HTML

In this lecture, we developed the structure of a website with the help of HTML tags that we studied in the previous lectures. After creating the website, we thoroughly understand how we build the structure of any website (with the help of HTML) before applying any styles to it.

All the tags we have learned so far are sufficient to build any website structure. But if you want to dig deeper into HTML, refer to the concepts discussed below.

MAP Tag

The <map> tag in HTML is used to create client-side image maps, allowing specific regions of an image to be clickable. This tag must have a name attribute that serves as a unique identifier for the map. It acts as a container for the clickable areas defined by the <area> tag.

Example:

```
<map name="mapName"></map>
```

Area Tag

The <area> tag is used within the <map> tag to define clickable areas on an image. It lets you specify each clickable area's shape, coordinates, and link.

Example:

Syntax for a Rectangle:

```
<area shape="rect" coords="x1,y1,x2,y2" alt="Description" href="link.html">
```

Syntax for a Circle:

```
<area shape="circle" coords="x,y,radius" alt="Description" href="link.html">
```

Syntax for a Polygon:

```
<area shape="poly" coords="x1,y1,x2,y2,x3,y3,..." alt="Description" href="link.html">
```

1. **shape:** Specifies the shape of the clickable area (rect for rectangle, circle for circle, poly for polygon).
2. **coords:** Defines the coordinates of the clickable area. The format varies based on the shape.
3. **alt:** Provides alternative text for accessibility purposes.
4. **href:** Specifies the URL to which the user will be directed when clicking the area.

Example of Image Map:

Let's consider an example using an image called "example.jpg" with three clickable areas:

```
<img src="example.jpg" usemap="#exampleMap" alt="Example Image Map">
<map name="exampleMap">
    <area shape="rect" coords="34,44,270,350" alt="Rectangle" href="rectangle.html">
  <area shape="circle" coords="290,172,120" alt="Circle" href="circle.html">
    <area shape="poly" coords="400,300,500,200,600,300" alt="Polygon" href="polygon.html">
</map>
```

In this example:

- The <img> tag displays the image "example.jpg" and references the map named "exampleMap."
- The <map> tag contains three <area> tags, each defining a clickable area with different shapes, coordinates, and associated links.

Canvas:

The <canvas> tag draws graphics, animations, and other visual elements on a web page. It provides a drawing space where you can use JavaScript to create and manipulate graphics dynamically. The <canvas> element has width and height attributes, specifying the dimensions of the drawing area.

Basic Syntax:

```
<canvas id="myCanvas" width="600" height="400" style="border:1px solid grey"></canvas>
```

<canvas>: The main tag for creating a drawing area.
id: A unique identifier for the canvas, useful for JavaScript manipulation.
width and height: Dimensions of the canvas in pixels.

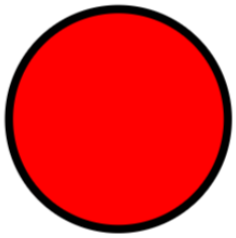
SVG (Scalable Vector Graphics):

SVG is an XML-based markup language for describing vector graphics. It is used to create graphics that can be scaled and styled without losing quality. SVG is well-suited for icons, illustrations, and other scalable visual elements.

Basic SVG Structure:

```
<svg width="100" height="100" xmlns="http://www.w3.org/2000/svg">
  <!-- SVG content goes here -->
  <circle cx="50" cy="50" r="40" stroke="black" stroke-width="3" fill="red" />
</svg>
```

Output:



<svg>: The root element for SVG graphics.
width and height: Specify the dimensions of the SVG viewport.
xmlns: Defines the XML namespace for SVG.

SVG Elements:

1. **Shapes (e.g., <circle>, <rect>, <line>, <path>):** Define various graphical shapes.
2. **Attributes (e.g., cx, cy, r for a circle):** Control the properties of SVG elements.
3. **Styling (e.g., stroke, fill, stroke-width):** Define colours, strokes, and fills.

Example with SVG:

```
<svg width="200" height="150" xmlns="http://www.w3.org/2000/svg">
  <rect width="100%" height="100%" fill="lightblue" />
  <circle cx="100" cy="75" r="50" fill="orange" />
    <text x="50%" y="50%" text-anchor="middle" dy=".3em" fill="white">SVG Example</text>
</svg>
```

Output:

Conclusion:

In summary, these notes delve into advanced HTML concepts that extend web development capabilities beyond basic structure and styling. The <map> and <area> tags facilitate the creation of client-side image maps, offering clickable regions on images. The <canvas> tag provides a dynamic space for drawing graphics and animations using JavaScript. Lastly, SVG (Scalable Vector Graphics) introduces an XML-based markup language for scalable and stylised vector graphics, allowing for the creation of intricate visual elements.

Developers can create more interactive and visually appealing web content by mastering these advanced HTML features. Image maps enable the definition of clickable areas on images, while the <canvas> tag opens the door to dynamic graphics and animations. SVG provides a versatile approach to scalable vector graphics, suitable for icons, illustrations, and other visual elements. Incorporating these concepts into web development enhances the overall user experience and allows for more creative and sophisticated designs.

References:

1. All HTML tags and Attributes: https://developer.mozilla.org/en-US/docs/Web/HTML/Reference
2. Map: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/map
3. Area: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/area
4. Canvas: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/canvas