Forms, Graphics, and Interactive Elements

ABSTRACT

Forms and input elements enable data collection through various input types and attributes. SVG graphics provide scalable shapes like lines, circles, rectangles, and polygons for visual design. Image maps using <map> and <area> add interactivity by linking specific image regions. Together, these elements enhance usability, graphics, and user interaction on the web.

HTML Forms <form>

- ✓ <form> is a HTML element to collect input data with containing interactive controls. It provides facilities to input text, number, values, email, password, and control fields such as checkboxes, radio buttons, submit buttons, etc
- ✓ Forms are generally used when you want to collect data from the user. For example, feedback form, login form, complaint form, inquiry form etc.

Form tags

Tag / Element	Description	Syntax / Example
<form>	Container used to collect and submit user input to a server.	<form action="/submit" method="post"> ... form elements ... </form>
<fieldset>	Groups related form elements together inside a box.	<fieldset> </fieldset>
<legend>	Caption/title for a <fieldset>.	<fieldset> <legend>Login Details</legend> </fieldset>
<input>	General input field (text, password, number, etc.).	<input type="text" > <input type="email" >
<textarea>	Multi-line text input field.	<textarea rows="20" cols="30">Enter your message</textarea>
<button>	Clickable button (can be submit, reset).	<button type="submit">Submit</button>
<label>	Defines a label for form elements; improves accessibility.	<label for="uname">Username:</label> <input type="text">
<datalist>	Provides a list of predefined suggestions for an <input>.	<input list="browsers" name="browser"> <datalist id="browsers"> <option value="Chrome"> <option value="Firefox"> </datalist>
<select>	Creates a drop-down list.	<select name="cars"> <option value="volvo">Volvo</option> <option value="bmw">BMW</option> </select>
<option>	Defines an item inside a <select> or <datalist>.	<select> <option value="html">HTML</option>

		<option value="css" selected>CSS</option> </select>
<optgroup>	Groups related <option> elements inside a <select>.	<select> <optgroup label="Frontend"> <option>HTML</option> <option>CSS</option> </optgroup> <optgroup label="Backend"> <option>PHP</option> <option>Python</option> </optgroup> </select>

Form Attributes

Attribute	Description	Default Value
action	URL where form data will be sent after submission.	"" (empty)
enctype	The enctype attribute specifies how the form-data should be encoded when submitting it to the server. **Note:** The enctype attribute can be used only if method="post".	application/x-www-form-urlencoded Form data is encoded as key-value pairs, joined with & and spaces converted to +.
method	HTTP method (GET or POST) used when submitting.	GET
type	Defines the type of <input> (text, password, etc.).	text
name	Name of the field (used as key when submitting data).	"" (empty)
id	Unique identifier for an element	"" (empty)
value	Default value for an input.	"" (empty)
placeholder	Hint text shown inside an input before typing.	"" (empty)
required	Makes the field mandatory before form submission.	false
readonly	Field is visible but not editable; value is submitted.	false
disabled	Field is visible but inactive; value is NOT submitted.	false
checked	Pre-selects a checkbox or radio button.	false
maxlength	Maximum number of characters allowed.	Unlimited
minlength	Minimum number of characters required.	0
step	Defines intervals for numeric/date inputs (e.g., step='2').	1 (depends on input type)
min	Minimum allowed value for number/date inputs.	No restriction
max	Maximum allowed value for number/date inputs.	No restriction
multiple	Allows selecting multiple values (e.g., multiple files or emails).	false
accept	Specifies file types accepted in a file upload input.	All file types

Attribute	Description	Default Value
autocomplete	Enables or disables autocomplete suggestions.	on
autofocus	Automatically focuses on an input field when the page loads.	false
rows	Defines visible rows in a <textarea>.	No fixed default (browser-dependent)
cols	Defines visible width (columns) in a <textarea>.	No fixed default (browser-dependent)
selected	Pre-selects an option in a <select> menu.	false
list	Connects an <input> with a <datalist> for suggestions.	None

<input> Types

Input Type	Description	Syntax / Example
text	Single-line text field.	<input type="text" name="username" id="uname" placeholder="Enter username" required maxlength="20">
password	Hides typed characters (for passwords).	<input type="password" name="pwd" id="pwd" placeholder="Enter password" required minlength="6">
email	Input for email addresses (validates format).	<input type="email" name="email" id="email" placeholder="name@example.com" required>
number	Numeric input with min/max/step controls.	<input type="number" name="age" id="age" min="1" max="100" step="1" value="18">
tel	Telephone number input (may show number pad on mobile).	<input type="tel" name="phone" id="phone" placeholder="1234567890">
checkbox	Allows selecting one or more options.	<input type="checkbox" name="hobby" value="music" checked> Music <input type="checkbox" name="hobby" value="sports"> Sports
radio	Allows selecting one option from a group.	<input type="radio" name="gender" value="male" checked> Male <input type="radio" name="gender" value="female"> Female
submit	Button to submit the form.	<input type="submit" value="Submit Form">
reset	Button to reset all form fields.	<input type="reset" value="Reset Form">
button	General button (custom actions via JavaScript).	<input type="button" value="Click Me">
file	Allows file selection/upload.	<input type="file" name="resume" id="resume" accept=".pdf,.doc" multiple>

hidden	Hidden input (not shown to user, but submitted).	\<input type="hidden" name="userid" value="12345">
date	Date picker (calendar).	\<input type="date" name="dob" id="dob" min="2000-01-01" max="2025-12-31" value="2020-05-15">
range	Slider control for selecting a numeric value.	\<input type="range" name="vol" id="vol" min="0" max="50" step="5" value="25">
color	Color picker input.	\<input type="color" name="favcolor" id="favcolor" value="#ff0000">

GET Method:
• Appends the form data to the URL, in name/value pairs
• NEVER use GET to send sensitive data! (the submitted form data is visible in the URL!)
• The length of a URL is limited (2048 characters)
• Useful for form submissions where a user wants to bookmark the result
• GET is good for non-secure data, like query strings in Google

POST Method:
• Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)
• POST has no size limitations, and can be used to send large amounts of data.
• Form submissions with POST cannot be bookmarked align: It aligns form in left or right or center.

❖ HTML \<select> tag
✓ \<select> tag is used to create a drop-down list with multiple options. The \<option> element is nested within \<select> tag for defining options in a list.
✓ The \<select> element is used to create a drop-down list. The \<select> element is most often used in a form, to collect user input.
✓ The \<optgroup> tag is used to group related options in a \<select> element (drop-down list). If you have a long list of options, groups of related options are easier to handle for a user.
✓ The \<option> tag defines an option in a select list.

Attributes
- **disabled:** Specifies that an option should be disabled
- **selected:** Specifies that an option should be pre-selected when the page loads.
- **hidden:** The option with the hidden attribute is not visible in the dropdown list, but it remains in the HTML code, and its value can still be accessed programmatically (e.g., through JavaScript).
- **Value:** Specifies the value to be sent to a server

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>
```
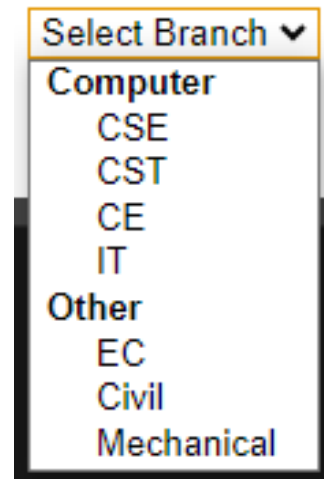
Selected attribute keeps the option selected at first time we load the web page.
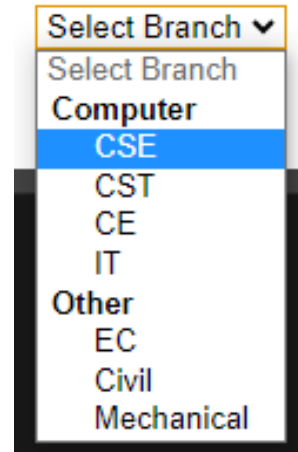
Disabled attribute disables the option. We cannot select the disabled option.

```
<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.

6

readonly,disabled and hidden

1. readonly

- **Behavior:** The field is visible and its value can be **submitted** with the form, but the user **cannot edit** it.
- **Use case:** Showing a value that should not be changed but still sent to the server.

2. disabled

- **Behavior:** The field is visible but **grayed out**, **cannot be edited**, and **is not submitted** with the form.
- **Use case:** Temporarily disabling a field based on a condition.

3. hidden

- **Behavior:** The field is **not visible at all** on the page, but its value **is submitted** with the form.
- **Use case:** Passing extra data to the server without showing it to the user (like user ID).

Example

```html
<h2>Form Example: readonly, disabled, hidden</h2>
<form action="/submit" method="POST">
  <!-- Readonly field -->
  <label>Username (readonly):</label>
  <input type="text" name="username" value="LJU" readonly>
  <br><br>
  <!-- Disabled field -->
  <label>Promo Code (disabled):</label>
  <input type="text" name="promo" value="SUMMER50" disabled>
  <br><br>
  <!-- Hidden field -->
  <input type="hidden" name="userId" value="12345">
  <!-- Editable field -->
  <label>Email:</label>
  <input type="email" name="email" placeholder="Enter your email" required>
  <br><br>
  <button type="submit">Submit</button>
</form>
```

Form Example: readonly, disabled, hidden

Username (readonly): LJU

Promo Code (disabled): SUMMER50

Email: Enter your email

Submit

Example

Write an HTML form that contains a date picker input. The date picker should only allow users to select dates between January 1, 2020 and December 31, 2022. Ensure that the date field is mandatory.

```html
<html> <body>

 <form action="#" method="get">
```

```html
<label for="date"> Choose a date (between 01-01-2020 and 31-12-2022) </label>
<input type="date" name="date" min="2020-01-01" max="2022-12-31" required />
<button type="submit">Submit</button>
</form>
</body></html>
```

Example

```html
<h2>Student Registration Form</h2>
<form action="submit.html" method="get">
  <!-- Personal Information -->
  <fieldset>
    <legend>Information</legend>

    <label for="uname">Username:</label>
    <input type="text" name="username" id="uname" autofocus autocomplete="on"
placeholder="Enter username" required maxlength="20"><br><br>

    <label for="pwd">Password:</label>
    <input type="password" name="pwd" id="pwd" placeholder="Enter password" required
minlength="6"><br><br>

    <label for="email">Email:</label>
    <input type="email" name="email" id="email" placeholder="name@example.com"
required><br><br>

    <label for="phone">Phone:</label>
    <input type="tel" name="phone" id="phone" placeholder="1234567890"><br><br>

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

    <p>Hobbies:</p>
    <input type="checkbox" name="hobby" value="music" checked> Music
    <input type="checkbox" name="hobby" value="sports"> Sports <br><br>
  </fieldset>

  <!-- Preferences -->
  <fieldset>
    <legend>Additional Information</legend>
```

```html
<label for="dob">Date of Birth:</label>
<input type="date" name="dob" id="dob" min="2025-09-07" max="2025-12-31"><br><br>

<label for="age">Age:</label>
<input type="number" name="age" id="age" min="1" max="100" step="1" value="18"><br><br>

<label for="favcolor">Favorite Color:</label>
<input type="color" name="favcolor" id="favcolor" value="#ff0000"><br><br>

<label for="browser">Preferred Browser:</label>
<input list="browsers" name="browser" id="browser">
<datalist id="browsers">
  <option value="Chrome">
  <option value="Firefox">
  <option value="Edge">
</datalist><br><br>

<label for="degree">Select Branch:</label>
<select name="degree_branch" id="degree">
  <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><br><br>
</fieldset>

<fieldset>
  <legend>Feedback</legend>
  <label for="msg">Your Message:</label><br>
  <textarea name="message" id="msg" rows="5" cols="30">Enter your
message</textarea><br><br>
  <label for="range">Satisfaction Level:</label>
  <input type="range" name="satisfaction" id="range" min="0" max="50" step="5"
value="25"><br><br>
```

```
    <!-- Hidden value -->
    <input type="hidden" name="userid" value="12345">
</fieldset>


<!-- Form Buttons -->
<fieldset>
    <legend>Submit Section</legend>
    <input type="submit" value="Submit Form">
    <input type="reset" value="Reset Form">
    <input type="button" value="Click Me">
    <button type="submit">Submit (Button Tag)</button>
</fieldset>
</form>
```

Student Registration Form

Information
Username: dfg
Password: ••••••••
Email: fh@dfg.gfjh
Phone: 1234567890

Gender:
◉ Male ○ Female

Hobbies:
☑ Music ☐ Sports

Additional Information
Date of Birth: 09/16/2025
Age: 18
Favorite Color: ▮
Preferred Browser: Firefox
Select Branch: Select Branch ⌄

Feedback
Your Message:
dgfsdgsdfg

Satisfaction Level: ●━━━

Submit Section
Submit Form   Reset Form   Click Me   Submit (Button Tag)

autofocus and autocomplete='on' added as attributes

Student Registration Form

Information
Username: [Enter username]

Password: [Enter p]

Email: [name@exa]

Phone: [123456789]

Gender:

● Male ○ Fem

Saved info   ×

dfg [Last used]
fh@dfg.gfjh · 1234567890 · Firefox

dfs
fh@dfg.gfjh · 1234567890 · Firefox

dfg

ghfmj

Example

Design an HTML form that allows users to **upload one or more files** to a server. The form should:

1. Include a file input field where the user can browse and select files.
2. Restrict the file types to only **PDF** (.pdf) and **Word Document** (.doc) formats.
3. Allow users to select **multiple files at once**.
4. Group the file input inside a **fieldset** for better structure.
5. Provide two buttons:
   - **Submit** → to send the selected file(s) to the server.
   - **Reset** → to clear the form and allow new input.
6. Use the **POST** method with enctype="multipart/form-data" so the files are correctly transferred in the request body.

```
<h1>File Upload</h1>
   <form action="submit.html" method="post" enctype="multipart/form-data">
   <!—Upload File -->
   <fieldset>
     <label for="resume">Upload Resume:</label>
     <input type="file" name="resume" id="resume" accept=".pdf,.doc" multiple><br><br>
   </fieldset>
   <br>
   <input type="submit"/>
   <input type="reset"/>
</form>
```

File Upload

Upload Resume: [Choose Files] No file chosen

[Submit] [Reset]

- The **accept attribute** is a **hint to the browser** about what file types should be selectable.
- <input type="file" accept=".pdf,.doc"> This tells the browser's file picker: *only show PDF and Word files as selectable*.
- The file dialog (file chooser window) will **filter** and only show .pdf and .doc files by default.
- But in many browsers, the user can still change the filter (e.g., "All Files .") and pick any file type (like .png, .exe).
- So accept is **not strict validation**. It's a UI convenience.

SVG - Scalable Vector Graphics

- SVG stands for Scalable Vector Graphics
- SVG is used to define vector-based graphics for the Web
- SVG defines the graphics in XML format
- Every element and every attribute in SVG files can be animated

SVG Advantages

Advantages of using SVG over other image formats (like JPEG and GIF) are:

- SVG images can be created and edited with any text editor
- SVG images can be searched, indexed, scripted, and compressed
- SVG images are scalable
- SVG images can be printed with high quality at any resolution
- SVG images are zoomable
- SVG graphics do NOT lose any quality if they are zoomed or resized
- SVG is an open standard
- SVG files are pure XML

1. SVG Line - <line>

```
<svg height="250" width="500" >
    <line x1="100" y1="10" x2="200" y2="200" stroke="blue" stroke-width="3"></line>
    <line x1="300" y1="50" x2="40" y2="220" stroke="red" stroke-width="3"/>
  </svg>
```
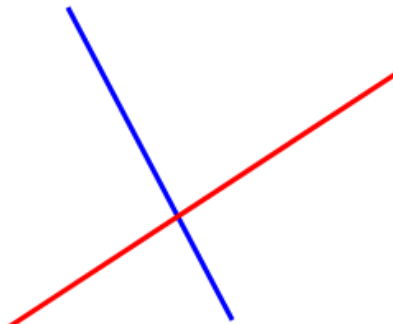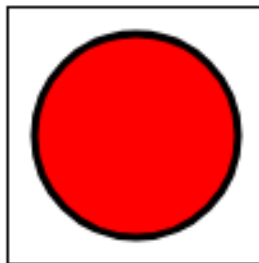

Code explanation:

- The x1 attribute defines the start of the line on the x-axis
- The y1 attribute defines the start of the line on the y-axis
- The x2 attribute defines the end of the line on the x-axis
- The y2 attribute defines the end of the line on the y-axis
- The stroke-width attribute defines the width of the border of the line
- The stroke attribute defines the color of the border of the line.

13

The stroke attribute must be set to a color (e.g., black, red, #000000) for the shape's outline to appear.The **default stroke** is not applied automatically, and an unspecified stroke means no visible stroke at all

Default value is not white. But we need to define a stroke attribute for the shape's outline to appear.

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>
```


Code explanation:

- The cx and cy attributes define the x and y coordinates of the center of the circle. If cx and cy are omitted, the circle's center is set to (0,0)
- The r attribute defines the radius of the circle.
- The fill attribute defines the fill color of the circle.
- The stroke-width attribute defines the width of the border of the circle.
- The stroke attribute defines the color of the border of the circle.

3. SVG Rectangle - <rect>

The <rect> element is used to create a rectangle and variations of a rectangle shape:
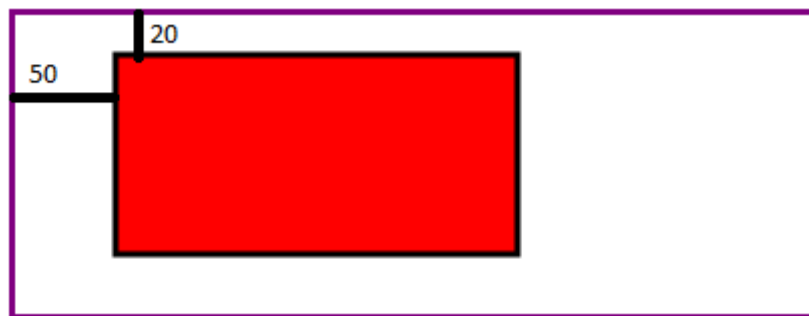
```
<svg width="400" height="120" style="border:3px solid purple">
  <rect width="300" height="100" stroke="black" stroke-width="3" fill="red"   />
 </svg>
```

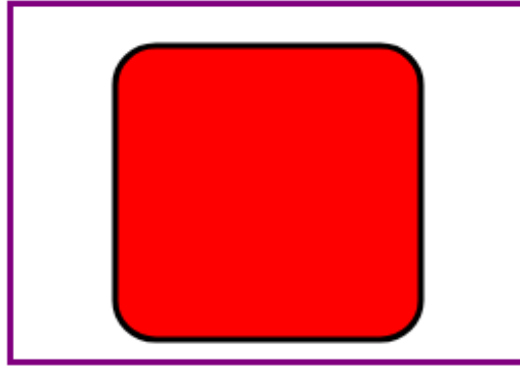In above example by default x and y coordinates are (0, 0)

- The width and height attributes of the <rect> element define the height and the width of the rectangle
- The fill attribute defines the fill color of the rectangle.
- The stroke-width attribute defines the width of the border of the rectangle.
- The stroke attribute defines the color of the border of the rectangle.

```
<svg width="400" height="150" style="border:3px solid purple">
  <rect x="50" y="20" width="200" height="100" stroke="black" stroke-width="3" fill="red"  />
</svg>
```


- The x attribute defines the left position of the rectangle (e.g. x="50" places the rectangle 50 px from the left margin)
- The y attribute defines the top position of the rectangle (e.g. y="20" places the rectangle 20 px from the top margin)

```
<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>
```
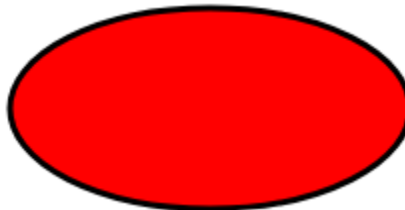
- The rx and the ry attributes rounds the corners of the rectangle

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>
```


Code explanation:

- The cx attribute defines the x coordinate of the center of the ellipse
- The cy attribute defines the y coordinate of the center of the ellipse
- The rx attribute defines the horizontal radius
- The ry attribute defines the vertical radius.
- The fill attribute defines the fill color of the ellipse.
- The stroke-width attribute defines the width of the border of the ellipse.
- The stroke attribute defines the color of the border of the ellipse.

5. Logo

```
<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>
```
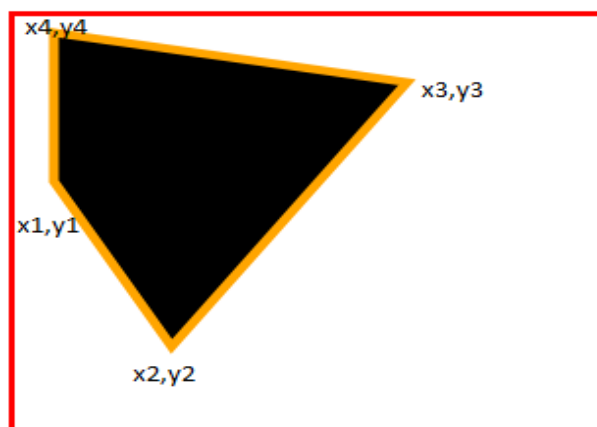

6. SVG Polygon - <polygon>

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>
```

Code explanation:

- The points attribute defines the x and y coordinates for each corner of the polygon
- The fill attribute defines the fill color of the polygon.
- The stroke-width attribute defines the width of the border of the polygon.
- The stroke attribute defines the color of the border of the polygon.

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>
```
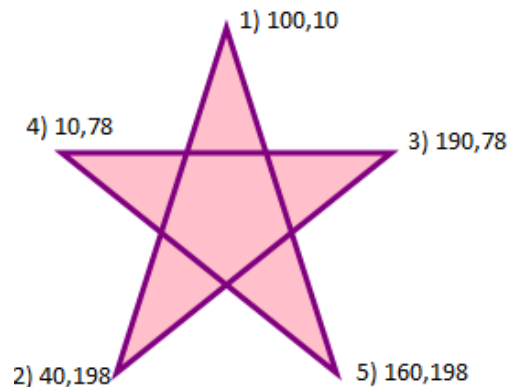

✓ The **fill-rule** attribute is a presentation attribute defining the algorithm to use to determine the *inside* part of a shape.

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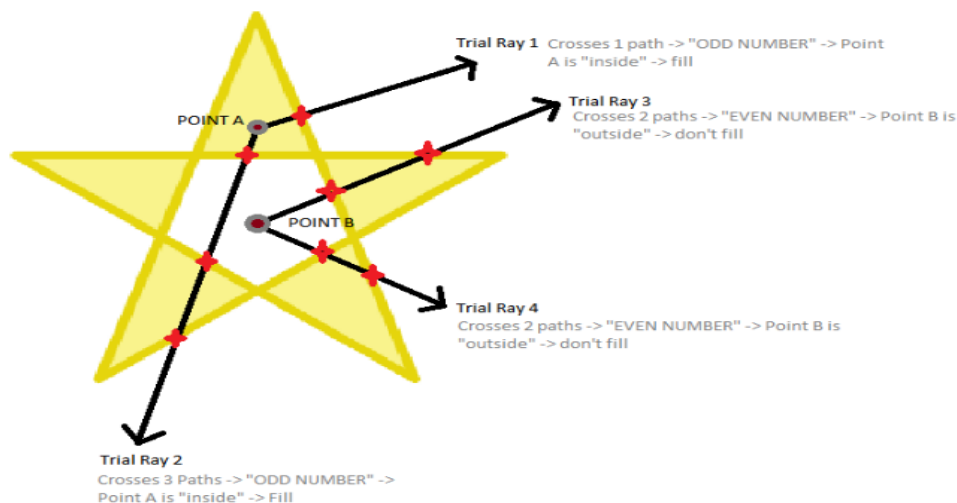
Change the fill-rule property to "evenodd":

```
<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>
```

This is how the mechanism works. Pick a point in the center, draw lines to infinity in any direction - they always cross an even number of path segments -which means that they are "outside" and their areas don't get filled.

Pick a point in the filled triangles - if you draw lines to infinity in any direction - they always cross an odd number of path segments and thus, they are "inside" and their area should be filled.

Trial Ray 1 Crosses 1 path -> "ODD NUMBER" -> Point A is "inside" -> fill

Trial Ray 3
Crosses 2 paths -> "EVEN NUMBER" -> Point B is "outside" -> don't fill

Trial Ray 4
Crosses 2 paths -> "EVEN NUMBER" -> Point B is "outside" -> don't fill

Trial Ray 2
Crosses 3 Paths -> "ODD NUMBER" -> Point A is "inside" -> Fill
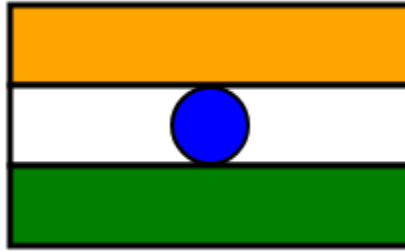
POINT A

POINT B

Note:
- ✓ If the **fill** attribute is not specified, then **default value is black**.
- ✓ **stroke** attribute is necessary to define in **line tag**, otherwise **line** will not be visible.
- ✓ In other tags like **rect, circle, polygon, ellipse** etc it will fill the shape with **default color black/mentioned color** without stroke (border of the shape).

SVG shapes like <circle>, <rect>, <line> etc do not render a stroke unless a stroke color is explicitly provided. **There is no default stroke automatically applied unless specified**.

Without the stroke attribute, the circle won't render any visible outline because no stroke color is applied. To make the circle visible, you need to explicitly define the stroke color.
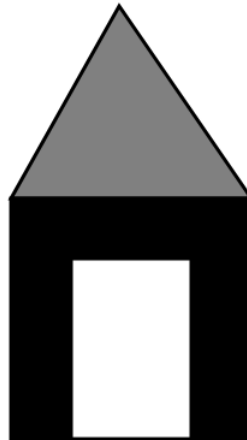
Default value is not white. But we need to define a stroke attribute for the shape's outline to appear.

Example 1


```
<svg height="250" width="500">
    <rect x='20' y='20' width="200" height="40" fill="orange" stroke="#000" stroke-width="3"/>
    <rect x='20' y='60' width="200" height="40" fill="#fff" stroke="#000" stroke-width="3"/>
    <rect x='20' y='100' width="200" height="40" fill="green" stroke="#000" stroke-width="3"/>
    <circle cx="120" cy="80" r="19" stroke="#000" stroke-width="2" fill="blue"/>
  </svg>
```
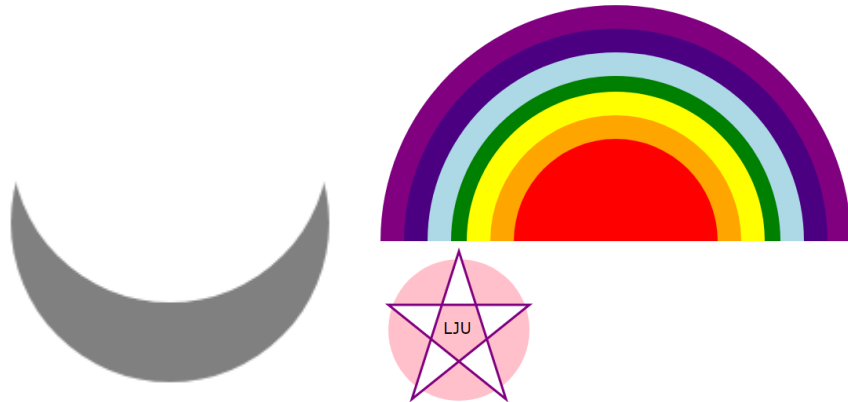
Example 2


```
<svg height="450" width="500">
    <polygon points="20,180 110,20 220,180" fill="grey" stroke="black" stroke-width="3" />
    <rect x='20' y='180' width="200" height="200" fill="black" stroke="#000" stroke-width="3"/>
    <rect x='70' y='230' width="100" height="150" fill="white" stroke="#000" stroke-width="3"/>
  </svg>
```

Example 3


```
<!—moon -- >
<svg width="200" height="200">
   <circle cx="120" cy="100" r="80" fill="grey"/>
   <circle cx="120" cy="60" r="80" fill="white"/>
</svg>


<!—rainbow half circle -- >
  <svg width="600" height="500">
   <circle cx="300" cy="500" r="300" fill="purple"/>
   <circle cx="300" cy="500" r="270" fill="indigo"/>
   <circle cx="300" cy="500" r="240" fill="lightblue"/>
   <circle cx="300" cy="500" r="210" fill="green"/>
   <circle cx="300" cy="500" r="190" fill="yellow"/>
   <circle cx="300" cy="500" r="160" fill="orange"/>
   <circle cx="300" cy="500" r="130" fill="red"/>
  </svg>


<!—star circle logo -- >
  <svg height="400" width="500">
   <circle cx="100" cy="110" r="90" fill="pink"></circle>
   <polygon points="100,10,40,198,190,78 10,78,160,198"
   stroke="purple" stroke-width="3" fill="white" fill-rule="evenodd" />
   <text fill="#000" font-size="20" x="80" y="115" font-family="verdana">LJU</text>
  </svg>
```
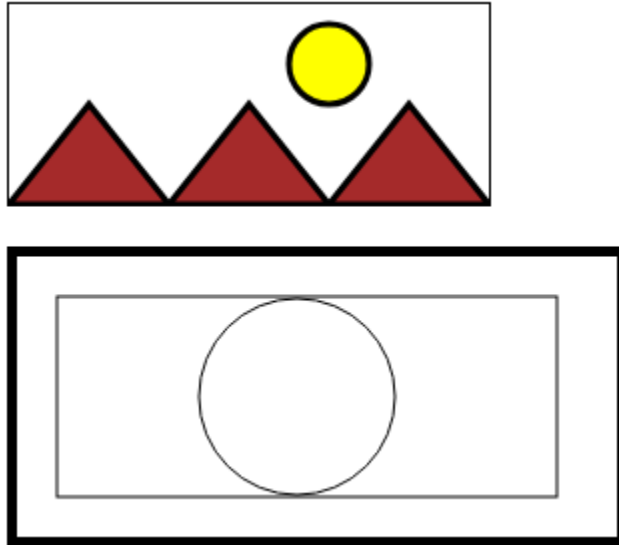
Example 4


```
<svg height="100" width="240" style="border: 1px solid black;">
   <polygon points="0,100,40,50,80,100,80,100,120,50,160,100,160,100,200,50,240,100" fill="brown"
stroke="black" stroke-width="3"/>
   <circle cx="160" cy="30" r="20" fill="yellow" stroke="black" stroke-width="3"/>
</svg>
<br>

<svg height="140" width="300" style="border: 5px solid black;">
   <rect x="20" y="20" width="250" height="100" fill="white" stroke="black" stroke-width=""/>
   <circle cx="140" cy="70" r="49" fill="#fff" stroke="#000"/>
</svg>
```

HTML Image Map

<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.

Note: The usemap attribute in <img> is associated with the <map> element's name attribute, and creates a relationship between the image and the map.

Attributes:
- **shape**: The shape to be mapped on the image, can be a "rect", a "circle" or a "poly".
- **coords**: The coordinates of the shape.
- **href**: The href is the link where the user will be directed to after clicking the mapped portion of the image.
- **alt**: Alternative text for a clickable area in an image map.
- **target**: Context in which to open the link resource.

x1,y1,x2,y2	Specifies the coordinates of the **top-left** and **bottom-right** corner of the rectangle **(shape="rect")**
x,y,radius	Specifies the coordinates of the circle center and the radius **(shape="circle")**
x1,y1,x2,y2,..,xn,yn	Specifies the coordinates of the edges of the polygon. If the first and last coordinate pairs are not the same, the browser will add the last coordinate pair to close the polygon **(shape="poly")**

Example:

```
<html>
   <head> <title>Area & Map</title> </head>
   <body>
      <h1> Map & Area </h1>
 <!-- Image Map Generated by http://www.image-map.net/ -->
<img src="test.jpg" usemap="#image-map">
```

```
<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>
```


To find coordinates reference website: **https://www.image-map.net/**