
- Never Skip the <title> Element
- Use Lowercase Element Names
  - Mixing uppercase and lowercase names looks bad
  - Developers normally use lowercase as it looks cleaner
- Close All HTML Elements even you do not have to close
- Add comments to explain sections or complex logic.

etc.).

Coding Conventions

- Always Quote Attribute/Property values

  -
    ```
    <a href="https://www.mypage.com/html/">Visit our HTML tutorial</a>
    ```
- Always Specify alt, width, and height for Images
  - This reduces flickering, because the browser can reserve space for the image before loading.

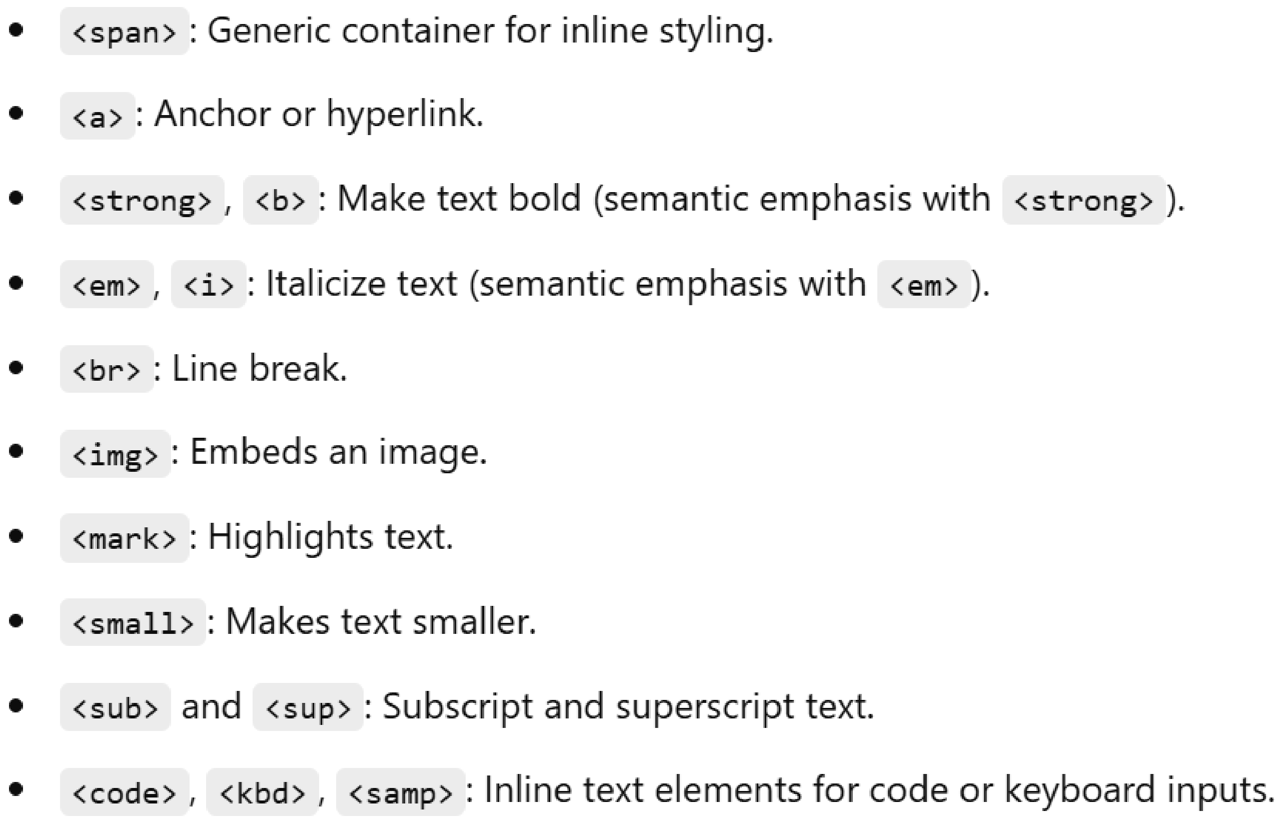
  -
    ```
    <img src=“apple.gif" alt=“APPLE" style="width:128px;height:128px">
    ```
- Do not add blank lines, spaces, or indentations without a reason.
- Use semantic tags where applicable (<header>, <footer>, <article>, <section>, etc
  - Use **semantic tags** to convey the meaning of content.
  - Avoid using non-semantic tags like <div> or <span> unless necessary

HTML

- Text Elements
- Block Elements
- Code-Related Elements
- Character References

Text Elements

- Inline elements do not start on a new line and they typically affect text or content styling and structure.



- `<span>` : Generic container for inline styling.
- `<a>` : Anchor or hyperlink.
- `<strong>` , `<b>` : Make text bold (semantic emphasis with `<strong>`).
- `<em>` , `<i>` : Italicize text (semantic emphasis with `<em>`).
- `<br>` : Line break.
- `<img>` : Embeds an image.
- `<mark>` : Highlights text.
- `<small>` : Makes text smaller.
- `<sub>` and `<sup>` : Subscript and superscript text.
- `<code>` , `<kbd>` , `<samp>` : Inline text elements for code or keyboard inputs.

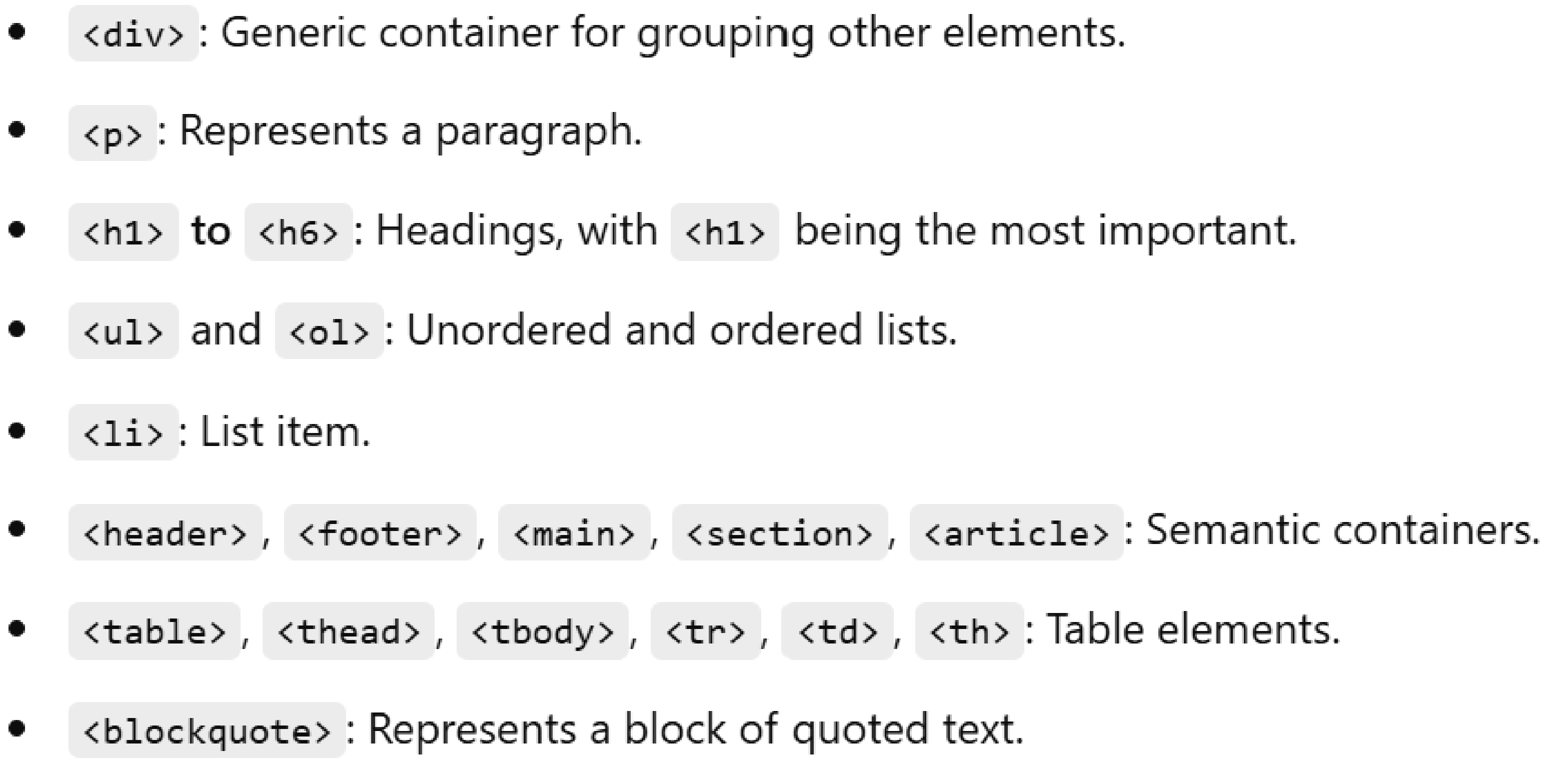
Text Formatting Tags

Tag	Description
<b>	Defines bold text **This text is bold.**
<big>	Defines big text
<em>	Defines emphasized text *This text is emphasized.*
<i>	Defines italic text *This text is italic.*
<mark>	displayed as marked with yellow ink *This text is italic.* marked
<small>	Defines small text
<strong>	Defines strong text **This text is strong.**

Tag	Description
<sub>	**Defines subscripted text** This is $_{\text{subscripted}}$ text.
<sup>	**Defines superscripted text** This is $^{\text{superscripted}}$ text.
<ins>	**Defines inserted text** text
<del>	**Defines deleted text** ~~blue~~
<strike>	**Deprecated. Use <del> instead**
<u>	**Deprecated. Use <ins> instead**

Block Elements

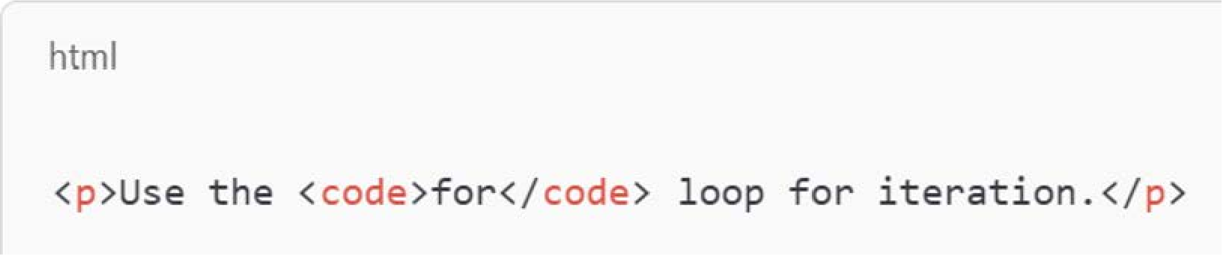
Block elements occupy the full width of their parent container by default and typically start on a new line.



- `<div>`: Generic container for grouping other elements.
- `<p>`: Represents a paragraph.
- `<h1>` to `<h6>`: Headings, with `<h1>` being the most important.
- `<ul>` and `<ol>`: Unordered and ordered lists.
- `<li>`: List item.
- `<header>`, `<footer>`, `<main>`, `<section>`, `<article>`: Semantic containers.
- `<table>`, `<thead>`, `<tbody>`, `<tr>`, `<td>`, `<th>`: Table elements.
- `<blockquote>`: Represents a block of quoted text.

Code-Related Elements

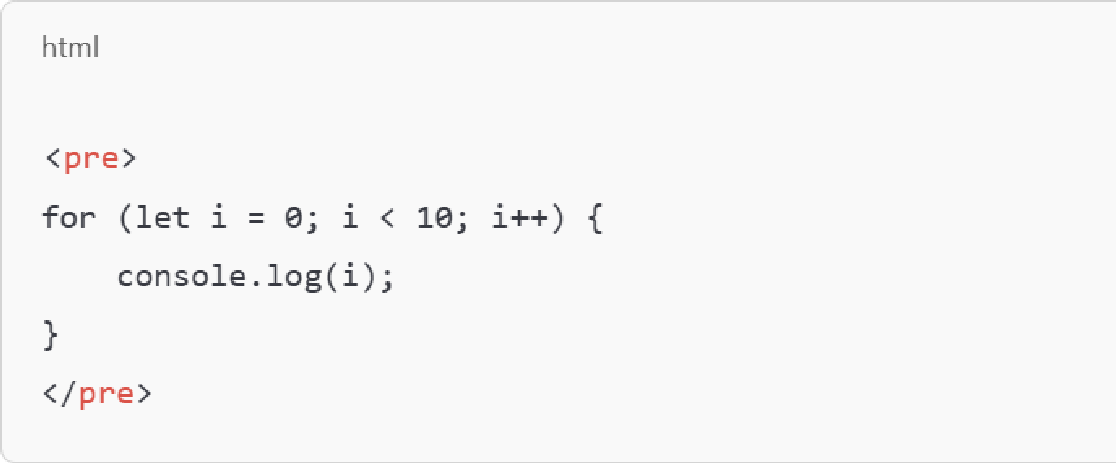
- These elements are designed to display code snippets, keyboard inputs, or output in a readable format.
- `<code>` : Represents inline code.



html

```html
<p>Use the <code>for</code> loop for iteration.</p>
```

- `<pre>` : Displays preformatted text with preserved whitespace.

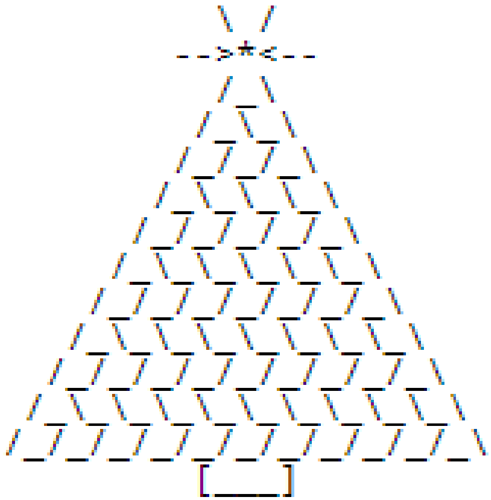


html

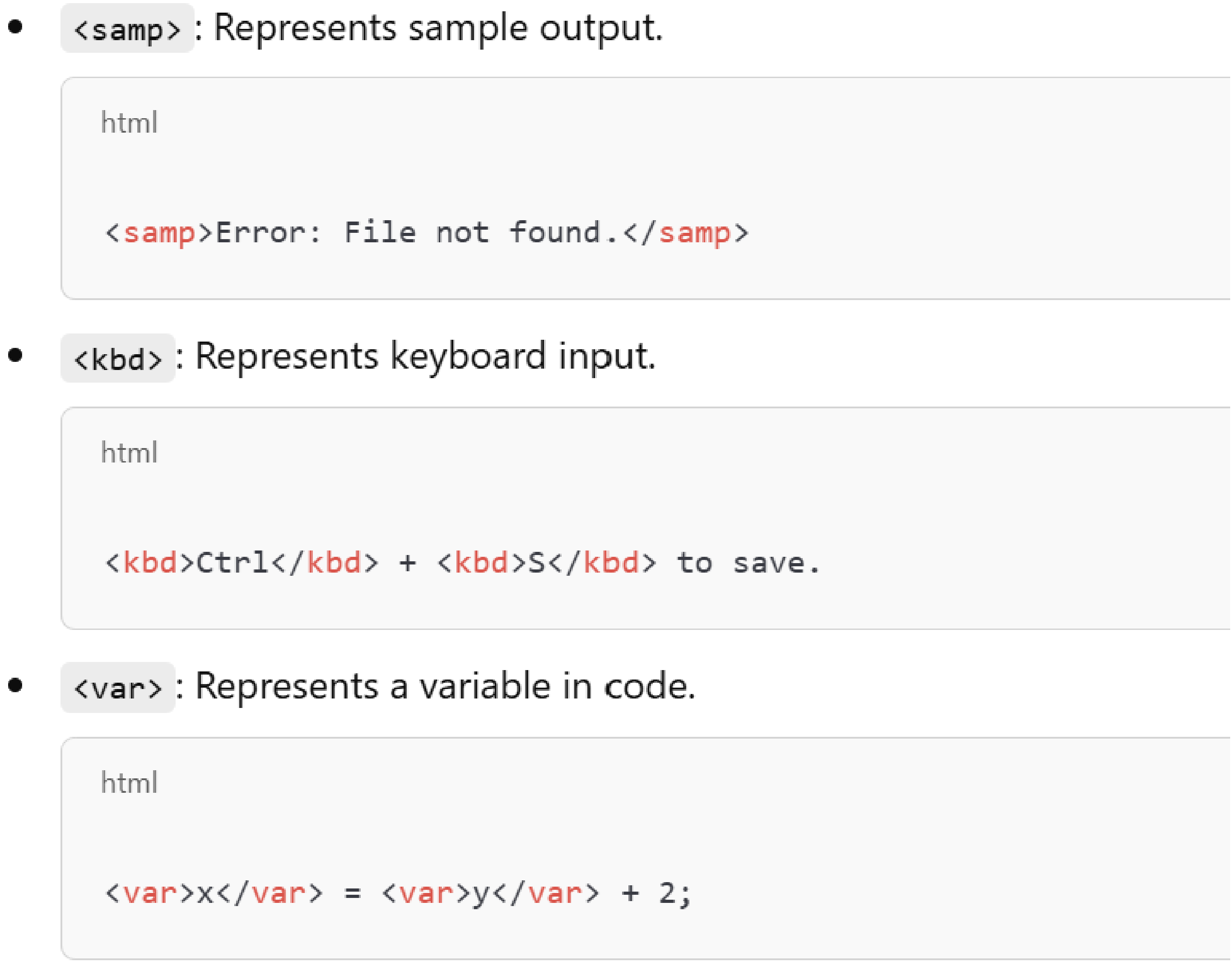
```html
<pre>
for (let i = 0; i < 10; i++) {
    console.log(i);
}
</pre>
```

PRE tag

- The <pre> tag defines preformatted text.
- Text in a <pre> element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks.



Code-Related Elements



- `<samp>` : Represents sample output.

html

```html
<samp>Error: File not found.</samp>
```

- `<kbd>` : Represents keyboard input.

html

```html
<kbd>Ctrl</kbd> + <kbd>S</kbd> to save.
```

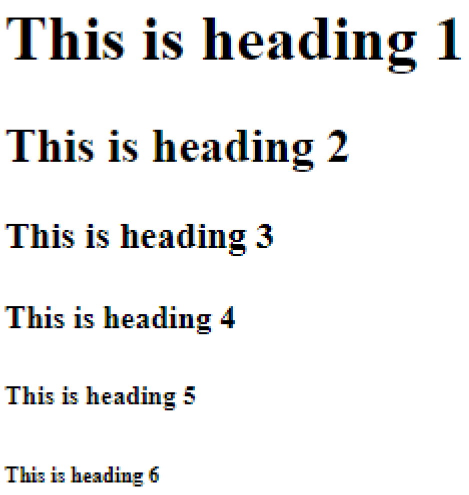
- `<var>` : Represents a variable in code.

html

```html
<var>x</var> = <var>y</var> + 2;
```

<h1>-<h6>

- The six different HTML headings:
- <h1>This is heading 1</h1> largest
  <h2>This is heading 2</h2>
  <h3>This is heading 3</h3>
  <h4>This is heading 4</h4>
  <h5>This is heading 5</h5>
  <h6>This is heading 6</h6> smallest



This is heading 1

This is heading 2

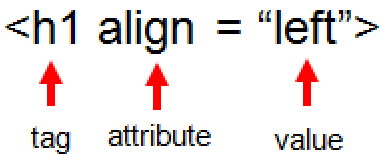
This is heading 3

This is heading 4

This is heading 5

This is heading 6

Attributes:



<h1 align = “left”>

tag attribute value

Attribute	Value	Description
align	left center right justify	Specifies the alignment of a heading

<img> tag

Attribute	Value	Description
src	*URL*	Specifies the URL of an image
alt	*text*	Specifies an alternate text for an image
align	top bottom middle left right	Specifies the alignment of an image according to surrounding elements
border	*pixels*	Specifies the width of the border around an image
height	*pixels*	Specifies the height of an image
width	*pixels*	Specifies the width of an image
ismap	ismap	Specifies an image as a server-side image-map
usemap	*#mapname*	Specifies an image as a client-side image-map

```
<img border=“5" src="buttonnext.gif" width="65" height="38">
```

Hr tag

Attribute	Value	Description
align	left center right	Specifies the alignment of a <hr> element
noshade	noshade	Specifies that a <hr> element should render in one solid color (noshaded), instead of a shaded color
size	*pixels*	Specifies the height of a <hr> element
width	*pixels%*	Specifies the width of a <hr> element

Obsolete in HTML5, use CSS instead

P tag

<p>This is some text in a paragraph.</p>

Attribute	Value	Description
align	left right center justify	Specifies the alignment of the text within a paragraph

Font tag

Attribute	Value	Description
face	*font_family*	Specifies the font of text
size	*number*	Specifies the size of text
color	*rgb(x,x,x)* *#xxxxxx* *colorname*	Specifies the color of text

```
<font size="3" color="red">This is some text!</font>
<font size="2" color="blue">This is some text!</font>
<font face="verdana" color="green">This is some text!</font>
```

Obsolete in HTML5, use CSS instead

address

- The <address> tag defines the contact information
- The text in the <address> element usually renders in *italic*.
- Most browsers will **add a line break before and after** the address element.

```
<address>
Visit us at:<br>
Example.com<br>
Box 564, Disneyland<br>
USA
</address>
```

```
<address>
Email: <a href="mailto:support@example.com">
support@example.com</a><br>
Phone: +1 (555) 123-4567
</address>
```

Center tag

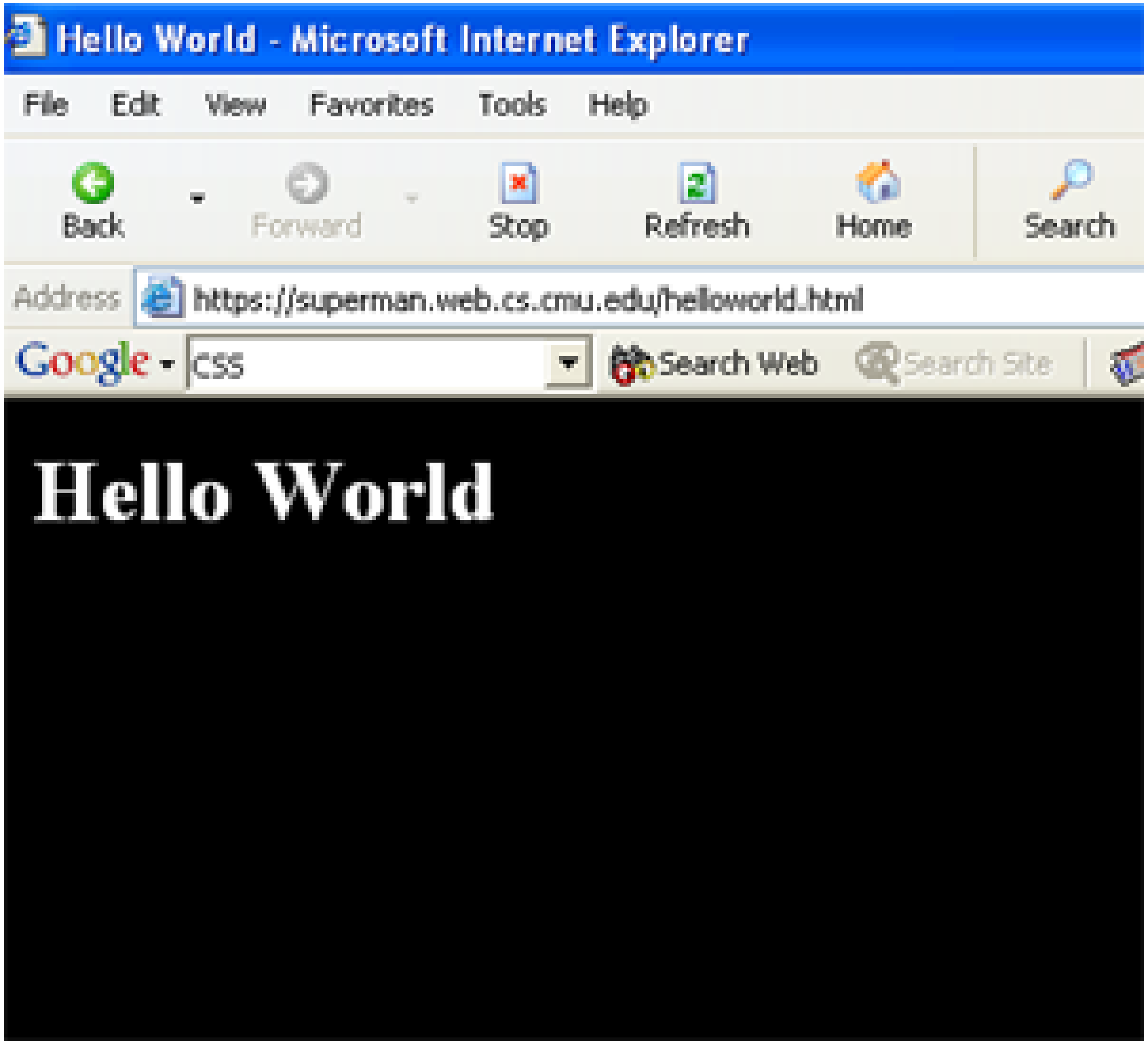
- You can use **<center>** tag to put any content in the center of the page.

Revision

Tag	Description
<html>	Defines an HTML document
<body>	Defines the document's body
<h1> to <h6>	Defines header 1 to header 6
<p>	Defines a paragraph
<br>	Inserts a single line break
<hr>	Defines a horizontal rule
<!-->	Defines a comment

Example HTML code:

```
<HTML>

<head>
<title>Hello World</title>
</head>

<body bgcolor = “#000000”>
<font color = “#ffffff”>
<H1>Hello World</H1>
</font>
</body>

</HTML>
```



Address https://superman.web.cs.cmu.edu/helloworld.html

css

Hello World

HTML Exercise

1. Create a Personal homepage
2. Design an E-Greeting page which has properly aligned paragraphs with images

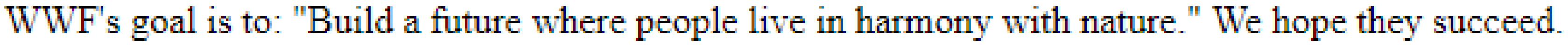
COMMONLY USED CHARACTERS

Result	Description	Entity Name
	non-breaking space	**&nbsp;**
<	less than	&lt;
>	greater than	&gt;
&	ampersand	&amp;
¢	cent	&cent;

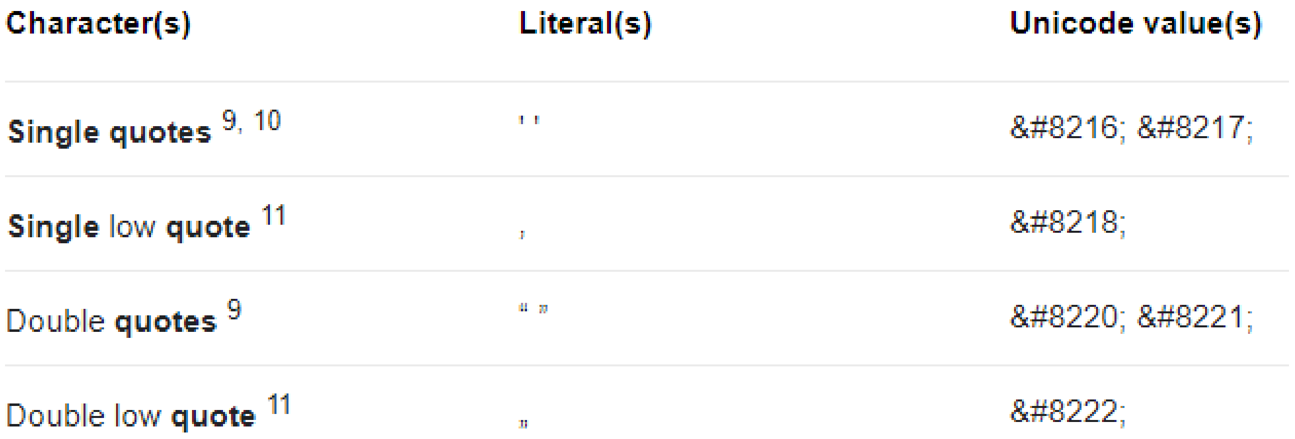
Result	Description	Entity Name
£	pound	&pound;
¥	yen	&yen;
€	euro	&euro;
§	section	&sect;
©	copyright	&copy;
®	registered trademark	&reg;
₹	Rupees	&#8377;

Q tag

- The <q> tag defines a **short quotation.**
- Browsers normally **insert quotation marks** around the text.
- <p>WWF's goal is to:
  <q>Build a future where people live in harmony with nature.</q>
  We hope they succeed.</p>



WWF's goal is to: "Build a future where people live in harmony with nature." We hope they succeed.



Character(s)	Literal(s)	Unicode value(s)
Single quotes [9, 10]	‘ ’	&#8216; &#8217;
Single low quote [11]	‚	&#8218;
Double quotes [9]	“ ”	&#8220; &#8221;
Double low quote [11]	„	&#8222;

Marquee tag

Attribute	Value	Description
behavior	scroll slide alternate	Defines the type of scrolling.
bgcolor	rgb(x,x,x) #xxxxxx colorname	*Deprecated*
direction	up down left right	Defines the direction of scrolling the content.
height	pixels or %	Defines the height of marquee.
hspace	pixels	Specifies horizontal space around the marquee.
loop	number	Specifies how many times to loop. The default value is INFINITE, which means that the marquee loops endlessly.
scrolldelay	seconds	Defines how long to delay between each jump.
scrollamount	number	Defines how how far to jump.
width	pixels or %	Defines the width of marquee.
vspace	pixels	Specifies vertical space around the marquee.

Obsolete in HTML5, use CSS instead

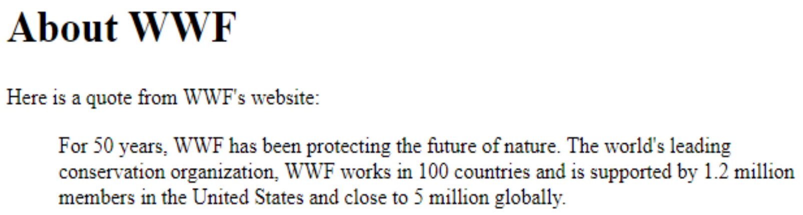
Div tag

- The <div> tag defines a division or a section in an HTML document

-
  ```
  <div style="color:#0000FF">
    <h3>This is a heading</h3>
    <p>This is a paragraph.</p>
  </div>
  ```
- <div id = "menu" align = "middle" >...</div>
- <div id = "content" align = "left" bgcolor = "white"> ....</div>

blockquote

- The <blockquote> tag specifies a section that is quoted from another source.
- Browsers usually **indent** <blockquote> elements.
- <html> <body>
- <h1>About WWF</h1>
- <p>Here is a quote from WWF's website:</p>
- <blockquote **cite**="http://www.worldwildlife.org/who/index.html">
- For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.
- </blockquote> </body> </html>



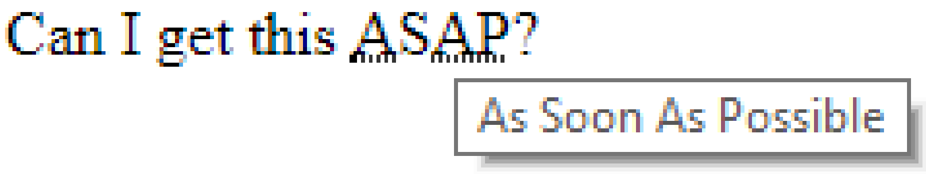
About WWF

Here is a quote from WWF's website:

> For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

acronym

- The <acronym> tag defines an acronym.
- An acronym can be spoken as if it were a word, example NATO, NASA, ASAP, GUI.
- **Specify a full description** of the term when hovered over using the **title** attribute.
- Can I get this <acronym **title**="As Soon As Possible">ASAP</acronym>?



Can I get this ASAP?

As Soon As Possible

anchor tag

- The <a> tag defines a hyperlink, which is used to link from one page to another.
- The most important attribute of the <a> element is the href attribute, which indicates the link's destination.
- By **default**, links will appear as follows in all browsers:
  - An unvisited link is underlined and blue
  - A visited link is underlined and purple
  - An active link is underlined and red

<a href="http://www.w3schools.com">Visit W3Schools.com!</a>

<a> tag attributes

href	*URL*	Specifies the URL of the page the link goes to
target	_blank _parent _self _top *framename*	Specifies where to open the linked document

An Image as a Link

```
<!DOCTYPE html>
<html>
<body>

<p>
An image as a link: <a href="http://www.w3schools.com">
<img border="0" alt="W3Schools" src="logo_w3s.gif" width="100" height="100">
</a>
</p>

</body>
</html>
```

anchor tag

Text as Hyper reference

<A HREF="APP.HTML">CLICK HERE </A>

Button as Hyper reference

<A HREF="APP.HTML"><button>Click me</button></A>

Image as Hyper reference

<A HREF="APP.HTML"><IMG SRC="sweet-home-icon.jpg"></A>

Create Link at the Same Page

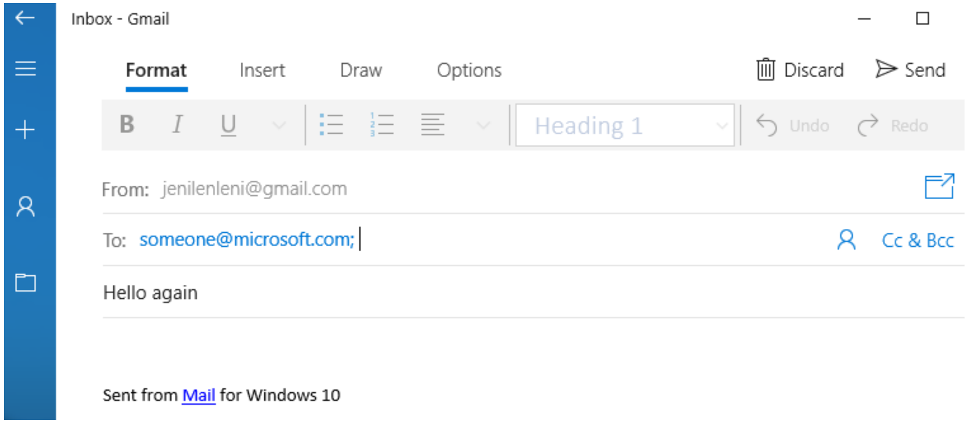
```
<!DOCTYPE html>
<html>
<body>
<h2><a id="top">There is a link at the bottom of the page!</a></h2>
<p>In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since.</p>
<p>"Whenever you feel like criticizing any one," he told me, "just remember that all the people in this world haven't had the advantages that you've had."</p>
…..
…..
…..
…..
<a href="#top">Go to top</a>
</body>
</html>
```

```
<p name="pname">….</p>
<a href="#pname">A simple link</a>
```

CREATE A MAILTO LINK

Open outlook mail with filled value

```
    <html>
<body>
<p>
This is a mail link:
<a href="mailto:someone@microsoft.com?subject=Hello%20again">
Send Mail</a>
</p>
</body>
</html>
```



From: jenilenleni@gmail.com

To: someone@microsoft.com;

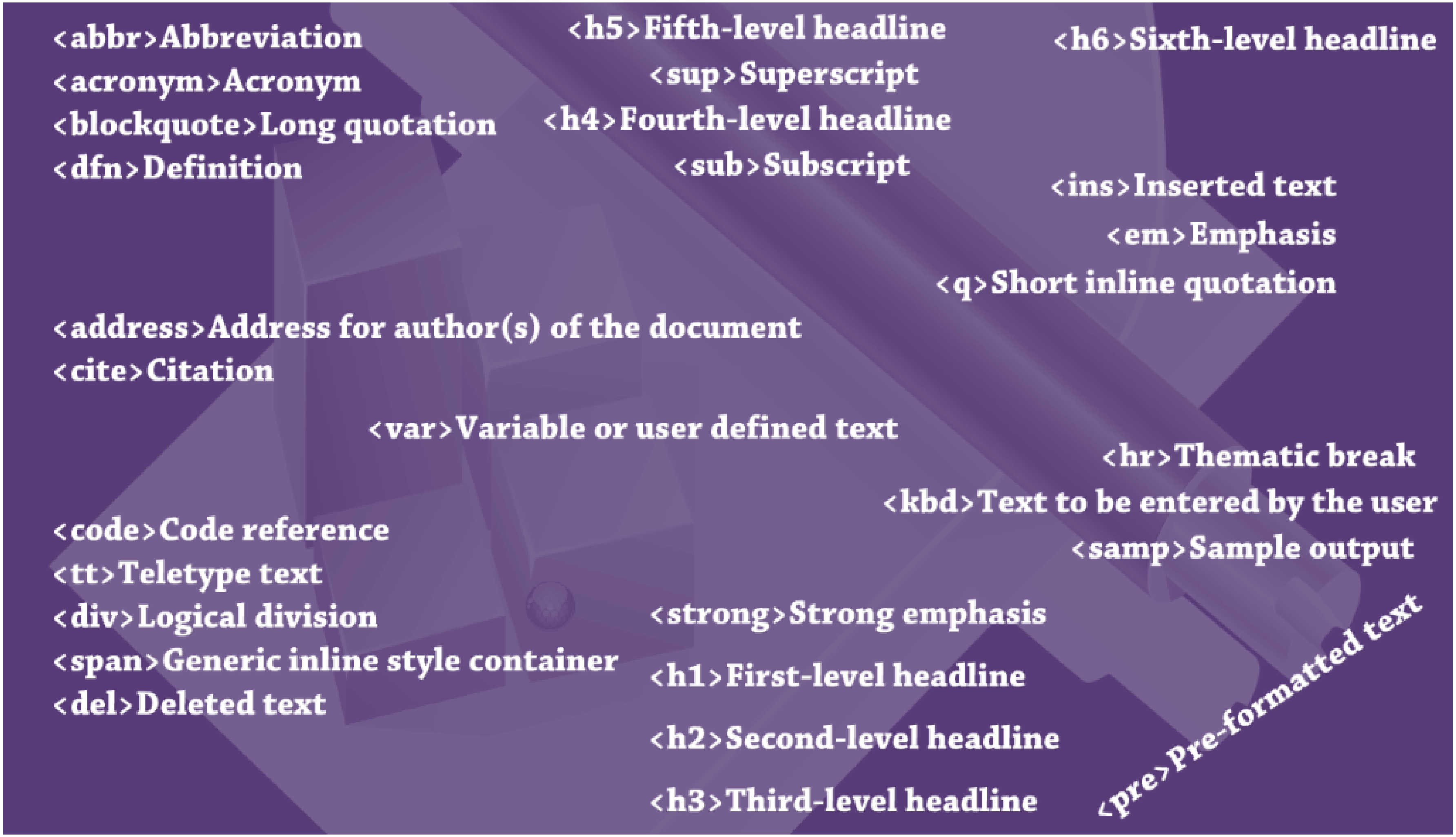
Hello again

Sent from Mail for Windows 10

"**%20**" represents a space in an encoded URL

"**%0A**" represents a blank line in an encoded URL

<a href="mailto:someone@example.com**?cc=**x@gmail.com**&bcc=**y@gmail.com**&**Subject=Hello%20again**&**body=I'm%20interested%20in%20booking%20a%20party%20of:%0A%0AThe%20dates:%20...">Send Mail</a>



<abbr>Abbreviation
<acronym>Acronym
<blockquote>Long quotation
<dfn>Definition

<h5>Fifth-level headline
<sup>Superscript
<h4>Fourth-level headline
<sub>Subscript

<h6>Sixth-level headline

<ins>Inserted text
<em>Emphasis
<q>Short inline quotation

<address>Address for author(s) of the document
<cite>Citation

<var>Variable or user defined text

<hr>Thematic break
<kbd>Text to be entered by the user
<samp>Sample output

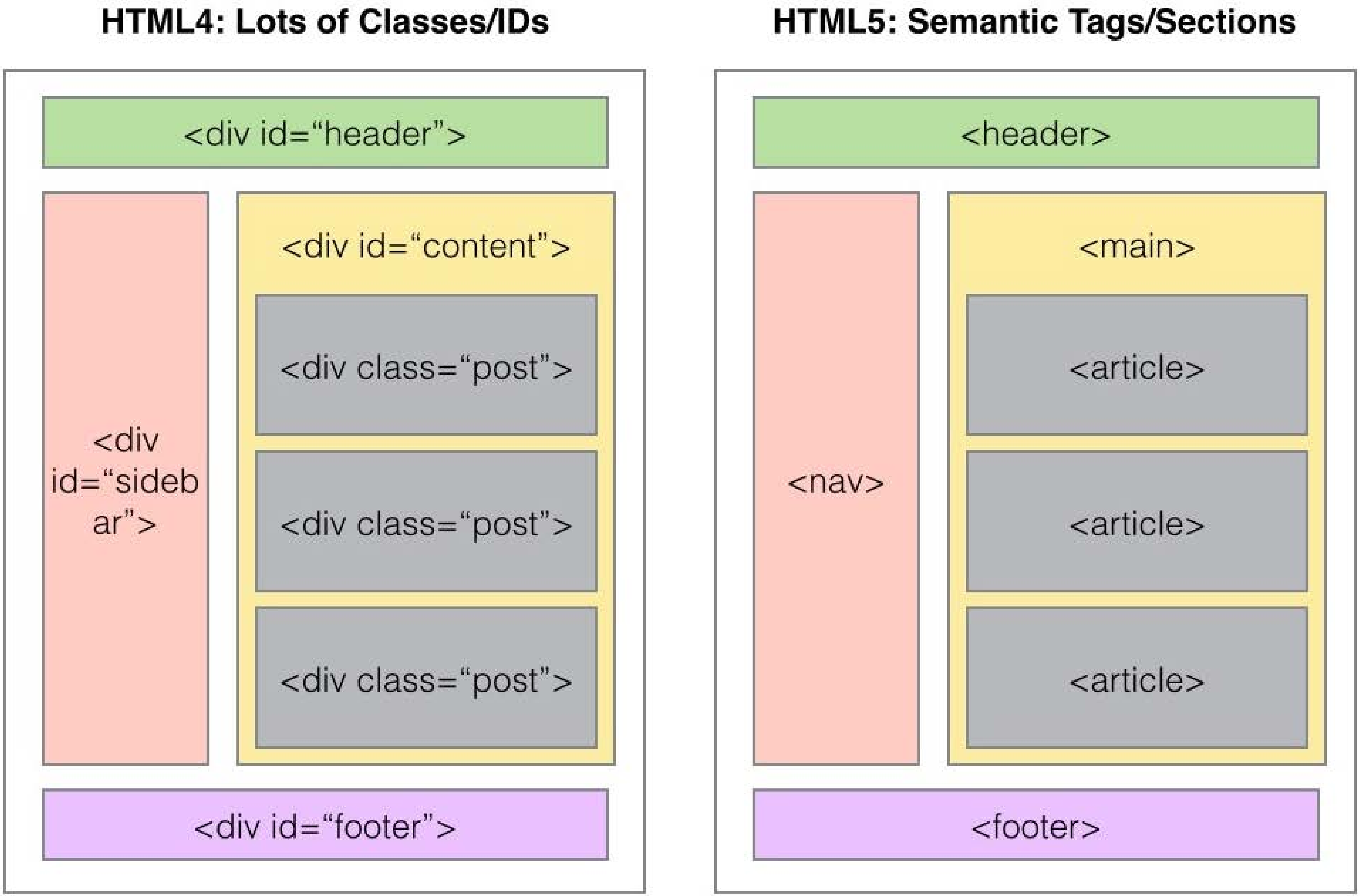
<code>Code reference
<tt>Teletype text
<div>Logical division
<span>Generic inline style container
<del>Deleted text

<strong>Strong emphasis
<h1>First-level headline
<h2>Second-level headline
<h3>Third-level headline

<pre>Pre-formatted text

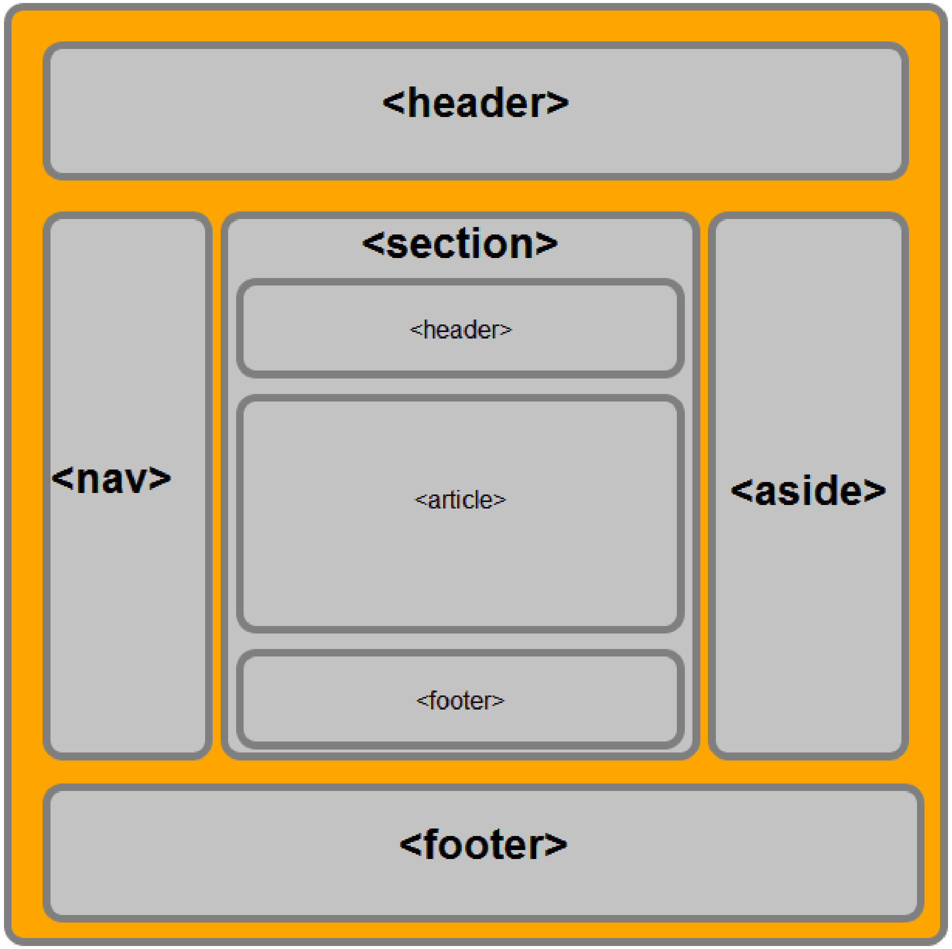
Source: https://www.lambdatest.com/blog/importance-of-semantic-html-in-modern-web-development/

HTML4 vs HTML5 Page Structure



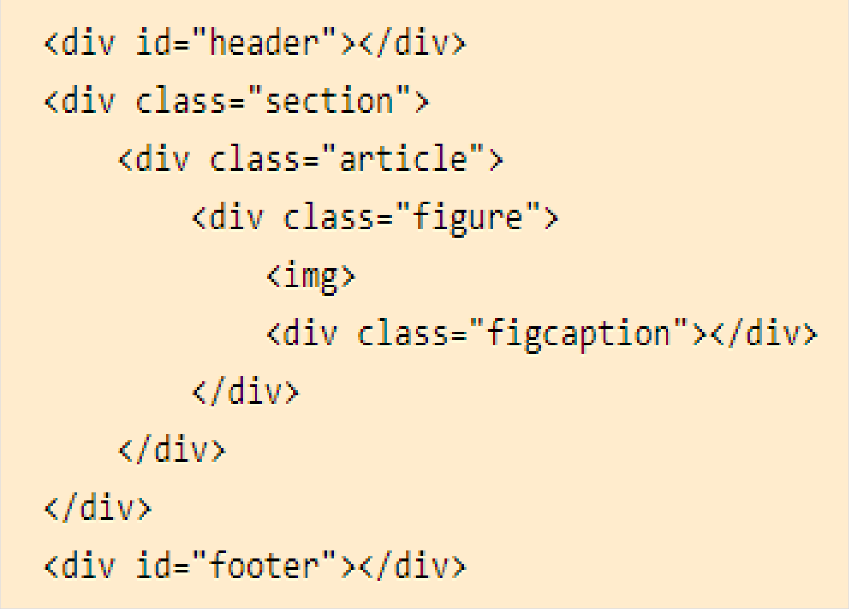
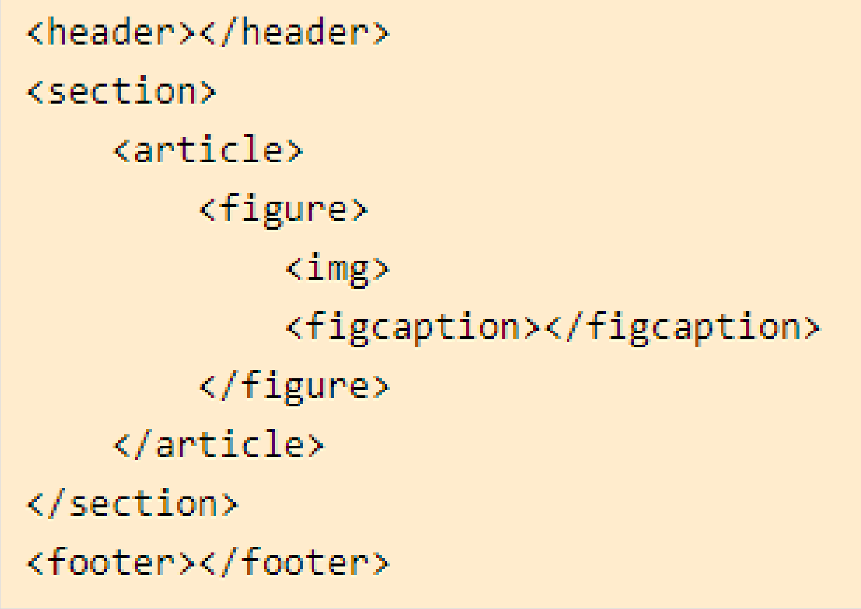
Source: https://www.vikingcodeschool.com/html5-and-css3/html5-semantic-tags

Semantic tags



Source: https://www.c-sharpcorner.com/UploadFile/b5be7f/working-with-semantic-elements-in-html5-with-layout-examples//

Non-Semantic Vs Semantic

Non-Semantic	Semantic
	

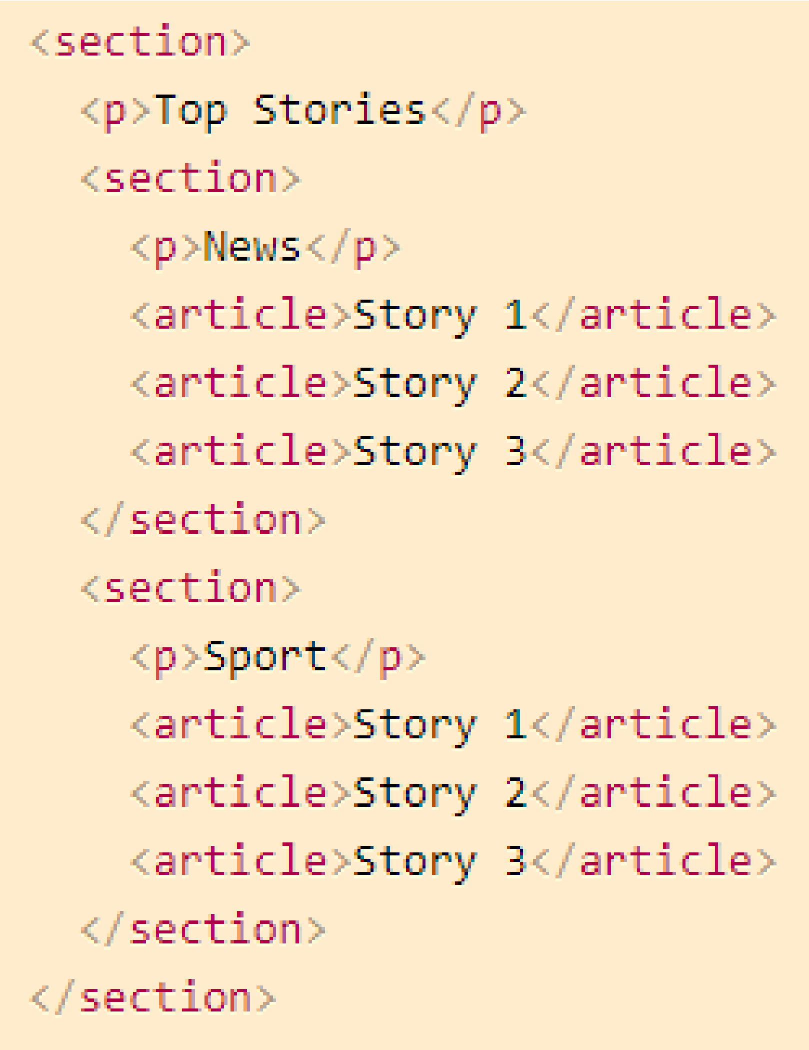
```
<div id="header"></div>
<div class="section">
    <div class="article">
        <div class="figure">
            <img>
            <div class="figcaption"></div>
        </div>
    </div>
</div>
<div id="footer"></div>
```

```
<header></header>
<section>
    <article>
        <figure>
            <img>
            <figcaption></figcaption>
        </figure>
    </article>
</section>
<footer></footer>
```

Source: https://www.freecodecamp.org/news/semantic-html5-elements/

Semantic Tags

- Multiple sections inside sections
- The <section> and <article> elements are conceptually similar and interchangeable.



```
<section>
  <p>Top Stories</p>
  <section>
    <p>News</p>
    <article>Story 1</article>
    <article>Story 2</article>
    <article>Story 3</article>
  </section>
  <section>
    <p>Sport</p>
    <article>Story 1</article>
    <article>Story 2</article>
    <article>Story 3</article>
  </section>
</section>
```

<nav>

<nav> defines a set of navigation links

Example:

```
<nav>
 <a href="/html/">HTML</a> |
 <a href="/css/">CSS</a> |
 <a href="/js/">JavaScript</a> |
 <a href="/jquery/">jQuery</a>
</nav>
```

Semantic Tags

<article> - article in the element

https://www.w3schools.com/tags/tag_article.asp

<header> - header for the section or article

<footer> - footer for the section or article

<section> - section in the document

<aside> - content aside from the page content

<main> - only one main tag in your document

Semantic Tags

<dialog> defines a dialog box or window

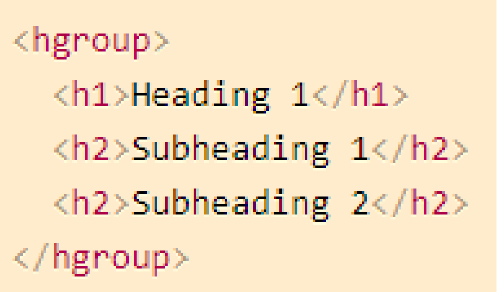
https://www.w3schools.com/tags/tag_dialog.asp

<figure> - self contained content like diagram

<figcaption> defines the caption for a <figure>

https://www.w3schools.com/tags/tag_figure.asp

<hgroup> main heading with one or more subheadings

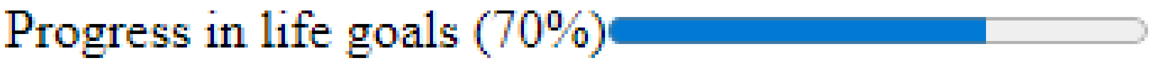


```
<hgroup>
  <h1>Heading 1</h1>
  <h2>Subheading 1</h2>
  <h2>Subheading 2</h2>
</hgroup>
```

Other Semantic Tags

<progress> - progress of a task

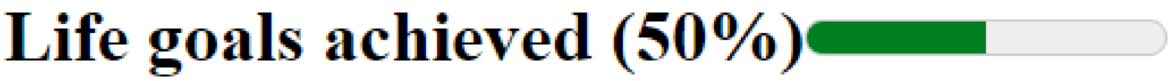
```
Progress in life goals (70%)
<progress max=100 value=70></progress>
```



Progress in life goals (70%)

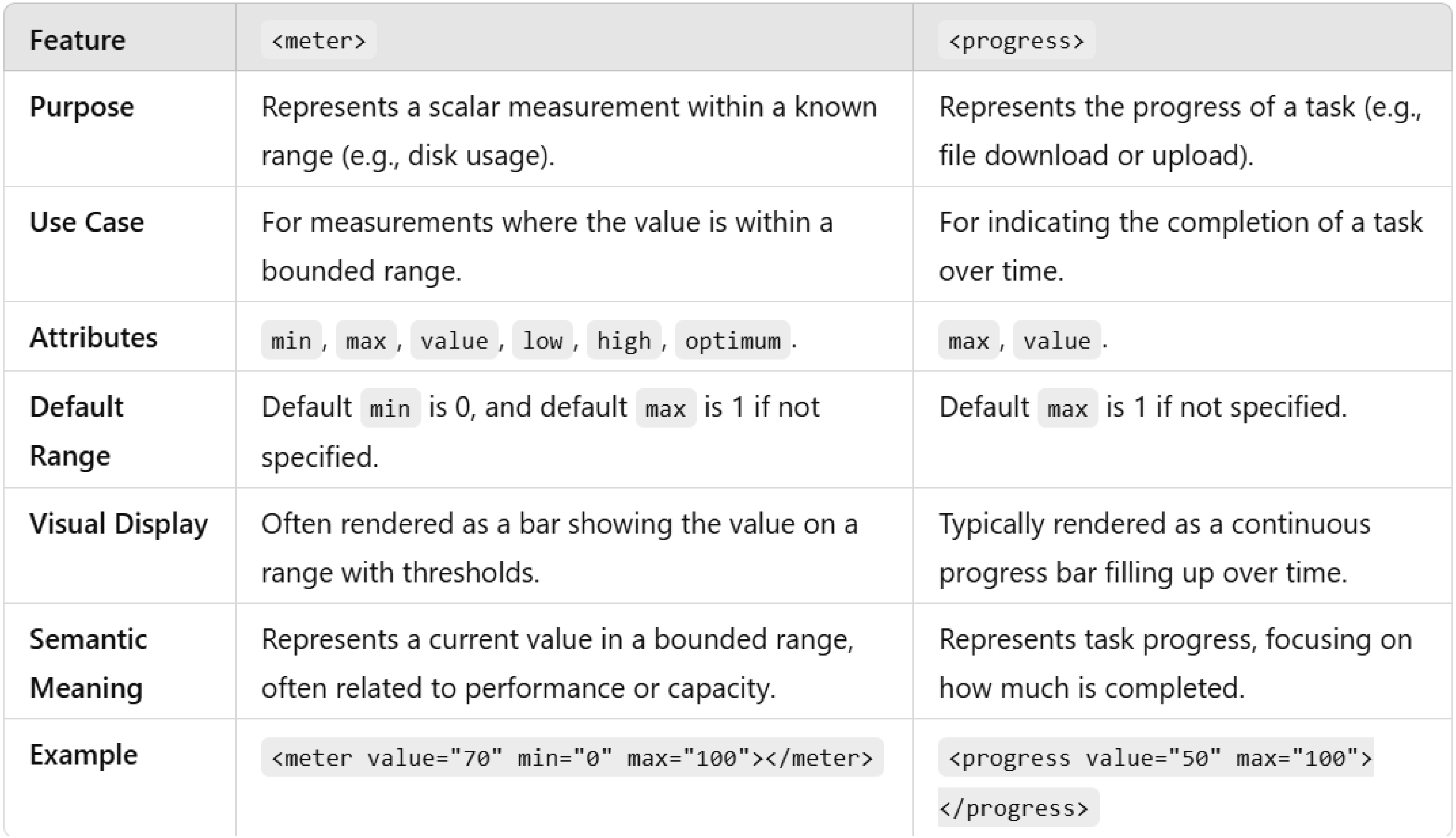
<meter> - scalar measurement with in a known range

```
Life goals achieved (50%)
<meter min=0 max=100 value=50></meter>
```



Life goals achieved (50%)

Meter Vs Progress



Feature	`<meter>`	`<progress>`
Purpose	Represents a scalar measurement within a known range (e.g., disk usage).	Represents the progress of a task (e.g., file download or upload).
Use Case	For measurements where the value is within a bounded range.	For indicating the completion of a task over time.
Attributes	`min`, `max`, `value`, `low`, `high`, `optimum`.	`max`, `value`.
Default Range	Default `min` is 0, and default `max` is 1 if not specified.	Default `max` is 1 if not specified.
Visual Display	Often rendered as a bar showing the value on a range with thresholds.	Typically rendered as a continuous progress bar filling up over time.
Semantic Meaning	Represents a current value in a bounded range, often related to performance or capacity.	Represents task progress, focusing on how much is completed.
Example	`<meter value="70" min="0" max="100"></meter>`	`<progress value="50" max="100"></progress>`

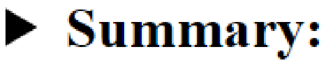
Other Semantic Tags

Details with Summary

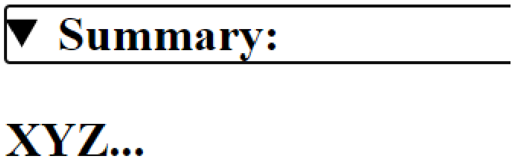
<details> defines additional details that people can view or hide (like a **tooltip**)

<summary> defines a visible heading for a <details> element

```
<details>
<summary>Summary:</summary>
<p>XYZ...</p>
</details>
```



▶ Summary:



▼ Summary:

XYZ...

Exercise

Write a program (Biodata) which will perform the following tasks:

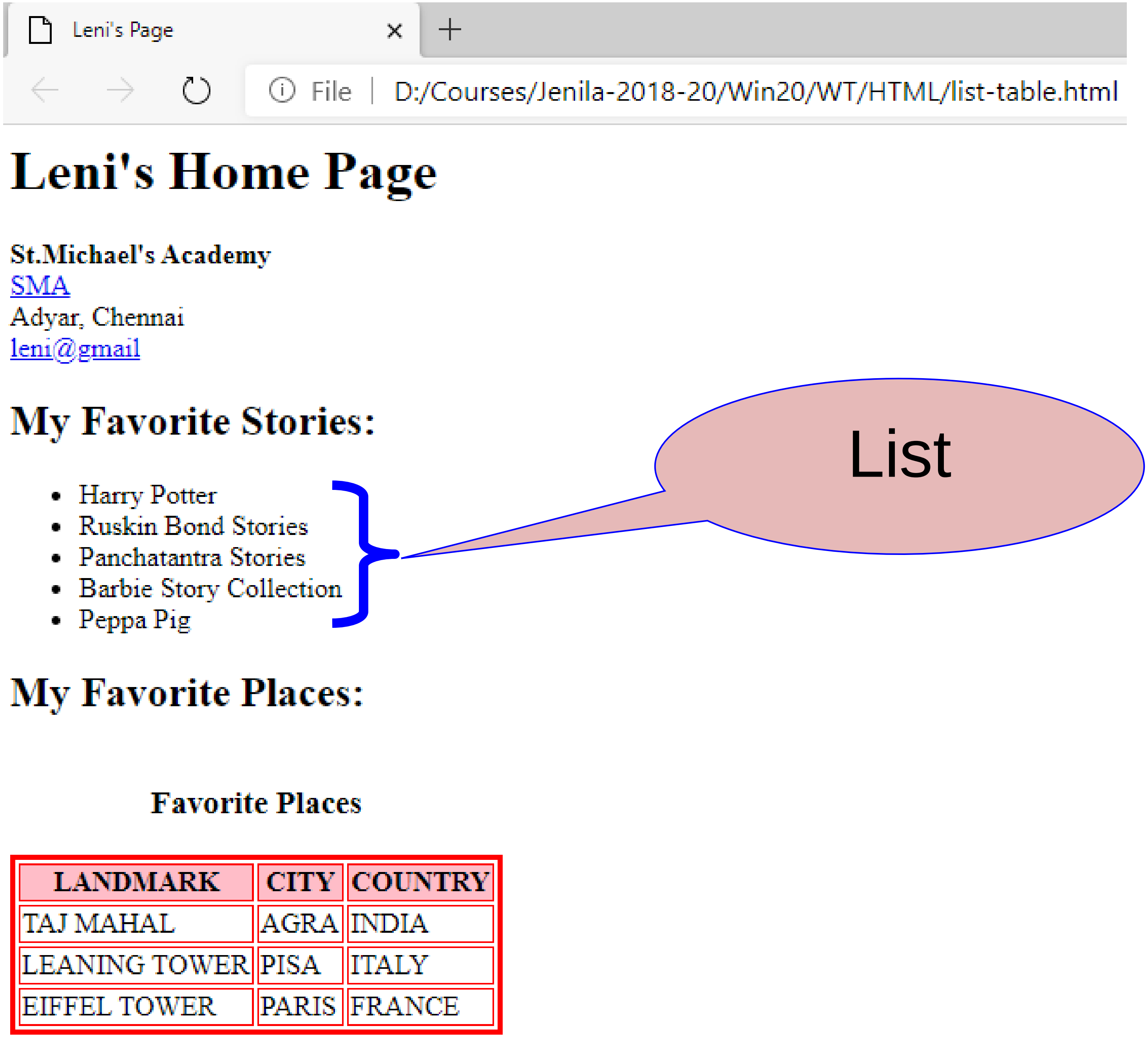
1. Use of <HTML><HEAD><TITLE><BODY> Tags
2. Use of <H1> to <H6> Tags
3. Use of <img> Tag
4. Use of <br>,<hr>,<p>,<div>,<blockquote>,<q><CENTER>,&nbsp Tags
5. Use of All Text Formatting Tags like <font>,<b>,<i>,<u>,<s>,<em>,<strong><big>,<small>,<sub>,<sup>, <code>, <address>,<del>,<ins>, <acronym>, <pre>,<q>
6. Use Of Hyperlink i.e. <a></a> Tag.
7. Use Semantic tags

Thank You

HTML Lists and Tables

Dr. L.M. Jenila Livingston

VIT Chennai



Leni's Home Page

St.Michael's Academy
SMA
Adyar, Chennai
leni@gmail

My Favorite Stories:

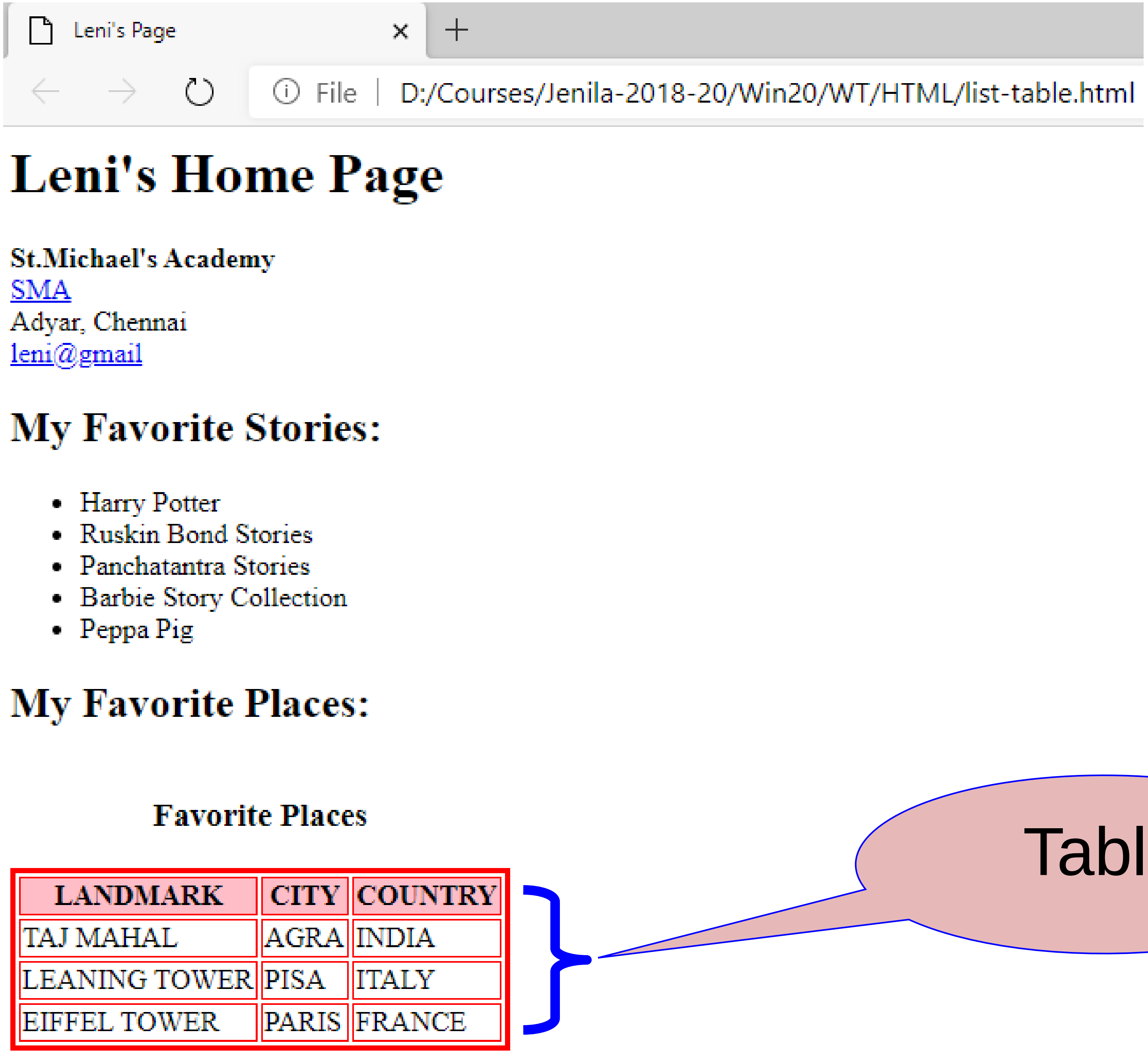
- Harry Potter
- Ruskin Bond Stories
- Panchatantra Stories
- Barbie Story Collection
- Peppa Pig

List

My Favorite Places:

Favorite Places

LANDMARK	CITY	COUNTRY
TAJ MAHAL	AGRA	INDIA
LEANING TOWER	PISA	ITALY
EIFFEL TOWER	PARIS	FRANCE



Leni's Home Page

St.Michael's Academy
SMA
Adyar, Chennai
leni@gmail

My Favorite Stories:

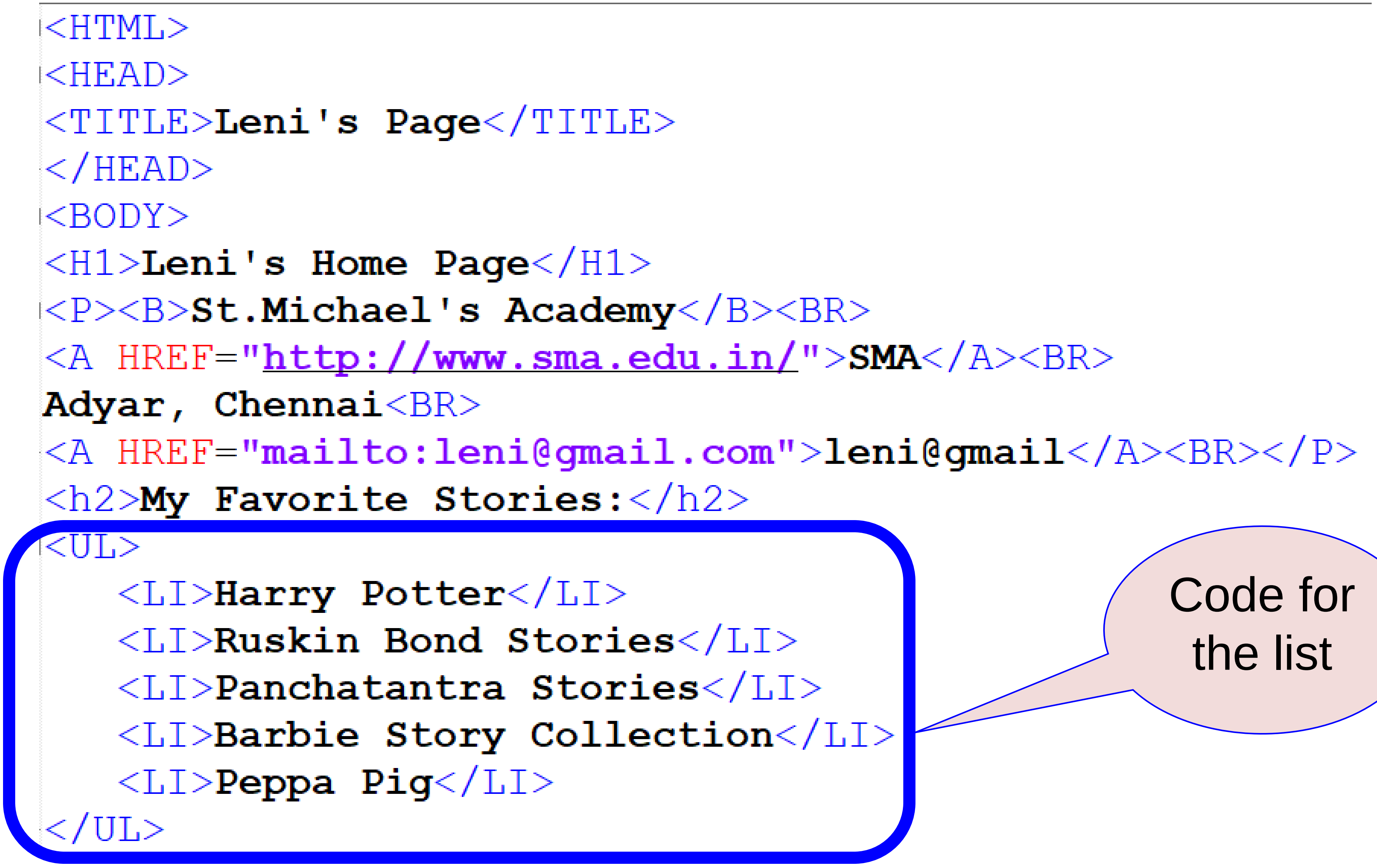
- Harry Potter
- Ruskin Bond Stories
- Panchatantra Stories
- Barbie Story Collection
- Peppa Pig

My Favorite Places:

Favorite Places

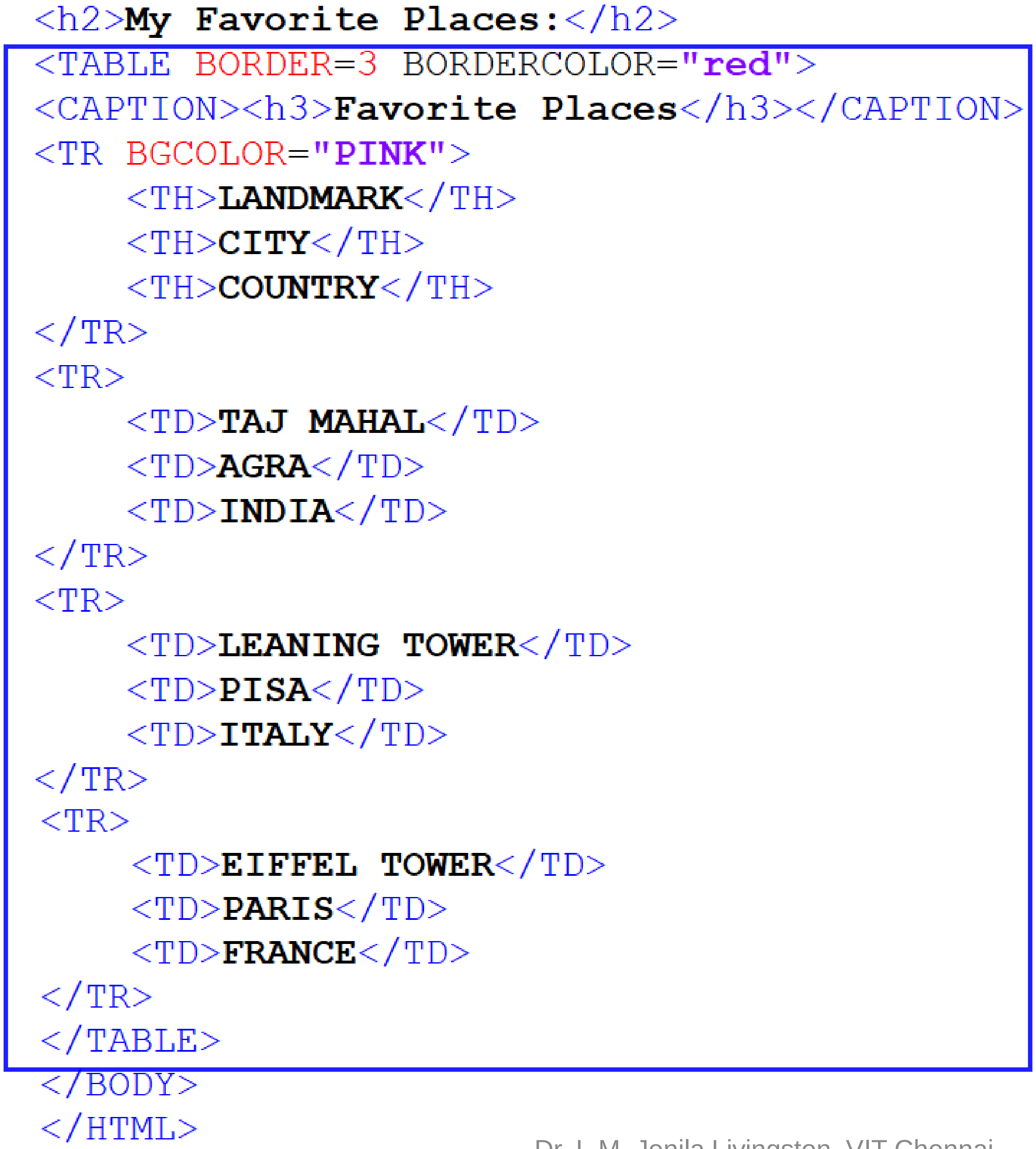
LANDMARK	CITY	COUNTRY
TAJ MAHAL	AGRA	INDIA
LEANING TOWER	PISA	ITALY
EIFFEL TOWER	PARIS	FRANCE

Table



```
<HTML>
<HEAD>
<TITLE>Leni's Page</TITLE>
</HEAD>
<BODY>
<H1>Leni's Home Page</H1>
<P><B>St.Michael's Academy</B><BR>
<A HREF="http://www.sma.edu.in/">SMA</A><BR>
Adyar, Chennai<BR>
<A HREF="mailto:leni@gmail.com">leni@gmail</A><BR></P>
<h2>My Favorite Stories:</h2>
<UL>
   <LI>Harry Potter</LI>
   <LI>Ruskin Bond Stories</LI>
   <LI>Panchatantra Stories</LI>
   <LI>Barbie Story Collection</LI>
   <LI>Peppa Pig</LI>
</UL>
```

Code for the list



```
<h2>My Favorite Places:</h2>
<TABLE BORDER=3 BORDERCOLOR="red">
<CAPTION><h3>Favorite Places</h3></CAPTION>
<TR BGCOLOR="PINK">
    <TH>LANDMARK</TH>
    <TH>CITY</TH>
    <TH>COUNTRY</TH>
</TR>
<TR>
    <TD>TAJ MAHAL</TD>
    <TD>AGRA</TD>
    <TD>INDIA</TD>
</TR>
<TR>
    <TD>LEANING TOWER</TD>
    <TD>PISA</TD>
    <TD>ITALY</TD>
</TR>
<TR>
    <TD>EIFFEL TOWER</TD>
    <TD>PARIS</TD>
    <TD>FRANCE</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Code for the Table

HTML Lists & Tables

- Lists are one way to present the data in an ordered or structured manner on Web pages

- Tables provide another - more customizable - way of displaying ordered information

Types of Lists

1. Unordered Lists
2. Ordered Lists
3. Definition List

Types of Lists

1. **Unordered Lists :** Displays items without any specific order (typically bullet points).
2. **Ordered Lists:** Displays items in a sequential or ranked order (numbers, letters, etc.).
3. **Definition List:** Used for definition terms and their descriptions.

Unordered List

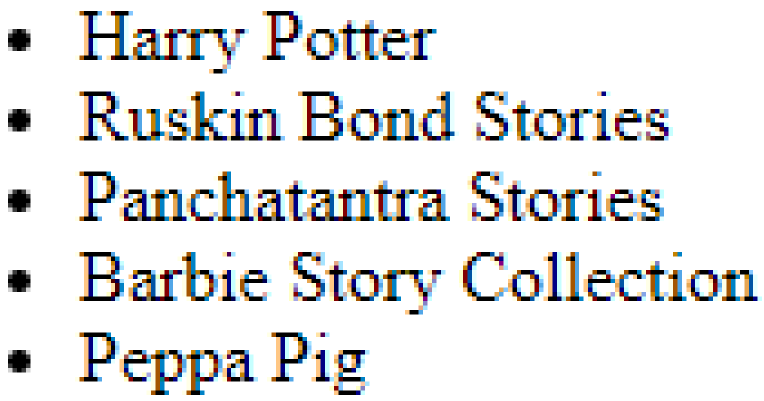
<UL>	Un-ordered List
<LI>	Line Item

Unordered List

HTML Code

```
<UL>
  <LI>Harry Potter</LI>
  <LI>Ruskin Bond Stories</LI>
  <LI>Panchatantra Stories</LI>
  <LI>Barbie Story Collection</LI>
  <LI>Peppa Pig</LI>
</UL>
```

Browser Display



- Harry Potter
- Ruskin Bond Stories
- Panchatantra Stories
- Barbie Story Collection
- Peppa Pig

Unordered List Types

type attribute

type	**Result**
“disc”	• disc (default)
“circle”	o circle
“square”	❑ square
“none”	Items will not be marked

UL Types

The default “bullet” for these lists is a “**disc**” (filled circle)

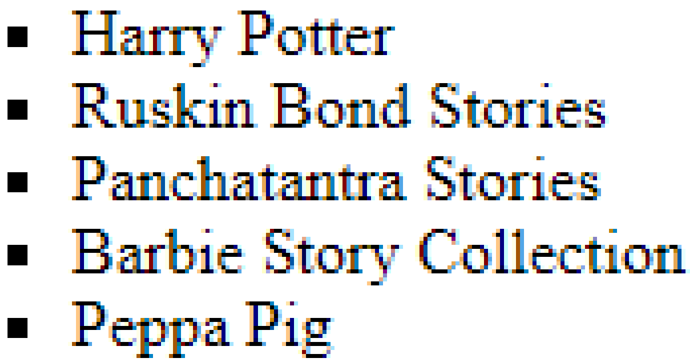
That, however, can be changed to a “**circle**” (open circle) or a “**square**” with the help of the **type** attribute

UL Square Type

HTML Code

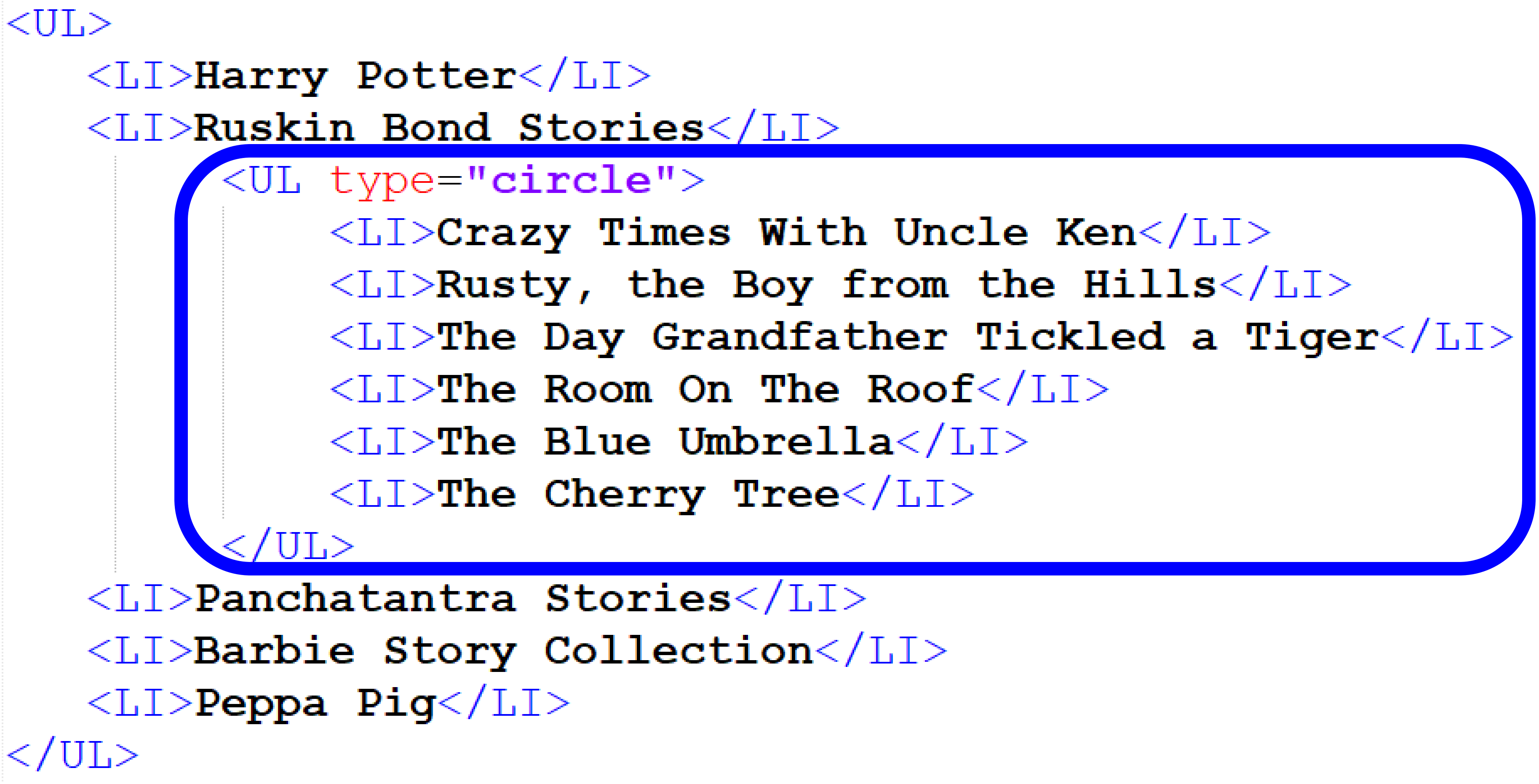
```
<UL type = “square”>
  <LI>Harry Potter</LI>
  <LI>Ruskin Bond Stories</LI>
  <LI>Panchatantra Stories</LI>
  <LI>Barbie Story Collection</LI>
  <LI>Peppa Pig</LI>
</UL>
```

Browser Display



▪ Harry Potter
▪ Ruskin Bond Stories
▪ Panchatantra Stories
▪ Barbie Story Collection
▪ Peppa Pig

Nested List

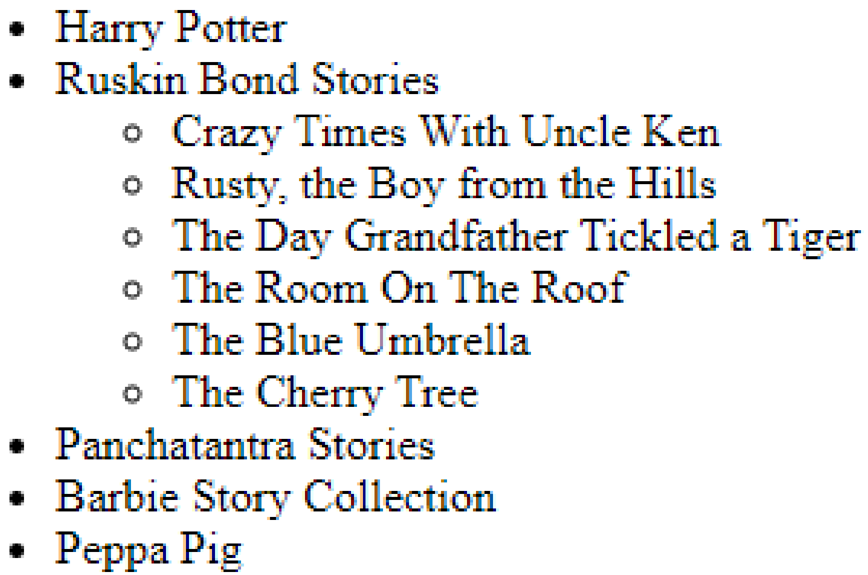


```
<UL>
    <LI>Harry Potter</LI>
    <LI>Ruskin Bond Stories</LI>
        <UL type="circle">
            <LI>Crazy Times With Uncle Ken</LI>
            <LI>Rusty, the Boy from the Hills</LI>
            <LI>The Day Grandfather Tickled a Tiger</LI>
            <LI>The Room On The Roof</LI>
            <LI>The Blue Umbrella</LI>
            <LI>The Cherry Tree</LI>
        </UL>
    <LI>Panchatantra Stories</LI>
    <LI>Barbie Story Collection</LI>
    <LI>Peppa Pig</LI>
</UL>
```

Nested List

1. Different bullets
2. Additional tab

Browser Display



- Harry Potter
- Ruskin Bond Stories
    - Crazy Times With Uncle Ken
    - Rusty, the Boy from the Hills
    - The Day Grandfather Tickled a Tiger
    - The Room On The Roof
    - The Blue Umbrella
    - The Cherry Tree
- Panchatantra Stories
- Barbie Story Collection
- Peppa Pig

Another list starts before the first list is finished, are called **Nested Lists**

Ordered List

<OL>	Ordered List
<LI>	Line Item

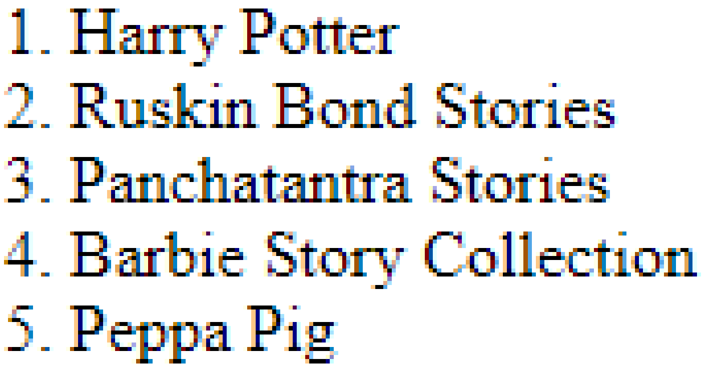
Ordered List

HTML Code

```
<OL>
  <LI>Harry Potter</LI>
  <LI>Ruskin Bond Stories</LI>
  <LI>Panchatantra Stories</LI>
  <LI>Barbie Story Collection</LI>
  <LI>Peppa Pig</LI>
</OL>
```

OL instead of UL

Browser Display



1. Harry Potter
2. Ruskin Bond Stories
3. Panchatantra Stories
4. Barbie Story Collection
5. Peppa Pig

Numbers instead of discs, circles or squares

Ordered List Types

type attribute

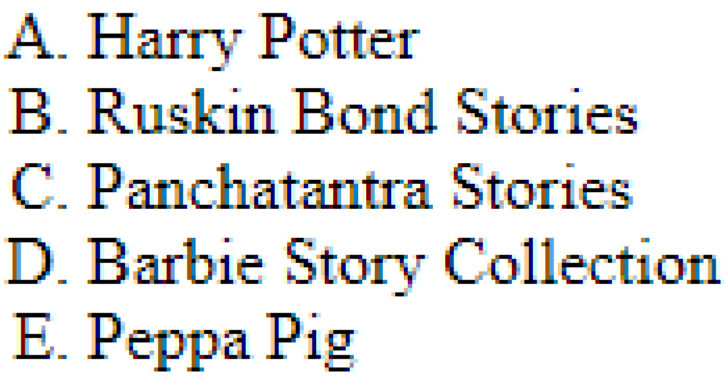
type	Result
“1”	1, 2, 3, … (default)
“A”	A, B, C, …
“a”	a, b, c, …
“I”	I, II, III, IV, …
“i”	i, ii, iii, iv, …

OL type="A"

HTML Code

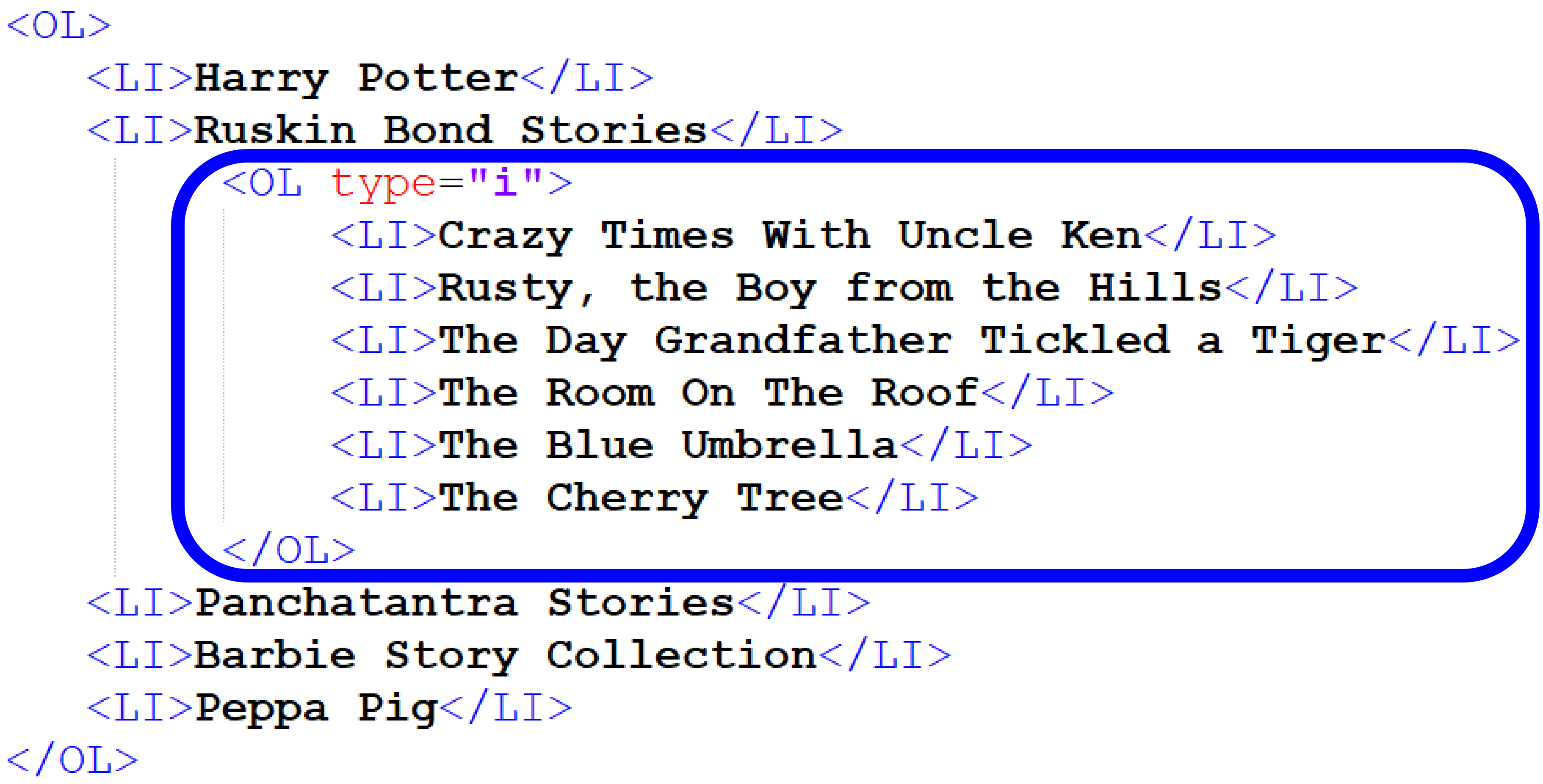
```
<OL type= "A" >
   <LI>Harry Potter</LI>
   <LI>Ruskin Bond Stories</LI>
   <LI>Panchatantra Stories</LI>
   <LI>Barbie Story Collection</LI>
   <LI>Peppa Pig</LI>
</OL>
```

Browser Display



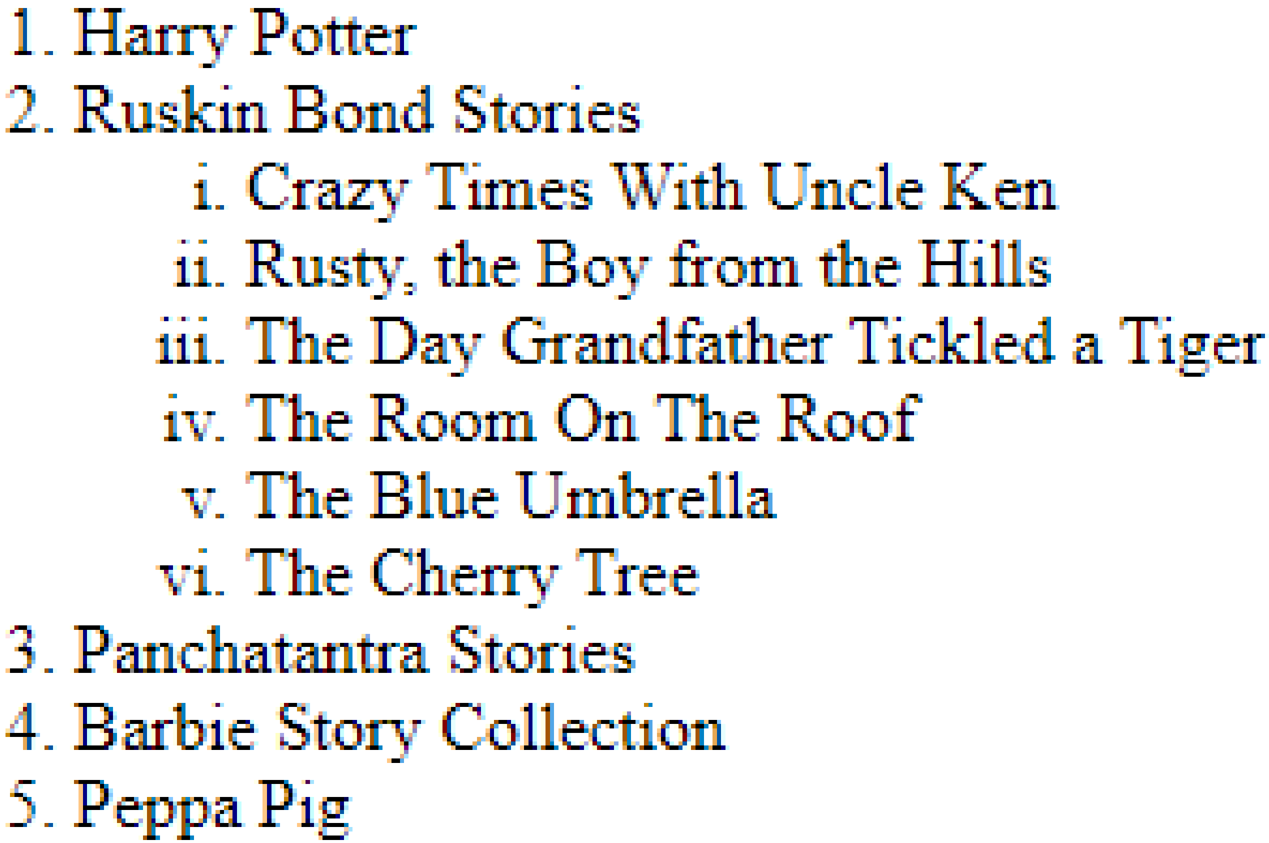
A. Harry Potter
B. Ruskin Bond Stories
C. Panchatantra Stories
D. Barbie Story Collection
E. Peppa Pig

OL-Nested List



```
<OL>
    <LI>Harry Potter</LI>
    <LI>Ruskin Bond Stories</LI>
        <OL type="i">
            <LI>Crazy Times With Uncle Ken</LI>
            <LI>Rusty, the Boy from the Hills</LI>
            <LI>The Day Grandfather Tickled a Tiger</LI>
            <LI>The Room On The Roof</LI>
            <LI>The Blue Umbrella</LI>
            <LI>The Cherry Tree</LI>
        </OL>
    <LI>Panchatantra Stories</LI>
    <LI>Barbie Story Collection</LI>
    <LI>Peppa Pig</LI>
</OL>
```

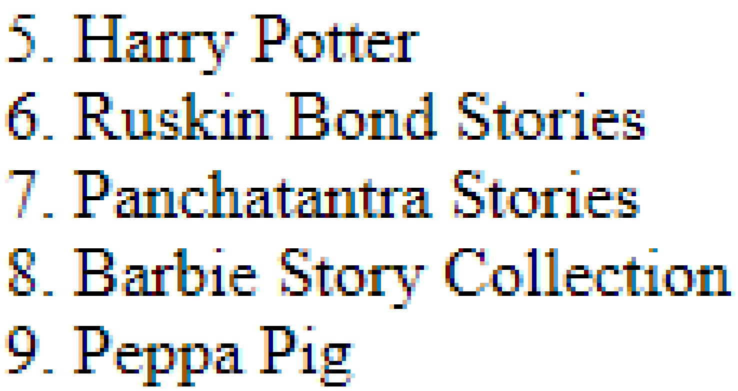
OL-Nested List



1. Harry Potter
2. Ruskin Bond Stories
    i. Crazy Times With Uncle Ken
    ii. Rusty, the Boy from the Hills
    iii. The Day Grandfather Tickled a Tiger
    iv. The Room On The Roof
    v. The Blue Umbrella
    vi. The Cherry Tree
3. Panchatantra Stories
4. Barbie Story Collection
5. Peppa Pig

Q: How would one start an ordered list with something other than 1

Browser Display



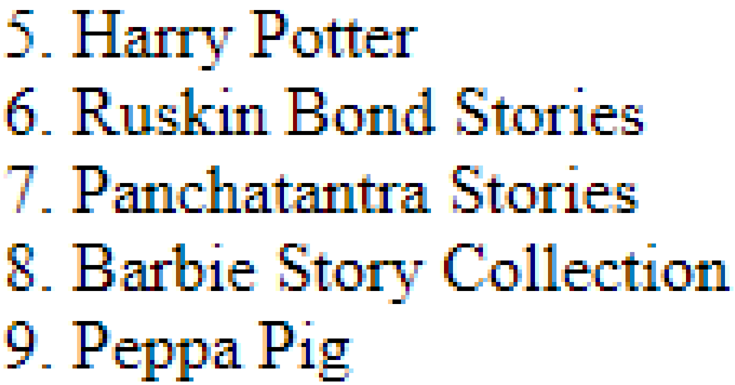
5. Harry Potter
6. Ruskin Bond Stories
7. Panchatantra Stories
8. Barbie Story Collection
9. Peppa Pig

OL start=“5”

HTML Code

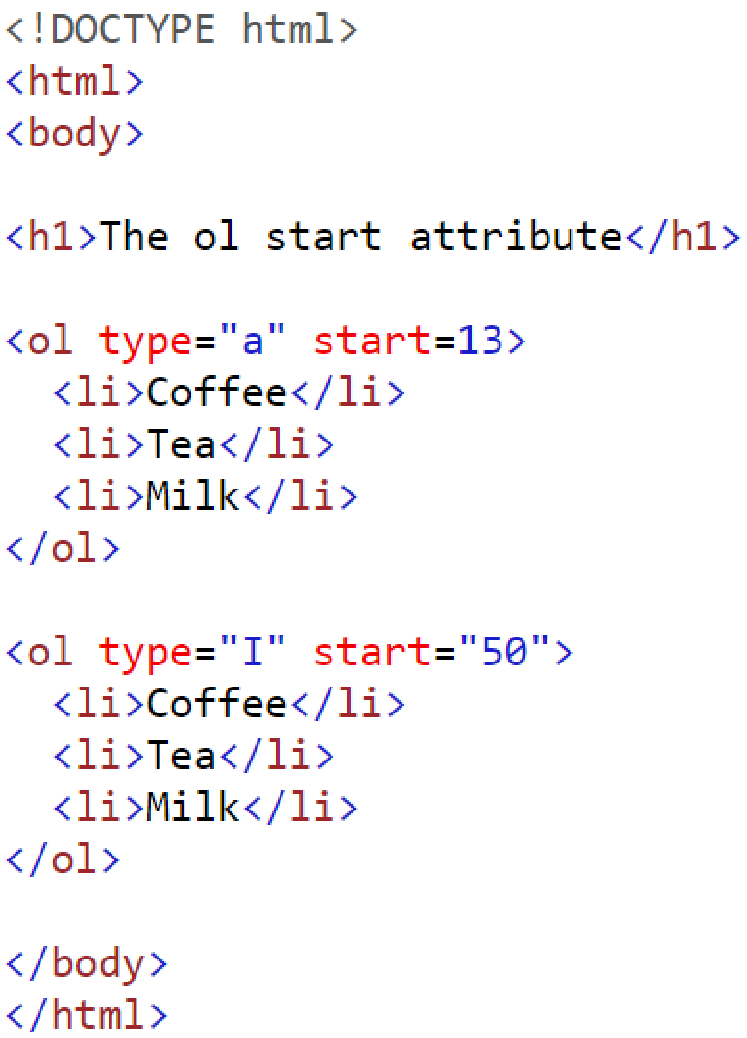
```
<OL start= “5” >
  <LI>Harry Potter</LI>
  <LI>Ruskin Bond Stories</LI>
  <LI>Panchatantra Stories</LI>
  <LI>Barbie Story Collection</LI>
  <LI>Peppa Pig</LI>
</OL>
```

Browser Display

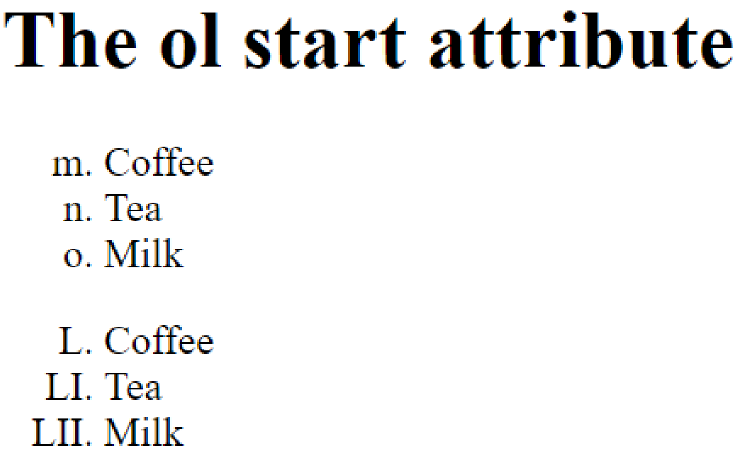


5. Harry Potter
6. Ruskin Bond Stories
7. Panchatantra Stories
8. Barbie Story Collection
9. Peppa Pig

OL start



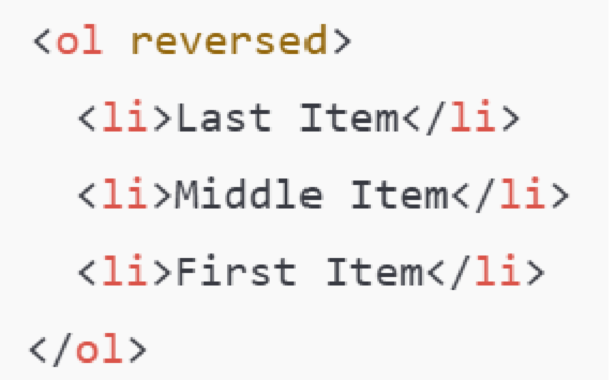
```
<!DOCTYPE html>
<html>
<body>

<h1>The ol start attribute</h1>

<ol type="a" start=13>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

<ol type="I" start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

</body>
</html>
```



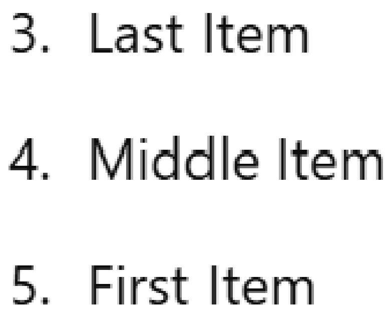
```
The ol start attribute

   m. Coffee
    n. Tea
    o. Milk

    L. Coffee
   LI. Tea
  LII. Milk
```

reversed Attribute

- Reverses the order of numbering, starting from the highest number.



```
<ol reversed>
  <li>Last Item</li>
  <li>Middle Item</li>
  <li>First Item</li>
</ol>
```



```
3. Last Item
4. Middle Item
5. First Item
```

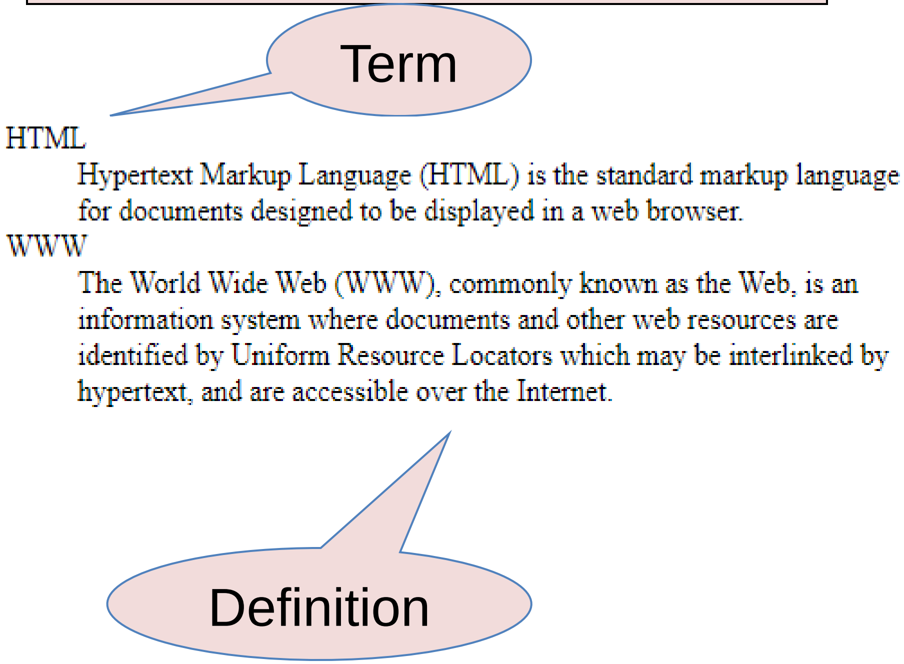
Definition List

<DL>	Definition List
<DT>	Term
<DD>	Definition

Definition List

```
<DL>
  <DT>HTML</DT>
   <DD>Hypertext Markup Language
     (HTML) is the standard markup
     language for documents designed
     to be displayed in a web
     browser. </DD>
  <DT>WWW</DT>
   <DD> The World Wide Web
     (WWW), commonly known as the
     Web, is an information system
     where documents and other web
     resources are identified by
     Uniform Resource Locators which
     may be interlinked by hypertext,
     and are accessible over the
     Internet.</DD>
</DL>
```

Browser Display



HTML

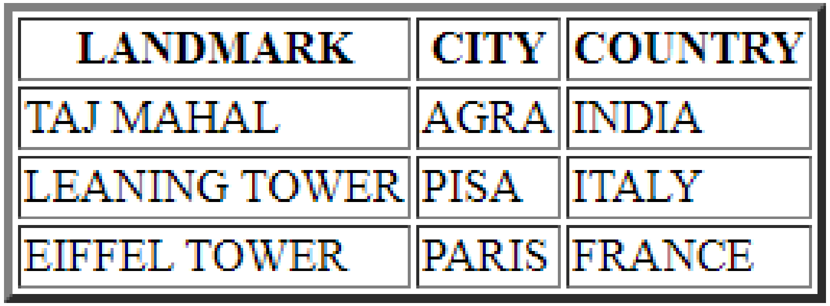
  Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser.

WWW

  The World Wide Web (WWW), commonly known as the Web, is an information system where documents and other web resources are identified by Uniform Resource Locators which may be interlinked by hypertext, and are accessible over the Internet.

HTML tables

Browser Display



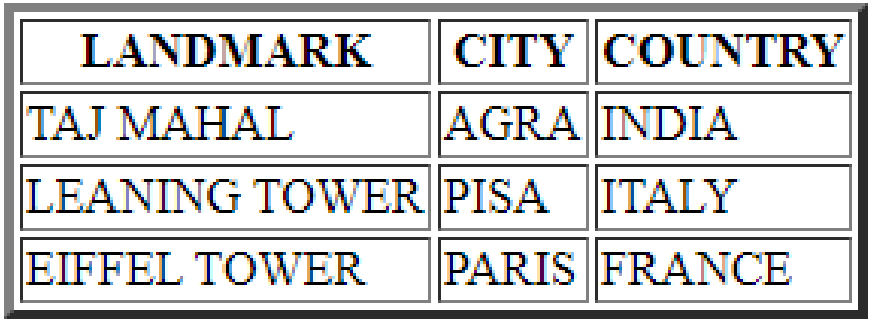
LANDMARK	CITY	COUNTRY
TAJ MAHAL	AGRA	INDIA
LEANING TOWER	PISA	ITALY
EIFFEL TOWER	PARIS	FRANCE

<TABLE>	Table (made up of rows)
<TR>	Row (made up of data cells)
<TH>	Heading Data Cell (Can contain paragraphs, images, lists, forms, tables)
<TD>	Data Cell (Can contain paragraphs, images, lists, forms, tables)

HTML Code

```
<TABLE BORDER=3>
<TR>
   <TH>LANDMARK</TH>
   <TH>CITY</TH>
   <TH>COUNTRY</TH>
</TR>
<TR>
   <TD>TAJ MAHAL</TD>
   <TD>AGRA</TD>
   <TD>INDIA</TD>
</TR>
<TR>
   <TD>LEANING TOWER</TD>
   <TD>PISA</TD>
   <TD>ITALY</TD>
</TR>
<TR>
   <TD>EIFFEL TOWER</TD>
   <TD>PARIS</TD>
   <TD>FRANCE</TD>
</TR>
</TABLE>
```

Browser Display

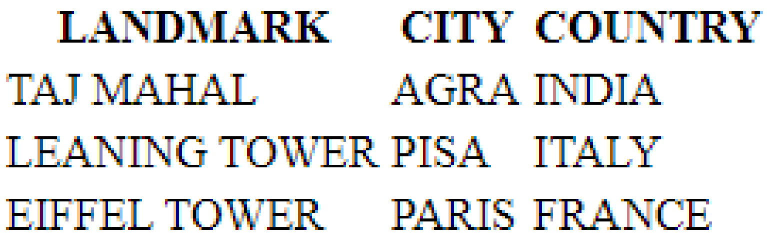


LANDMARK	CITY	COUNTRY
TAJ MAHAL	AGRA	INDIA
LEANING TOWER	PISA	ITALY
EIFFEL TOWER	PARIS	FRANCE

HTML Code

```
<TABLE>
<TR>
   <TH>LANDMARK</TH>
   <TH>CITY</TH>
   <TH>COUNTRY</TH>
</TR>
<TR>
   <TD>TAJ MAHAL</TD>
   <TD>AGRA</TD>
   <TD>INDIA</TD>
</TR>
<TR>
   <TD>LEANING TOWER</TD>
   <TD>PISA</TD>
   <TD>ITALY</TD>
</TR>
<TR>
   <TD>EIFFEL TOWER</TD>
   <TD>PARIS</TD>
   <TD>FRANCE</TD>
</TR>
</TABLE>
```

Browser Display



LANDMARK	CITY	COUNTRY
TAJ MAHAL	AGRA	INDIA
LEANING TOWER	PISA	ITALY
EIFFEL TOWER	PARIS	FRANCE

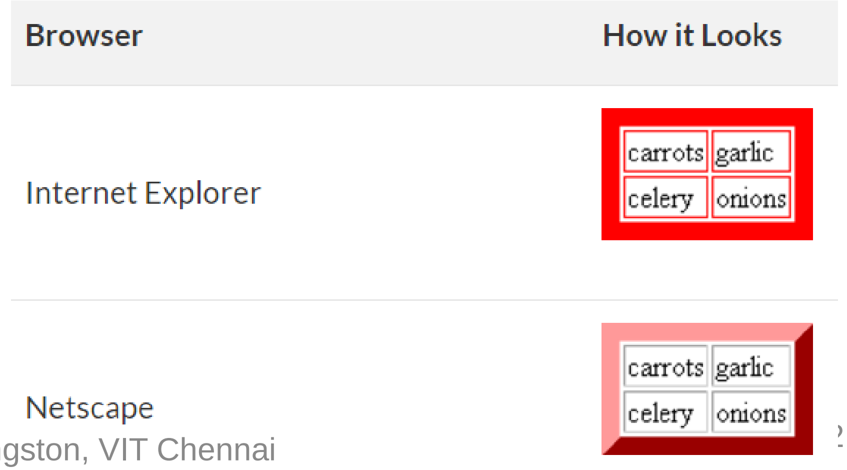
Table without Border

<TABLE> Attributes

- BORDER
  - Determines the thickness of the table border
  - Example: <TABLE BORDER = “2”>

- BORDERCOLOR
  - The color of the table borders as a whole is set with the BORDERCOLOR attribute

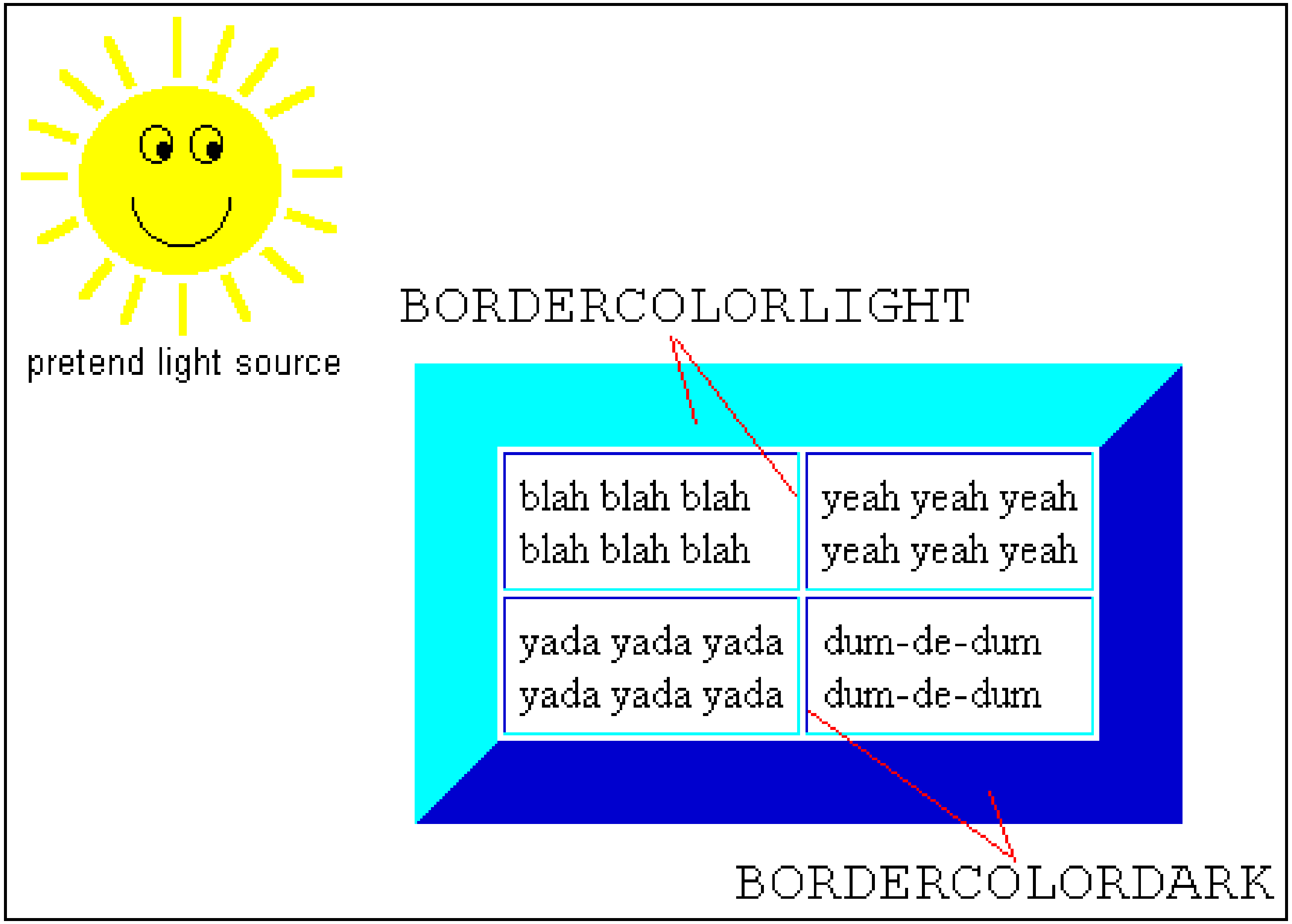
```
<TABLE BORDER=10 BORDERCOLOR=RED>
```

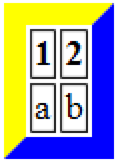


Browser	How it Looks
Internet Explorer	carrots garlic celery onions
Netscape	carrots garlic celery onions

Netscape maintains the 3-D appearance.

BORDERCOLORLIGHT, BORDERCOLORDARK



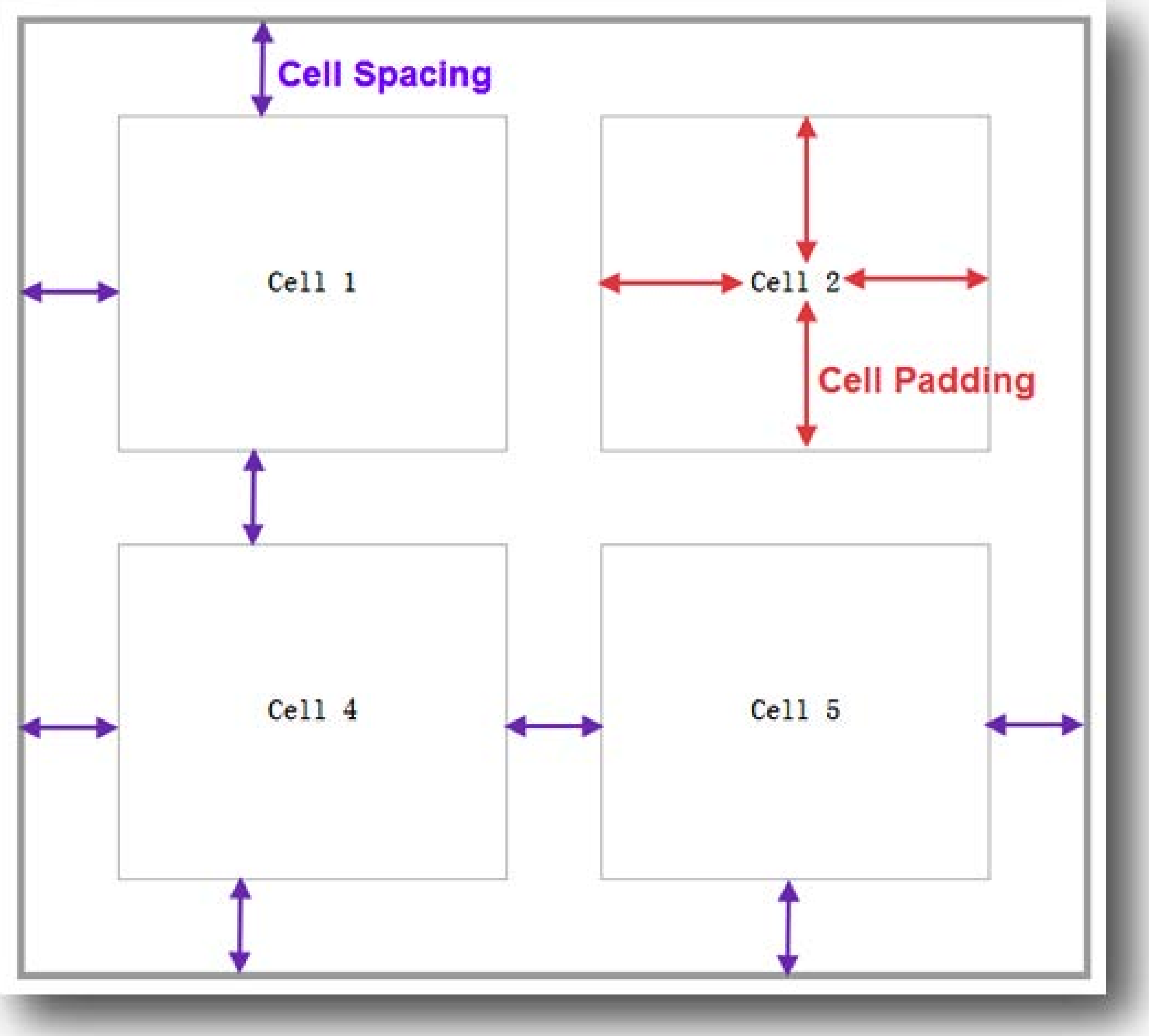


```
<TABLE BORDER=10 BORDERCOLORLIGHT=YELLOW BORDERCOLORDARK=BLUE>
```

<TABLE> Attributes

- CELLPADING
  - Determines the distance between the border of a cell and the contents of the cell
  - Example: <TABLE CELLPADDING = “3”>

- CELLSPACING
  - Determines the empty spacing between the borders of two adjacent cells
  - Example: <TABLE CELLSPACING = “1”>


Cell Spacing
Cell 1
Cell 2
Cell Padding
Cell 4
Cell 5

<TABLE>,<TR>,<TH>,<TD> Attributes

- ALIGN
  - Possible values: Center, Left, Right
  - Example: <TH ALIGN = “center”>
- BGCOLOR
  - Example: <TH BGCOLOR = “green”>
- BORDERCOLOR
  - Example: <TD BORDERCOLOR = “red”>
- WIDTH
  - Example: <TR WIDTH = “40%”>
- HEIGHT
  - Example: <TABLE HEIGHT = “200”>

<TR> Attributes

- VLAIGN
  - Determines the vertical alignment of the contents of all of the cells in a particular row
  - Possible values: Top, Middle, Bottom
  - Example: <TR VALIGN = “bottom”>

<TH> & <TD> Attributes

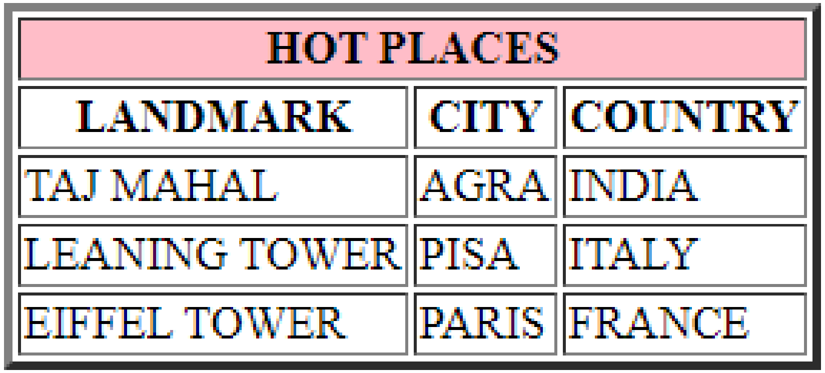
- NOWRAP
  - Extend the width of a cell, if necessary, to fit the contents of the cell in a single line
  - Example: <TD NOWRAP>
- COLSPAN
  - No. of rows the current cell should extend itself downward
  - Example: <TD COLSPAN = “2”>
- ROWSPAN
  - The number of columns the current cell should extend itself
  - Example: <TD ROWSPAN = “5”>
- VALIGN
  - Same as that for <TR>

HTML Code

```
<TABLE BORDER=3>
<TR BGCOLOR="PINK">
   <TH COLSPAN=3>HOT PLACES</TH>
</TR>
<TR>
   <TH>LANDMARK</TH>
   <TH>CITY</TH>
   <TH>COUNTRY</TH>
</TR>
<TR>
   <TD>TAJ MAHAL</TD>
   <TD>AGRA</TD>
   <TD>INDIA</TD>
</TR>
<TR>
   <TD>LEANING TOWER</TD>
   <TD>PISA</TD>
   <TD>ITALY</TD>
</TR>
<TR>
   <TD>EIFFEL TOWER</TD>
   <TD>PARIS</TD>
   <TD>FRANCE</TD>
</TR>
</TABLE>
```

colspan

Browser Display



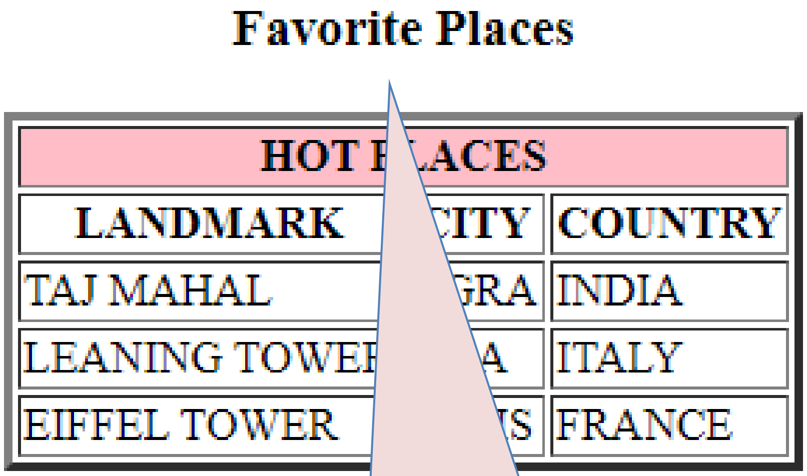
HOT PLACES		
LANDMARK	CITY	COUNTRY
TAJ MAHAL	AGRA	INDIA
LEANING TOWER	PISA	ITALY
EIFFEL TOWER	PARIS	FRANCE

HTML Code

```
<TABLE BORDER=3>
<CAPTION><h3>Favorite Places</h3>
</CAPTION>
<TR BGCOLOR="PINK" >
   <TH COLSPAN=3>HOT PL          H>
</TR>
<TR>
   <TH>LANDMARK</TH>
   <TH>CITY</TH>
   <TH>COUNTRY</TH>
</TR>
<TR>
   <TD>TAJ MAHAL</TD>
   <TD>AGRA</TD>
   <TD>INDIA</TD>
</TR>
<TR>
   <TD>LEANING TOWER</TD>
   <TD>PISA</TD>
   <TD>ITALY</TD>
</TR>
<TR>
   <TD>EIFFEL TOWER</TD>
   <TD>PARIS</TD>
   <TD>FRANCE</TD>
</TR>
</TABLE>
```

Caption

Browser Display



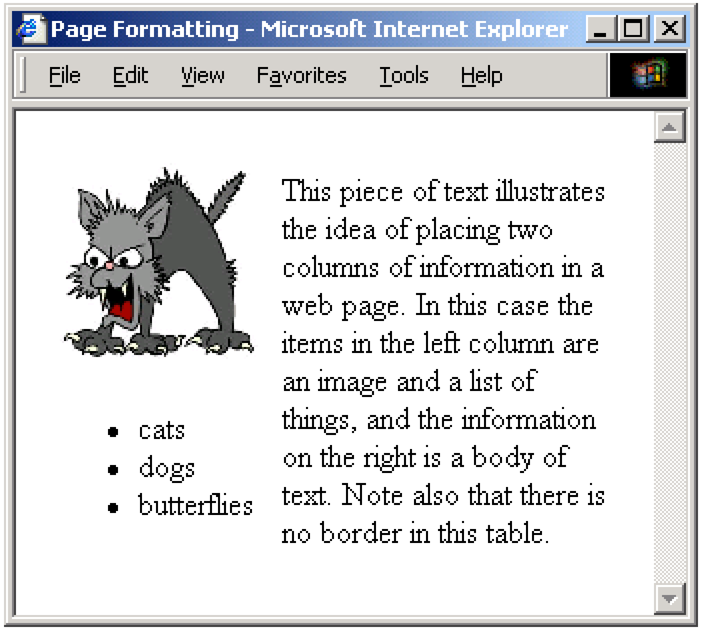
Favorite Places

HOT P LACES		
LANDMARK	ITY	COUNTRY
TAJ MAHAL	GRA	INDIA
LEANING TOWEF	A	ITALY
EIFFEL TOWER	IS	FRANCE

Caption. Placed immediately after the<TABLE> tag

Page formatting

- Tables can be used to organise the layout of the web page itself

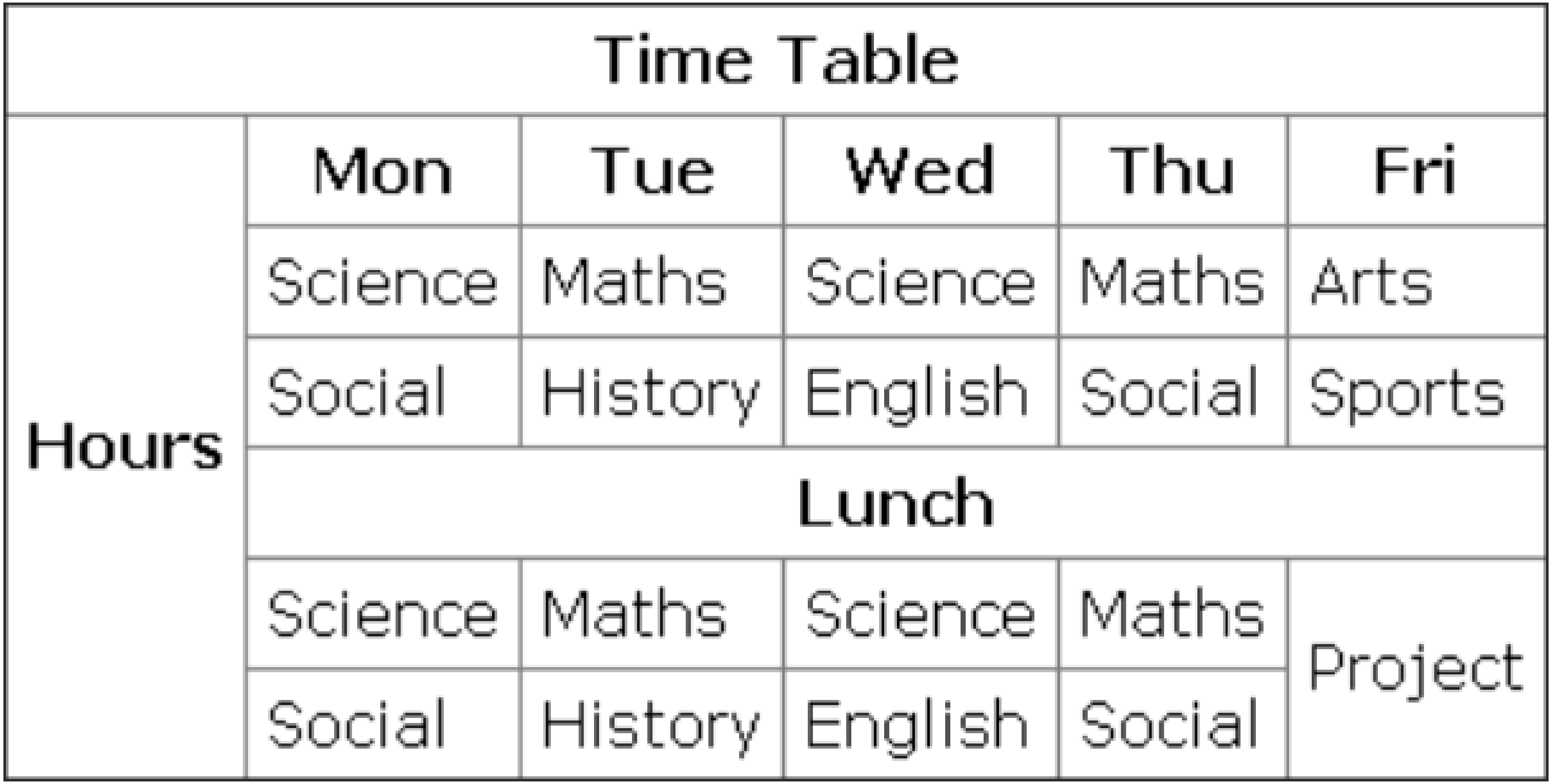


- cats
- dogs
- butterflies

This piece of text illustrates the idea of placing two columns of information in a web page. In this case the items in the left column are an image and a list of things, and the information on the right is a body of text. Note also that there is no border in this table.

```
</body>
 <table border="0" cellspacing="10">
  <tr>
   <td>
    <img src="cat.gif" alt="cat">
    <ul>
     <li>cats</li>
     <li>dogs</li>
     <li>butterflies</li>
    </ul>
   </td>
   <td>
    This piece of text illustrates
    the idea of placing two columns
    of information in a web page...
    Note also that there is no
    border in this table.
   </td>
  </tr>
 </table>
</body>
```

Year	Quarter	Expenses		Income	
		Quetta	Dubai	Quetta	Dubai
2021	1	1,900	8,650	9,000	7,780
	2	2,230	8,650	8,500	8,670
	3	4,000	8,650	9,900	9,870
	4	2,200	8,650	9,800	9,900
2022	1	7,780	8,650	7,780	9,000
	2	8,670	8,650	8,670	8,500
	3	9,870	8,650	9,870	9,900
	4	9,900	8,650	9,900	9,800
2013	****				



Time Table					
Hours	Mon	Tue	Wed	Thu	Fri
	Science	Maths	Science	Maths	Arts
	Social	History	English	Social	Sports
	Lunch				
	Science	Maths	Science	Maths	Project
	Social	History	English	Social	

References

HTML Lists & Tables
www.altafkhan.com › html-lists-tables

Image Maps

Dr. L.M. Jenila Livingston

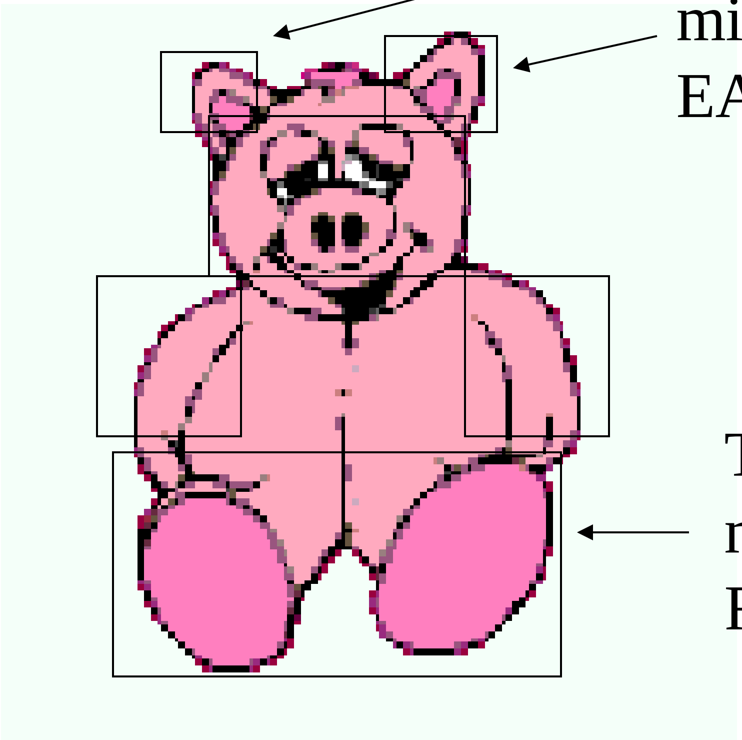
VIT Chennai

Image Maps

- An image that contains hyperlinked, clickable regions, sometimes called "hot spots"
- Each hot spot is defined by a set of coordinates (indicating its position on the image) and a URL reference
- Two types of image maps:
  - Client-side (commonly used)
  - Server-side (rarely used)

Image Maps

- If you could see an image with its map, it might look like this:



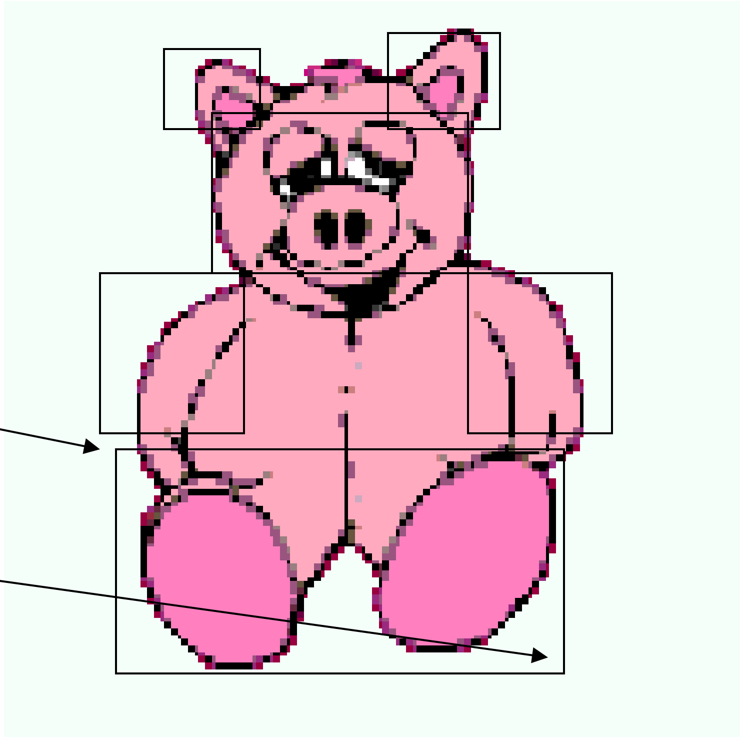
These areas might link to EAR.HTM

Each area could be a separate link.

This area might link to FEET.HTM

Image Maps

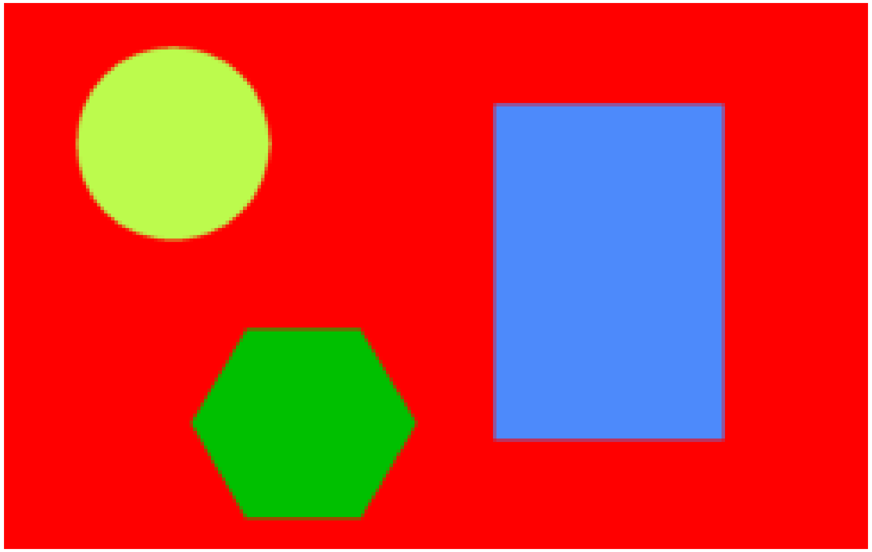
- Each area is defined by coordinates in pixels.
- For rectangular areas, it is the upper left and lower right.



This area might be defined as 2,200 (upper left) 205,300 (lower right)

Defining Hot Spots

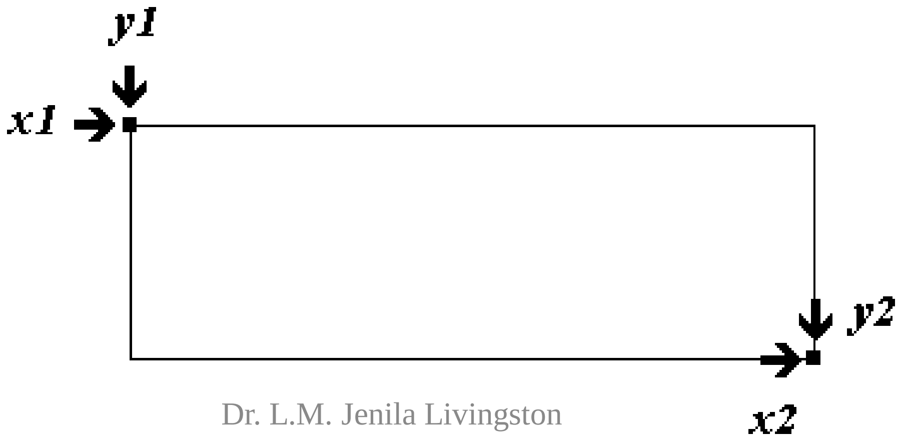
- Three shapes for hot spots:
  - – Rectangle
  - – Circle
  - – Polygon



Rectangle Hot Spot

- Any two points can define a rectangle
- Each point is represented by a horizontal (x) coordinate and a vertical (y) coordinate
- Rectangles are defined by four coordinate values representing the upper-left and bottom-right corners of the rectangle
  - Code:

```
<area shape="rect" coords="1,52,33,96" href="hand.htm“ />
```

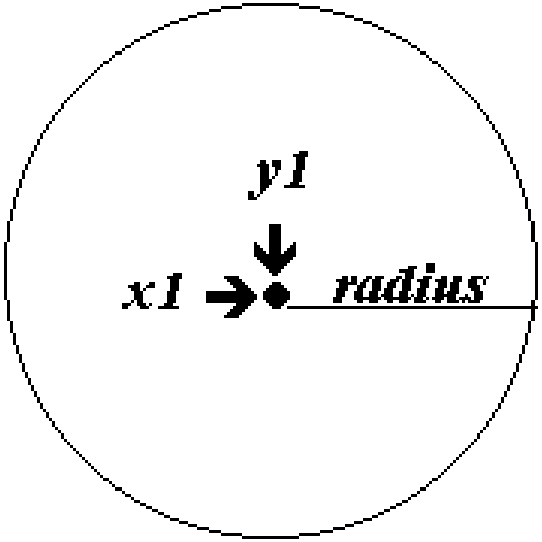


Circle Hot Spot

- Circles are defined by two coordinates and a radius
  - Code:

```
<area shape="circle" coords="x1,y1,radius" href="url"/>
```

- The pair of coordinates speci the circle's center
- A third number specifies the desired radius



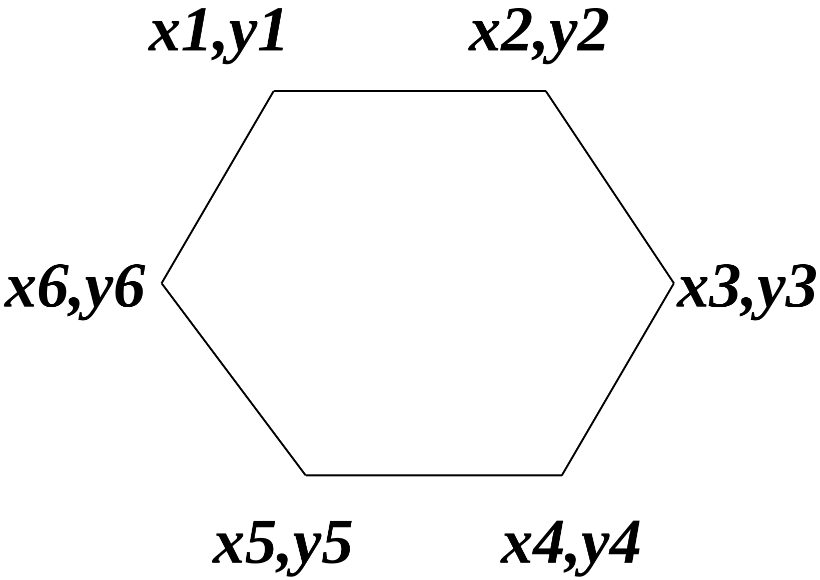
Polygon Hot Spot

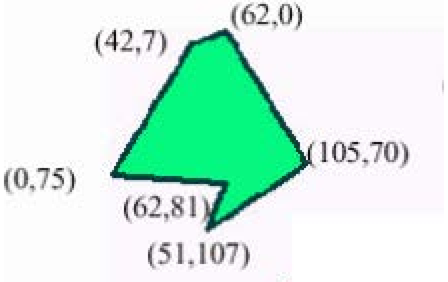
<AREA SHAPE="poly"

COORDS="*x1,y1,x2,y2,x3,y3,x4,y4,...*"

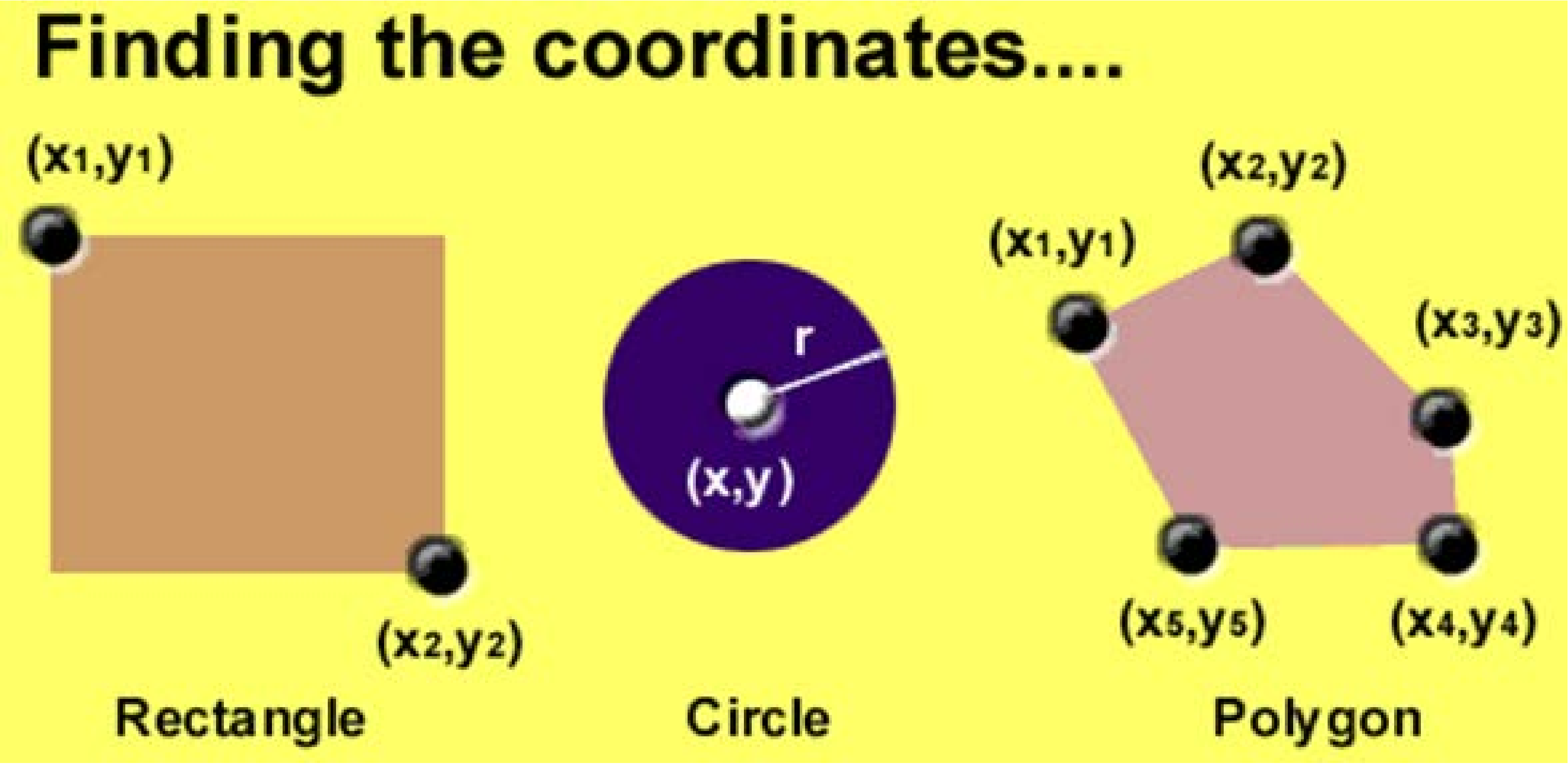
HREF="*url*">

- Replace *x1,y1* with two numbers: horizontal and vertical position of a point on the edge of the polygon
- Add more *x,y* pairs until you have traced around the desired shape





3 types of Hot Spot



Four Steps to Create a Client-Side Image Map

1. Select image to use
2. Define areas of image map
   - Three shapes: circle, rectangle, polygon
   - X and Y coordinates in pixels
   - Get coordinates from a graphics program such as Paint Shop Pro or Photoshop
   - <AREA> tags identify shapes in map
     - SHAPE attribute – circle, rect, poly
     - HREF attribute – URL to load
     - COORDS – x and y coordinates of shape

Four Steps to Create a Client-Side Image Map

3. Include map information in HTML document:
   - <MAP></MAP> - NAME attribute names the map
   - <MAP NAME="books">
4. Connect image with map information
   - USEMAP attribute in <IMG> tag
   - USEMAP value is map name with a # symbol before it
   - <img src="image.gif" usemap="#books">

Creating Client-Side Image Maps

- Can use target=window_name for frames

```
<map name="books">
<area shape="poly" coords="70,0, 0,37, 0,183, 27,192,
   27,48, 103, 9" href="file.html“ alt=“Books”
   title=“Books”>
</map>

<img src="image.gif" usemap="#books">
```

alt: Specifies an alternate text.

title: hover: display the text on hover

onClick=fun(): Include Script function

Example 1

Example 1

```
<img src="tmahal.jpeg" usemap = #imgmap
   border=0>
   <map name=imgmap>
   <area shape=Rect Coords=30,30,59,59
   Href="taj.jpg" Target="_blank">
   <area shape=Rect Coords=100,60,129,80
   Href="elephant.gif" Target="_blank" >
   </map>
```

Find Coordinates

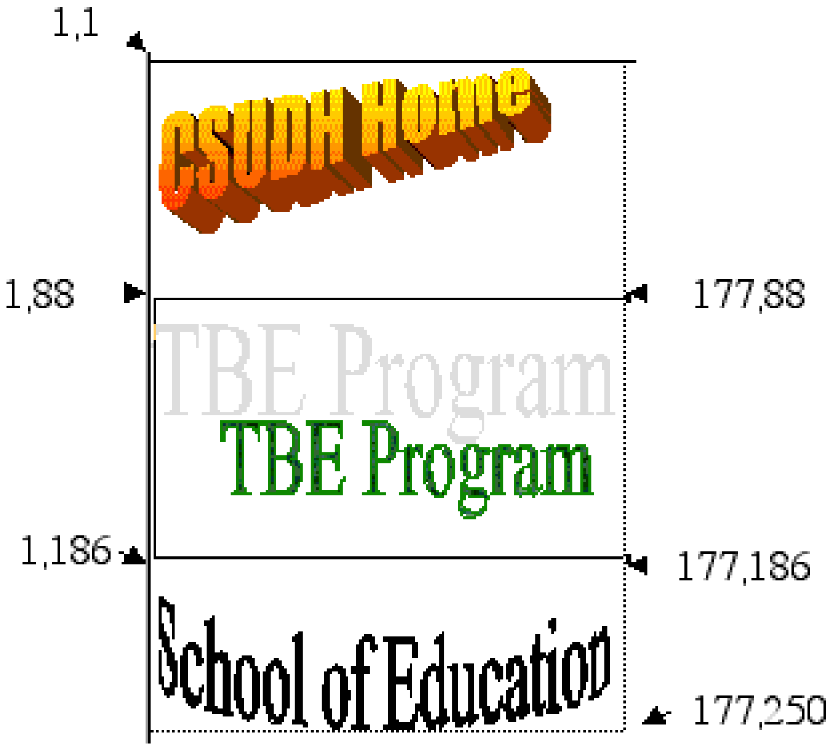
- use a graphics program, if it shows the position of the cursor with “pixels”.

- Move the cursor over the picture and write down the appropriate coordinates.

- https://imagemap.org/
- https://www.image-map.net/

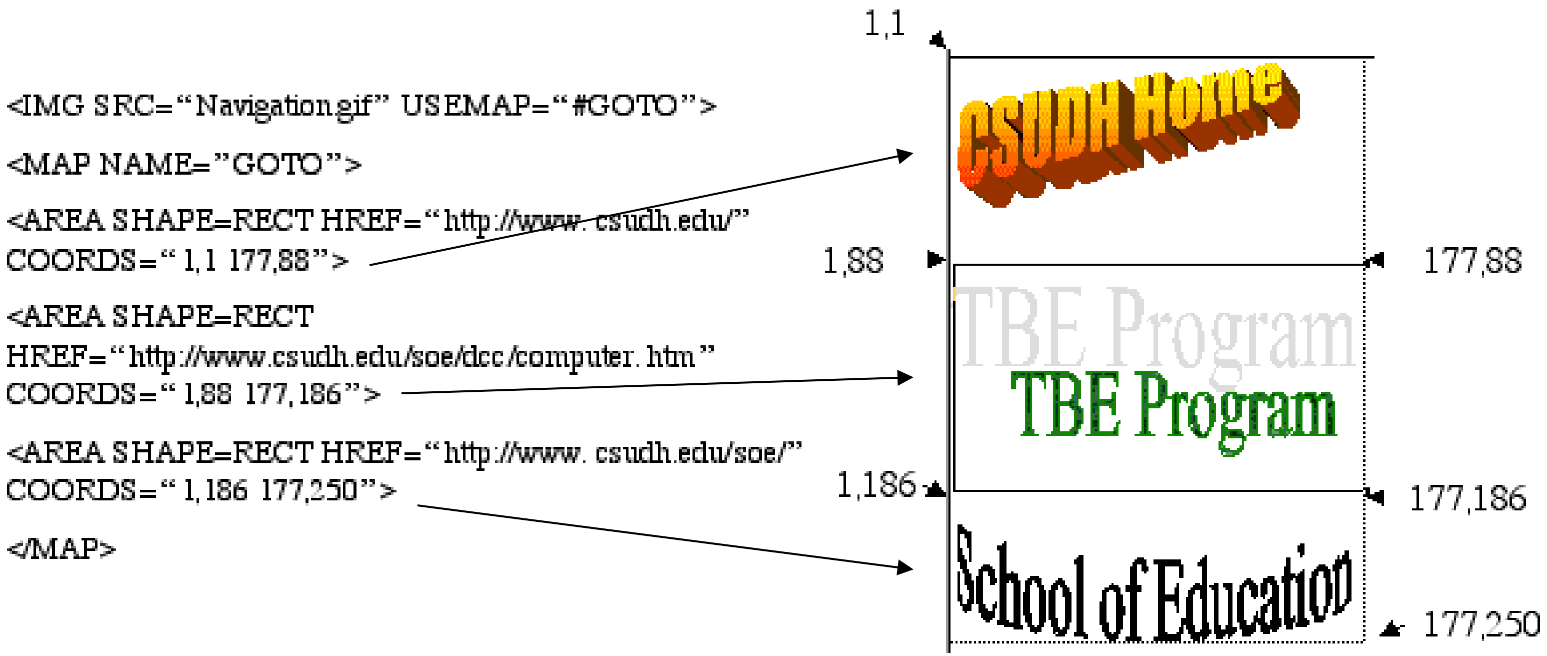
Example 2

- Here is a simpler graphic with coordinates:



Example 2

- The accompanying HTML code for the image map:



```
<IMG SRC=“Navigation.gif” USEMAP=“#GOTO”>
<MAP NAME=”GOTO”>
<AREA SHAPE=RECT HREF=“http://www. csudh.edu/”
COORDS=“ 1,1 177,88”>
<AREA SHAPE=RECT
HREF=“http://www.csudh.edu/soe/dcc/computer. htm”
COORDS=“ 1,88 177,186”>
<AREA SHAPE=RECT HREF=“http://www. csudh.edu/soe/”
COORDS=“ 1,186 177,250”>
</MAP>
```

HTML Media elements

Dr. L.M. Jenila Livingston

Professor

VIT Chennai

Multimedia Elements

- Video
- Audio
- Graphics
  - Canvas
  - **SVG**
- Google Map

Multimedia Elements

- Multimedia on the web is sound, music, videos, movies, and animations.

- Different formats:
- images, music, sound, videos, records, films, animations, and more.

HTML5: Video

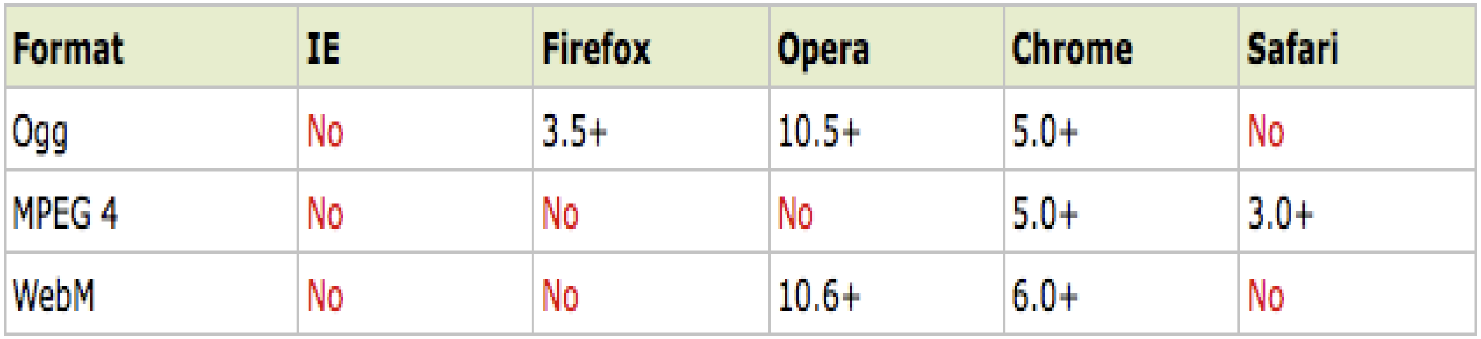
Until now, there hasn't been a standard for showing video on a web page.

Today, most videos are shown through a plugin (like Flash). However, not all browsers have the same plugins.

HTML5 specifies a standard way to include video with the video element.

HTML5: Video

Currently, there are 3 supported video formats for the video element:



Format	IE	Firefox	Opera	Chrome	Safari
Ogg	No	3.5+	10.5+	5.0+	No
MPEG 4	No	No	No	5.0+	3.0+
WebM	No	No	10.6+	6.0+	No

HTML5: Video

```
<video width="320" height="240" controls loop autoplay
muted poster="thumbnail.jpg">
  <source src="movie.ogg" type="video/ogg" />
  <source src="movie.mp4" type="video/mp4" />
  <source src="movie.webm" type="video/webm" />
Your browser does not support the video tag.
</video>
```

Note: To ensure that your video works across all browsers, it's a good idea to include multiple formats of the same video.

Video Attributes

- **width and height**: These attributes control the size of the video player.
- **controls**: This attribute adds play, pause, volume, and full-screen options to the video player.
- **autoplay**: Starts playing the video as soon as it is ready.
- **loop**: Loops the video when it reaches the end.
- **muted**: Starts the video with the audio muted.
- **poster**: Specifies an image to be shown before the video starts
- **<source>**: Specifies the video file to be used.

```
<html>
<body>
<button onclick="playVid()" type="button">Play Video</button>
<button onclick="pauseVid()" type="button">Pause Video</button>
<video id="myVideo" width="320" height="176">
  <source src="mov_bbb.mp4" type="video/mp4">
  <source src="mov_bbb.ogg" type="video/ogg">
  Your browser does not support HTML5 video.
</video>

<script>
var vid = document.getElementById("myVideo");
function playVid() {
  vid.play();
}
function pauseVid() {
  vid.pause();
}
</script>
</body>
</html>
```

HTML5: Audio

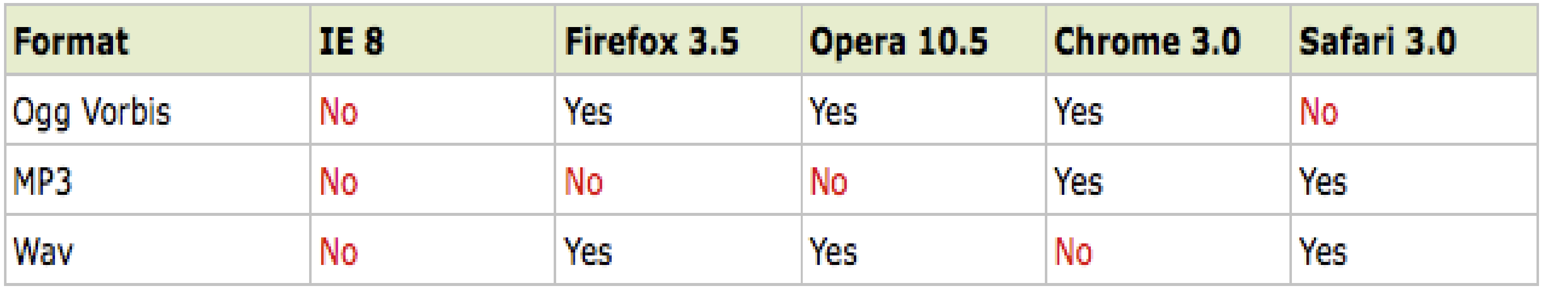
Until now, there has never been a standard for playing audio on a web page.

Today, most audio is played through a plugin (like Flash). However, not all browsers have the same plugins.

HTML5 specifies a standard way to include audio, with the audio element. The audio element can play sound files, or an audio stream.

HTML5: Audio

Currently, there are 3 supported formats for the audio element:



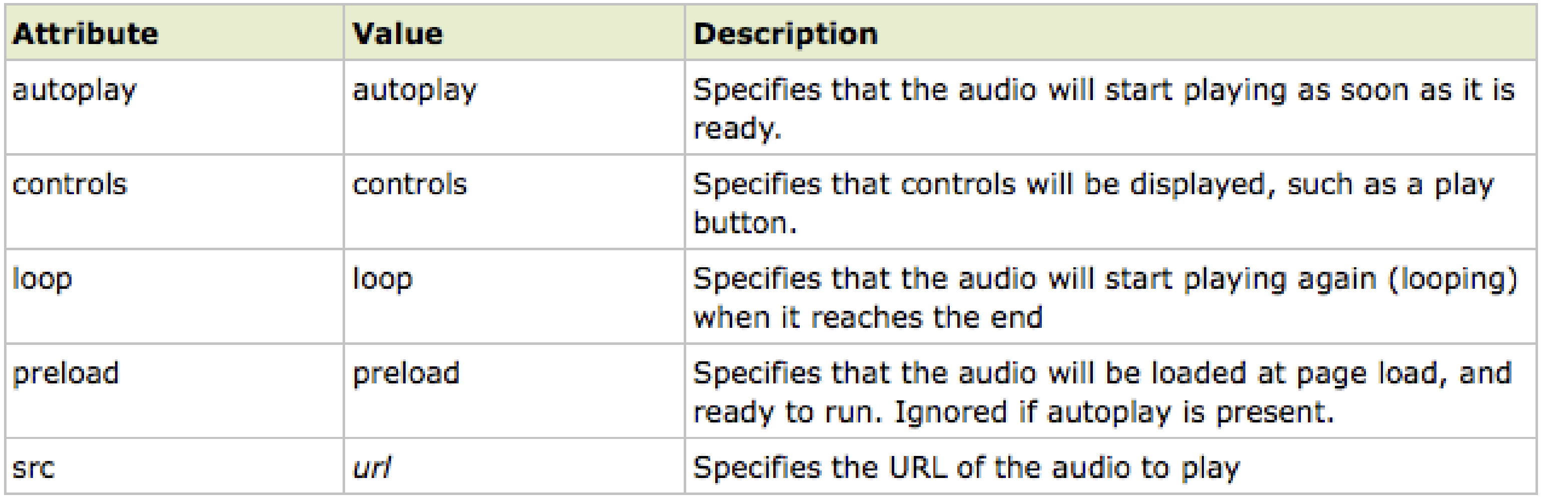
Format	IE 8	Firefox 3.5	Opera 10.5	Chrome 3.0	Safari 3.0
Ogg Vorbis	No	Yes	Yes	Yes	No
MP3	No	No	No	Yes	Yes
Wav	No	Yes	Yes	No	Yes

HTML5: Audio

```
<audio controls="controls">
  <source src="song.ogg" type="audio/ogg" />
  <source src="song.mp3" type="audio/mpeg" />
Your browser does not support the audio element.
</audio>
```

Note: To ensure that your audio works across all browsers, it's a good idea to include multiple formats of the same audio.

HTML5: Audio



Attribute	Value	Description
autoplay	autoplay	Specifies that the audio will start playing as soon as it is ready.
controls	controls	Specifies that controls will be displayed, such as a play button.
loop	loop	Specifies that the audio will start playing again (looping) when it reaches the end
preload	preload	Specifies that the audio will be loaded at page load, and ready to run. Ignored if autoplay is present.
src	*url*	Specifies the URL of the audio to play

HTML5 Graphics

Tag	Description
<canvas>	Defines graphic drawing using JavaScript
<svg>	Defines graphic drawing using SVG

Canvas

The HTML5 canvas element uses JavaScript to draw graphics on a web page.

A canvas is a rectangular area, and you control every pixel of it.

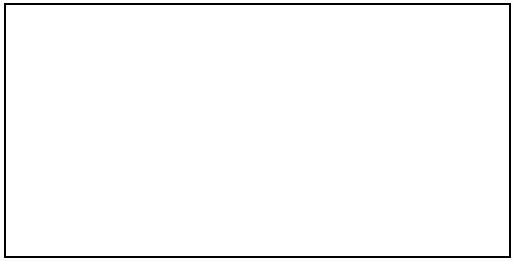
The canvas element has several methods for drawing paths, boxes, circles, characters, and adding images.

Canvas

Adding a canvas element to the HTML5 page.

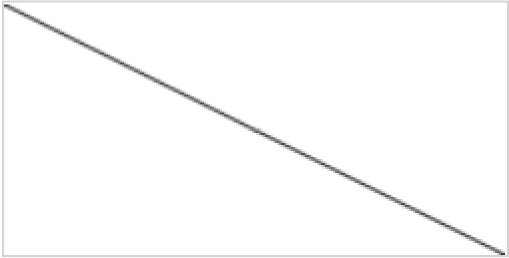
- Specify the id, width, height of the element
- To add a border, use the style attribute.

```
<canvas id="myCanvas" width="200" height="100"
style="border:1px solid #000000;">
</canvas>
```



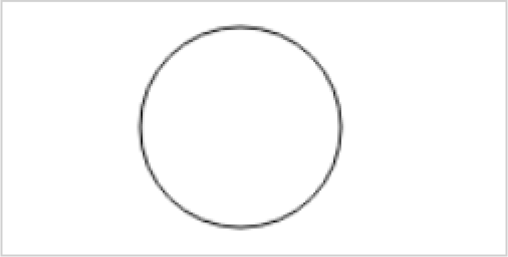
Draw a line

```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0,0);
ctx.lineTo(200,100);
ctx.stroke();
```



Draw a Circle

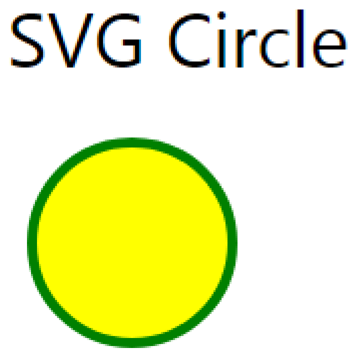
```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95,50,40,0,2*Math.PI);
ctx.stroke();
```



SVG

Scalable Vector Graphics

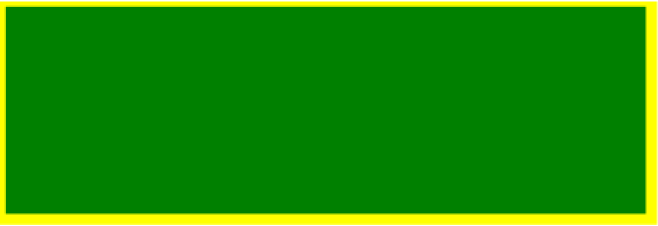
```
<!DOCTYPE html>
<html>
<body>

<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green"
 stroke-width="4"fill="yellow" />
</svg>

</body>
</html>
```



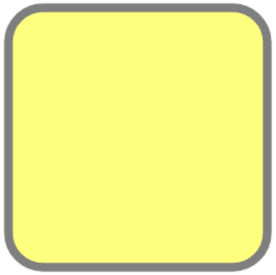
SVG Circle

SVG

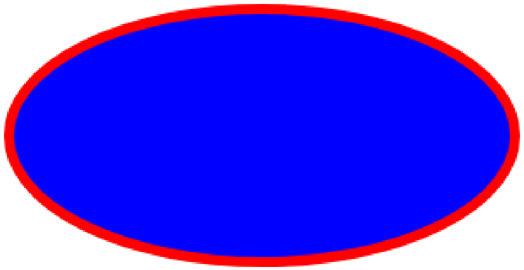
```
<svg width="400" height="110">
  <rect x="50" y="130“ width="300" height="100"
style="fill:green;stroke-width:5;stroke:yellow" />
</svg>
```



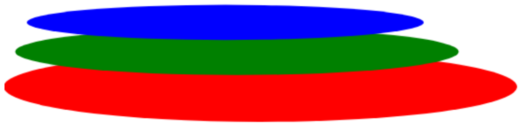
```
<svg width="400" height="180">
  <rect x="50" y="20" rx="20" ry="20" width="150"
height="150" style="fill:yellow;stroke:black;stroke-
width:5;opacity:0.5" />
</svg>
```



```
<svg height="200" width="500">
  <ellipse cx="200" cy="80" rx="100" ry="50"
style="fill:blue;stroke:red;stroke-width:4" />
</svg>
```

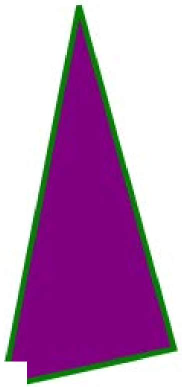


```
<svg height="150" width="500">
  <ellipse cx="240" cy="100" rx="220" ry="30" style="fill:red" />
  <ellipse cx="220" cy="70" rx="190" ry="20" style="fill:green" />
  <ellipse cx="210" cy="45" rx="170" ry="15" style="fill:blue" />
</svg>
```

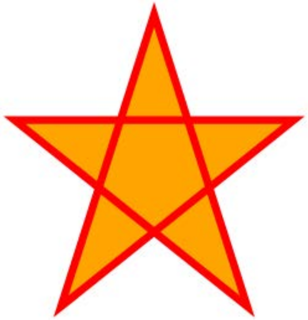


SVG

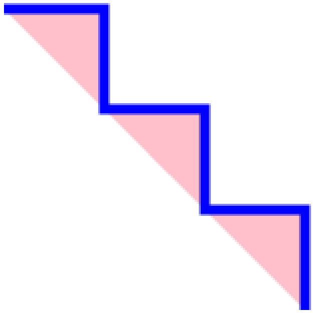
```
<svg height="250" width="500">
  <polygon points="200,10 250,190 160,210"
style="fill:purple;stroke:green;stroke-width:3" />
</svg>
```



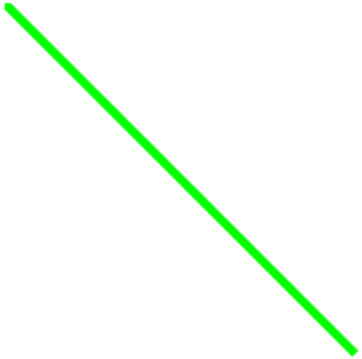
```
<svg height="250" width="500">
  <polygon points="100,10 40,198 190,78 10,78
160,198" style="fill:orange;stroke:red;stroke-
width:5;fill-rule:nonzero;"/>
</svg>
```



```
<svg height="180" width="500">
  <polyline points="0,40 40,40 40,80 80,80 80,120
120,120 120,160" style="fill:pink;stroke:blue;stroke-
width:4" />
</svg>
```

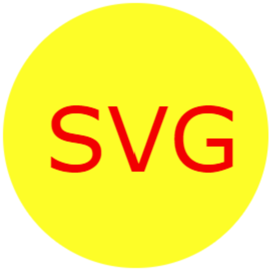


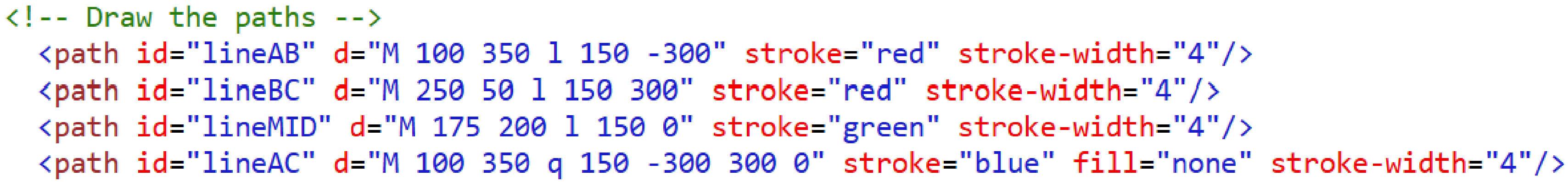
```
<svg height="250" width="500">
  <line x1="0" y1="0" x2="200" y2="200"
style="stroke:rgb(0,255,0);stroke-width:5" />
</svg>
```



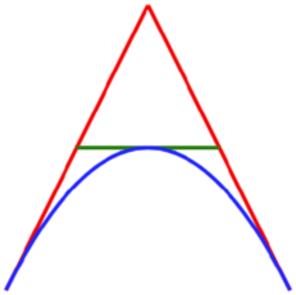
SVG

```
<SVG>
<circle cx="95" cy="66" r=“66" fill="yellow" />
<text fill="#ff0000" x="50" y="84" font-size="45"
font-family="Verdana">
</text>
</SVG>
```





```
<!-- Draw the paths -->
  <path id="lineAB" d="M 100 350 l 150 -300" stroke="red" stroke-width="4"/>
  <path id="lineBC" d="M 250 50 l 150 300" stroke="red" stroke-width="4"/>
  <path id="lineMID" d="M 175 200 l 150 0" stroke="green" stroke-width="4"/>
  <path id="lineAC" d="M 100 350 q 150 -300 300 0" stroke="blue" fill="none" stroke-width="4"/>
```



Coordinate points for SVG elements

Circle	
r	Required. The radius of the circle
cx, cy	center of the circle (x, y). Default is 0,0
Ellipse	
rx	Required. The x radius of the ellipse
ry	Required. The y radius of the ellipse
cx, cy	center of the ellipse (x, y). Default is 0,0
Rectangle	
width	Required. The width of the rectangle
height	Required. The height of the rectangle
x, y	The x-y position for the top-left corner of the rectangle
rx	The x radius of the corners of the rectangle (used to round the corners). Default is 0
ry	The y radius of the corners of the rectangle (used to round the corners). Default is 0

Coordinate points for SVG elements – contd..

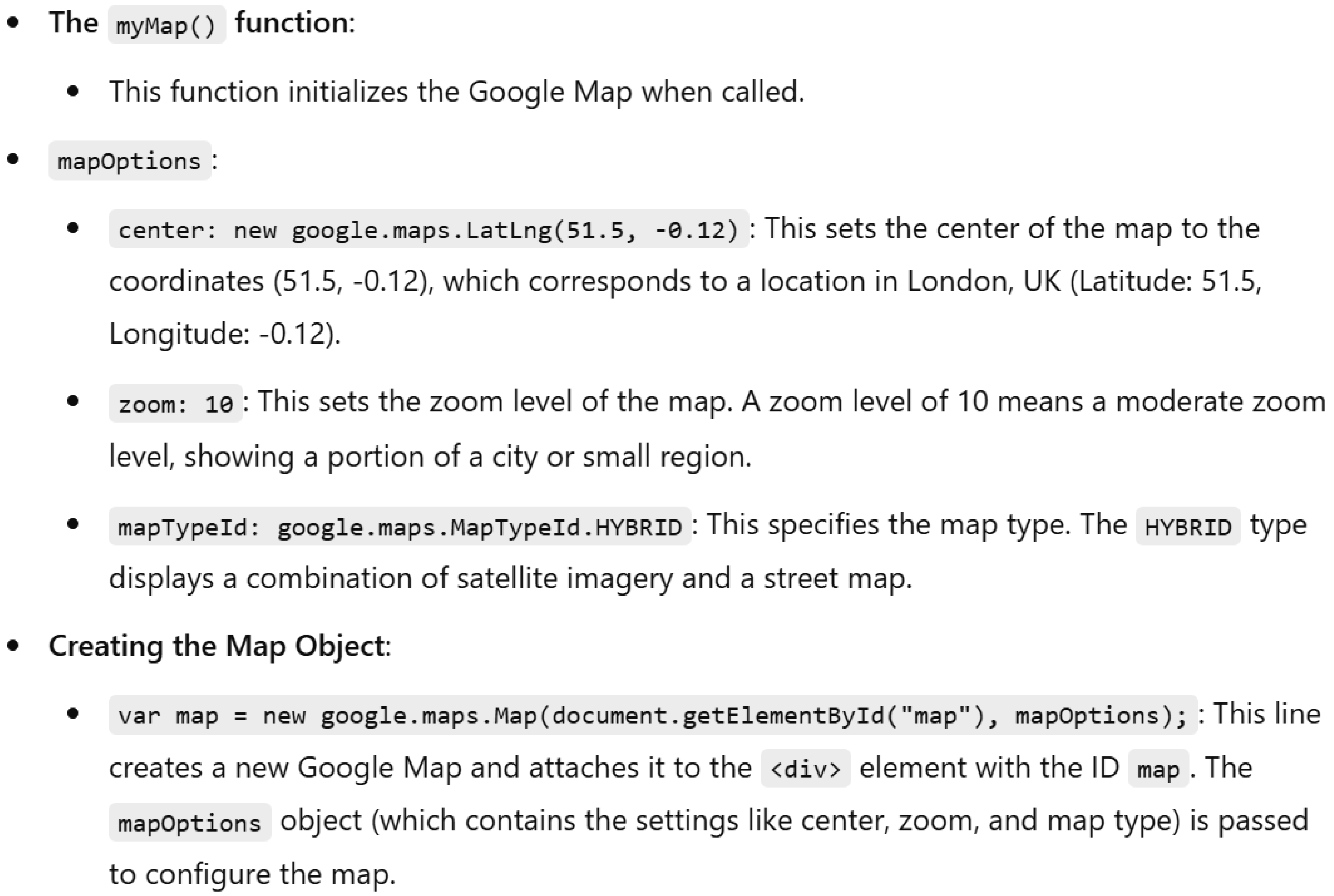
Line	
x1,y1	The starting point of the line
x2,y2	The ending point of the line
Polyline	
The <polyline> element is used to create any shape that consists of only straight lines.	
points	Required. xn,yn coordinates of different points to be given
Polygon	
The <polygon> element is used to create a graphic that contains at least three sides	
points	Required. xn,yn coordinates of different points to be given
Text	
x,y	The (x,y) position of the start of the text
rotate	The rotation (in degrees) applied to each letter of text
Path	
M: M50 50	Move the "pen" to the point (50, 50).
L: L150 150	Draw a line from the current position to the point (150, 150).
E	elliptical Arc (create a elliptical arc)
C	curveto (create a curve
Z	Close the path by drawing a line back to the starting point (50, 50)

Embedding Google map

```
<body>
<h1>My First Google Map</h1>
<div id="map" style="width:400px;height:400px;"></div>
<script>
function myMap() {
 var mapOptions = {
 center: new google.maps.LatLng(51.5, -0.12), // Latitude and Longitude for UK
 zoom: 10,
 mapTypeId: google.maps.MapTypeId.HYBRID
 }
var map = new google.maps.Map(document.getElementById("map"), mapOptions);
}
</script>
<!-- Add the Google Maps API script with your API key -->
<script
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=myMap">
</script>
</body>
```

```
LatLng(20.5937, 78.9629), // Latitude and Longitude for India
```

Details



- **The `myMap()` function**:
    - This function initializes the Google Map when called.
- `mapOptions`:
    - `center: new google.maps.LatLng(51.5, -0.12)`: This sets the center of the map to the coordinates (51.5, -0.12), which corresponds to a location in London, UK (Latitude: 51.5, Longitude: -0.12).
    - `zoom: 10`: This sets the zoom level of the map. A zoom level of 10 means a moderate zoom level, showing a portion of a city or small region.
    - `mapTypeId: google.maps.MapTypeId.HYBRID`: This specifies the map type. The `HYBRID` type displays a combination of satellite imagery and a street map.
- **Creating the Map Object**:
    - `var map = new google.maps.Map(document.getElementById("map"), mapOptions);`: This line creates a new Google Map and attaches it to the `<div>` element with the ID `map`. The `mapOptions` object (which contains the settings like center, zoom, and map type) is passed to configure the map.

Thank you

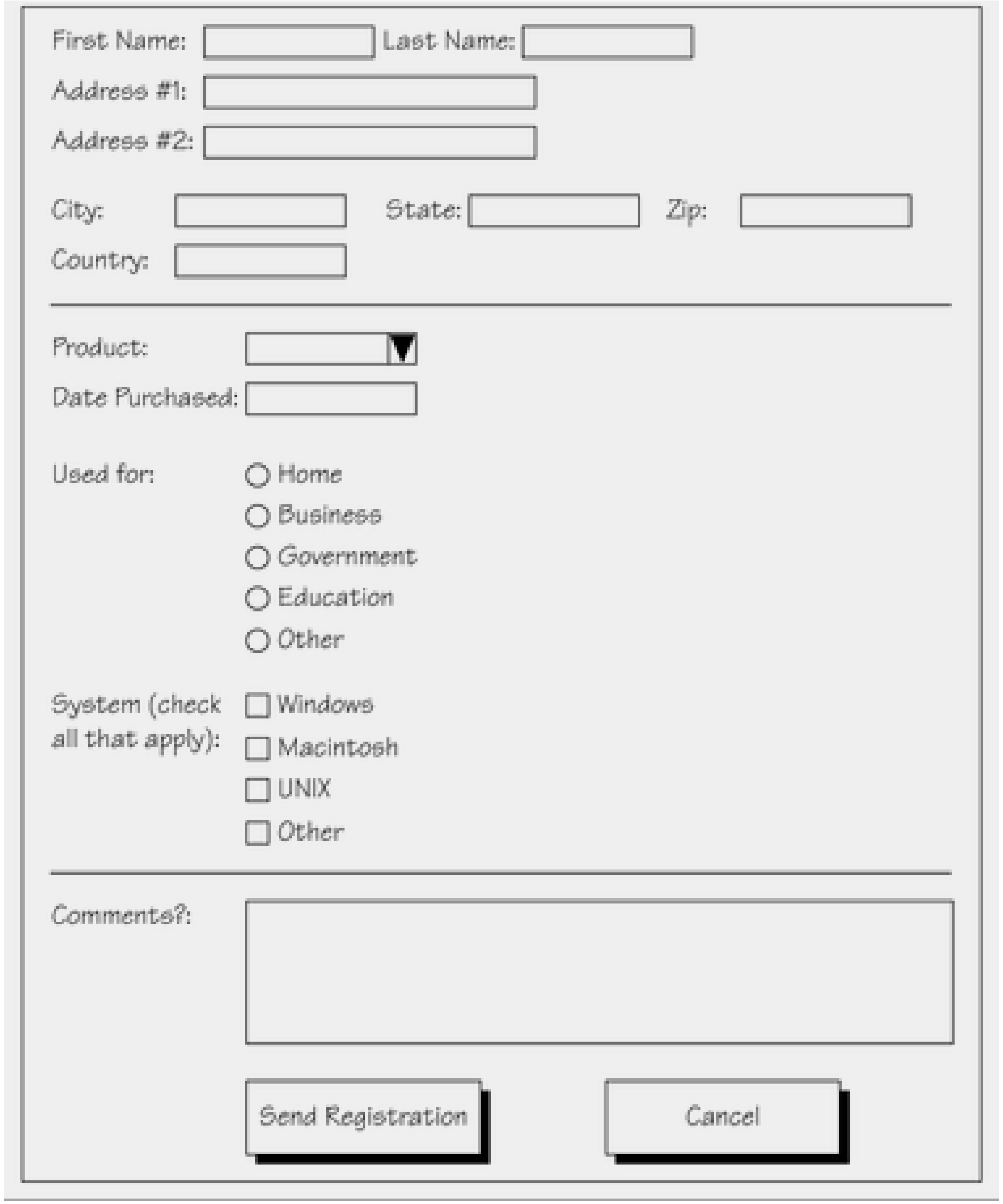
Forms

Dr. L.M. Jenila Livingston

Professor

VIT Chennai

An Example of a Form



First Name: ______ Last Name: ______

Address #1: ______

Address #2: ______

City: ______ State: ______ Zip: ______

Country: ______

Product: ______ ▼

Date Purchased: ______

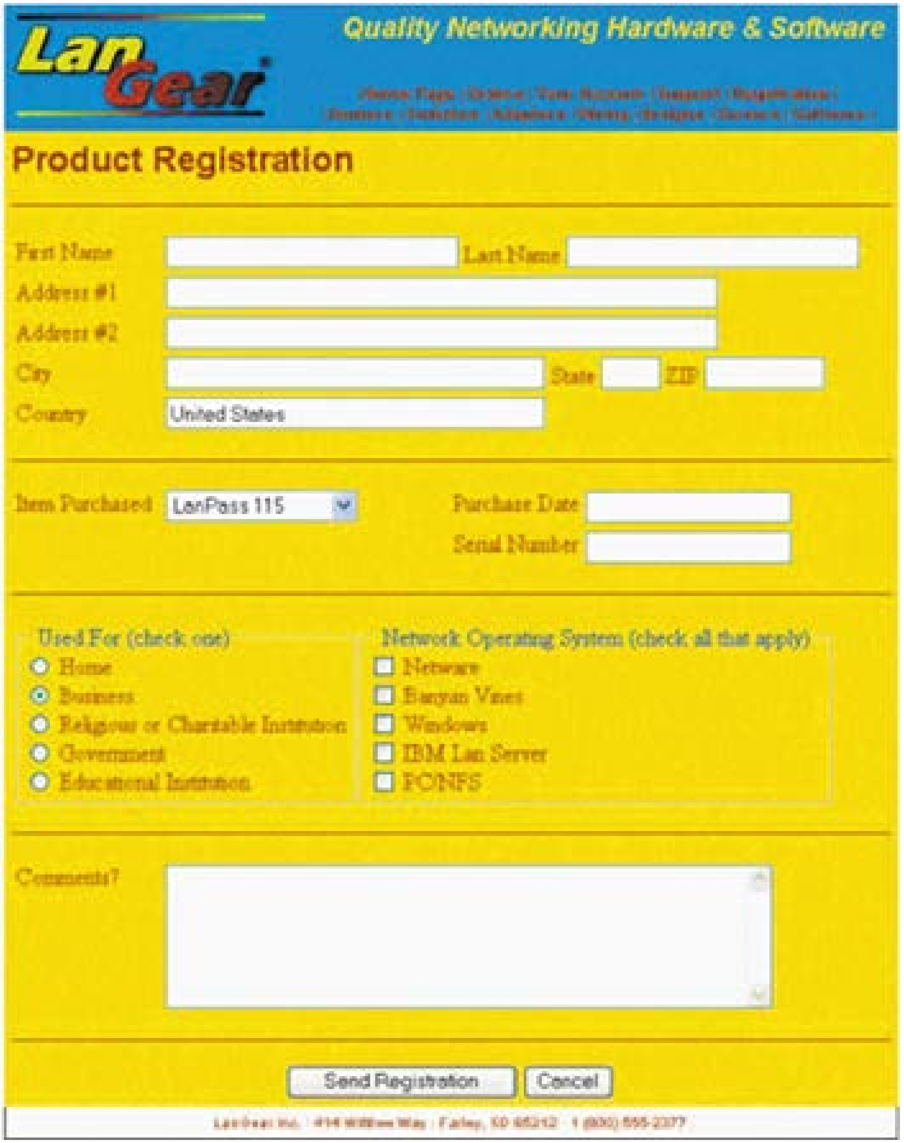
Used for:
○ Home
○ Business
○ Government
○ Education
○ Other

System (check all that apply):
□ Windows
□ Macintosh
□ UNIX
□ Other

Comments?: ______

Send Registration Cancel

Registration Form



LanGear

Quality Networking Hardware & Software

Home Page | Orders | Your Account | Support | Registration |
Routers | Switches | Adapters | Wiring | Bridges | Servers | Software |

Product Registration

contact information

First Name ____ Last Name ____

Address #1 ____

Address #2 ____

City ____ State ____ ZIP ____

Country United States

product information

Item Purchased LanPass 115 ▾

Purchase Date ____

Serial Number ____

usage information

Used For (check one)

- ○ Home
- ● Business
- ○ Religious or Charitable Institution
- ○ Government
- ○ Educational Institution

Network Operating System (check all that apply)

- □ Netware
- □ Banyan Vines
- □ Windows
- □ IBM Lan Server
- □ PC/NFS

comments

Comments? ____

buttons

Send Registration | Cancel

LanGear Inc. · 414 Willow Way · Farley, SD 65212 · 1 (800) 555-2377

The <form> Tag

- A single web page contains multiple forms, but you cannot nest one form inside another.
- Syntax:

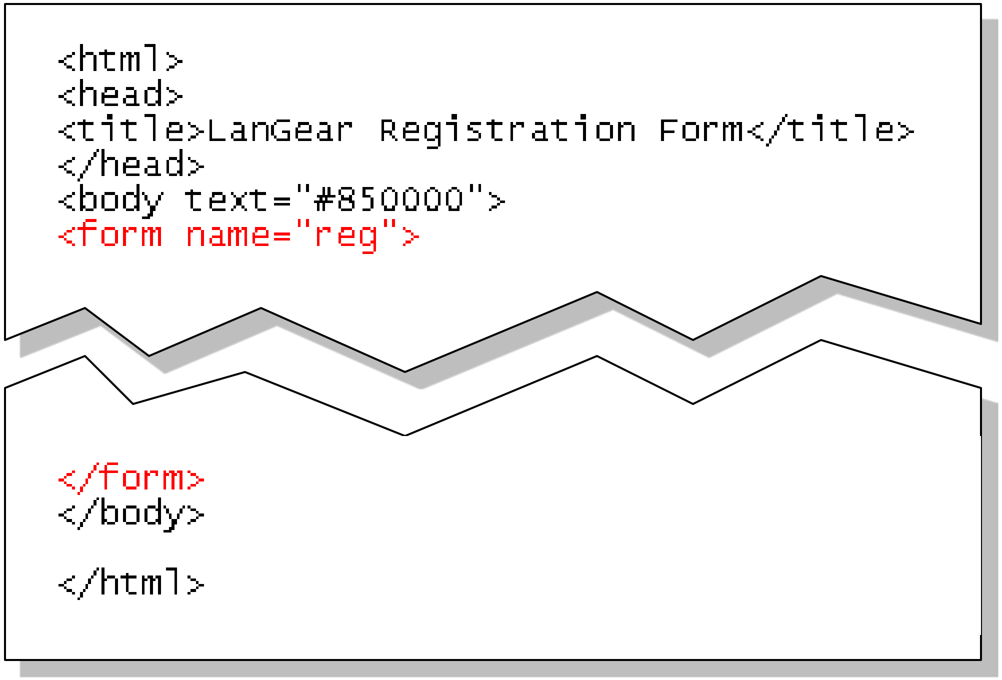
```
<form attributes>
   form elements and layout tags
</form>
```

- **<form>** tag includes the **name** attribute. It identifies each form on the page.
- The name attribute is also needed for programs that retrieve values from the form.

The <form> Tag

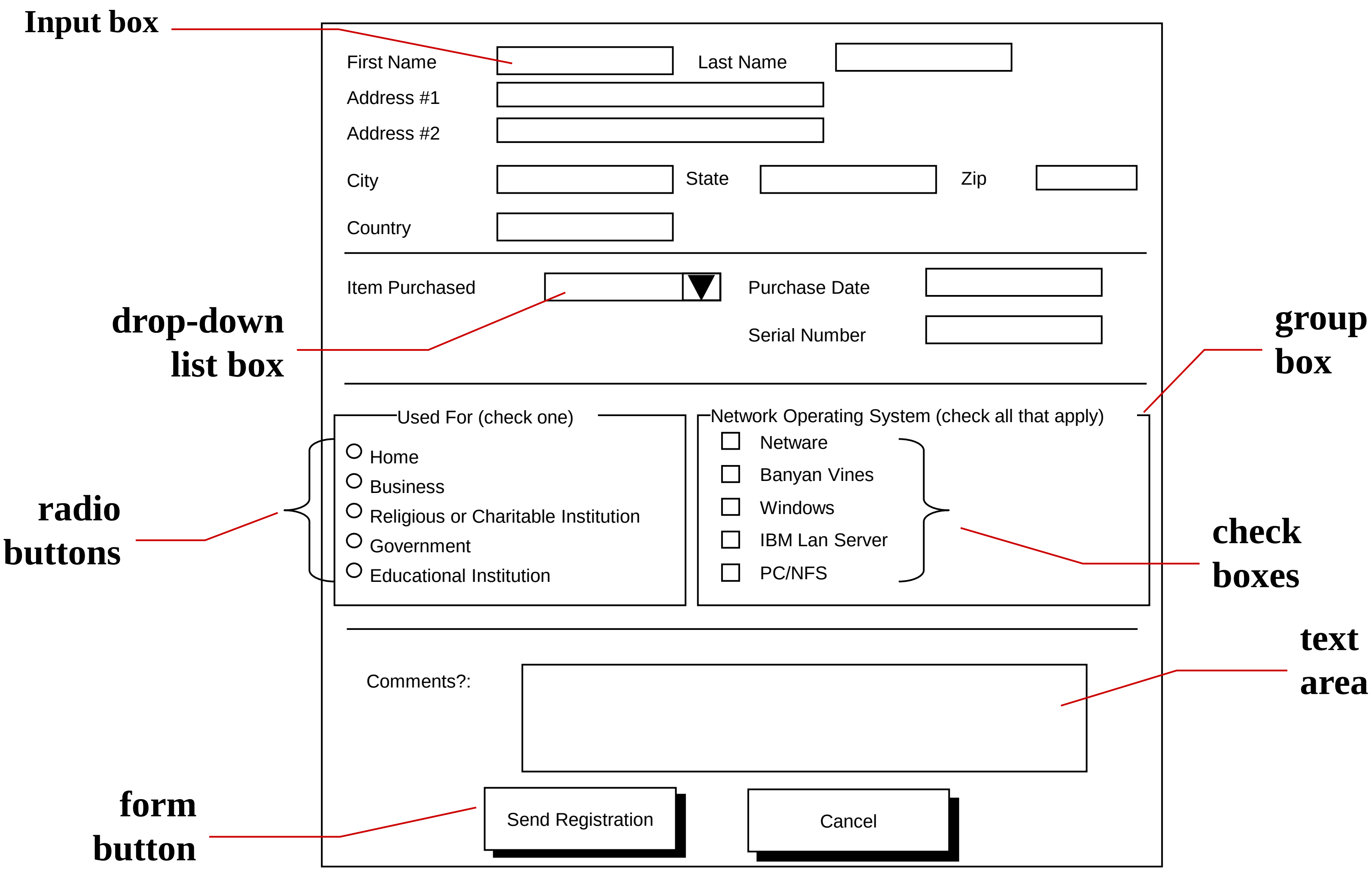
The <form> tag includes information on what **Common Gateway Interface (CGI)** script to use, how the data is to be transferred to the script, and so forth.

This figure shows the form name “reg.”



```
<html>
<head>
<title>LanGear Registration Form</title>
</head>
<body text="#850000">
<form name="reg">


</form>
</body>

</html>
```

Basic Form Elements



Input box

First Name ▭ Last Name ▭

Address #1 ▭

Address #2 ▭

City ▭ State ▭ Zip ▭

Country ▭

drop-down list box

Item Purchased ▭▼ Purchase Date ▭

Serial Number ▭

group box

Used For (check one)

radio buttons

○ Home
○ Business
○ Religious or Charitable Institution
○ Government
○ Educational Institution

Network Operating System (check all that apply)

□ Netware
□ Banyan Vines
□ Windows
□ IBM Lan Server
□ PC/NFS

check boxes

text area

Comments?: ▭

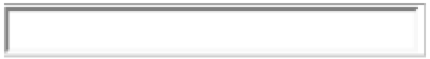
form button

Send Registration

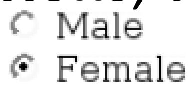
Cancel

Form Elements

- Form elements :
  - **text boxes** for text and numerical entries

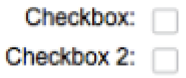


  - **radio buttons**, also called **option buttons**, to select a single option from a predefined list



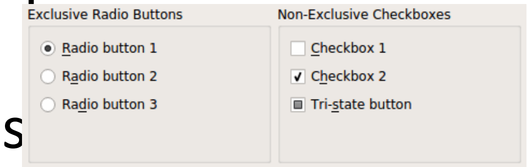
○ Male
● Female

  - **check boxes** to specify an item as either present or absent



Checkbox: □
Checkbox 2: □

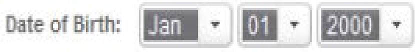
  - **groups boxes** to organize form elements



Exclusive Radio Buttons
● Radio button 1
○ Radio button 2
○ Radio button 3

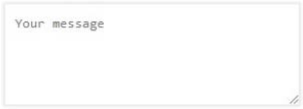
Non-Exclusive Checkboxes
□ Checkbox 1
☑ Checkbox 2
▣ Tri-state button

  - **combo box- selection lists** for long lists of options, usually appearing in a **drop-down list box**



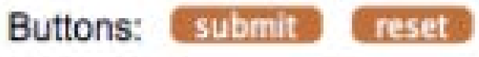
Date of Birth: Jan ▾ 01 ▾ 2000 ▾

  - **text areas** for extended entries that can include several lines of text



Your message

  - **buttons** that can be clicked to start processing the form



- Each element in which the user can enter information is called a **field**.

Input Type

<input type=“type”>

Type	Description
type=”button”	Display a button which can be clicked to perform an action from a script
type=”checkbox”	Display a check box
type=”file”	Display a browse button to locate and select a file
type=”hidden”	Create a hidden field, not viewable on the form
type=”image”	Display an inline image which can be clicked to perform an action from a script
type=”password”	Display a text box in which hides text entered by the user
type=”radio”	Display a radio (option) button
type=”reset”	Display a button which resets the form when clicked
type=”submit”	Display a button which submits the form when clicked
type=”text”	Display a text box in which displays text entered by the user

Input Type

- Input fields are created using the **<input>** tag.
- Syntax:

  <input type=“type” name=“name” id=“id”>

  - ***type*** specifies the type of input field
  - ***name*** and ***id*** attributes **identifies the input field for the CGI script**
- To create a text box, you would enter the tag:

  <input type=“text”>

The <INPUT> tag attributes

- ***TYPE***: Defines the type of data used in the field.
- ***NAME***: The name of the particular element.
- ***ID***: The identifier of the particular element.
- ***MAXLENGTH***: The maximum number of characters that can be entered by users in a text field.
- ***SIZE***: Specifies the size of the field and depends on its type.
- ***SRC***: Denote URL for an image.
- ***VALUE***: Contain the initial value displayed to users.
- ***CHECKED***: Indicates that a checkbox or radio button is selected.
- ***DISABLED***: Prevents the field from receiving focus.
- ***ALIGN***: Alignment if image is used. (left, right, middle, top, bottom)
- ***READONLY***: Prevents modification of the contents of the field.
- ***PATTERN***: specifies a regular expression that the <input> element's value is checked against on form submission (regular expression, validation)
- ***PLACEHOLDER*** attribute specifies a short hint that describes the expected value of an input field (e.g. a sample value or a short description of the expected format).

1. Text box

<input type=“text” id=“id” value=“value” size=“value” maxlength=“value”>

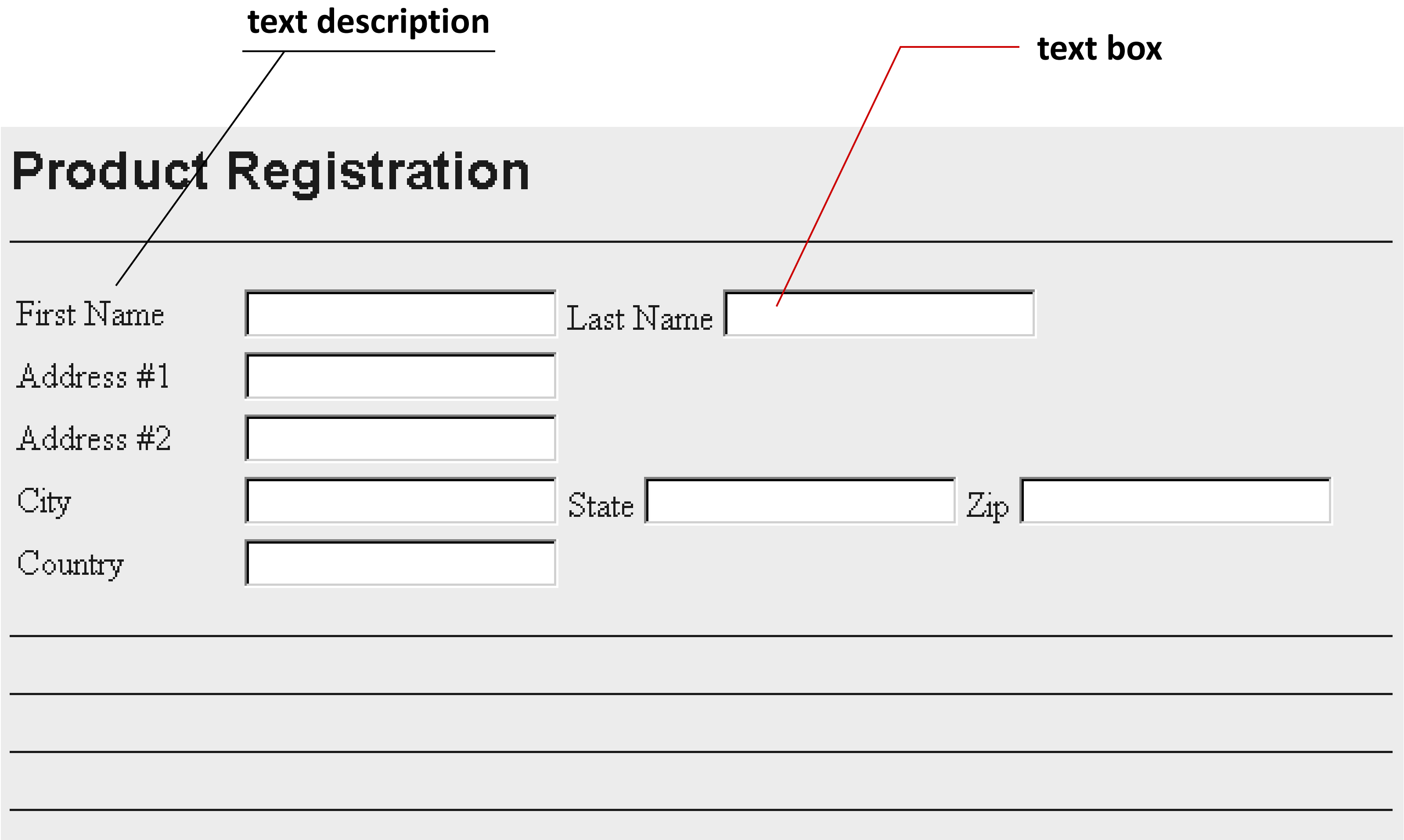
– ***name*** and ***id*** attributes identify the field

– ***value*** - a default value to the text box

– ***size*** - the width of the text box in number of characters

– ***maxlength*** - the maximum number of characters allowed in the field

Text Boxes on the Form



text description

text box

Product Registration

First Name

Last Name

Address #1

Address #2

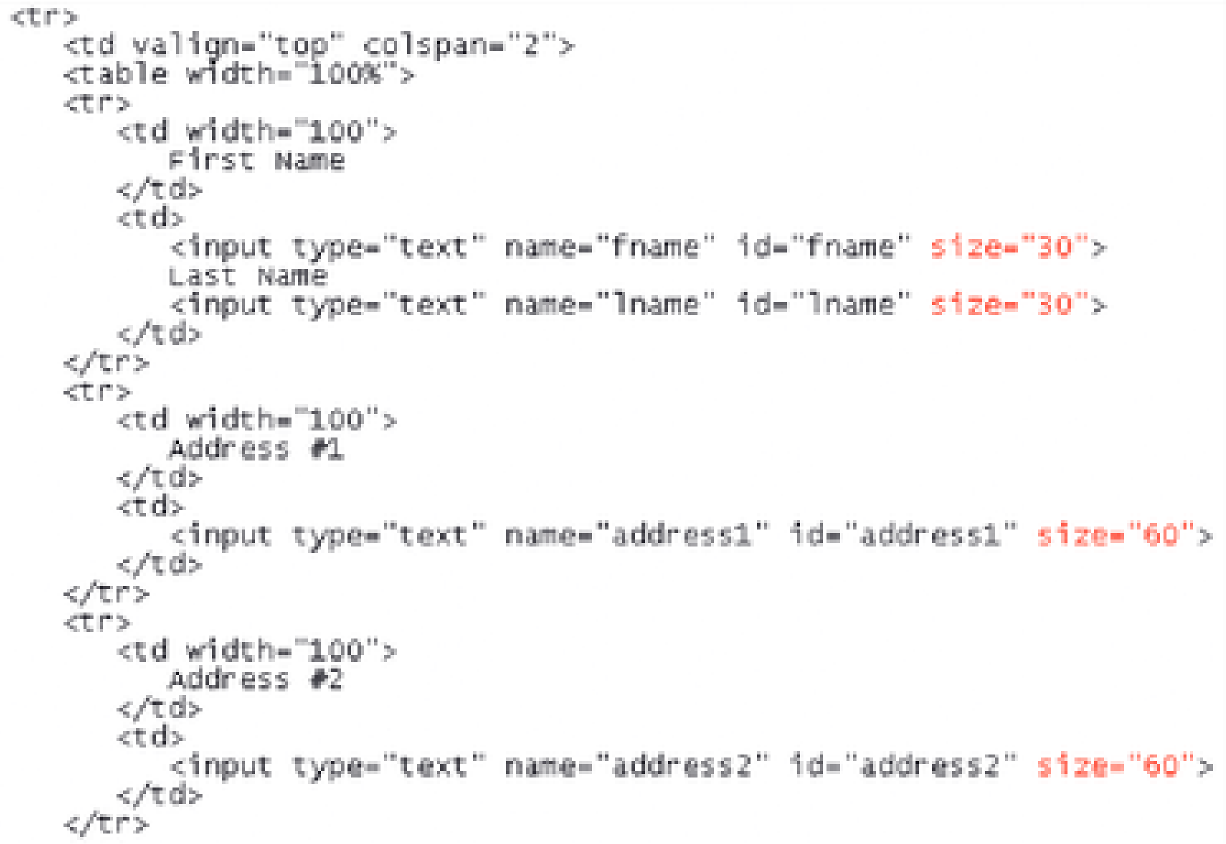
City

State

Zip

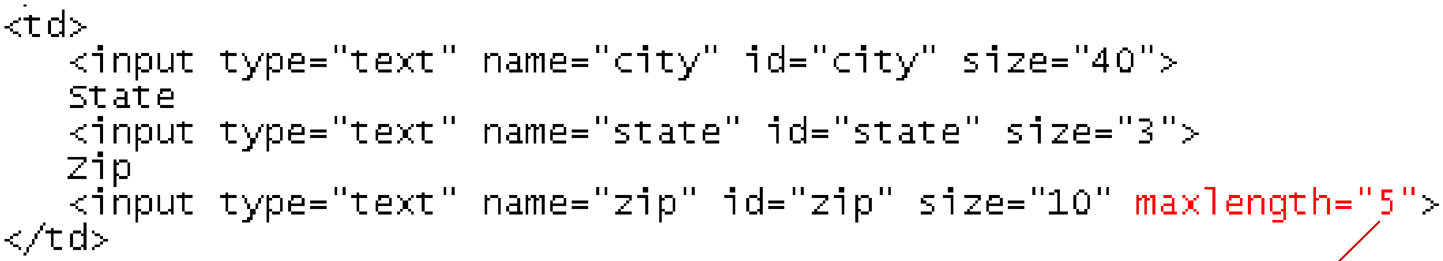
Country

Setting the size (width) of Text Boxes



```
<tr>
    <td valign="top" colspan="2">
    <table width="100%">
    <tr>
        <td width="100">
            First Name
        </td>
        <td>
            <input type="text" name="fname" id="fname" size="30">
            Last Name
            <input type="text" name="lname" id="lname" size="30">
        </td>
    </tr>
    <tr>
        <td width="100">
            Address #1
        </td>
        <td>
            <input type="text" name="address1" id="address1" size="60">
        </td>
    </tr>
    <tr>
        <td width="100">
            Address #2
        </td>
        <td>
            <input type="text" name="address2" id="address2" size="60">
        </td>
    </tr>
```

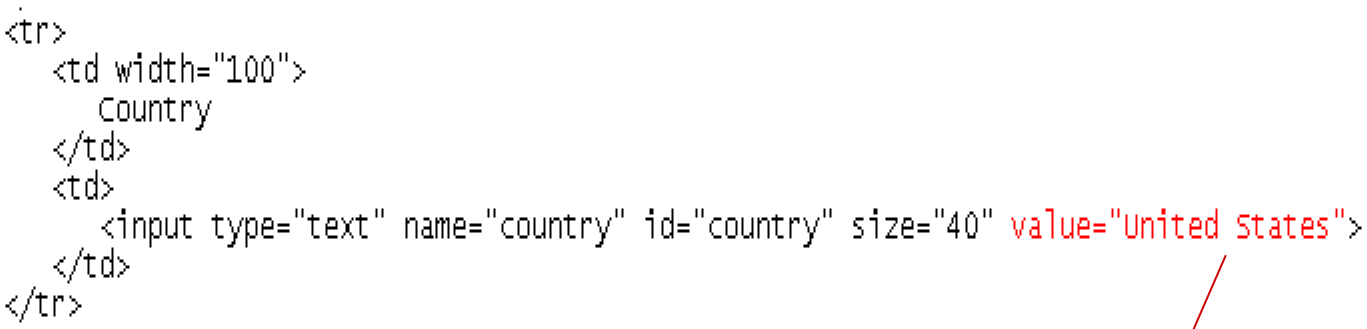
Maximum Length of a text box



```
<td>
   <input type="text" name="city" id="city" size="40">
   State
   <input type="text" name="state" id="state" size="3">
   Zip
   <input type="text" name="zip" id="zip" size="10" maxlength="5">
</td>
```

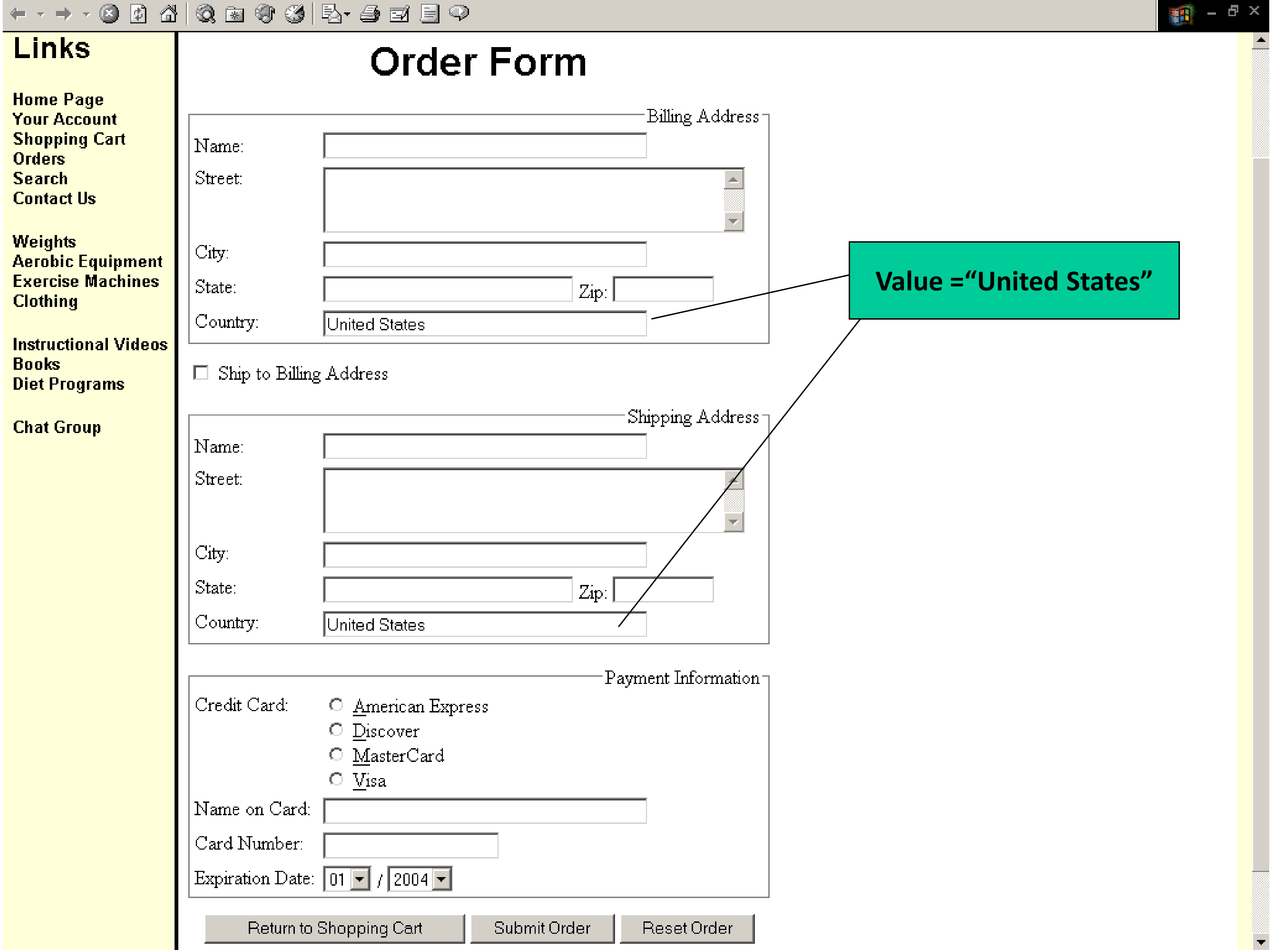
no more than 5 characters are allowed in this text box

Default Value for a Text box



```
<tr>
   <td width="100">
      Country
   </td>
   <td>
      <input type="text" name="country" id="country" size="40" value="United States">
   </td>
</tr>
```

default value



Links

Home Page
Your Account
Shopping Cart
Orders
Search
Contact Us

Weights
Aerobic Equipment
Exercise Machines
Clothing

Instructional Videos
Books
Diet Programs

Chat Group

Order Form

Billing Address

Name:
Street:
City:
State: Zip:
Country: United States

□ Ship to Billing Address

Shipping Address

Name:
Street:
City:
State: Zip:
Country: United States

Payment Information

Credit Card:
○ American Express
○ Discover
○ MasterCard
○ Visa

Name on Card:
Card Number:
Expiration Date: 01 / 2004

Value =“United States”

2. Password Field

- A **password field** is a text box in which the characters typed by the user are displayed as **bullets** or **asterisks** i.e. ****.
- Syntax:

  <input type="password">

- Using a password field should not be confused with having a secure connection. It is not encrypted.
- The password field only acts as a mask for a field entry as it is entered.

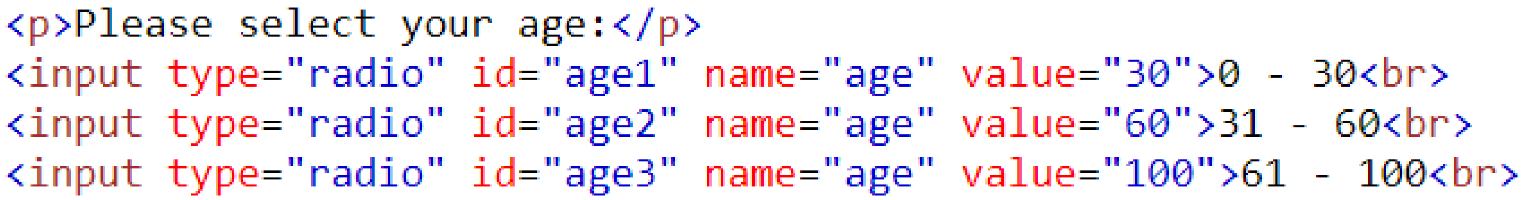
3. Radio Buttons

- **Radio buttons** display a list of choices from which a user makes a selection.
  - **Only one radio button can be selected at a time.**
- Syntax:

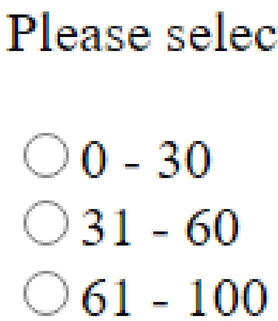
<input type=“radio” name=“name” id=“id” value=“value”>

  - ***name*** - the field containing the radio button (required)
  - ***id*** - the specific option. Only required if you intend to use a field label with the radio button
  - ***value*** - sent to the CGI script, if that radio button is selected by the user

Creating Radio Buttons



```
<p>Please select your age:</p>
<input type="radio" id="age1" name="age" value="30">0 - 30<br>
<input type="radio" id="age2" name="age" value="60">31 - 60<br>
<input type="radio" id="age3" name="age" value="100">61 - 100<br>
```



Please select your age:

○ 0 - 30
○ 31 - 60
○ 61 - 100

4. Check Boxes

- A **check box** is either selected or not
- Syntax:

  <input type="checkbox" name="name" id="id" value="value">

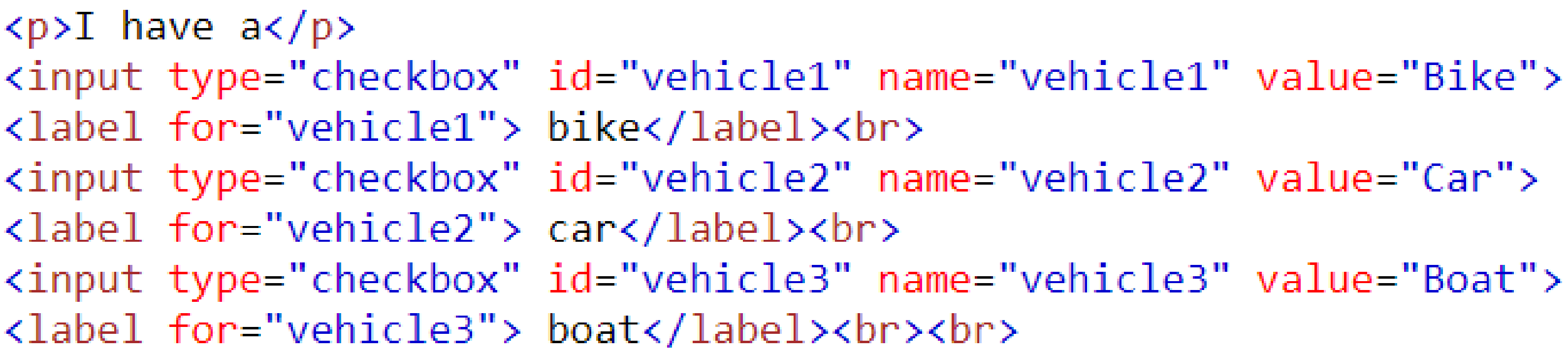
  – ***name*** and ***id*** - identify the check box

  – the ***value*** - is sent to the CGI script when the check box is selected
- Check boxes are not selected by default.

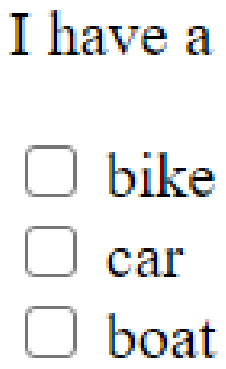
  – to do this, add the **checked** attribute to the **<input>** tag

  – <input type ="checkbox" checked = "checked">
    <input type ="checkbox" checked = "true">

Check Boxes



```
<p>I have a</p>
<input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
<label for="vehicle1"> bike</label><br>
<input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
<label for="vehicle2"> car</label><br>
<input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
<label for="vehicle3"> boat</label><br><br>
```



I have a

□ bike
□ car
□ boat

5. Field Set (Group Box)

- A **group box** labels an entire collection of radio/ check boxes.
- A group box is a box placed around a set of fields that indicates that they belong to a common group.
- the <legend> tag is used to display a legend on the group box
- Syntax:

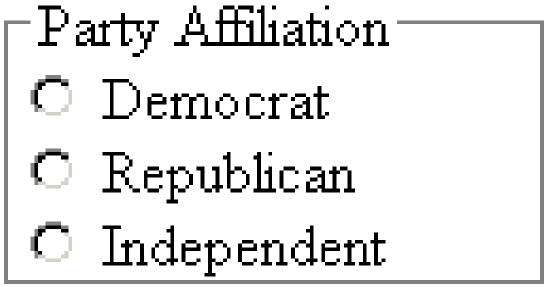
```
<fieldset>
  <legend align="top">legend text</legend>
  collection of fields
</fieldset>
```

Creating a Group Box and Legend

start of group box

group box legend

```
<fieldset>
   <legend align="top">Party Affiliation</legend>

   <input type="radio" name="party" id="dem" value="dem">
   <label for="dem">Democrat</label><br>

   <input type="radio" name="party" id="gop" value="gop">
   <label for="gop">Republican</label><br>

   <input type="radio" name="party" id="ind" value="ind">
   <label for="ind">Independent</label>
</fieldset>
```



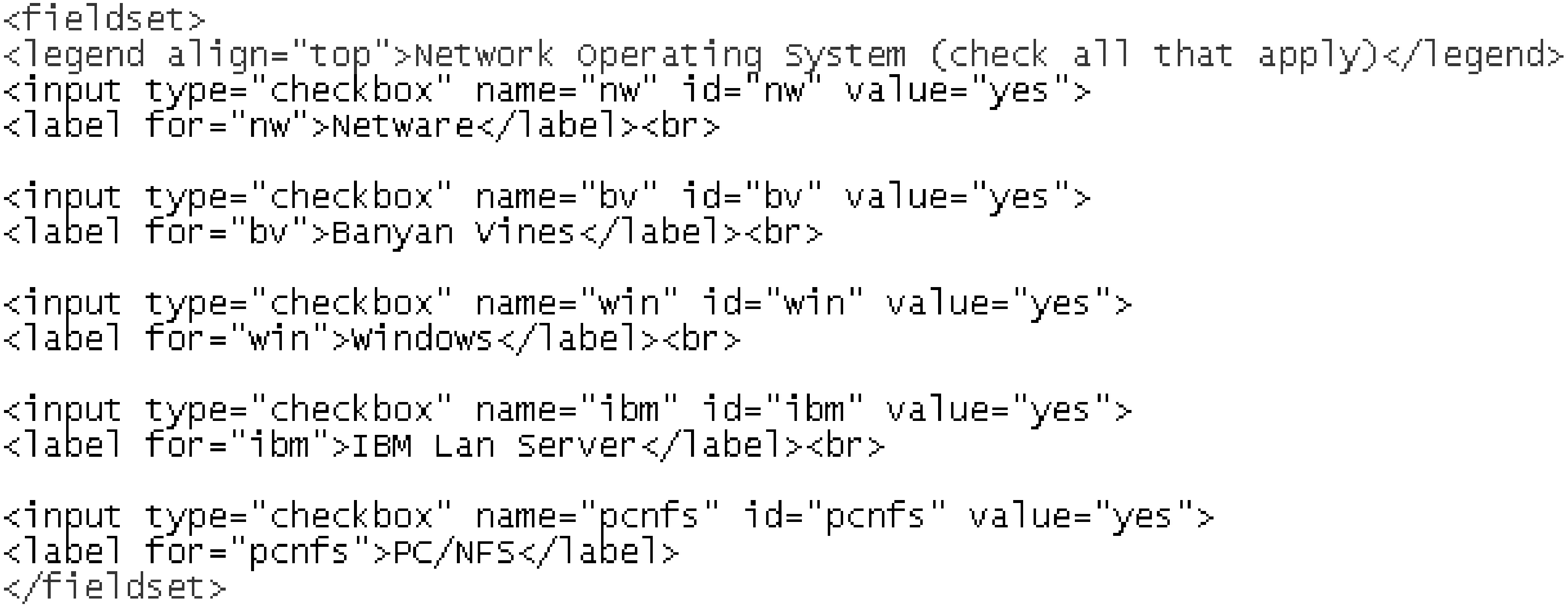
Party Affiliation

○ Democrat

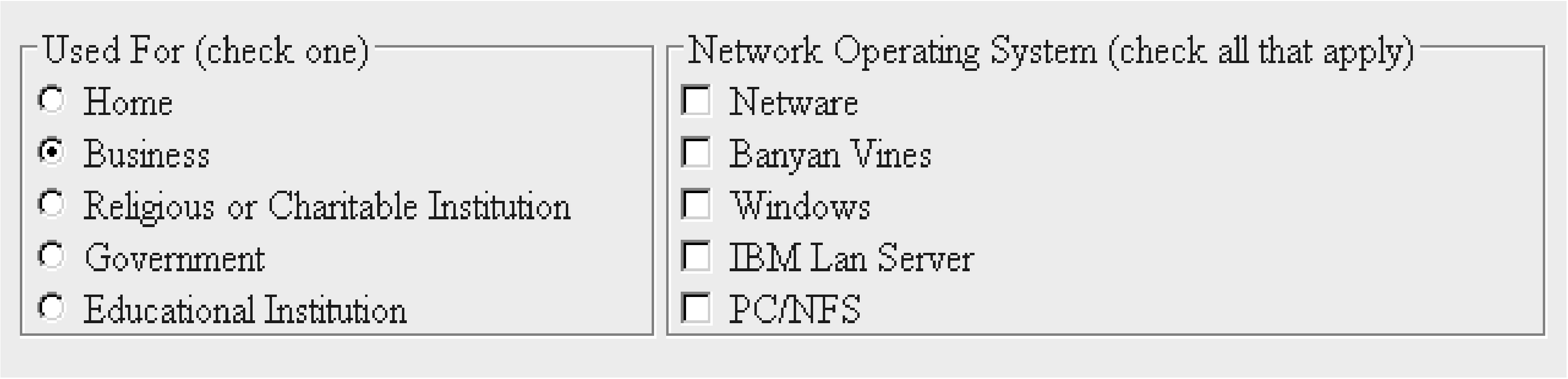
○ Republican

○ Independent

Group Boxes for Radio Buttons and Check Boxes



```
<fieldset>
<legend align="top">Network Operating System (check all that apply)</legend>
<input type="checkbox" name="nw" id="nw" value="yes">
<label for="nw">Netware</label><br>

<input type="checkbox" name="bv" id="bv" value="yes">
<label for="bv">Banyan Vines</label><br>

<input type="checkbox" name="win" id="win" value="yes">
<label for="win">Windows</label><br>

<input type="checkbox" name="ibm" id="ibm" value="yes">
<label for="ibm">IBM Lan Server</label><br>

<input type="checkbox" name="pcnfs" id="pcnfs" value="yes">
<label for="pcnfs">PC/NFS</label>
</fieldset>
```



Used For (check one)

○ Home
● Business
○ Religious or Charitable Institution
○ Government
○ Educational Institution

Network Operating System (check all that apply)

□ Netware
□ Banyan Vines
□ Windows
□ IBM Lan Server
□ PC/NFS

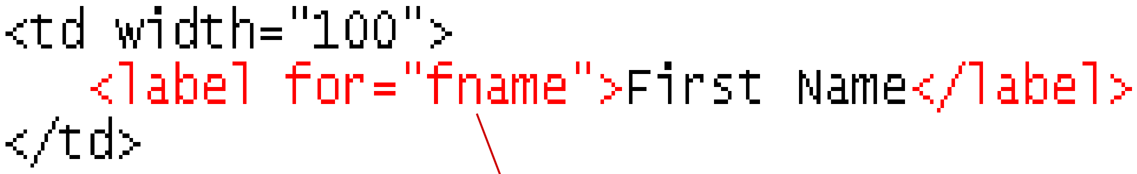
6. Form Labels

- For scripting purposes, HTML allows you to formally attach a label to a related text element.
- Syntax:

  <label for="id">label text</label>

  - *id* is the value of the id attribute for a field on the form
  - *label* text is the text of the label
  - you must bind the label to the id attribute of the field and not the name attribute

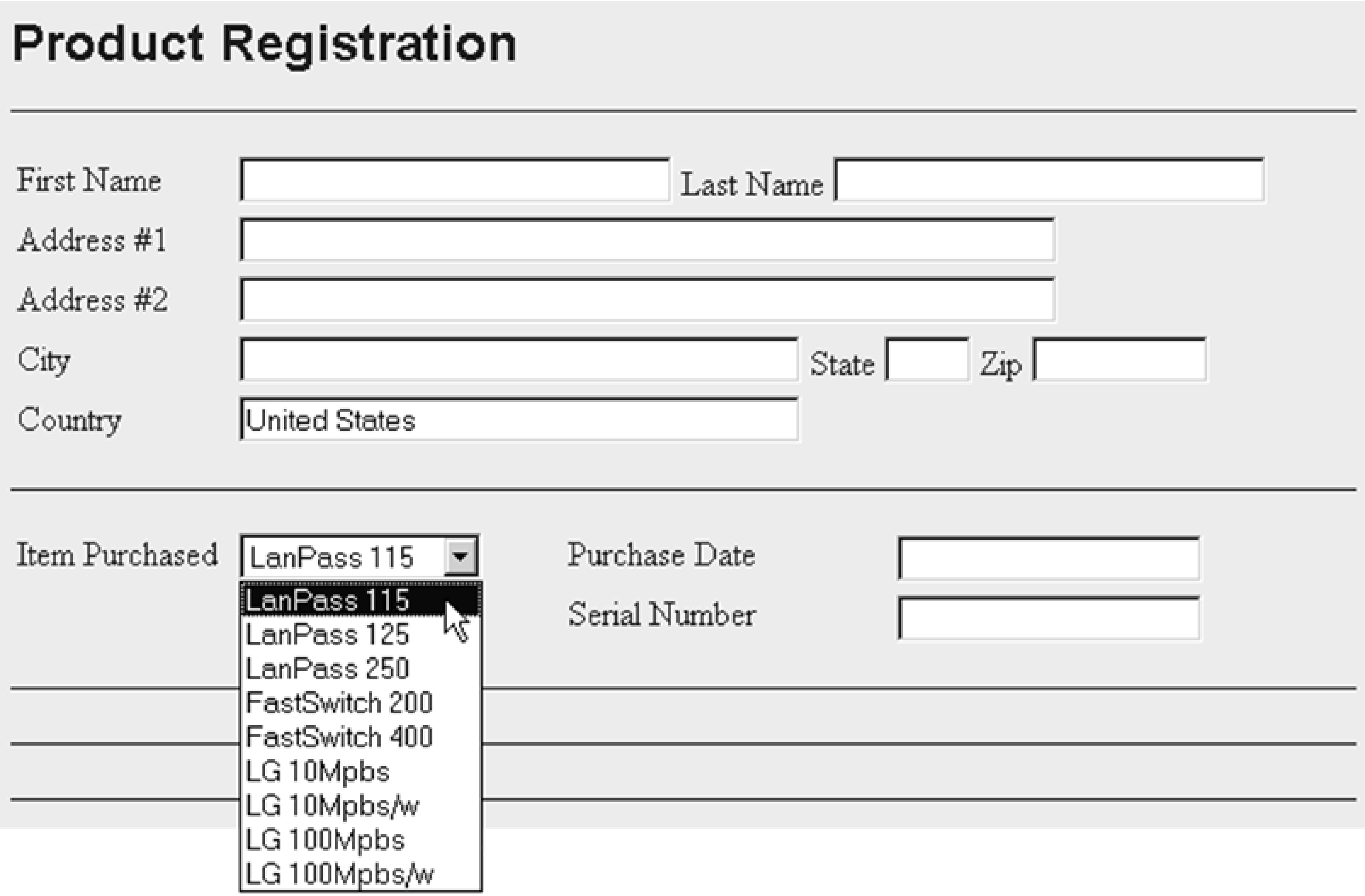
Creating a Label for the fname Field



```
<td width="100">
   <label for="fname">First Name</label>
</td>
```

value of the id
attribute for the
first name field

7. Selection List



Product Registration

First Name

Last Name

Address #1

Address #2

City

State

Zip

Country: United States

Item Purchased: LanPass 115

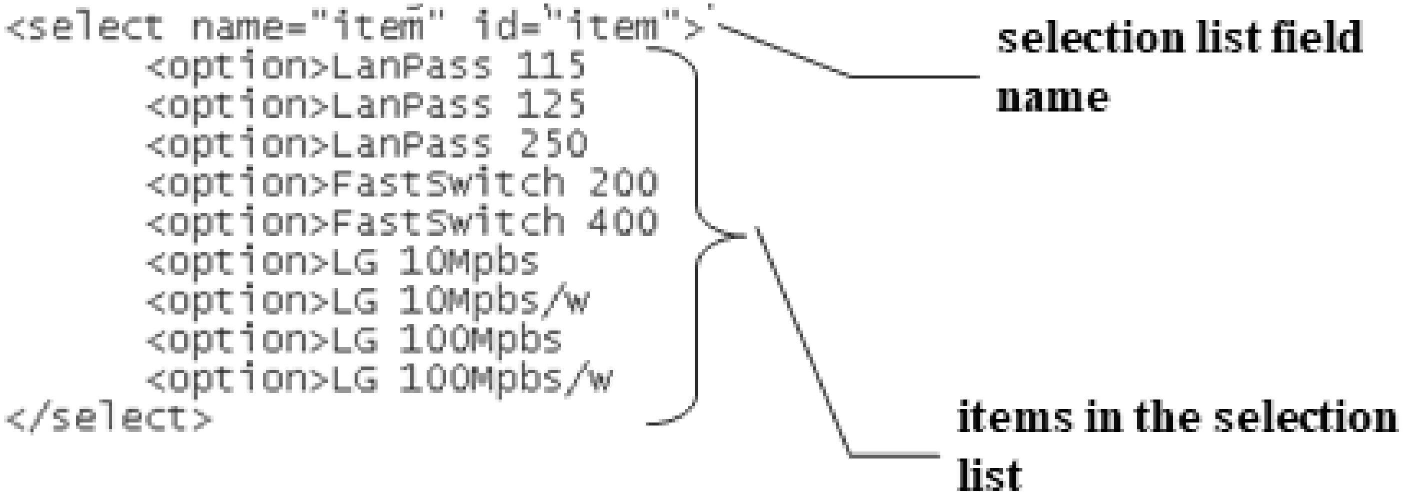
Purchase Date

Serial Number

Selection List

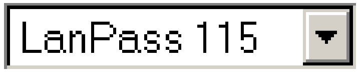
- A **selection list** is a list box from which a user selects a particular value or set of values.
- Selection lists are useful to use when a set of possible options is fixed.
- Selection lists help eliminate spelling errors and invalid entries.
- A selection list is created using the **<select>** tag.
- The **<option>** tag is used to specify individual selection items.

Creating a Selection List



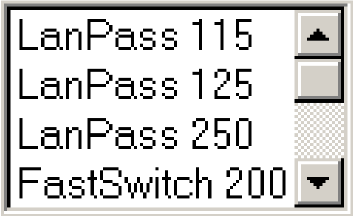
```
<select name="item" id="item">
      <option>LanPass 115
      <option>LanPass 125
      <option>LanPass 250
      <option>FastSwitch 200
      <option>FastSwitch 400
      <option>LG 10Mpbs
      <option>LG 10Mpbs/w
      <option>LG 100Mpbs
      <option>LG 100Mpbs/w
</select>
```

Selection Lists with Different Size Values



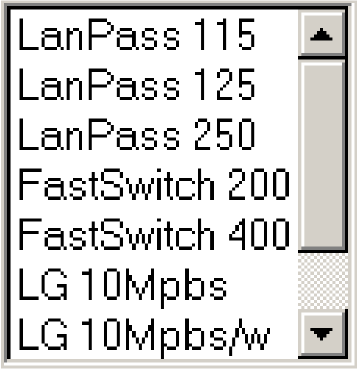
LanPass 115

size = "1"



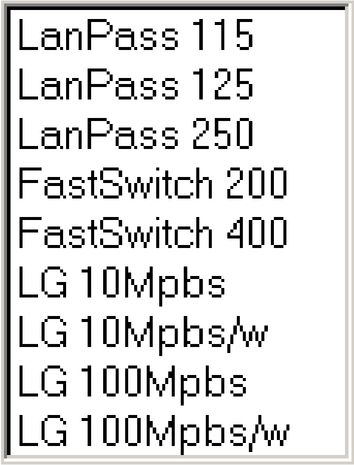
LanPass 115
LanPass 125
LanPass 250
FastSwitch 200

size = "4"



LanPass 115
LanPass 125
LanPass 250
FastSwitch 200
FastSwitch 400
LG 10Mpbs
LG 10Mpbs/w

size = "7"



LanPass 115
LanPass 125
LanPass 250
FastSwitch 200
FastSwitch 400
LG 10Mpbs
LG 10Mpbs/w
LG 100Mpbs
LG 100Mpbs/w

size = "9"

Option Groups

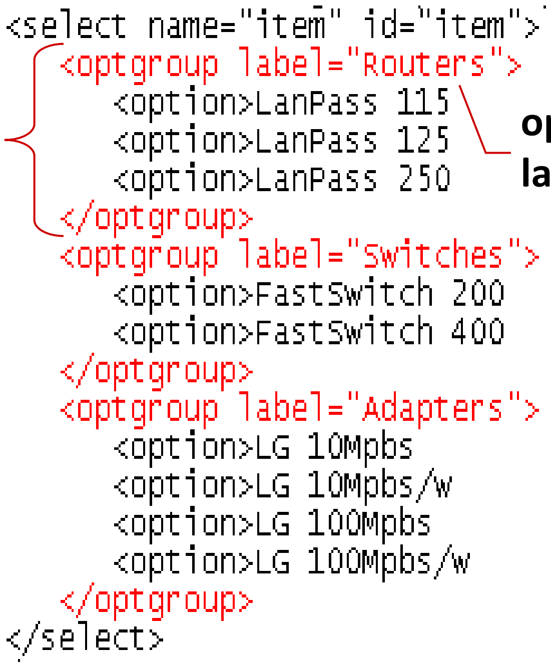
- **Option groups** allow you to organize selection lists into distinct groups.
- Syntax:

  <optgroup label=“label”>

  - ***label*** is the label assigned to the option group
  - the text for the label appears in the selection list above each group of items but is not a selectable item from the list

Option Groups

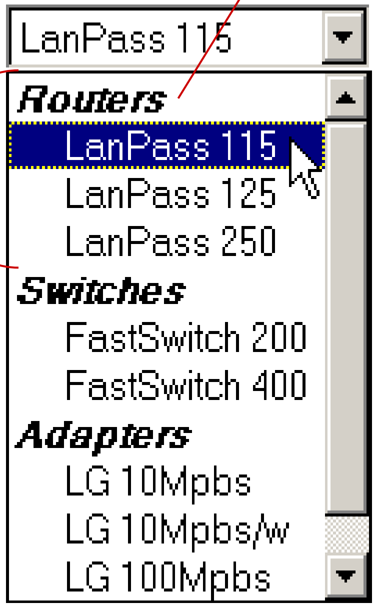
a single option group



```
<select name="item" id="item">
  <optgroup label="Routers">
    <option>LanPass 115
    <option>LanPass 125
    <option>LanPass 250
  </optgroup>
  <optgroup label="Switches">
    <option>FastSwitch 200
    <option>FastSwitch 400
  </optgroup>
  <optgroup label="Adapters">
    <option>LG 10Mpbs
    <option>LG 10Mpbs/w
    <option>LG 100Mpbs
    <option>LG 100Mpbs/w
  </optgroup>
</select>
```

option group label

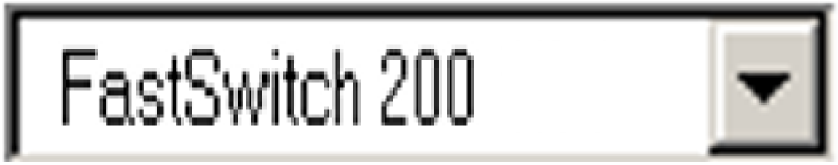
a single option group



option group label

selected attribute

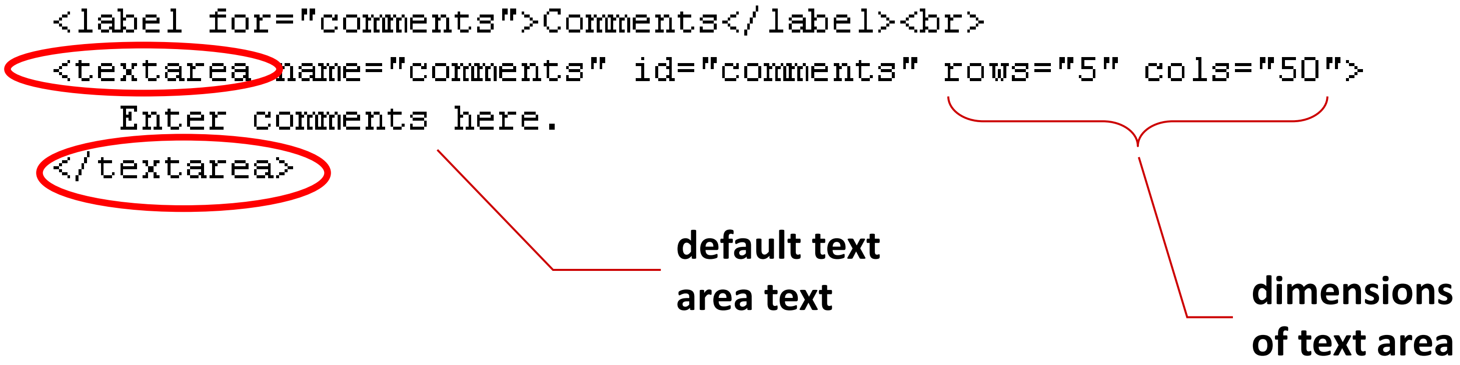
To display “FastSwithch 200” in a combo box by default when the page loads


FastSwitch 200

<option selected>FastSwitch 200</option>

8. Text Area

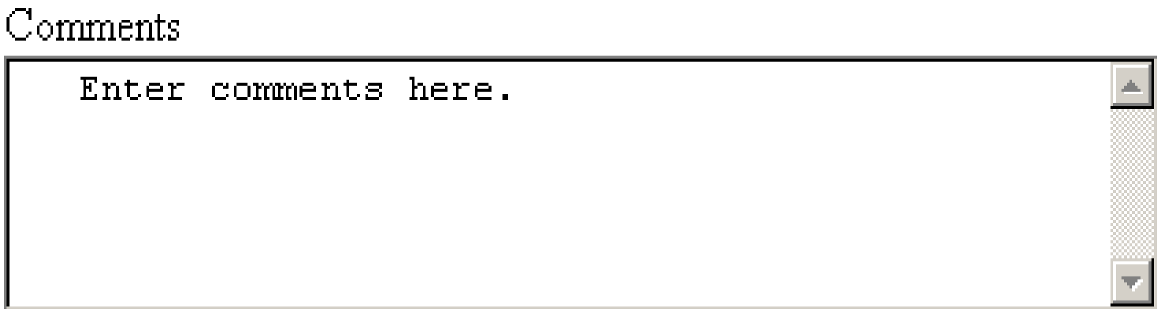
- The <**textarea**> tag defines a multi-line text input control.
- The <**textarea**> element is often used in a form, to collect user inputs like comments or reviews.



```
<label for="comments">Comments</label><br>
<textarea name="comments" id="comments" rows="5" cols="50">
   Enter comments here.
</textarea>
```

default text area text

dimensions of text area



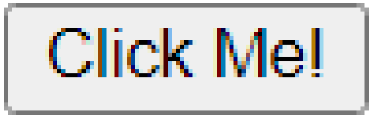
Comments

```
   Enter comments here.
```

9. Buttons

- The **<button>** tag defines a clickable button.
- It is used to submit forms or anywhere in a document for accessible, standard button functionality.

- <button type="button">Click Me!</button>



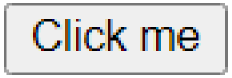
Push Buttons

- **Push button** is created using the <input> tag
- Syntax:

```
<input type=“button” value=“text”>
```

  - ***text*** is the text that appears on the button
- Push buttons do not perform any actions by themselves on the web page.
- To create an action, write a script or program that runs automatically when the button is clicked.

```
<input type="button" value="Click me" onclick="msg()">
```



Submit and Reset Buttons

- A **submit button** is a button that submits the form to the CGI script for processing.
- A **reset button** resets the form to its original values.
- The syntax for creating these two buttons is:

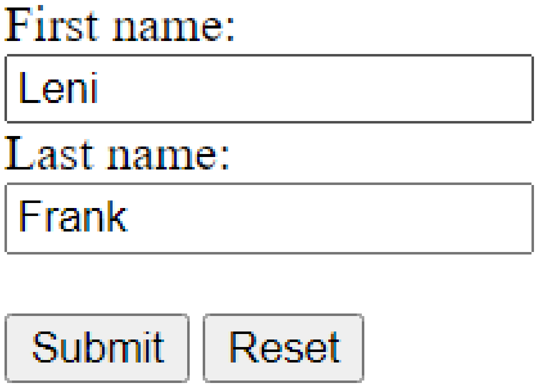
  <input type="submit" value="text">

  <input type="reset" value="text">

  – ***value*** attribute defines the text that appears on the button

Submit and Reset Buttons

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value=“Leni"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value=“Frank"><br><br>
  <input type="submit" value="Submit">
  <input type="Reset" value="Reset">
</form>
```

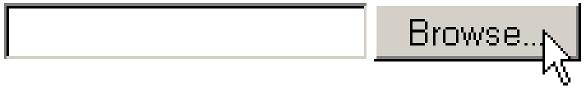


First name:
Leni
Last name:
Frank

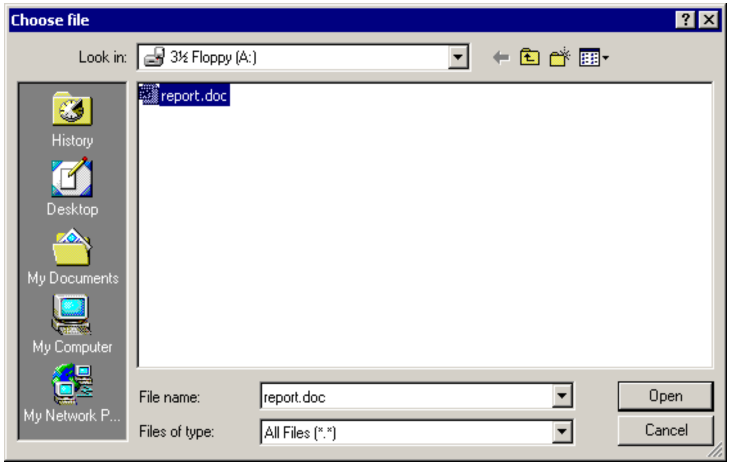
10. File Button

```
<input type="file" name="file_name">
```

1. User clicks the Browse button



2. Selects a file from the Choose File dialog box


Choose file

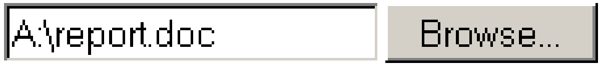
Look in: 3½ Floppy (A:)

report.doc

File name: report.doc

Files of type: All Files (*.*)

3. The filename and location are automatically placed in the text box


A:\report.doc

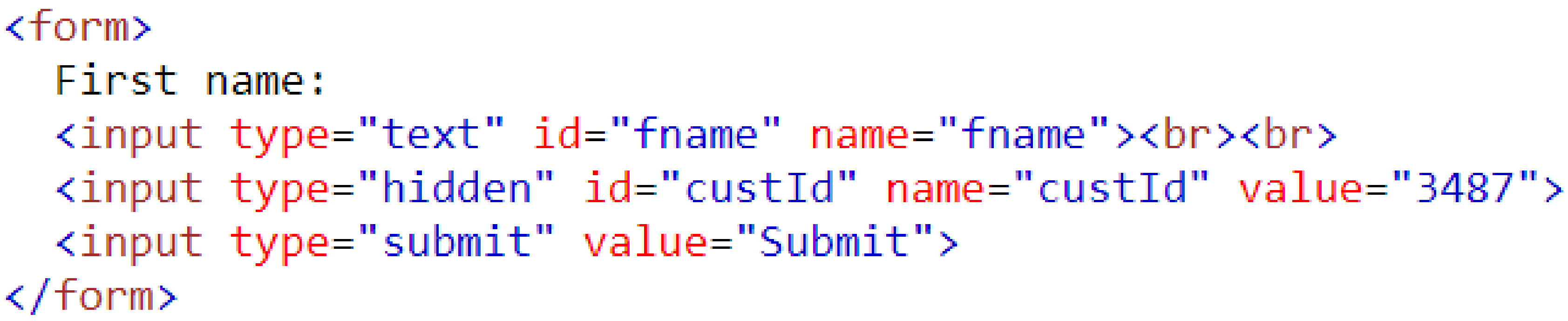
11. Hidden Fields

- A **hidden field** is added to the form but not displayed in the Web page.
- Because the field is hidden, it can be placed anywhere between the opening and closing **<form>** tag.
- Syntax:

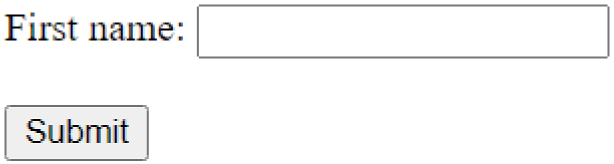
  <input type="hidden" name="name" value="value>

- Place all hidden fields in one location to make it easier to read and interpret the HTML code.
- Include a comment describing the purpose of the field.

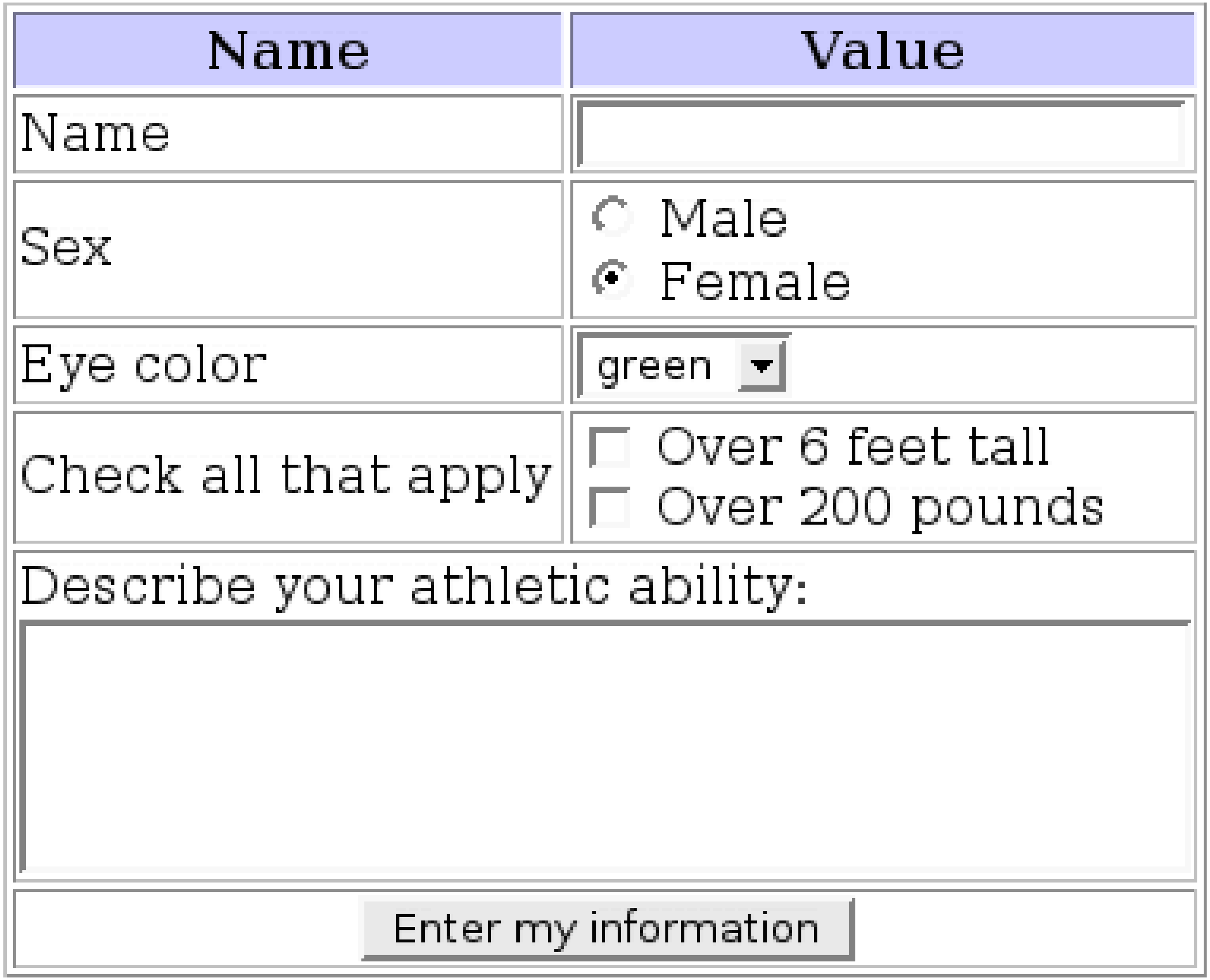
Hidden Field



```
<form>
  First name:
  <input type="text" id="fname" name="fname"><br><br>
  <input type="hidden" id="custId" name="custId" value="3487">
  <input type="submit" value="Submit">
</form>
```



Exercise



Name	Value
Name	
Sex	○ Male ● Female
Eye color	green ▾
Check all that apply	□ Over 6 feet tall □ Over 200 pounds
Describe your athletic ability:	
Enter my information	

References

- https://www.slideshare.net/NamanJoshi20/creating-web-formfor-college-seminars

Form Elements

Dr. L.M. Jenila Livingston
Professor
VIT Chennai

HTML5: New Form elements

HTML5 has several new elements for forms.

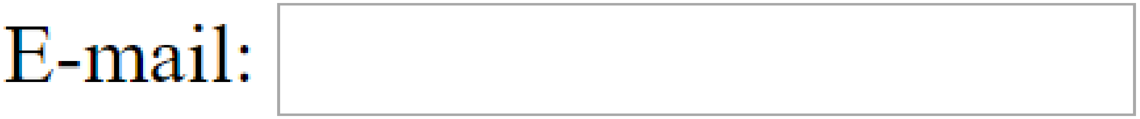
- input types
  - > email
  - > url
  - > tel
  - > number
  - > range
  - > output
  - > date pickers
  - > search
  - > color
  - > datalist
  - > keygen

- Form Validation attributes

HTML5: Input - e-mail

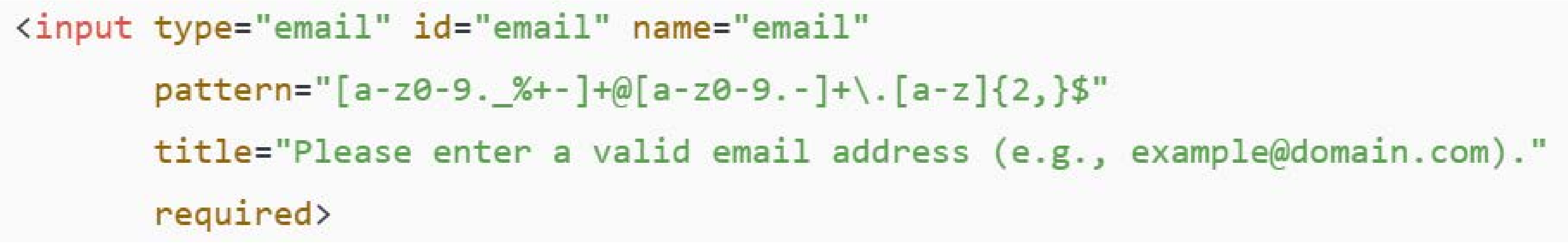
- The email type is used for input fields and will have an email address.
- Depending on browser support, the e-mail address can be automatically validated when submitted.

```
E-mail: <input type="email" name="user_email" />
<input type="email" value="default@example.com">
<input type="email" size="32" minlength="3" maxlength="64">
<input type="email" placeholder="sophie@example.com">
```



input type - email

- Perform basic built-in validation
- Allows formats like abc@domain (without a top-level domain).
- Doesn't enforce stricter rules for valid characters or specific domain formats.
- Strict Email Validation with pattern



```
<input type="email" id="email" name="email"
       pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}$"
       title="Please enter a valid email address (e.g., example@domain.com)."
       required>
```

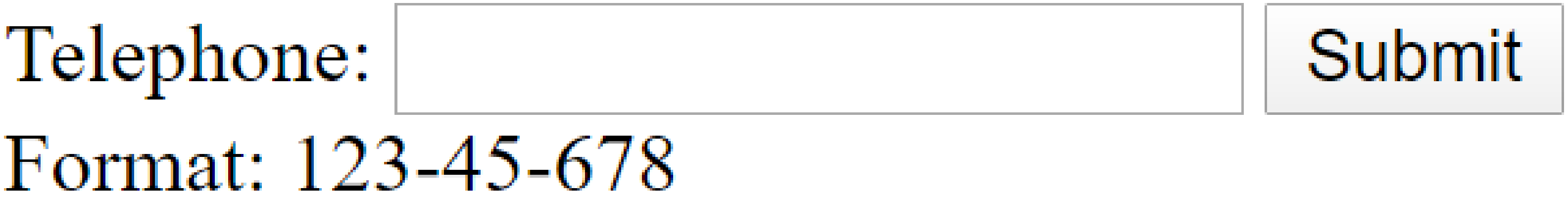
Input - url

- The url type is used for input fields and will have a URL address.
- The value of the url field is automatically validated when the form is submitted.

```
Homepage: <input type="url" name="user_url" />
```

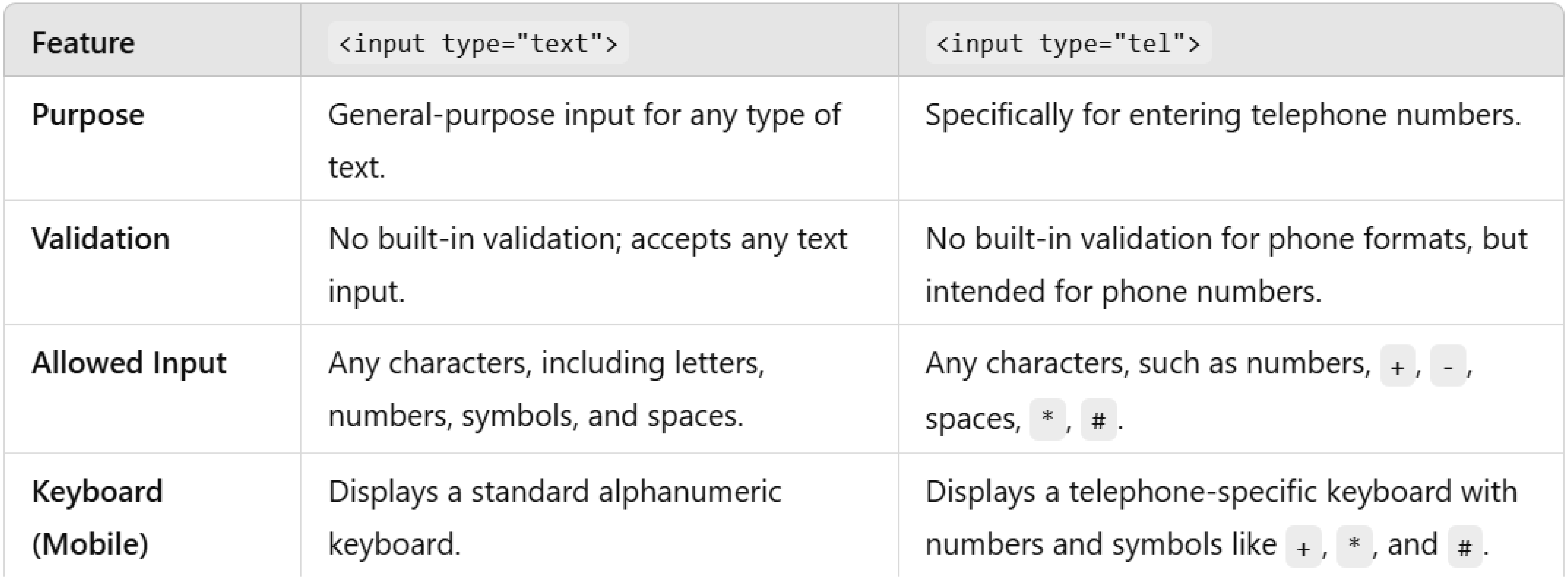
tel

```
<form>
Telephone: <input type="tel" name="phone"
pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}" required>
<input type="submit">
<span>Format: 123-45-678</span>
</form>
```



Format: 123-45-678

text vs tel

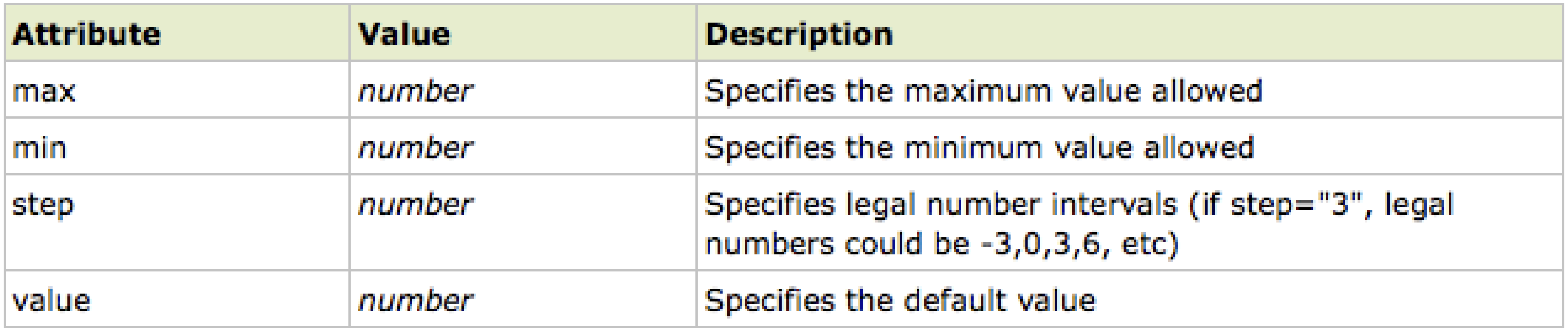


Feature	`<input type="text">`	`<input type="tel">`
Purpose	General-purpose input for any type of text.	Specifically for entering telephone numbers.
Validation	No built-in validation; accepts any text input.	No built-in validation for phone formats, but intended for phone numbers.
Allowed Input	Any characters, including letters, numbers, symbols, and spaces.	Any characters, such as numbers, `+`, `-`, spaces, `*`, `#`.
Keyboard (Mobile)	Displays a standard alphanumeric keyboard.	Displays a telephone-specific keyboard with numbers and symbols like `+`, `*`, and `#`.

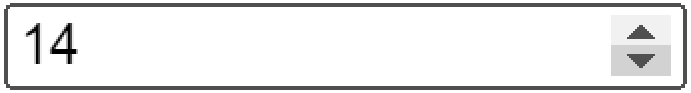
Input - number

- defines a numeric input field
- Set restrictions on what numbers are accepted:

Points: <input type="number" name="points" min="1" max="10" />

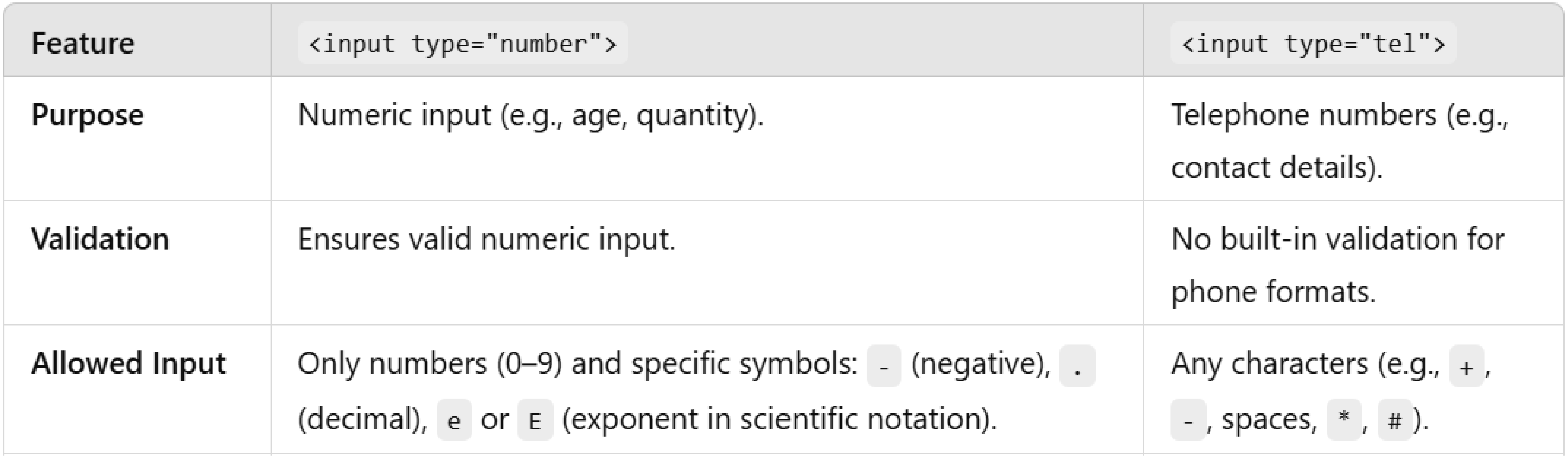


Attribute	Value	Description
max	*number*	Specifies the maximum value allowed
min	*number*	Specifies the minimum value allowed
step	*number*	Specifies legal number intervals (if step="3", legal numbers could be -3,0,3,6, etc)
value	*number*	Specifies the default value



14

number Vs tel

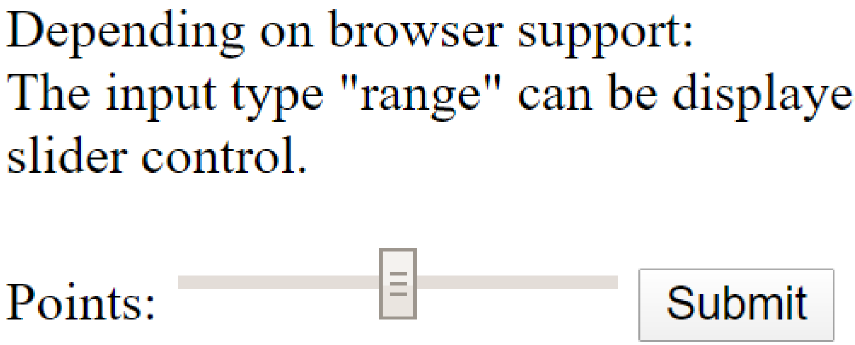


Feature	`<input type="number">`	`<input type="tel">`
Purpose	Numeric input (e.g., age, quantity).	Telephone numbers (e.g., contact details).
Validation	Ensures valid numeric input.	No built-in validation for phone formats.
Allowed Input	Only numbers (0–9) and specific symbols: `-` (negative), `.` (decimal), `e` or `E` (exponent in scientific notation).	Any characters (e.g., `+`, `-`, spaces, `*`, `#`).

Input - range

- Slider control
- Default range is 0 to 100
- The range type is used for input fields that should contain a value from a range of numbers.
- You can also set restrictions on what numbers are accepted:

```
<input type="range" name="points" min="1" max="10" />
```



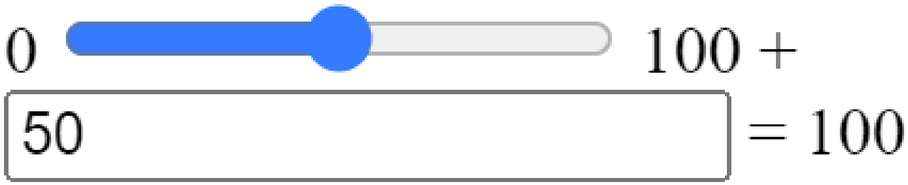
Depending on browser support:
The input type "range" can be displayed as a slider control.

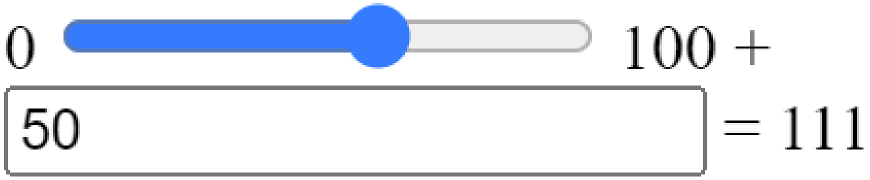
output

The **output** element is used for different types of output, like calculations or script output:

```
<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">
0<input type="range" id="a" value="50">100 + <br>
<input type="number" id="b" value="50">
= <output id="x"></output>
</form>
```

The **oninput** attribute ensures that the output is updated dynamically when either input changes.


0 100 +
50 = 100


0 100 +
50 = 111

Input – date pickers

input type for selecting date and time:

> date - Selects date, month and year

> month - Selects month and year

> week - Selects week and year

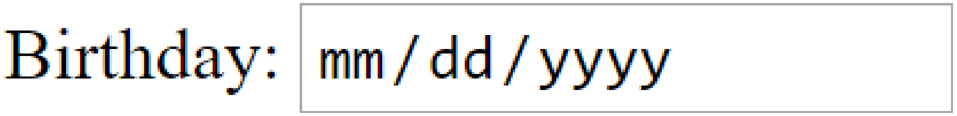
> time - Selects time (hour and minute)

> datetime - Selects time, date, month and year

> datetime-local - Selects time, date, month and year (local time)

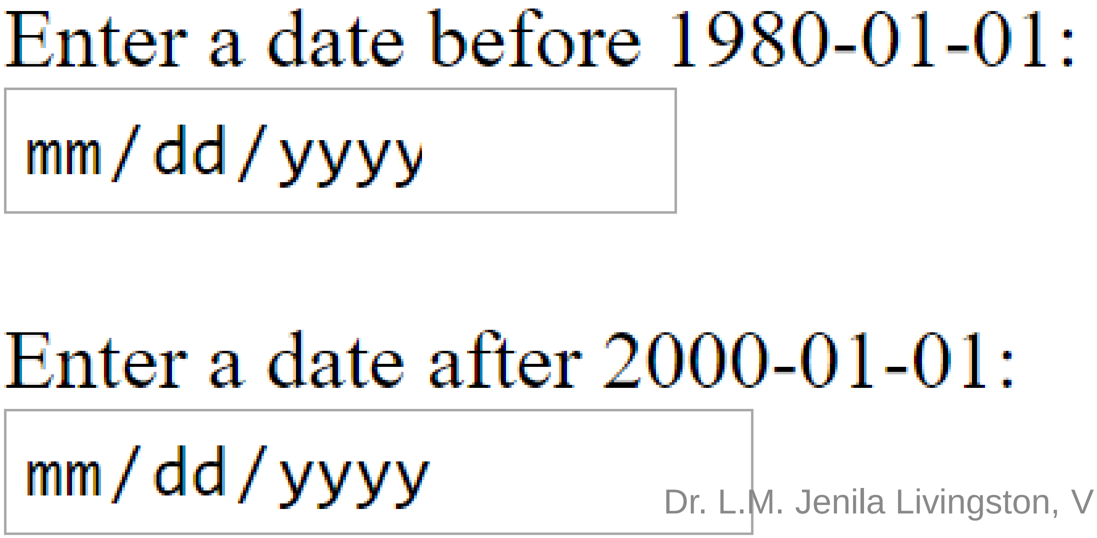
Date

Birthday:
  <input type="date" name="bday">



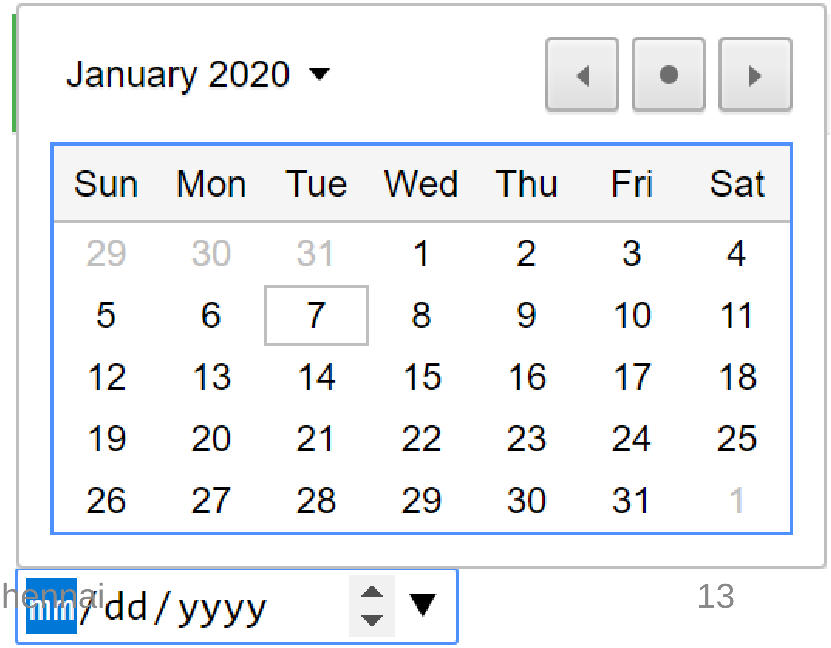
Birthday: mm/dd/yyyy

```
<form>
Enter a date before 1980-01-01:<br>
<input type="date" name="bday" max="1979-12-31"><br><br>
Enter a date after 2000-01-01:<br>
<input type="date" name="bday" min="2000-01-02"><br><br>
</form>
```



Enter a date before 1980-01-01:
mm/dd/yyyy

Enter a date after 2000-01-01:
mm/dd/yyyy



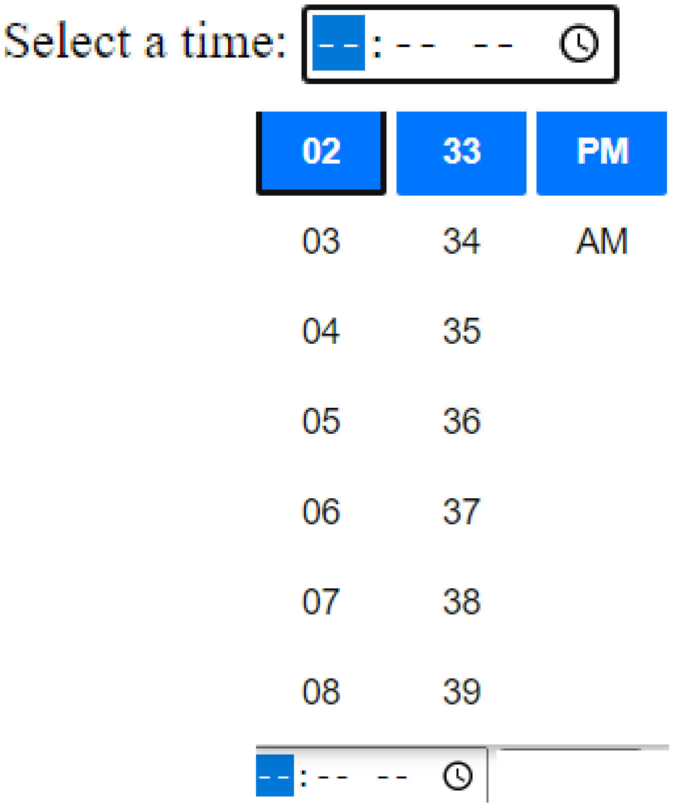
January 2020 ▾ ◂ ● ▸

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

mm/dd/yyyy ▾

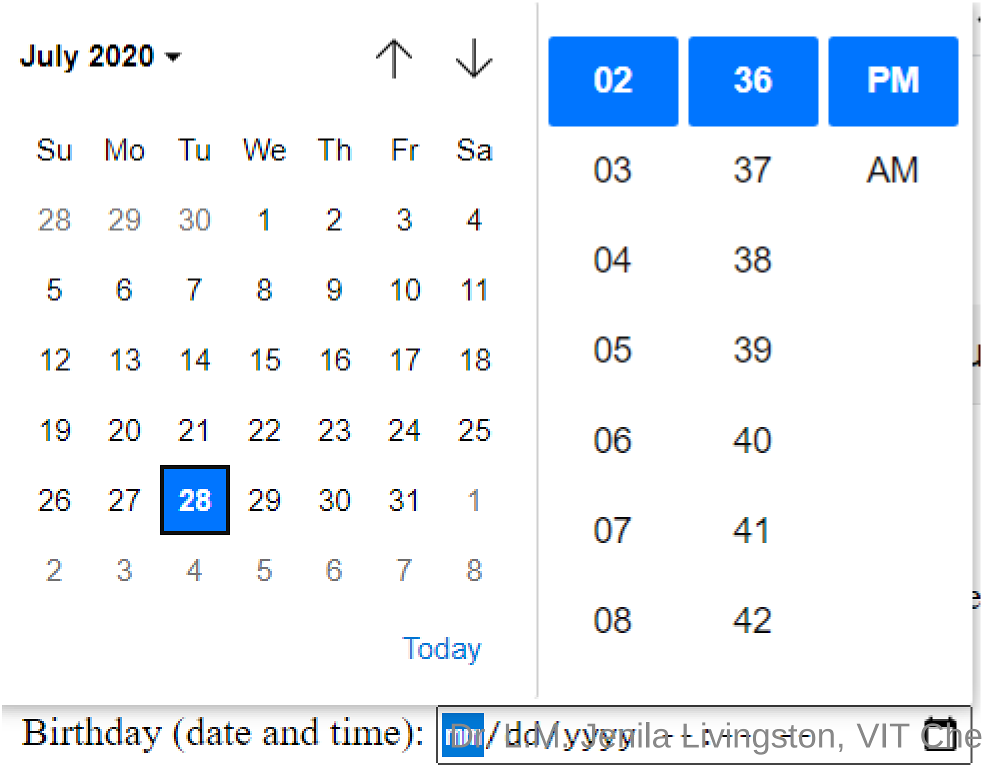
Time, Date

<input type="time" id="tnputtime" name=“inputtime">



Select a time: --:-- --

Birthday (date and time):
<input type="datetime-local" name="bdaytime">



July 2020

Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Today

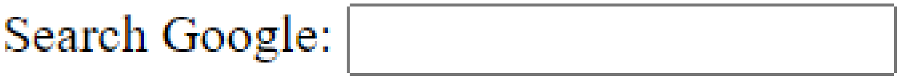
Birthday (date and time): mm/dd/yyyy --:-- --

Input - search

- The search type is used for search fields like a site search or Google search.

- The search field behaves like a regular text field.

```
Search Google:<input type="search" id="gsearch" name="gsearch">
```



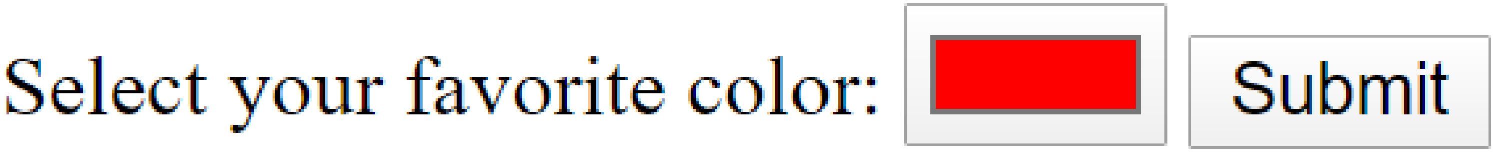
Input – color picker

- input field that contains a color.
- allow you to select a color from a color picker. Default: black color

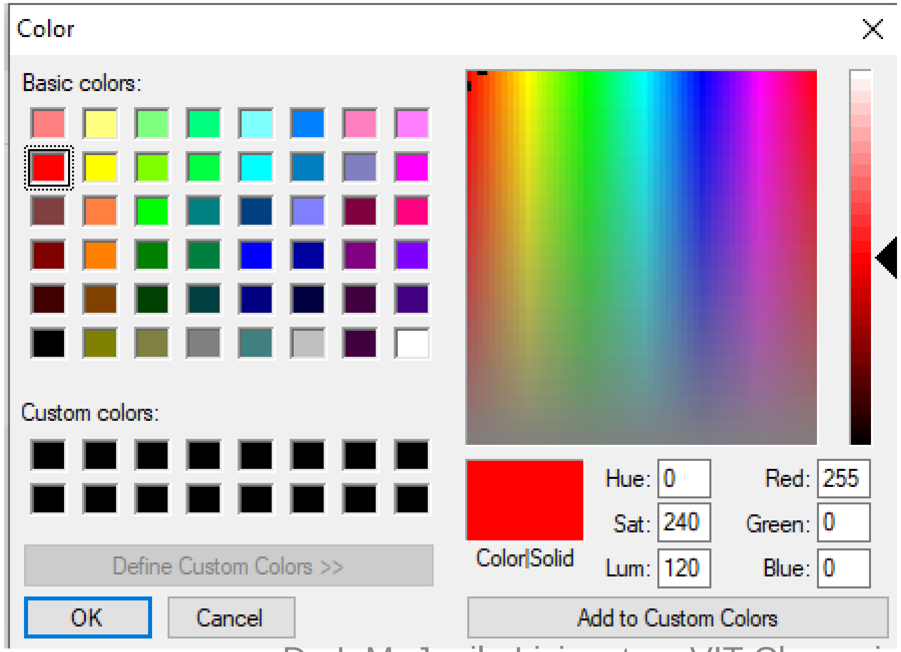
```
Color: <input type="color" name="user_color" />
```

Color Picker

```
<form>
  Select your favorite color:
  <input type="color" name="favcolor" value="#ff0000">
  <input type="submit">
</form>
```



Select your favorite color: 🟥 Submit

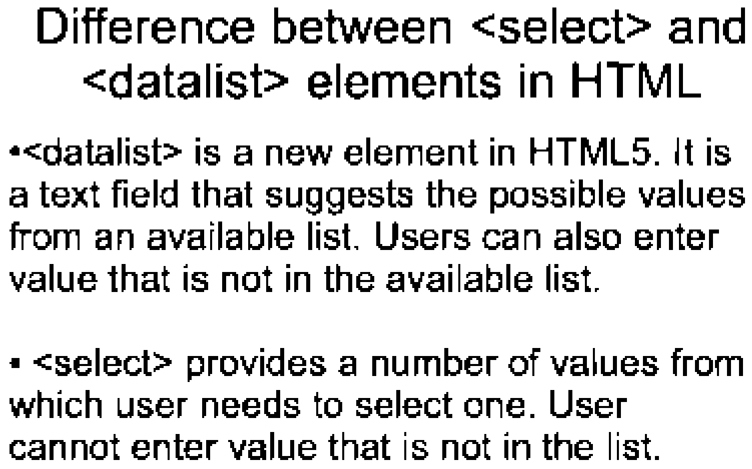


Hue: 0
Sat: 240
Lum: 120
Red: 255
Green: 0
Blue: 0

datalist

- The **datalist** element specifies a list of options for an input field. The list is created with option elements inside the datalist. (similar to SELECT tag)
- The **user can enter value that is not available in the list** also

```
Eg: <datalist id="browsers">
        <option value="Internet Explorer">
        <option value="Firefox">
     </datalist>
```



Difference between <select> and <datalist> elements in HTML

•<datalist> is a new element in HTML5. It is a text field that suggests the possible values from an available list. Users can also enter value that is not in the available list.

• <select> provides a number of values from which user needs to select one. User cannot enter value that is not in the list.

keygen

The purpose of the **keygen** element is to provide a secure way to authenticate users. (keytype:rsa,dsa,ec)

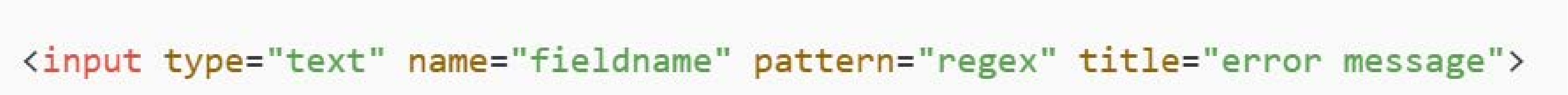
Eg: <keygen name="security" keytype="rsa">

RSA - Rivest–Shamir–Adleman
DSA - Digital Signature Algorithm)
EC - Elliptic Curve

Form validation attributes

required	Specifies that an input field is required (must be filled out)
pattern	Specifies a regular expression to check the input value against

pattern attribute



```
<input type="text" name="fieldname" pattern="regex" title="error message">
```

Example 1: **Validating a Phone Number**

Phone Number (10 digits):
<input type="tel" id="phone" name="phone" **pattern="[0-9]{10}**" title="Please enter a valid 10-digit phone number." required>

Example 2: **Validating a Password**

<input type="password" id="password" name="password" **pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[\W_]).{8,}"** title="Password must be at least 8 characters long, contain uppercase and lowercase letters, a number, and a special character." required>

Example 3: **Validating a Pancard**

<input type="text" id="pan" name="pan" **pattern="[A-Z]{5}[0-9]{4}[A-Z]{1}"** title="Please enter a valid PAN number (e.g., ABCDE1234F)." required>

Thank you!

HTML Frames

Dr. Jenila Livingston L.M.

Professor

VIT Chennai

Frames

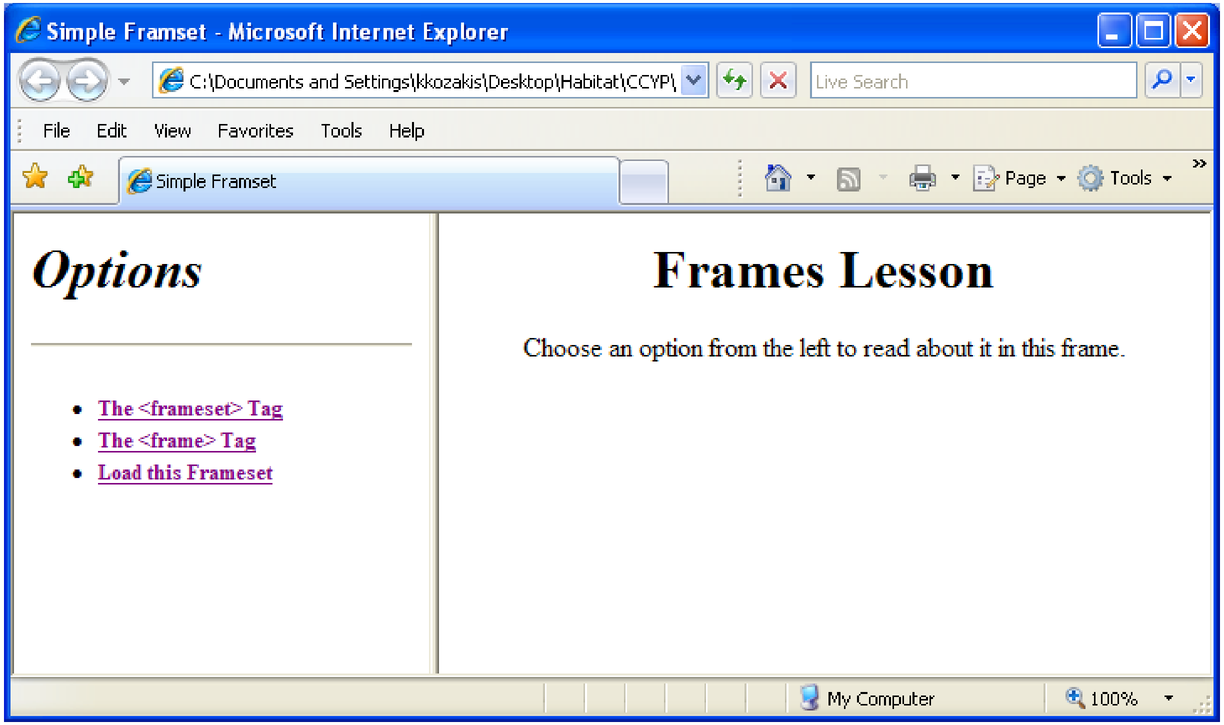
- **HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document.**

- **A collection of frames in the browser window is known as a frameset.**

The <frameset> Tag

- **<frameset> requires a closing </frameset> tag**
- **Determines the frame types and sizes on the page**
- **Two frameset types:**
  - **Columns**
  - **Rows**

Columns Example


Simple Framset - Microsoft Internet Explorer
C:\Documents and Settings\kkozakis\Desktop\Habitat\CCYP\
Live Search
File
Edit
View
Favorites
Tools
Help
Simple Framset
Page
Tools
My Computer
100%

Options

- The <frameset> Tag
- The <frame> Tag
- Load this Frameset

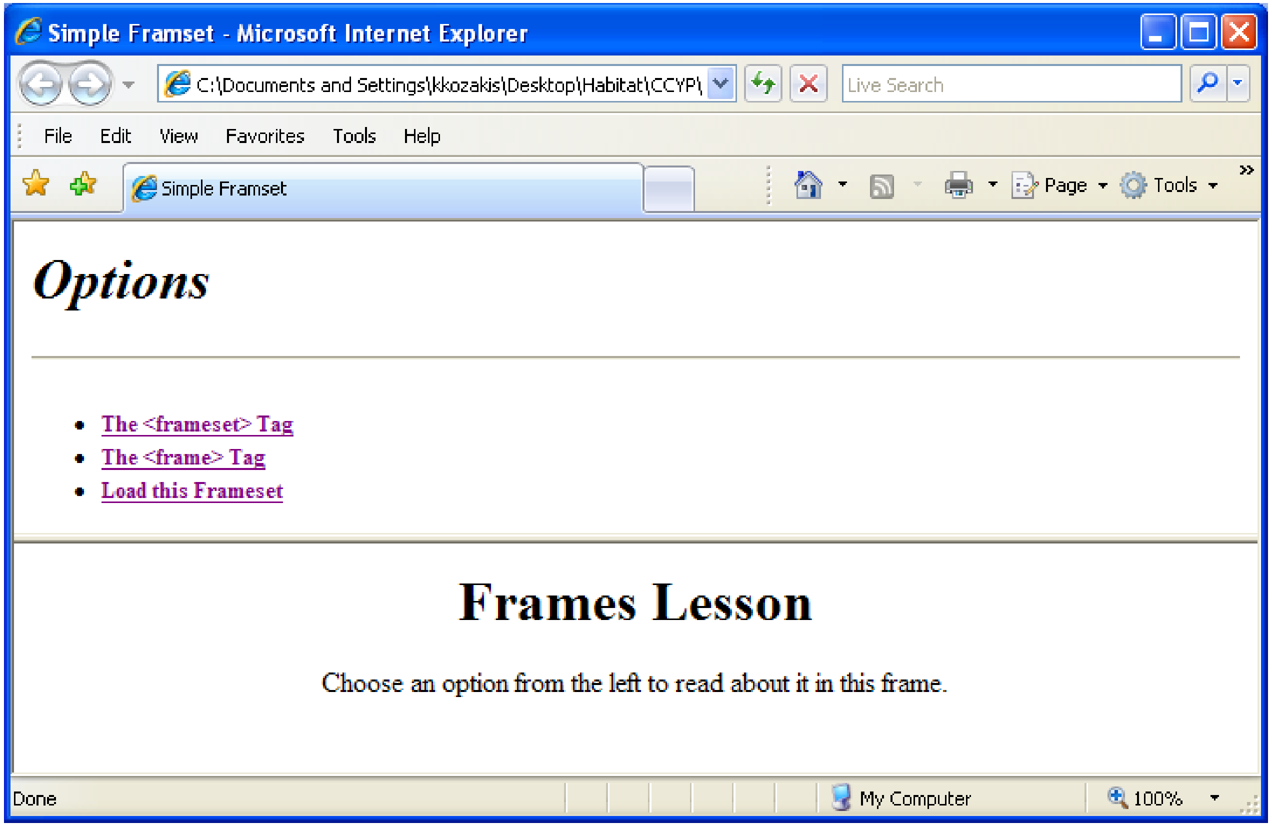
Frames Lesson

Choose an option from the left to read about it in this frame.

- **This frameset was created by the following code:**
- **<frameset cols="35%,65%"> </frameset>**

Absolute values in pixels. For example to create three vertical frames, use cols="100, 500,100".

Rows Example



Simple Framset - Microsoft Internet Explorer
C:\Documents and Settings\kkozakis\Desktop\Habitat\CCYP\
Live Search
File
Edit
View
Favorites
Tools
Help
Simple Framset
Page
Tools
Done
My Computer
100%

Options

- The <frameset> Tag
- The <frame> Tag
- Load this Frameset

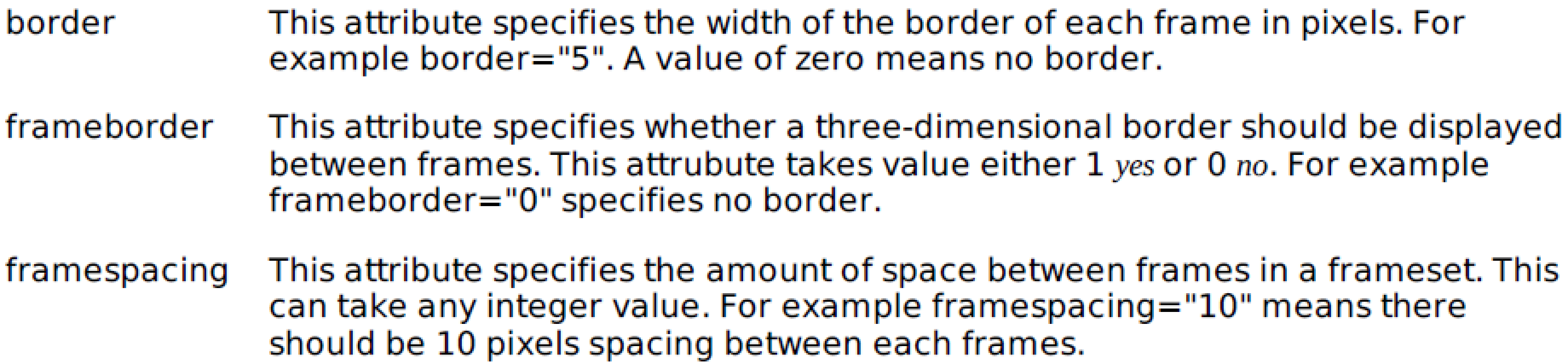
Frames Lesson

Choose an option from the left to read about it in this frame.

- **This frameset was created by the following code:**
- **<frameset rows=“65%,*"> </frameset>**

<frameset attributes>

- **rows**
- **cols**
- **border**
- **frameborder**
- **framespacing**



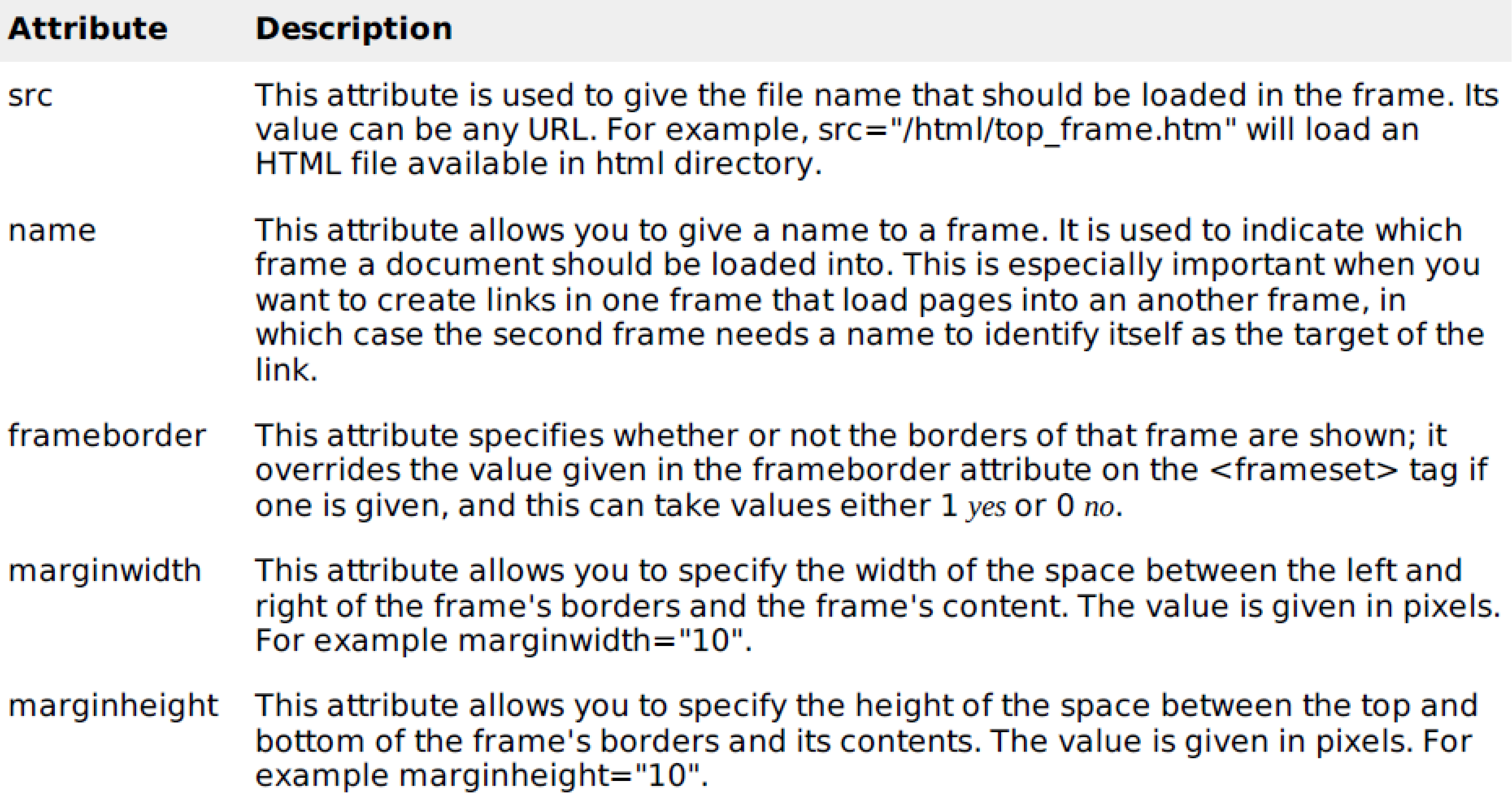
border	This attribute specifies the width of the border of each frame in pixels. For example border="5". A value of zero means no border.
frameborder	This attribute specifies whether a three-dimensional border should be displayed between frames. This attrubute takes value either 1 *yes* or 0 *no*. For example frameborder="0" specifies no border.
framespacing	This attribute specifies the amount of space between frames in a frameset. This can take any integer value. For example framespacing="10" means there should be 10 pixels spacing between each frames.

The <frame> Tag

- **Defines the content in each frame**
- **Placed between the <frameset> </frameset> tags**
- **The *src* attribute specifies the file that appears in the frame**
- **In the following example, the page that will appear in the top frame is the file fl-toc.html, and the page that will appear in the lower frame is fl-second.html.**

```
<frameset rows=“70%,*">
      <frame src="fl-toc.html"/>
      <frame src="fl-second.html"/>
</frameset>
```

<frame> Attributes



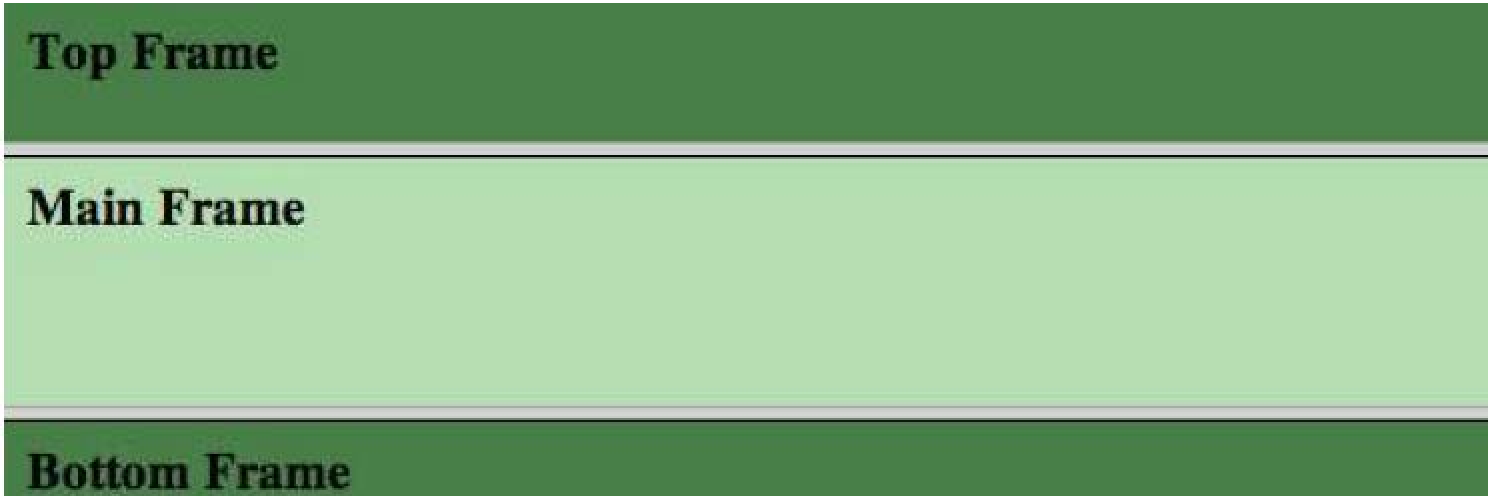
Attribute	Description
src	This attribute is used to give the file name that should be loaded in the frame. Its value can be any URL. For example, src="/html/top_frame.htm" will load an HTML file available in html directory.
name	This attribute allows you to give a name to a frame. It is used to indicate which frame a document should be loaded into. This is especially important when you want to create links in one frame that load pages into an another frame, in which case the second frame needs a name to identify itself as the target of the link.
frameborder	This attribute specifies whether or not the borders of that frame are shown; it overrides the value given in the frameborder attribute on the <frameset> tag if one is given, and this can take values either 1 *yes* or 0 *no*.
marginwidth	This attribute allows you to specify the width of the space between the left and right of the frame's borders and the frame's content. The value is given in pixels. For example marginwidth="10".
marginheight	This attribute allows you to specify the height of the space between the top and bottom of the frame's borders and its contents. The value is given in pixels. For example marginheight="10".

<frame> Attributes

noresize
By default you can resize any frame by clicking and dragging on the borders of a frame. The noresize attribute prevents a user from being able to resize the frame. For example noresize="noresize".

scrolling
This attribute controls the appearance of the scrollbars that appear on the frame. This takes values either "yes", "no" or "auto". For example scrolling="no" means it should not have scroll bars.

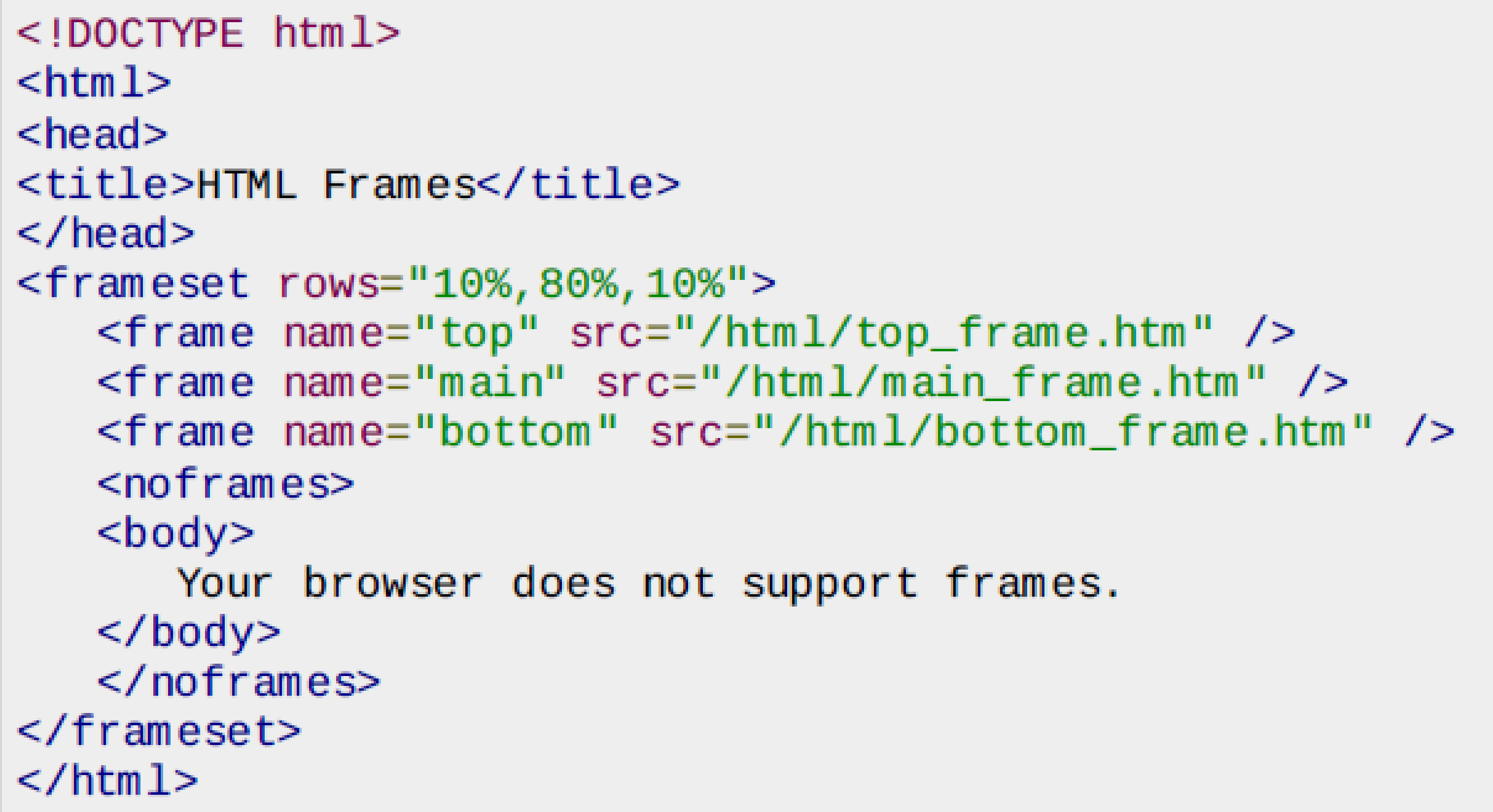
longdesc
This attribute allows you to provide a link to another page containing a long description of the contents of the frame. For example longdesc="framedescription.htm"



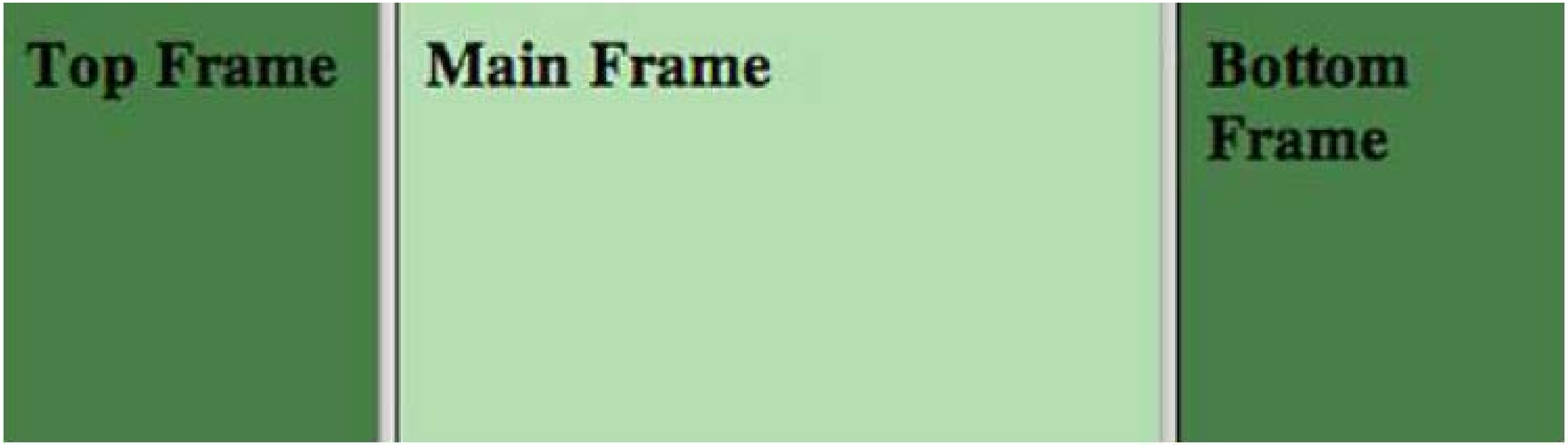
Top Frame

Main Frame

Bottom Frame



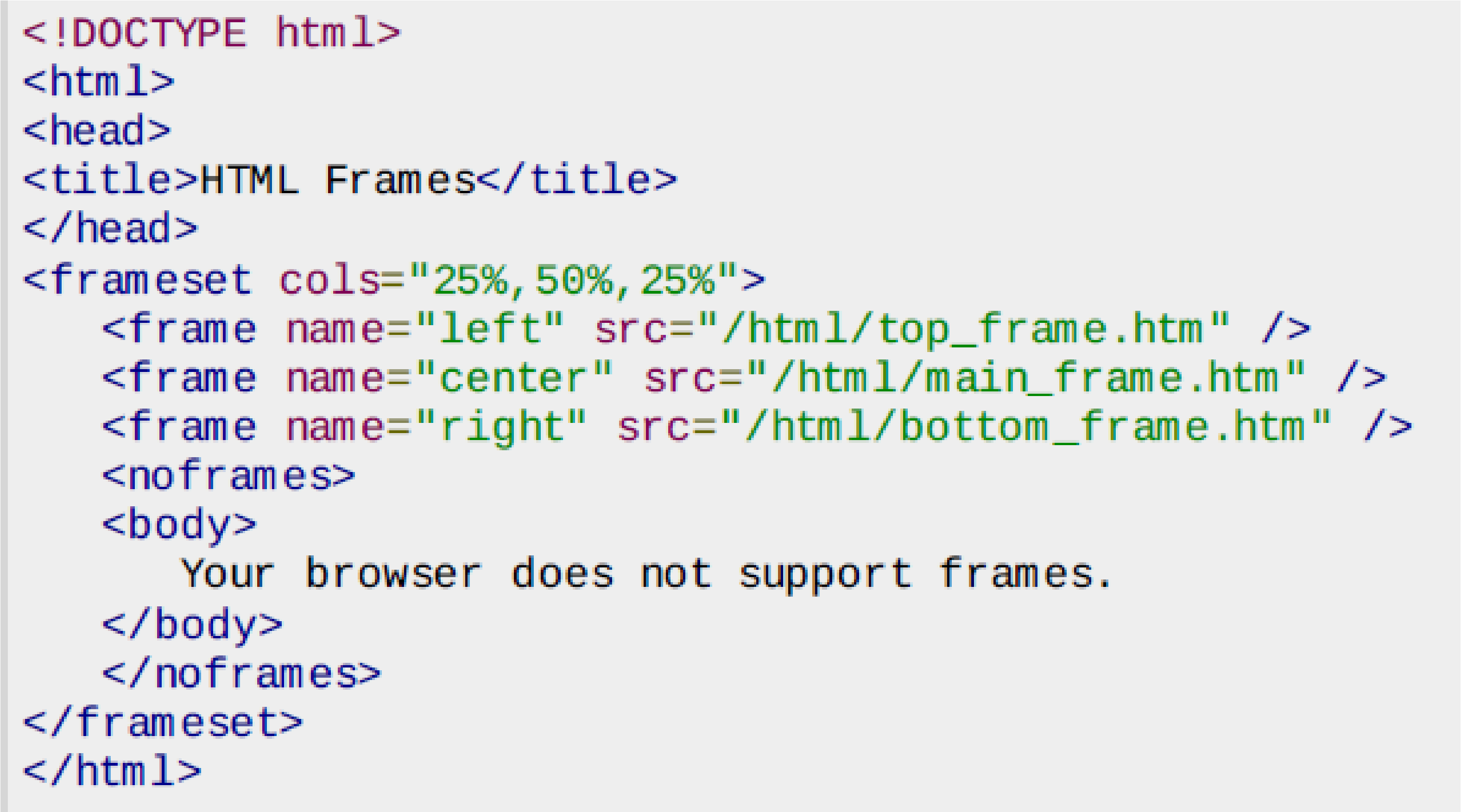
```html
<!DOCTYPE html>
<html>
<head>
<title>HTML Frames</title>
</head>
<frameset rows="10%,80%,10%">
   <frame name="top" src="/html/top_frame.htm" />
   <frame name="main" src="/html/main_frame.htm" />
   <frame name="bottom" src="/html/bottom_frame.htm" />
   <noframes>
   <body>
      Your browser does not support frames.
   </body>
   </noframes>
</frameset>
</html>
```



Top Frame

Main Frame

Bottom Frame



```
<!DOCTYPE html>
<html>
<head>
<title>HTML Frames</title>
</head>
<frameset cols="25%,50%,25%">
   <frame name="left" src="/html/top_frame.htm" />
   <frame name="center" src="/html/main_frame.htm" />
   <frame name="right" src="/html/bottom_frame.htm" />
   <noframes>
   <body>
      Your browser does not support frames.
   </body>
   </noframes>
</frameset>
</html>
```

The <noframes> Tag

- **If a user is using any old browser or any browser which does not support frames then <noframes> element should be displayed to the user.**
- **So you must place a <body> element inside the <noframes> element because the <frameset> element is supposed to replace the <body> element, but if a browser does not understand<frameset> element then it should understand what is inside the <body> element which is contained in a <noframes> element.**

Targeting Frames with Hyperlinks

- **Use the *name* attribute to name a frame, then target the frame name with hyperlinks**
- **The syntax for naming a frame is as follows:**

  <frame src="*url*" name="*framename*"/>

- **The following code names a frame:**

  <frame src="james.html" name="authors"/>

- **The following code targets this frame:**

  <a href="james.html" target= "authors"> Visit James </a>

- **If a user clicks the Visit James link, the James page will open in the Authors frame**

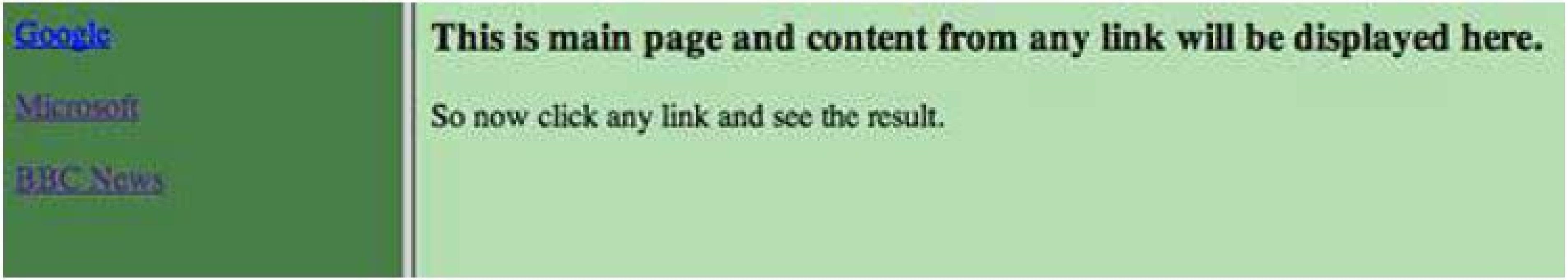
Base Target

- **A base target automatically sets a default target frame for all links in a page**
- **Created using the <base> tag**
- **Code:**

```
<base target="main" href="page.html"/>
```

- **This code will cause all linked pages to open in the frame named Main**
- **The *href* attribute is optional**

Example Page



Google

Microsoft

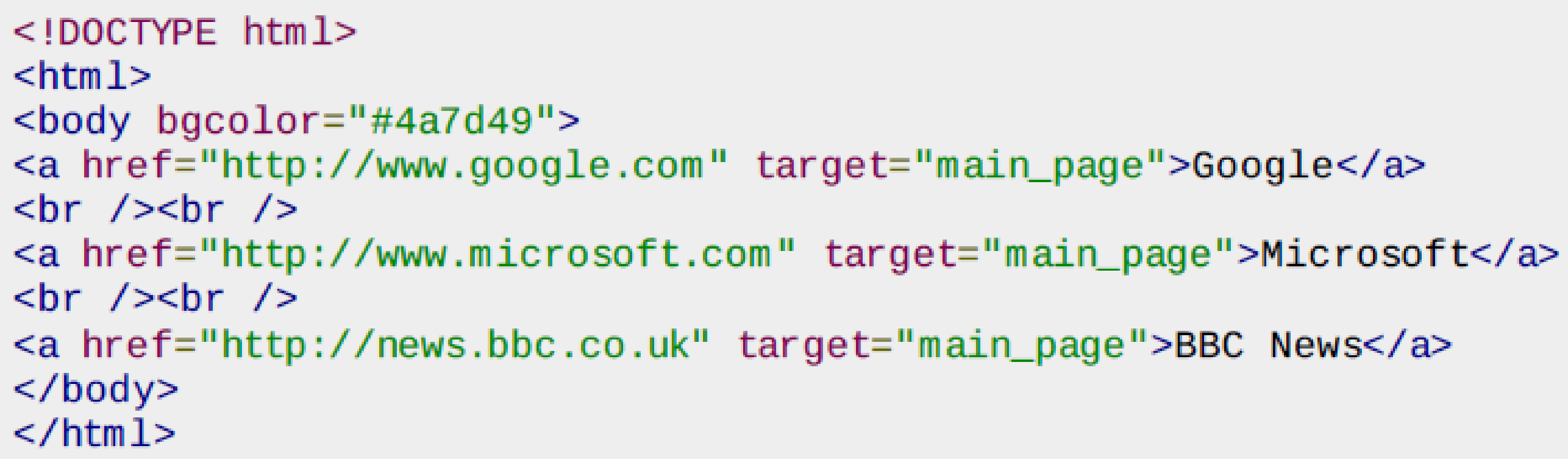
BBC News

This is main page and content from any link will be displayed here.

So now click any link and see the result.

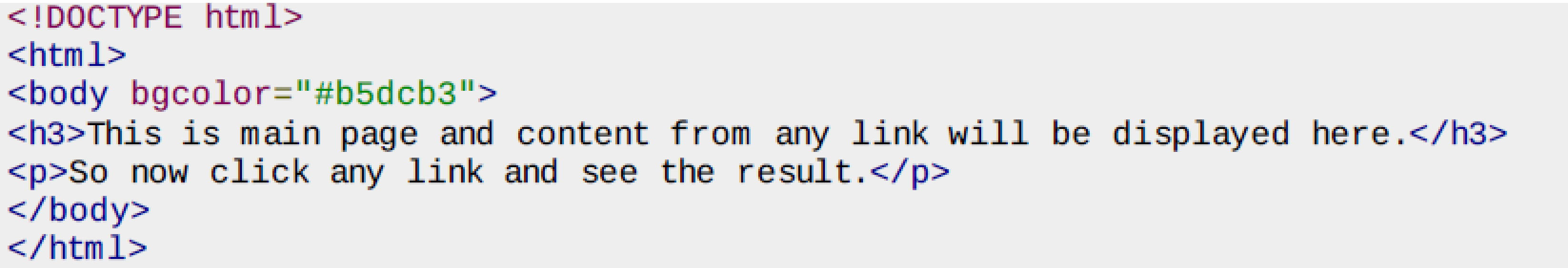
```
<frameset cols=“30%,*”>
        <frame name=“nav” src=“menu.html”>
        <frame name=“main_page” src=“content.html”>
</frameset>
```

menu.html



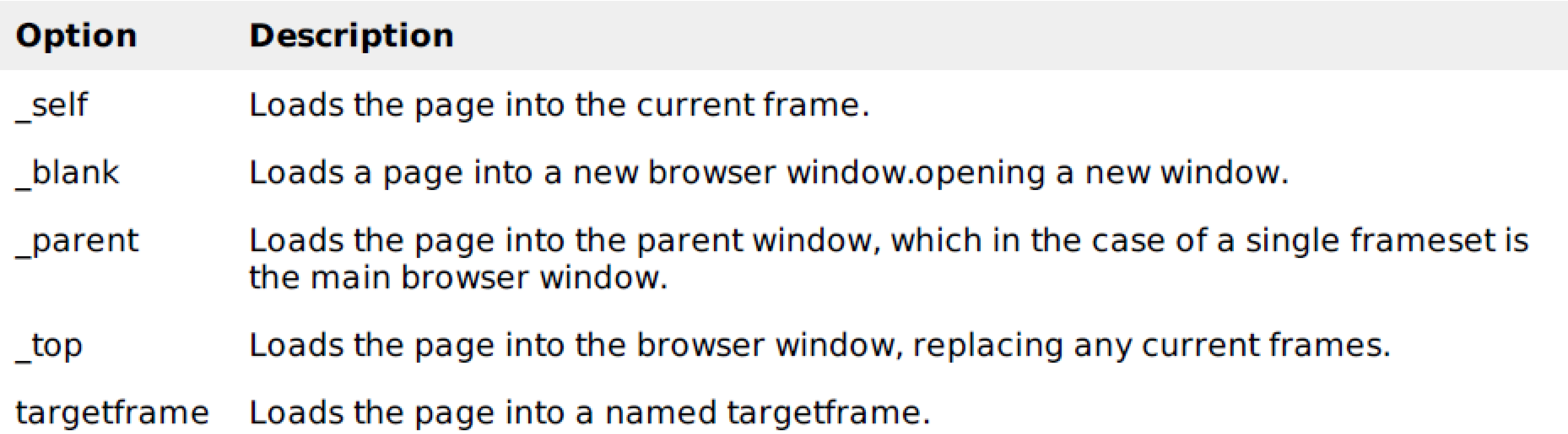
```
<!DOCTYPE html>
<html>
<body bgcolor="#4a7d49">
<a href="http://www.google.com" target="main_page">Google</a>
<br /><br />
<a href="http://www.microsoft.com" target="main_page">Microsoft</a>
<br /><br />
<a href="http://news.bbc.co.uk" target="main_page">BBC News</a>
</body>
</html>
```

Content.html

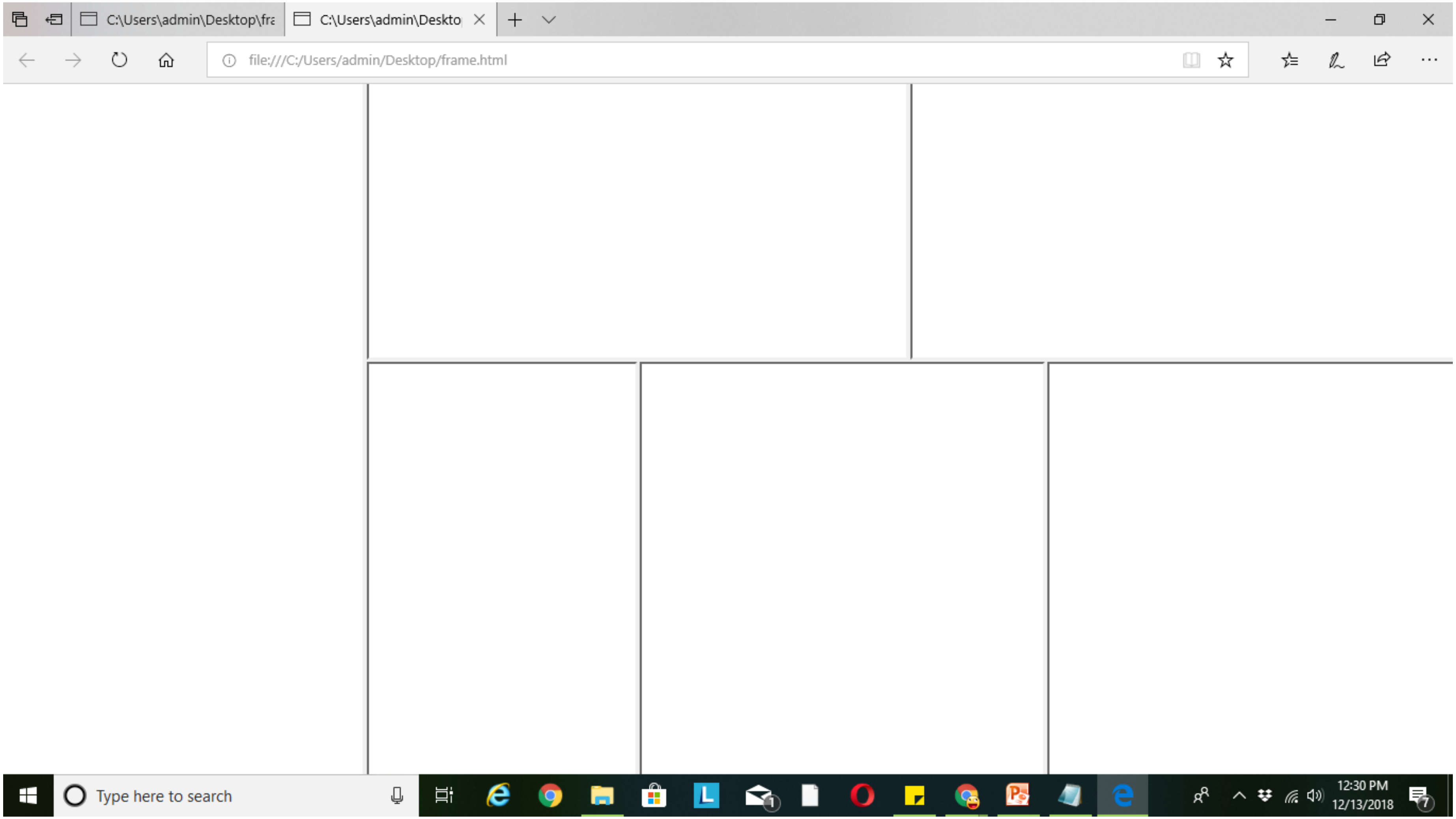


```
<!DOCTYPE html>
<html>
<body bgcolor="#b5dcb3">
<h3>This is main page and content from any link will be displayed here.</h3>
<p>So now click any link and see the result.</p>
</body>
</html>
```

Anchor tag target values



Option	Description
_self	Loads the page into the current frame.
_blank	Loads a page into a new browser window.opening a new window.
_parent	Loads the page into the parent window, which in the case of a single frameset is the main browser window.
_top	Loads the page into the browser window, replacing any current frames.
targetframe	Loads the page into a named targetframe.



C:\Users\admin\Desktop\fra
C:\Users\admin\Deskto
file:///C:/Users/admin/Desktop/frame.html
Type here to search
12:30 PM
12/13/2018

Program

```
<frameset cols="25%,*">
    <frame name="f1">
    <frameset rows="40%,60%">
        <frameset cols="50%,50%">
            <frame name="f2">
            <frame name="f3">
        </frameset>
        <frameset cols="25%,*,*">
            <frame name="f4">
            <frame name="f5">
            <frame name="f6">
        </frameset>
    </frameset>
</frameset>
```

Inline Frames

- **Inserts an HTML or XHTML document inside another**
- **Also called "floating frames"**
- **Created with the <iframe> tag**
- **The browser reads the <iframe> tag from the file, then makes a separate request to the server for the embedded file**

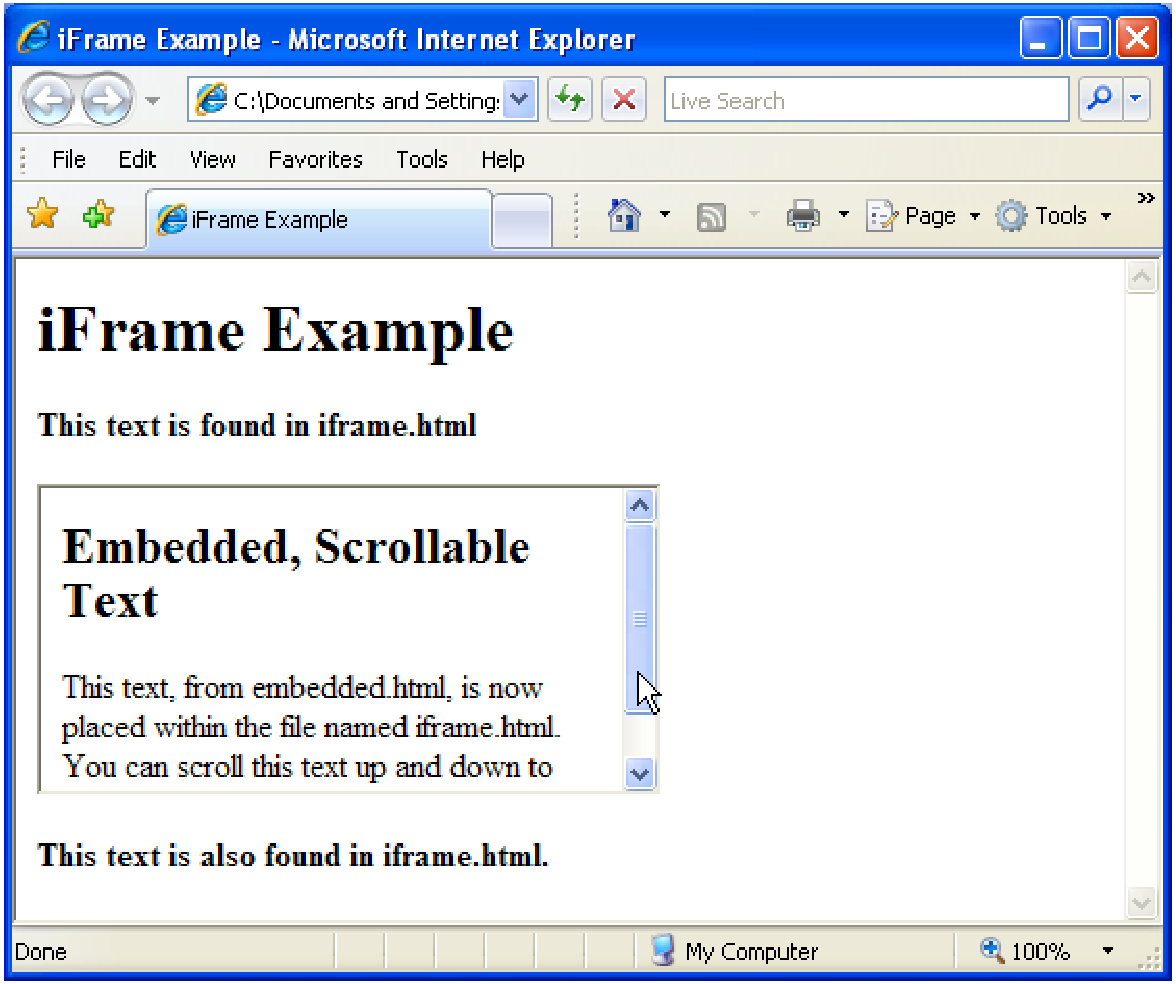
Inline Frames *(cont’d)*

- **Simple XHTML page with inline frame:**

```
<h1>iFrame Example</h1>
<p><strong>This text is found in
iframe.html</strong><p/>

<iframe src="embedded.html" scrolling= "yes">
        Your browser does not support frames.
</iframe>

<p><strong>This text is also found in iframe.html.
</strong></p>
```

- **The next slide shows the results of this code…**

Inline Frames *(cont'd)*



iFrame Example

This text is found in iframe.html

Embedded, Scrollable Text

This text, from embedded.html, is now placed within the file named iframe.html. You can scroll this text up and down to

This text is also found in iframe.html.