

Chapter 1

- ✓ **World Wide Web (WWW)**
- ✓ **ARPANET (Advanced Research Projects Agency Network)**
- ✓ **Hyper Text Transfer Protocol(HTTP)**
- ✓ **Uniform Resource Locator (URL)**
 - A client that wants to access the document in an internet needs an address and to facilitate the access of documents, the HTTP uses the concept of Uniform Resource Locator (URL).

*In 1989, **Tim Berners Lee** began to develop a technology for sharing information via hyperlinked text documents. In **October-1994 Tim Berners Lee** founded an organization called **W3C (World Wide Web Consortium)** devoted to developing technologies for www.*

Messages:

Request Message: The request message is sent by the client that consists of a **request line**, **headers**, and **sometimes a body**.

Initial Request Line

Syntax:

GET /path/to/file/index.html HTTP/1.0

Response Message: The response message is sent by the server to the client that consists of a **status line**, **headers**, and **sometimes a body**.

Initial Response Line/Status Line

Example:

HTTP/1.0 200 OK

or

HTTP/1.0 404 Not Found

Here,
The HTTP version of the response line and request line are the same as "HTTP/x.x"

HTTP/1.1 200 OK

HTTP	Status	Reason
version	Code	Phrase

HTTP status codes

HTTP defines these standard status codes that can be used to convey the results of a client's

request. The status codes are divided into five categories.

- ✓ 1xx: Informational – Communicates transfer protocol-level information.
- ✓ 2xx: Success – Indicates that the client's request was accepted successfully.
- ✓ 3xx: Redirection – Indicates that the client must take some additional action in order to complete their request.
- ✓ 4xx: Client Error – This category of error status codes points the finger at clients.
- ✓ 5xx: Server Error – The server takes responsibility for these error status codes.

<u>200 OK</u>	<u>301 Moved Permanently</u>	<u>400 Bad Request</u>
<u>401 Unauthorized</u>	<u>402 Payment Required</u>	<u>403 Forbidden</u>
<u>404 Not Found</u>	<u>502 Bad Gateway</u>	<u>503 Service Unavailable</u>

Chapter 2

- **Define keywords for search engines:**

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

- **Define a description of your web page:**

```
<meta name="description" content="Web tutorials for HTML">
```

- **Define the author of a page:**

```
<meta name="author" content="LJU">
```

- **Set the viewport to make your website responsive and look good on all devices:**

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- **Refresh: Defines a time interval for the document to refresh itself.**

```
<meta http-equiv="refresh" content="300">
```

❖ HTML Comments

```
<!-- Write your comments here -->
```

❖ HTML Headings

HTML headings are titles or subtitles that you want to display on a webpage.

```
<h1>Heading 1</h1>
```

```
<h2>Heading 2</h2>
```

```
<h3>Heading 3</h3>
```

```
<h4>Heading 4</h4>
```

```
<h5>Heading 5</h5>
```

```
<h6>Heading 6</h6>
```

❖ The HTML <pre> tag and <p> tag

Example

```
<!-- Example with <pre> -->
```

```
<pre>
```

```
This is preformatted
text, with line breaks
```

```
and extra spaces retained.
</pre>
```

```
<!-- Example with <p> -->
```

```
<p>
This is paragraph text,
with line breaks and extra spaces
ignored.
</p>
```

Output:

This is preformatted
text, with line breaks
and extra spaces retained.

This is paragraph text, with line breaks and extra spaces ignored.

❖ Data Formatting tags

1. ****: Defines bold text
2. ****: The HTML element defines text with strong importance. The content inside is typically displayed in bold.

In summary, use for purely stylistic bolding and when the text has more importance or needs to be semantically emphasized.

3. **<i>**: The content inside is displayed in *italic*.
4. ****: It is also one of the element of HTML used in formatting texts. It is used to define emphasized text or statements. The content inside is displayed in *italic*.

Note:

Despite the fact that both and <i> tags give the same visual effect, i.e. italic text, search engine robots pay more attention to the first one. That's why it is recommended to use the **** tag for **website optimization**. tag informs Google crawlers that the selected content is particularly important and deserves attention.

Same for the and tags. Both tags give the same visual effect, but **** tag informs Google crawlers that the selected content is particularly important and deserves attention.

5. ****: The tag in HTML plays an important role in the web page to create an attractive and readable web page. The font tag is used to change the color, size, and style of a text. The base font tag is used to set all the text to the same size, color and face.

Default size = 3, Range of size = 1 to 7

```
<body>
  <font size="5" face="Comic sans MS" color="green"> Welcome to LJU </font>
</body>
```

Output: Welcome to LJU

6. **<u>**: If you write anything within <u>.....</u> element, is shown in underlined text.
7. **<ins>**: The <ins> tag defines a text that has been inserted into a document. Browsers will usually underline inserted text.
8. ****: The tag defines text that has been deleted from a document. Browsers will usually strike a line through deleted text.

Example:

```
<h1>My favorite color is <del>blue</del> <ins>red</ins>!</h1>
```

Output:

My favorite color is ~~blue~~ red!

Note: <u> is purely for **visual underlining** without meaning. <ins> is for **indicating added or inserted content**, with underlining being the default style. In modern web development, it's generally better to use semantic tags like <ins> when conveying meaning and <u> only when you want underlining purely for design.

9. **<hr>**: **Horizontal Rule.** The <hr> element is most often displayed as a horizontal rule that is used to separate content in an HTML page.

Attribute	Value	Description
align	left center right	Used to specify the alignment of the horizontal rule. Default is center.
noshade	noshade	Used to specify the bar without shading effect.
size	pixels	Used to specify the height of the horizontal rule.
width	Pixels/percentage	Used to specify the width of the horizontal rule.

```
<h3>Textt1</h3>
```

```
<hr size="10" width="40%">
<h3>Textt2</h3>
<hr size="20" width="50%" noshade align="right">
<h3>Textt3</h3>
<hr size="2">
```

Output:

Textt1

Textt2

Textt3

**10.
:**

- The **
** tag inserts a single line break.
- This tag is an empty tag which means that it has no end tag.

```
<body>
<p>This is sample example</p>
<p>This is <br> sample example</p>
</body>
```

Output:

This is sample example

This is
sample example

11.<center>: **Not Supported in HTML5.** The **<center>** tag was used in HTML4 to align text in center.

12.<sup>: The **<sup>** tag is used to add a superscript text to the HTML document.

```
<h3>2<sup>2</sup> + 5<sup>2</sup> = 29</h3>
```

Output:

$2^2 + 5^2 = 29$

13.<sub>: The **<sub>** tag is used to add a subscript text to the HTML document.

```
<h4>CH<sub>3</sub></h4>
```

```
<h4>H<sub>2</sub>SO<sub>4</sub></h4>
```

Output:

CH₃

H₂SO₄

14. **<mark>**: The `<mark>` tag defines text that should be marked or highlighted.

```
<h1>This is <mark>Highlighted text!!</mark></h1>
```

Output:

This is **Highlighted text!!**

15. **<small>**: The `<small>` tag defines smaller text (like copyright and other side-comments).

Tip: This tag is not deprecated, but it is possible to achieve richer (or the same) effect with CSS.

16. **<big>**: **Not Supported in HTML5.** The `<big>` tag was used to define bigger text.

Example:

```
<p>This is a normal paragraph.</p>
<p><small>This is a normal paragraph.</small></p>
<p><big>This is a normal paragraph.</big></p>
```

Output:

This is a normal paragraph.

This is a normal paragraph.

This is a normal paragraph.

❖ HTML Link: Anchor tag (<a>)

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.

Attributes:

- ✓ **href:** Specifies the URL of the page the link goes to
- ✓ **target:** Specifies where to open the linked document

```
<a href="url" target="_blank | _self | _parent | _top " name="a_link">Test</a>
```

_blank = Opens the linked document in a new window or tab

_self = Opens the linked document in the same frame as it was clicked (this is default)

_parent = Opens the linked document in the parent frame

_top = Opens the linked document in the full body of the window

```
<a href="https://www.example.com" target="_blank">Click here</a>
```

❖ Images: tag

The tag is used to embed an image in an HTML page. Images are not technically inserted into a web page; images are linked to web pages. The tag creates a holding space for the referenced image.

```

```

❖ HTML Lists:

There are two types of lists:

- 1) **Unordered list:** An unordered list starts with the tag. Each list item starts with the tag. The list items will be marked with disc (small black circles) by default.

Attribute:

type: type = " disc/square/circle/none"

```
<ul>
```

```
<li>Test 1</li>
```

```
<li>Test 2</li>
```

```
<ul type="square">
```



```

    <li>Test 3</li>
    <li>Test 4</li>
  </ul>
</ul>

```

Output:

- Test 1
- Test 2
 - Test 3
 - Test 4

type="disc":

Items are marked with **filled circles** as bullet points. This is default.

- Item 1
- Item 2
- Item 3

type="square":

Items are marked with **squares**.

- Item 1
- Item 2
- Item 3

type="circle":

Items are marked with **hollow circles**.

- Item 1
- Item 2
- Item 3

type="none":

No bullet points are shown for the list items.

- Item 1
- Item 2
- Item 3

2) **Ordered list:** An ordered list starts with the tag. Each list item starts with the tag. The list items will be marked with numbers by default:

Attribute:

type: type=" 1/ i/ l/ a/ A"

start: Specifies the start value of an ordered list

reversed: Specifies that the list order should be reversed (9,8,7...)

```
<ol start="5">
  <li>Test 1</li>
  <li>Test 2</li>
  <li>Test 3</li>
  <ol type="A">
    <li>Test 4</li>
    <li>Test 5</li>
    <ol reversed type="i">
      <li>Test 6</li>
      <li>Test 6</li>
    </ol>
  </ol>
</ol>
```

Output:

```
5. Test 1
6. Test 2
7. Test 3
  A. Test 4
  B. Test 5
    ii. Test 6
    i. Test 6
```

For type 'a' 27,28,29.. letters are aa,ab,ac... respectively.

❖ HTML Definition List:

The <dl> tag defines a description list. The <dl> tag is used in conjunction with <dt> (defines terms/names/title) and <dd> (describes each term/name/data).

- <dl> : definition list
- <dt> : definition term
- <dd> : definition detail

❖ HTML Table:

- ✓ The <table> tag defines an HTML table.
- ✓ An HTML table consists of one <table> element and one or more <tr>, <th>, and <td> elements.
- ✓ The <tr> element defines a table row, the <th> element defines a table header, and the <td> element defines a table cell.
- ✓ An HTML table may also include <caption>, <thead>, <tfoot>, and <tbody> elements.

Tags for table:

<caption>	The <caption> tag defines a table caption.
<table>	Defines a table.
<th>	Defines a header cell in a table
<tr>	Defines a row in a table
<td>	Defines a cell in a table
<thead>	Groups the header content in a table
<tbody>	Groups the body content in a table
<tfoot>	Groups the footer content in a table

HTML <table> tag attributes:

Attribute	Value	Description
align	right left center justify char	Deprecated – Visual alignment.

bgcolor	rgb(x,x,x) #hexcode colorname	Deprecated – Specifies the backgroundcolor of the table.
border	pixels	Deprecated – Specifies the border width.A value of "0" means no border.
cellpadding	pixels or %	Deprecated – Specifies the space between the cell borders and their contents.
cellspacing	pixels or %	Deprecated – Specifies the space between cells.
rules	none groups rows cols all	Deprecated – The HTML <table> rules Attribute is used to <i>specify which parts of the inside borders that should be visible</i> .
width	pixels or %	Deprecated – Specifies the width of the table.

<td><th><tr> tags attributes:

Attribute	Value	Description
align	right left center justify char	Deprecated – Visual alignment.
bgcolor	rgb(x,x,x) #hexcode colorname	Deprecated – Specifies the backgroundcolor of the cell.
colspan	Number of columns to merge	Number of columns a header cell should span
rowspan	Number of rows to merge	Set the number of rows a header cell should span.

Example 1:

Table

Name	Cost
A	20
B	30
Total	50

```
<table border="4" cellpadding="10" rules="all" bgcolor="pink" width="50%" align="center">
  <caption>Table</caption>
  <thead>
    <tr>
      <th>Name</th>
      <th>Cost</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>A</td>
      <td>20</td>
    </tr>
    <tr>
      <td>B</td>
      <td>30</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <th>Total</th>
      <td>50</td>
    </tr>
  </tfoot>
</table>
```

Example 2

C	A
	b

```
<table border="2" cellspacing="4" cellpadding="30">
  <tr>
    <td rowspan="2">C</td>
    <td>A</td>
  </tr>
  <tr>
    <td>b</td>
  </tr>
</table>
```

Example3:

A	
B	C

```
<table border="2" rules="rows" cellpadding="30">
  <tr>
    <td colspan="2" align="center">A</td>
  </tr>
  <tr>
    <td>B</td>
    <td>C</td>
  </tr>
</table>
```

❖ HTML Frame:

- ✓ 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 window is divided into frames in a similar way the tables are organized: into rows and columns.

The `<frame>` tag has been **deprecated in HTML5** and should no longer be used in modern web development. In older versions of HTML (like HTML4), if you use `<frameset>` to define a layout with frames, the **main HTML page should not include a `<body>` tag**. If you include the `<body>` tag, it will ignore the `<frameset>` and only display the content inside the `<body>` tag.

Here's a simple frame document:

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
<TITLE>A simple frameset
document</TITLE>
</HEAD>
<FRAMESET cols="20%, 80%">
  <FRAMESET rows="10%, *">
    <FRAME src="frame1.html">
    <FRAME src="frame2.gif">
  </FRAMESET>
  <FRAME src="frame3.html">
</FRAMESET>
</HTML>
```



The page is split into two main **columns**:

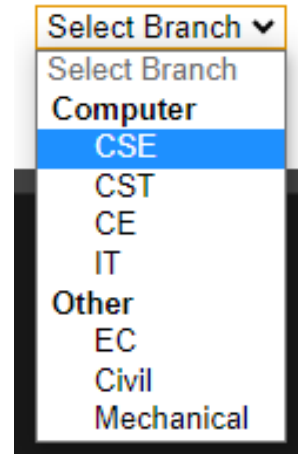
- The left column occupies **20%** of the page width and is further split into **two rows**:
 - The top row (10% height) displays the content of frame1.html.
 - The bottom row (remaining height) displays the image frame2.gif.
- The right column occupies **80%** of the page width and displays the content of frame3.html.

❖ HTML Forms <form>:

Optgroup and selected,hidden,disabled example

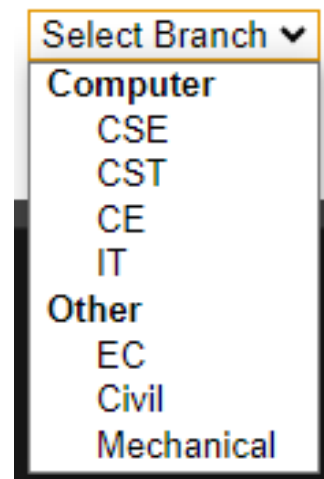
```
<select name="degree_branch">
  <option value="select" selected disabled>
Select Branch</option>
  <optgroup label="Computer">
    <option value="cse"> CSE </option>
    <option value="cst"> CST </option>
    <option value="ce"> CE </option>
    <option value="it"> IT </option>
  </optgroup>
  <optgroup label="Other">
    <option value="ec"> EC </option>
    <option value="civil"> Civil </option>
    <option value="Mech"> Mechanical </option>
  </optgroup>
</select>
```

Selected attribute keeps the option selected at first time we load the web page.



Disabled attribute disables the option. We cannot select the disabled option.

```
<select name="degree_branch">
  <option value="select" selected hidden>
Select Branch</option>
  <optgroup label="Computer">
    <option value="cse"> CSE </option>
    <option value="cst"> CST </option>
    <option value="ce"> CE </option>
    <option value="it"> IT </option>
  </optgroup>
  <optgroup label="Other">
    <option value="ec"> EC </option>
    <option value="civil"> Civil </option>
    <option value="Mech"> Mechanical </option>
  </optgroup>
</select>
```



Hidden attribute hides the option.

Example

```

<form method="get" action="n1.html" align="left">
  <fieldset>
    <legend align="left"> Sign-Up</legend>
    <label for="uname"> User Name <input id="uname" type="text"
placeholder="Username" name="uname" autofocus autocomplete required/> </label>

    <label for="pass"> Password <input type="password" id = "pass" maxlength="8"
minlength="3" value="123" disabled/> </label> <br/><br/>

    <label> Your Email ID <input type="email" value="email" readonly placeholder="this
is readonly field" /> </label> <br/><br/>

    <label >Age:<input type="range" id="age" name="age" min="20" max="70"
step="2"></label><br/><br/>

    <label for="age">age <input type="number" id="age" name="number" min="10"
max="20" ></label><br/><br/>

    <label> Gender <input type="radio" checked name="gender" value="male"/> Male
</label>
    <label><input type="radio" name="gender" value="female"/> Female </label>
<br/><br/>

    <label> Subjects</label>
    <input type="checkbox" name="subject" value="DS"/> DS
    <input type="checkbox" name="subject" value="DBMS" /> DBMS
    <input type="checkbox" name="subject" value="CN"/> CN
    <input type="checkbox" name="subject" value="FSD"/> FSD <br/><br/>

    <label>DOB <input type="date" name="dob" min="2023-10-01" > <!-- yyyy-mm-
dd --></label><br/><br/>

    <label for="s1">search <input type="search" name="s" id="s1"></label><br/><br/>

    <label> Department </label>
    <select name="department">
      <option value=" " selected hidden> Select Department</option>
      <optgroup label="CE">
        <option> CSE </option>
        <option> CST </option>
        <option> CSD </option>

```

```

    <option disabled> RAI </option>
  </optgroup>
  <optgroup label="Mech">
    <option> CAD </option>
    <option> CAM </option>
    <option> Thermal </option>
  </optgroup>
</select> <br/><br/>

```

```

<label> Address <textarea name="addr" rows="6" cols="20">Enter Your
Address</textarea> </label><br/><br/>

```

```

<label> Upload CV <input type="file" name="f1" multiple/></label> <br/><br/>

```

```

<input type="image" src="submit.jpg" alt="Photo" width="150"
height="100"><br/><br/>

```

```

<input type="submit" value="Click Me"/>
<input type="button" value="button submit"><br/><br/>
<input type="reset" value="Reset"/>
</fieldset>
</form>

```

❖ HTML Block and Inline Elements

Every HTML element has a default display value, depending on what type of element it is. There are two display values: block and inline.

Block-level Elements

- A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.
- A block-level element always takes up the full width available (stretches out to the left and right as far as it can).
- Two commonly used block elements are: **<p> and <div>**.
- The <p> element defines a paragraph in an HTML document.
- The <div> element defines a division or a section in an HTML document.
- Other examples are ,<pre>,,,<section>,<dl>,<dd>,<dt> etc

Inline Elements

- An inline element does not start on a new line.
- An inline element only takes up as much width as necessary.
- For an example **** element inside a paragraph.
- Other examples are <a>,,<i>,<sup>,<sub> etc

Chapter 3

New Semantic elements added

Semantic Elements: Semantic elements have meaningful names which tells about type of content. For example, header, footer, table, ... etc. HTML5 introduces many semantic elements as mentioned below which make the code easier to write and understand for the developer as well as instructs the browser on how to treat them.

- article
- aside
- footer
- header
- nav
- section

<iframe>

```
<iframe
src="https://www.youtube.com/embed/LXb3EKWsInQ?playlist=2 kAzyaX7SU,oHdecbMr
cbl,LXb3EKWsInQ,kVxTrhojpFI&loop=1&autoplay=1&mute=1&controls=0"
title="YouTubevideo playlist" allowfullscreen>
</iframe>
```

<audio>

```
<audio controls loop autoplay muted>
  <source src="a1.mp3" type="audio/mp3">
  Your browser does not support the video tag.
</audio>
```

<video>

```
<video width="320" height="240" loop autoplay muted controls>
  <source src="v1.mp4" type="video/mp4">
  <source src="v1.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

Write an HTML script to play a YouTube video looping forever & autoplaying as soon as the page loads.

```
<!-- YouTube video iframe embedded -->
<iframe width="560"height="315"
src="https://www.youtube.com/embed/VIDEO_ID?autoplay=1&loop=1&playlist=VIDEO_ID&mute=1" title="YouTube video player" allowfullscreen>
</iframe>
```

1. SVG Line - <line>

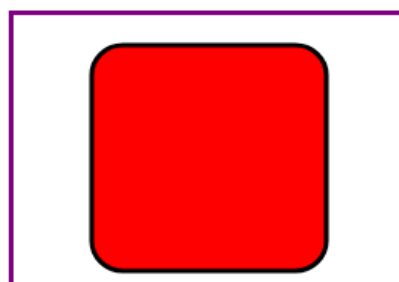
```
<svg height="250" width="500" >
  <line x1="100" y1="10" x2="200" y2="200" stroke=rgb(0,230,255) stroke-
width="3"></line>
  <line x1="300" y1="50" x2="40" y2="220" stroke=red stroke-width="3"/>
</svg>
```

2. SVG Circle

```
<svg height="100" width="100" style="border:1px solid black">
  <circle cx="50" cy="50" r="40" stroke="black" stroke-width="3" fill="red" />
</svg>
```

3. SVG Rectangle - <rect>

```
<svg width="250" height="180" style="border:3px solid purple">
  <rect x="50" y="20" rx="20" ry="20" width="150" height="150" stroke="black" stroke-
width="3" fill="red"/>
</svg>
```



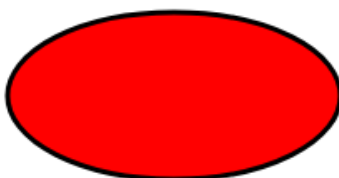
- The rx and the ry attributes rounds the corners of the rectangle

4. SVG Ellipse - <ellipse>

The <ellipse> element is used to create an ellipse.

An ellipse is closely related to a circle. The difference is that an ellipse has an x and a y radius that differs from each other, while a circle has equal x and y radius:

```
<svg height="140" width="500">
  <ellipse cx="200" cy="80" rx="100" ry="50" stroke="black" stroke-width="3" fill="red"/>
</svg>
```



Logo using ellipse

```
<svg height="140" width="500">
  <ellipse cx="200" cy="80" rx="100" ry="50" stroke="black" stroke-width="3"
  fill="pink"/>
  <text fill="#000" font-size="50" x="155" y="100" font-family="verdana">LJU</text>
</svg>
```

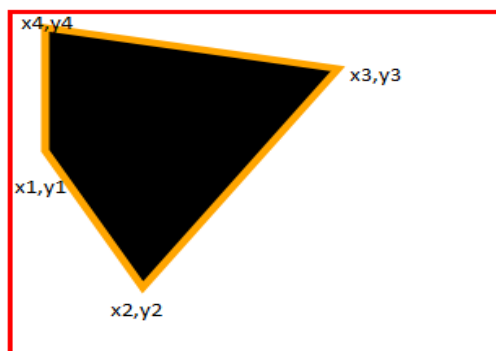


5. SVG Polygon - <polygon>

The <polygon> element is used to create a graphic that contains at least three sides.

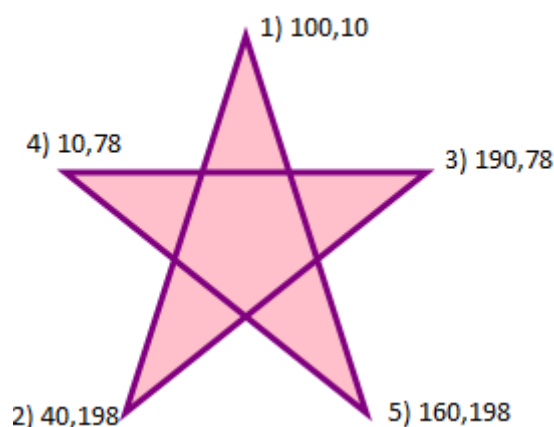
Polygons are made of straight lines, and the shape is "closed" (all the lines connect up).

```
<polygon points="x1,y1,x2,y2,x3,y3,x4,y4....xn,yn" stroke="color" stroke-width="5"
fill="black" />
<svg height="250" width="300" style="border: 3px solid red;">
  <polygon points="20,100,80,200,200,40,20,10" stroke="orange" stroke-width="5"
  fill="black" />
</svg>
```



Use the `<polygon>` element to create a **Star**:

```
<svg height="210" width="500">
  <polygon points="100,10,40,198,190,78 10,78,160,198"
    stroke="purple" stroke-width="3" fill="pink" fill-rule="nonzero" />
</svg>
```



- ✓ The **fill-rule** attribute is a presentation attribute defining the algorithm to use to determine the *inside* part of a shape.
- ✓ After counting the crossings paths, if the result is zero then the point is outside the path. Otherwise, it is inside. **Default value is "nonzero"**

Change the fill-rule property to **"evenodd"**:

```
<svg height="210" width="500">
  <polygon points="100,10,40,198,190,78 10,78,160,198"
    stroke="purple" stroke-width="3" fill="pink" fill-rule="evenodd" />
</svg>
```



❖ HTML Image Maps

<area> and <map>

- The <area> tag defines an area inside an image map (an image map is an image with clickable areas).
- <area> elements are always nested inside a <map> tag.

Example:

```
<html>
  <head> <title>Area & Map</title> </head>
  <body>
    <h1> Map & Area </h1>
    <!-- Image Map Generated by http://www.image-map.net/ -->
    

    <map name="image-map">
      <area target="_blank" alt="laptop" title="laptop" href="" coords="0,3,339,523"
shape="rect">

      <area target="_blank" alt="tea" title="tea" href="" coords="437,382,85"
shape="circle">

      <area target="_blank" alt="book" title="book" href="" coords="465,261,
578,191,668,234,775,393,617,496" shape="poly">

      <area target="_blank" alt="flower" title="flower" href="" coords="322,0,324,0
414,149,609,202,783,112,800,3" shape="poly">
    </map>
  </body>
```


</html>