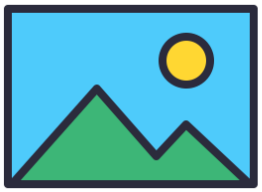
HTML Media

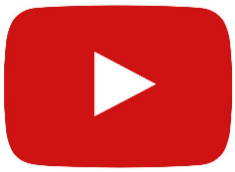
Topics Covered

- Introduction to HTML Media
- Image
- Audio
- Video
- iFrame
- SVG (Scalable Vector Graphics)

Introduction to HTML Media

In this topic, we will study how to insert an image, audio, or video in an HTML document. We will also learn how to embed another document (e.g. YouTube video, Google Maps, another website) into our web page. At last, we will explore more media options like Canvas and SVG.





Image

Image tag <img/>

If we want to include an image on our web page, we will use the image tag **<img/>.**

Syntax

```
Unset
<img src="image.jpg" alt="img tag syntax"/>
```

Here, the **src** attribute can be a path to an image or URL of an image, and the **alt** attribute is used to provide alternative text for an image

Example: Include a random image

```
Unset
<body>
    <h1>Heading </h1>
    <p>This is a random photo</p>
    <img src="https://picsum.photos/200" alt="image" />
</body>
```

Browser Output:

Suppose, we have given the wrong path to the src attribute, the image will not be displayed we call it a broken image.

Example: Broken image with alt attribute

```
Unset
<body>
    <h1>Heading</h1>
    <p>This is a random photo</p>
```

```
    <img src="broken-image.jpg" alt="image alt text" />
</body>
```

Browser Output

Yes!! It will show the **alt** text that we provided in the **<img>** tag. It is important to include **alt** text for images because it provides information to users who may be unable to see the image due to a slow internet connection, or a broken image link.

Image tag attributes

1. **src**: The src attribute is an HTML attribute used to specify the URL of an image file to be displayed on a web page.

2. **alt**: The alt attribute is an HTML attribute used to provide alternative text for an image in case the image cannot be displayed.

3. **width and height:** These attributes specify the width and height of the image in pixels.

```
Unset
<img src="image.jpg" width="200" height="150">
```

The above image has 200px width and 150px height.

Note: Initially, attributes like width & height were preferred to use, but later when CSS came, styles were separated from HTML for better code clarity and separation of concerns. We will study more about CSS in later modules.

4. **loading:** The loading attribute is an HTML attribute used to control the loading behavior of an image on a web page. The attribute has two possible values:
    - **lazy** which defers the loading, and loads only when the image is needed.
    - **eager** which loads the image immediately. This is the default behavior.

Supported Image Formats

The **<img> tag** in HTML can display various types of image files. The most commonly supported image formats are:

- .jpeg
- .png
- .svg
- .gif
- .webp

Picture tag <picture>

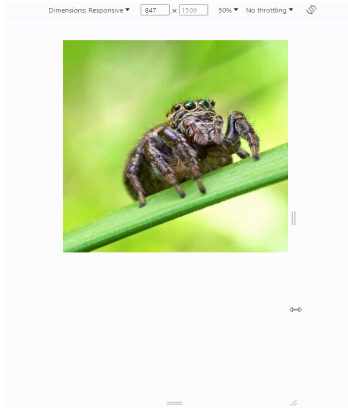
The <picture> element holds one <img> element and zero or more <source> elements. The browser checks every <source> and selects the most compatible one, if it does not find a suitable match, it will select the URL in the src attribute in <img> tag and display it.

Browse selects one source on the basis of srcset, media, and type attribute of the <source> tags. srcset defines image URL, media defines media query(will study in later modules), and type defines media format(like mp3, mp4, etc.) to understand it better let's see one example,

Example

```
Unset
<picture>
      <source
srcset="https://cdn.pixabay.com/photo/2023/07/30/11/30/spider-8158656_12
80.jpg"
        media="(min-width: 700px)"
      />
      <source
srcset="https://cdn.pixabay.com/photo/2023/07/04/10/30/dragonfly-8105988
_1280.jpg"
        media="(min-width: 450px)"
      />
      <img
src="https://cdn.pixabay.com/photo/2023/08/06/07/19/field-8172323_1280.j
pg"
        alt="photos"
      />
</picture>
```

Browser Output



You can see in the above example, the browser displays different images at different screen sizes, because it selects the most compatible one.

Audio

Audio tag <audio/>

The Audio tag includes audio on our web page. **<audio/>** tag is introduced in HTML5.

Syntax

```
Unset
<audio src="audio.mp3" controls/>
```

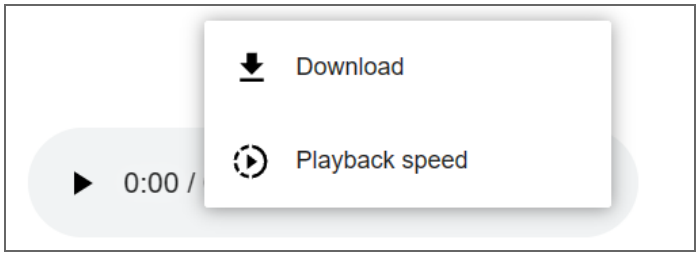
Here, the **src** attribute can be a URL of an audio file, and the **controls** attribute enables play, pause, and volume controls.

Example: Embed an audio

```
Unset
<body>
  <audio
        controls
src="https://interactive-examples.mdn.mozilla.net/media/cc0
-audio/t-rex-roar.mp3">
  </audio>
</body>
```

Browser Output:

If you click on 3 dots, it will show options to download and set playback speed.



Audio tag Attributes

1. **src:** This attribute specifies the source URL (Uniform Resource Locator) of the audio file that should be played.

2. **controls:** When this attribute is present, it displays the default audio controls, such as play, pause, and volume, allowing users to interact with the audio player.

3. **autoplay:** When this attribute is present, the audio will start playing automatically when the page loads.

4. **loop:** The audio will loop and play repeatedly when this attribute is present.

5. **preload:** This attribute specifies how the audio file should be preloaded for a better user experience. It can have values like,
    - **auto** - loads the audio file automatically, even if the user does not require it immediately.
    - **metadata** - loads only the metadata of the audio file (e.g. length of the audio file)
    - **none** - does not preload the audio file

6. **controlslist:** This attribute specifies the controls that should be displayed in the audio player's controls. It can have values like
    - **nodownload -** disable download button
    - **nofullscreen** - disable fullscreen button
    - **nodownload nofullscreen** - disable both download and fullscreen buttons.

```
Unset
<audio src="audio.mp3" controls
controlsList="nodownload"></audio>
```

7. **muted:** When this attribute is present, the audio will be muted by default.

```
Unset
<audio src="audio.mp3" muted></audio>
```

Audio with multiple sources
Not all browsers or devices support the same file format, and using multiple **<source>** tags allows the browser to choose the best-suited file format that it can play.

To specify multiple sources, we use **<source>** tags within **<audio>** tag. Each source tag has a **type** attribute, which identifies the format of the audio file.

Example:

```
Unset
<audio controls>
 <source src="audio.opus" type="audio/ogg; codecs=opus" />
 <source src="audio.ogg" type="audio/ogg; codecs=vorbis" />
```

```
 <source src="audio.mp3" type="audio/mpeg" />
</audio>
```

The browser will try to play the first file format specified in the **<source> tag**, and if it cannot play it, it will try the next one until it finds a compatible file format.

Video
Video tag <video/>
The **<video>** Tag embeds a video player into our webpage. **<video>** tag is introduced in HTML5.

Syntax

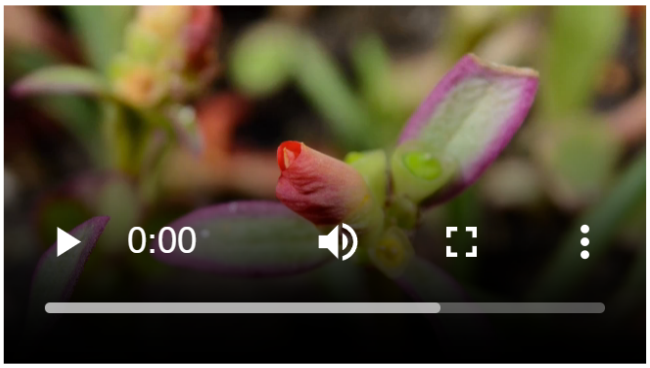
```
Unset
<video src="video.mp4" controls"></video>
```

Here, the **src** attribute can be a URL of a video file, and the **controls** attribute enables play, pause, and volume controls.

Example: Embed a video

```
Unset
<body>
    <video
src="https://interactive-examples.mdn.mozilla.net/media/cc0
-videos/flower.webm" controls width="250">
    </video>
</body>
```

Browser Output:-



Video Tag attributes

1. **src:** This attribute specifies the source URL (Uniform Resource Locator) of the video file that should be played.

2. **controls:** When this attribute is present, it displays the default video controls, such as play, pause, and seek, allowing users to interact with the video player.

3. **autoplay:** When this attribute is present, the video will start playing automatically when the page loads.

4. **loop:** The video will loop and play repeatedly when this attribute is present. For example;

5. **preload:** This attribute specifies how the video file should be preloaded. It can have values like,
   - **auto** - loads the video file automatically, even if the user does not require it immediately.
   - **metadata** - loads only the metadata (e.g. length) of the video file
   - **none** - does not preload the video file.

6. **width and height:** These attributes specify the width and height of the video player in pixels.

```
Unset
<video src="video.mp4" width="640" height="360"></video>
```

Note: Initially, attributes like width & height were preferred to use, but later when CSS came, styles were separated from HTML for better code clarity and separation of concerns. We will study more about CSS in later modules.

7. poster: This attribute specifies an image URL that should be displayed as a poster frame before the video starts playing.

```
Unset
<video src="video.mp4" poster="movie-poster.jpg"
width="640" height="360" controls></video>
```

8. controlsList: This attribute specifies the controls that should be displayed in the video player's controls. It can have values like,

- **nodownload** - disables download button.
- **nofullscreen** - disables the fullscreen button.
- **nodownload nofullscreen** - disables both download and fullscreen buttons.

9. muted: When this attribute is present, the video will be muted by default.

Video with multiple sources
Just like the **<audio> tag**, the **<video> tag** can use the **<source> tag** to provide alternative file formats to ensure that the video can be played on a wide range of devices and web browsers.

Example

```
Unset
<video width="620" controls>
     <source src="video.ogv" type="video/ogg" />
     <source src="video.avi" type="video/avi" />
     <source src="video.mp4" type="video/mp4" />
     Sorry, your browser doesn't support embedded videos.
</video>
```

iFrame
iFrame tag <iframe/>
The **<iframe>** tag is an HTML element used to embed another HTML document within the current document.

Syntax

```
Unset
<iframe src="document_url"></iframe>
```

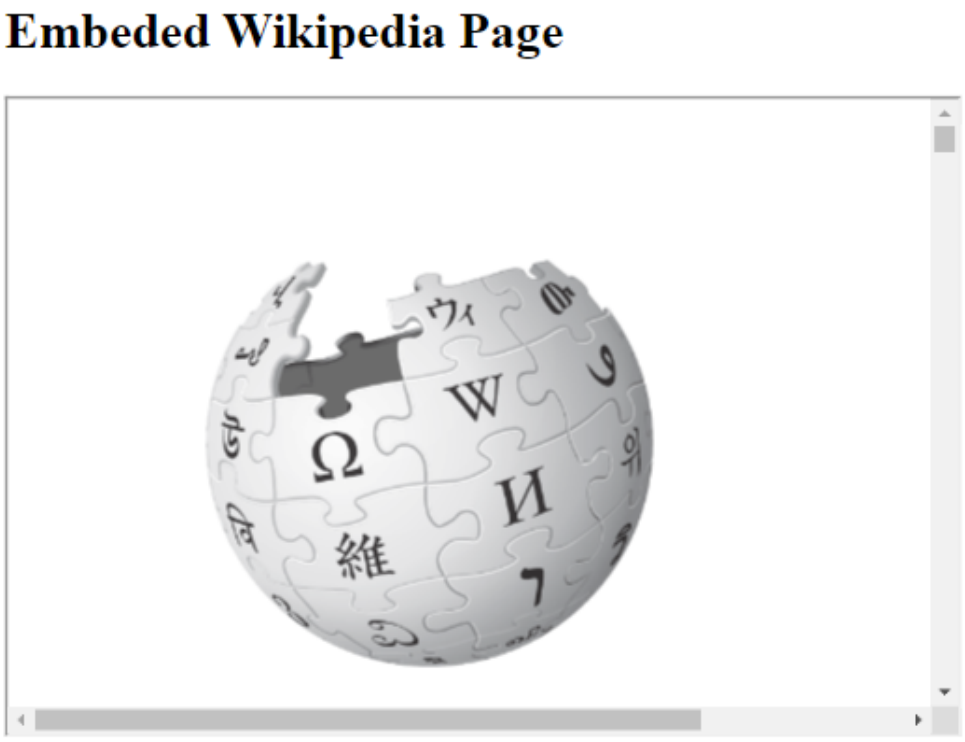
Here, the **src** attribute specifies the URL of the web page document that you want to display within the **<iframe>.**

Example: Embed Wikipedia page.

```
Unset
<body>
    <h1>Embeded Wikipedia Page</h1>
```

```
    <iframe
      src="https://www.wikipedia.org/"
      frameborder="1"
      width="600"
      height="400"
    ></iframe>
</body>
```

Browser Output,



iFrame tag attributes

1. **src:** This attribute specifies the URL of the web page that you want to display within the **<iframe>.**

2. **width and height:** These attributes specify the width and height of the **<iframe>**, respectively. They can be specified in pixels or as a percentage of the available width or height.

Note: Initially, attributes like width & height were preferred to use, but later when CSS came, styles were separated from HTML for better code clarity and separation of concerns. We will study more about CSS in later modules.

3. **sandbox:** This attribute allows you to define a sandbox environment for the content within the **<iframe>**. It can be used to restrict the capabilities of the embedded content, such as preventing it from running JavaScript or accessing certain features of the parent document.

4. **allow:** This attribute specifies what features are allowed to be used by the content within the **<iframe>**. It can be used to enable specific permissions, such as access to the microphone, camera, battery, web-share, allow fullscreen, allows payment, etc. (This whole list of permissions is available at (Link)

5. **loading:** The loading attribute is an HTML attribute used to control the loading behavior of an iframe on a web page. The attribute has two possible values:
    - **lazy** which defers the loading, and loads only when the iframe is needed.
    - **eager** which loads the iframe immediately. This is the default behavior..

6. **scrolling**: This attribute specifies whether or not to display scrollbars within the **<iframe/>**. It can have values of **yes**, **no, or auto**.
7. **name:** A targatable name for iframe, that can be used to place as value of target attribute in <a>, <form>. It is useful, when we want to refer any iframe on link clicks or form submits.

Example:

```
Unset
<iframe src="https://www.wikipedia.org/" name="wiki"></iframe>
<a href="https://www.w3schools.com" target="wiki">wikipedia.org</a>
```

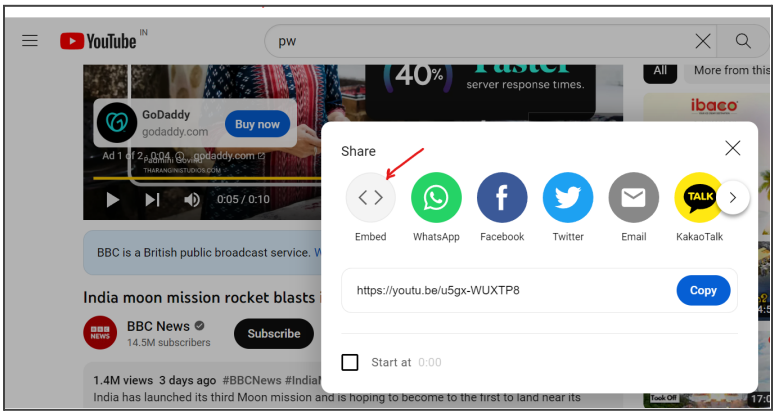
Browser Output

wikipedia.org

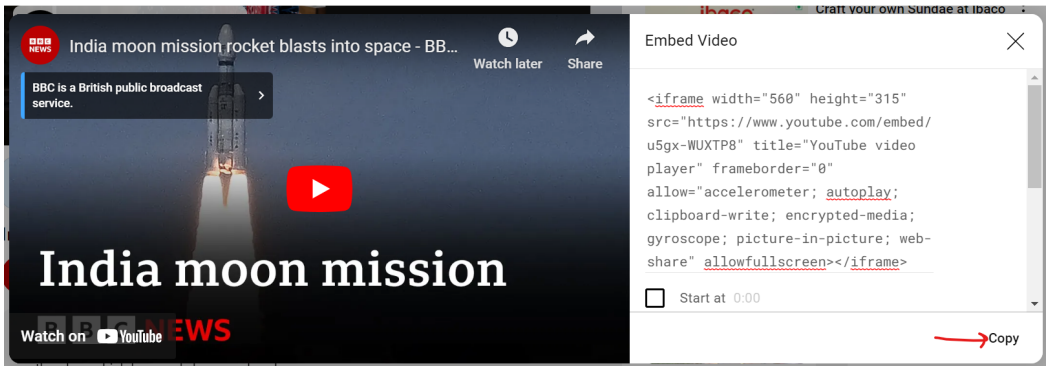
Some Usages of iframe

1. **Embed a YouTube video.**

We can embed a YouTube video using iframe. To do so, you open any YouTube video that you want to embed in your webpage. Then click the share button, and you will see an option to embed.

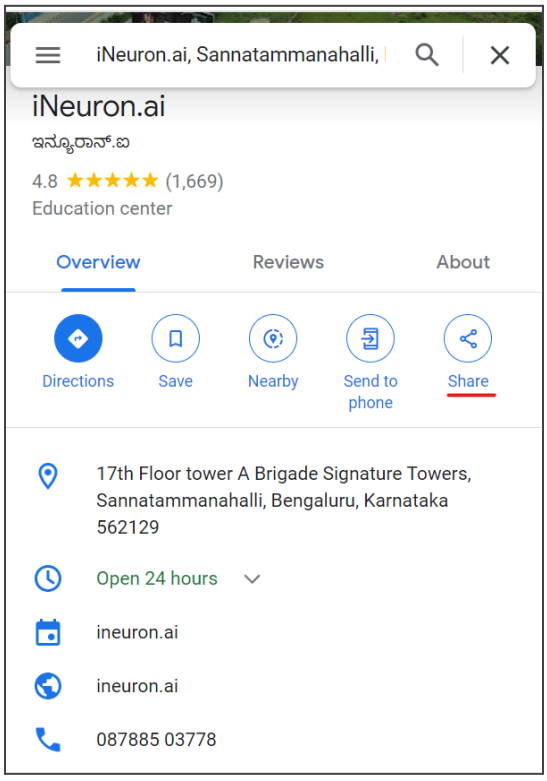


Click on embed, it will show you an iframe code, Copy that code and paste it into the HTML document where you want to embed it.

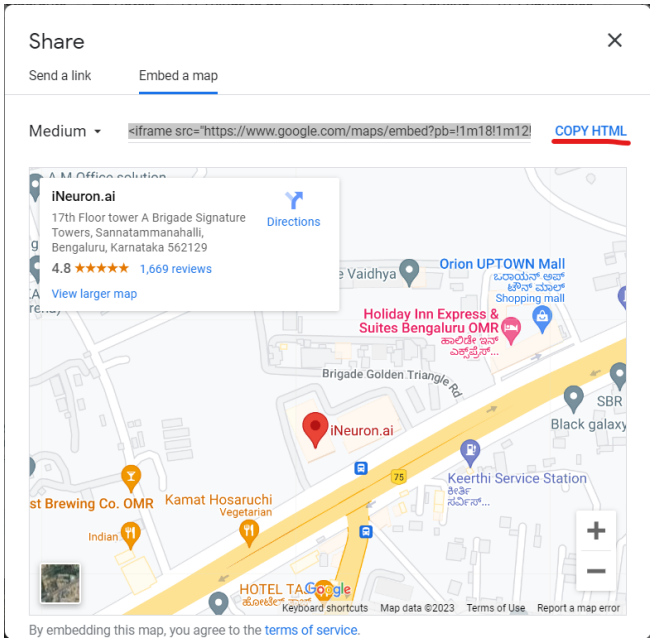


2. Embed Google Maps

We can embed Google Maps also. Go to Google Maps, and search for the place that you want to embed. You will have a share option, click on it.

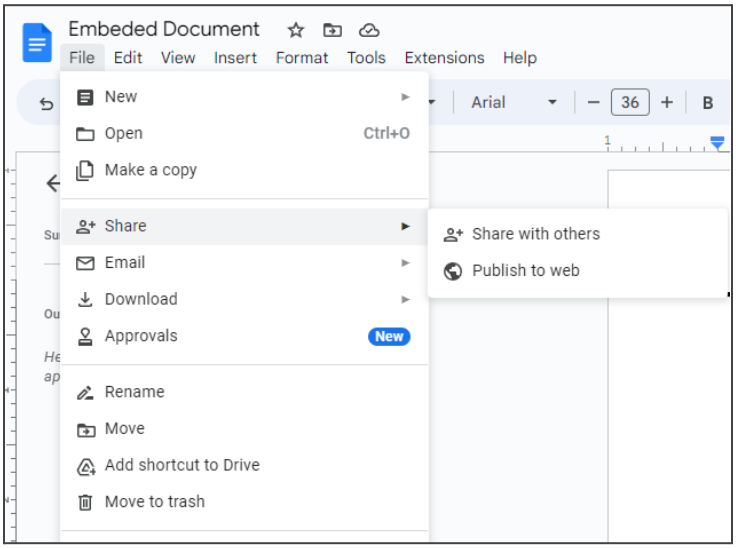


Once you click share it will show an option of **“Embed map”.** Open it, copy the iframe code, and paste it into your HTML document where you want to display the map.

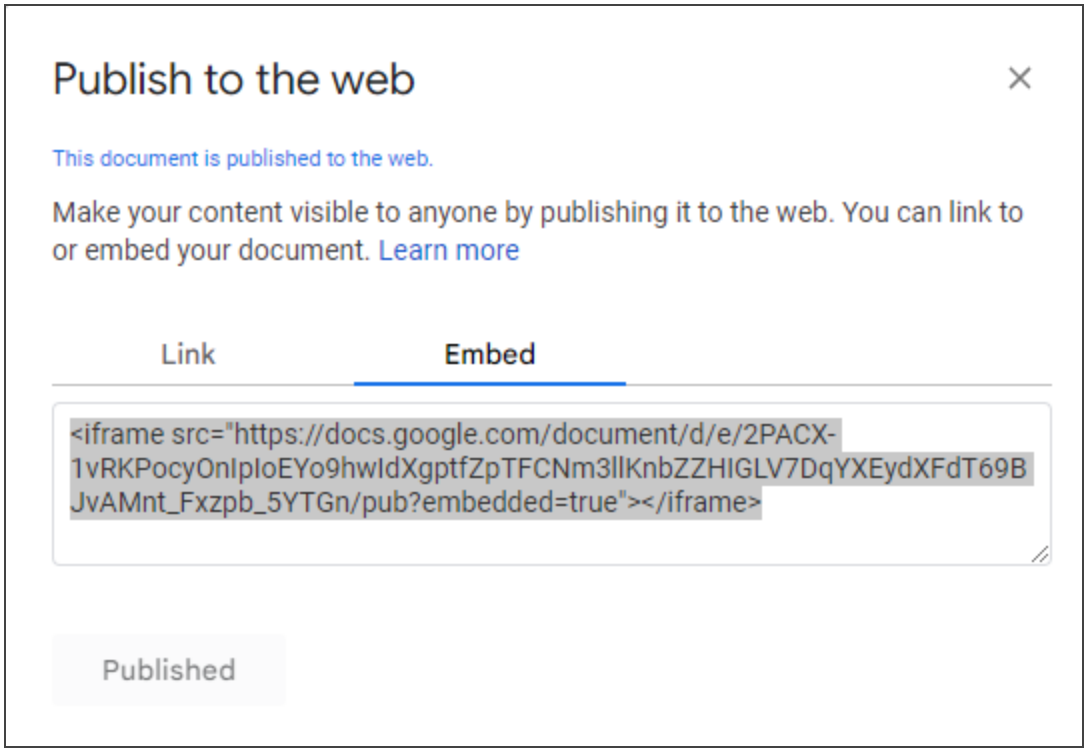


3. Embed Google Docs

Yes, you can embed Google Docs as well in a similar way. Go to doc, which you want to embed, open **file > share > Publish to web.**



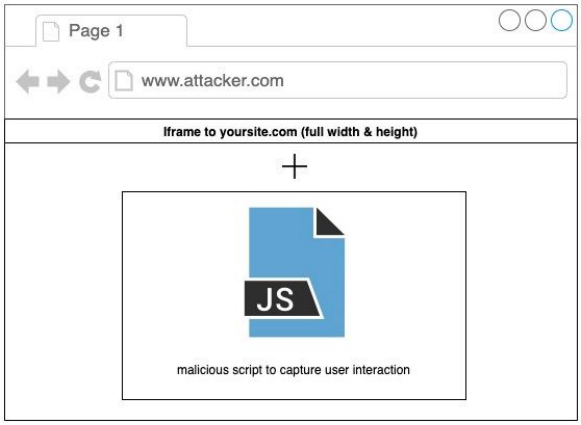
You will see an option to embed, copy the code, and paste it into the HTML document where you want to display your document.



Similarly, you can embed social media like tweets, Facebook or Instagram posts, etc.

Issues with Iframe and solution

Iframe can cause security issues, if not used properly. Let's discuss it with a scenario. Suppose an attacker embeds your website in his own website.



If any user mistakenly visits the attacker's website, the attacker's website will load an iframe of your website, along with some malicious JavaScript

code, which can steal users' personal information (like username and password).

This is commonly known as **Cross-Frame Scripting (XFS)**: It is a malicious attack where an **iframe** is used to load a genuine page while incorporating harmful JavaScript, aiming to surreptitiously extract data from an unaware user.

To prevent users from Cross-Frame Scripting, we can use **Content-Security-Policy: frame-ancestors 'none';** It prevents attackers from embedding our website in another webpage.

To use CSP(Content-Security-Policy), we can configure our web server to return the CSP HTTP header in response.

```
Unset
Content-Security-Policy: frame-ancestors 'none';
```

SVG (Scalable Vector Graphics)
What is SVG?
SVG is an XML-based language, used to draw Vector Graphics. SVGs are useful because they are scalable to any size without losing the quality of the image. See one example of an SVG image below,

Like HTML have different tags for different purpose (e.g. , <img>, <a>, etc.), similarly, SVG also has its own set of tags to define different types of shape and paths (line, circle, rectangle, etc.)

Syntax

```
Unset
<svg
    version="1.1"
    width="300"
    height="300"
    xmlns="http://www.w3.org/2000/svg"
>
      <!-- Here shapes SVG elements will come -->
</svg>
```

Here, **<svg>** tag defines an SVG container, it will contain all other SVG elements (shapes, text, paths, etc.). Here **version** attribute denotes the SVG version, **xmlns** defines the namespace for the SVG element. (Read more about namespaces here). **Width and height** attributes define SVG container size.
Example: A circle with text inside.

```
Unset
<body>
    <svg
      width="300"
      height="300"
      xmlns="http://www.w3.org/2000/svg"
    >
      <circle cx="150" cy="100" r="80" fill="blue" />
      <text x="150" y="125" font-size="60"
text-anchor="middle" fill="white">
        SVG
      </text>
    </svg>
</body>
```

Browser Output

Attributes of SVG

1. **height :** Defines height of the svg container. It can be in length or in %.
2. **width :** Defines the width of the svg container. It can be in length or in %.
3. **viewBox :** Defines viewport (visible to the user) coordinates and size. It is defined as follows,

```
Unset
viewBox=” min-x min-y width height“
```

Here **min-x** and **min-y** denote minimum x and y coordinates from the top left corner of the viewport. **width** and **height** define the size of the viewport.

Example:

```
Unset
  <body>
    <div>
      <svg
        version="1.1"
        width="300"
        height="300"
        xmlns="http://www.w3.org/2000/svg"
        viewBox="0 200 300 300"
      >
```

```
        <circle cx="200" cy="200" r="100px" fill="blue" />
      </svg>
    </div>
  </body>
```

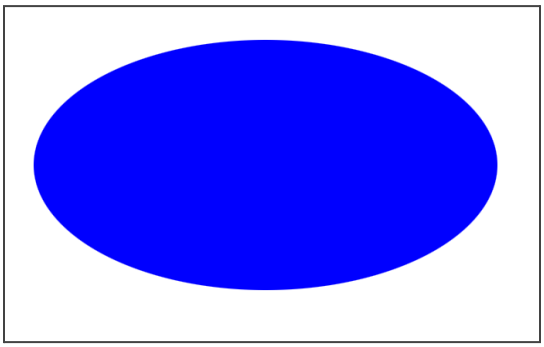
4. **x:** x-coordinate from the left of the parent container.
5. **y:** y-coordinate from the top of the parent container.
6. **preserveAspectRatio:** It defines how the SVG element will be resized if it is displayed with a different aspect ratio.

Examples of SVG

Example 1: An Eclipse

```
Unset
  <body>
    <div>
      <svg width="500" height="500"
xmlns="http://www.w3.org/2000/svg">
        <ellipse cx="200" cy="80" rx="100" ry="50"
style="fill: blue" />
      </svg>
    </div>
  </body>
```

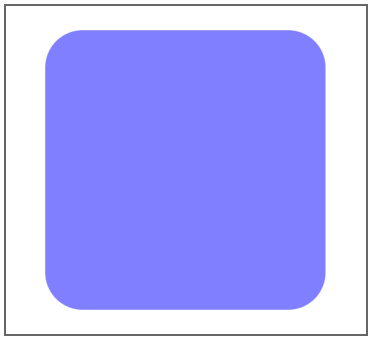
Browser Output



Example 2: A square with rounded corners.

```
Unset
  <body>
    <div>
      <svg width="500" height="500"
xmlns="http://www.w3.org/2000/svg">
        <rect
          x="50"
          y="20"
          rx="20"
          ry="20"
          width="150"
          height="150"
          style="fill: blue; opacity: 0.5"
        />
      </svg>
    </div>
  </body>
```

Browser Output



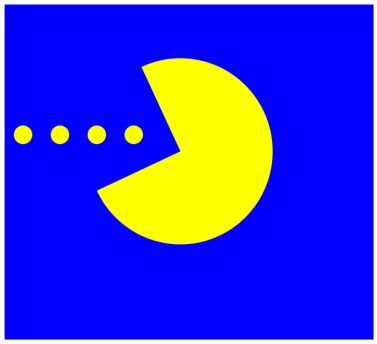
Example 4: Pacman

```
Unset
  <body>
    <div>
      <h1>PACMAN</h1>
      <svg width="500" height="500"
xmlns="http://www.w3.org/2000/svg">
        <rect width="40%" height="36%" fill="blue" />

        <g transform="rotate(-25 50 100)">
          <circle cx="100" cy="100" r="50px" fill="yellow"
/>
          <rect width="20%" height="20%" fill="blue" />
        </g>
        <circle cx="10" cy="70" r="5px" fill="yellow" />
        <circle cx="30" cy="70" r="5px" fill="yellow" />
        <circle cx="50" cy="70" r="5px" fill="yellow" />
        <circle cx="70" cy="70" r="5px" fill="yellow" />
      </svg>
    </div>
  </body>
```

Browser Output

PACMAN



Canvas

Canvas is also used to develop graphics on the web. Similar to SVG, we define our **canvas** container as below,

```
Unset
<canvas width="120" height="120">
  <!-- render shapes, lines, etc. here -->
</canvas>
```

Here, **width** and **height** attributes define the size of the canvas container. Unlike svg, we draw shapes and lines using JavaScript, and we will learn about JavaScript in later modules.

Canvas vs SVG

SVG (Scalable Vector Graphics) and Canvas are both widely used technologies for rendering graphics on the web, but they have different approaches.

SVG	Canvas

SVG graphics are vector-based, which means they are made up of paths (eg. lines).	Canvas graphics are raster-based, which means they are made up of pixels.
It is better scalable.	Less scalable.
Suitable for higher resolution.	Not suitable for higher resolution.
It can be modified via Javascript API and also CSS.	It can only be modified using Javascript.
It is part of the DOM tree.	It is rendered like a single image and can be saved to PNG or JPG format.