Exp.no: 3.a

AIM: Write a HTML program, that makes use of <article>, <aside>, <figure> <figcaption> <footer>, <header>, <main>, <nav>, <section>, , tags.

DESCRIPTION:

This HTML document is a demo webpage that showcases the use of HTML5 semantic elements such as

, , , , , , , , and to structure a blog-style website.

1. DOCTYPE & Metadata

- <!DOCTYPE html> declares the document type as HTML5.
- <html lang="en"> specifies that the page is in English.
- <meta charset="UTF-8"> ensures the webpage supports special characters.
- <title> sets the title of the browser tab.

2. CSS Styling (<style>)

- Defines layout, background colors, fonts, and spacing.

- Uses flexbox in <main> to align article and aside side by side.

-Navigation links styled to look neat and user-friendly.

3. Header (<header>)

-Displays the blog's title: "My Blog Website".

4. Navigation (<nav>)

-Provides links to different sections: Home, Articles, Gallery, Contact.

-Styled with spacing and white text for readability.

5. Main Content (<main>)

-Divided into two parts:

-Article (<article>): Contains the main blog content.

-A heading: Understanding HTML5 Semantic Tags.

-A paragraph explaining semantic elements.

-Figure (<figure>): Contains an image with a caption (<figcaption>).

-Section (<section>): Explains Why Use Semantic Tags?

-Aside (<aside>): Sidebar with related posts (Introduction to HTML, CSS Basics, JavaScript).

6. Footer (<footer>)

-Displays copyright:

PROGRAM:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>HTML 5 Semantic elements demo</title>

        <style>
                body {
                        font-family: Arial, sans-serif;
                        margin: 0;
                        }
                header, nav, footer {
                        background-color: #333;
                        color: white;
                        padding: 1em;
                        }
                nav a {
                        color: white;
                        margin: 0 10px;
                        text-decoration: none;
                        }
                main {
                        display: flex;
                        padding: 20px;
                        }
                article {
                        flex: 3;
                        padding: 20px;
                        background-color: #f9f9f9;
```

```
                }
            aside {
                flex: 1;
                padding: 20px;
                background-color: #e0e0e0;
                }
            figure {
                margin: 0;
                text-align: center;
                }
            figcaption {
                font-style: italic;
                font-size: 0.9em;
                }
            footer{
                text-align: center;
                }
        </style>
</head>
<body>
        <header>
            <h1>My Blog Website (24B11CS171)</h1>
        </header>
        <nav>
            <a href="#">Home</a>
            <a href="#">Articles</a>
            <a href="https://photos.google.com/?pli=1">Gallery</a>
            <a href="#">Contact</a>
        </nav>
        <main>
```

```
        <article>
            <h2> Understanding HTML 5 Semantic Tags </h2>
            <p> <span style="font-weight: bold;">HTML5</span>
            introduced semantic elements that make your code more readable and accessible.. </p>
        <figure>
            <img src= "C:\Users\krish\Pictures\dodge car.jpg" height=500 width=800>
            <figcaption>Figure: Illustration of HTML5 layout </figcaption>
        </figure>
        <section>
            <h3>Why Use Semantics Tags?</h3>
            <p>Semantic tags help search engines and screen readers understand the structure of your web page</p>
        </section>
        </article>
        <aside>
            <h3>Related posts</h3>
            <ul>
                <li><a href="#">Introduction to HTML</a></li>
                <li><a href="#">CSS Basics</a></li>
                <li><a href="#">Java Script for Beginners</a></li>
            </ul>
        </aside>
    </main>

    <footer>
        <p>&copy; 2025 My Blog. All rights reserved.</p>
    </footer>
</body>
</html>
```

OUTPUT:

text-align: center; }

My Blog Website (24B11CS171)

Home Articles Gallery Contact

Understanding HTML 5 Semantic Tags

HTML5 introduced semantic elements that make your code more readable and accessible..

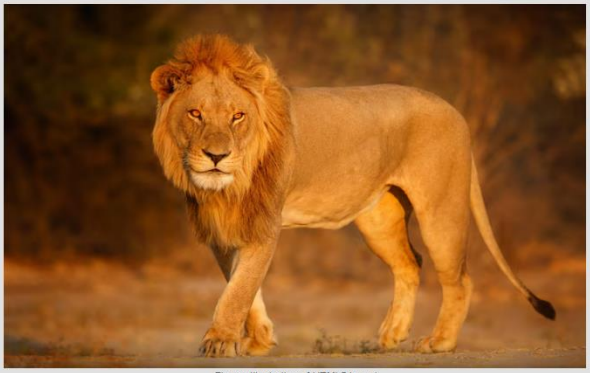


Figure: Illustration of HTML5 layout

Why Use Semantics Tags?

Semantic tags help search engines and screen readers understand the structure of your web page

Related posts

Introduction to HTML
CSS Basics
Java Script for Beginners

RESULT:

Exp.no: 3.b

AIM: Write a HTML program, to embed audio and video into HTML web page.

DESCRIPTION:

This HTML page demonstrates how to embed audio and video files into a web page using the <audio> and <video> tags in HTML5.

1. Document Setup
    - <!DOCTYPE html> declares HTML5 document type.
    - <html lang="en"> specifies the language as English.
    - <meta charset="UTF-8"> ensures correct character encoding.
    - <title> sets the title of the webpage to “Audio and Video Embedding”.
2. Page Content (<body>)
    - Main Heading (<h1>): "Embedded Audio and Video in HTML".
3. Audio Section (<audio>)
    - <audio controls> creates an audio player with built-in play, pause, volume controls.
    - <source> elements specify different audio files (MP3 format in this case):
        - .mp3 file
        - .mp3 file
    - If the browser cannot play audio, the fallback text “Your browser does not support the audio element.” is shown.
4. Video Section (<video>)
    - <video width="640" height="360" controls> creates a video player with playback controls and fixed size.
    - <source> elements specify two video files (MP4 format):
        - .mp4 file
        - .mp4 file
    - If the browser cannot play video, the fallback text “Your browser does not support the video element.” is shown.

PROGRAM:

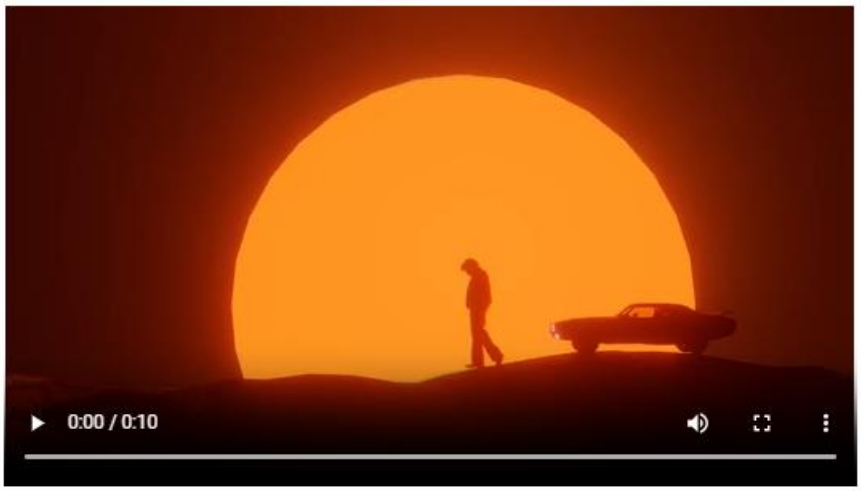
```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Audio and Video</title>
</head>
<body >
   <h1>Embedded audio and video in html (24B11CS171)</h1>
   <!-- Audio section -->
         <h2>Sample Audio</h2>
         <audio controls>
                  <source src="C:\Users\krish\Downloads\Fire Storm.mp3type="audio/mpeg">
                  <source src="C:\Users\krish\Downloads\Fire Storm.mp3" type="audio/ogg">
                  Your browser does not support the audio element.
         </audio >
    <!-- Video section -->
         <h2>Sample Video</h2>
      <video width="640" height="360" controls>
                  <source src="C:\Users\krish\Downloads\Fire Storm.mp3"type="audio/mpeg">
                  <source src="C:\Users\krish\Downloads\Fire Storm.mp3" type="audio/ogg">
                  Your browser does not support the video element.
         </video >

</body>
</html>
```

OUTPUT:

Embedded audio and video in html (24B11CS171)

Sample Audio

Sample Video



RESULT:

Exp.no: 3.c

AIM: Write a program to apply different types (or levels of styles or style specificatio formats) - inline, internal, external styles to HTML elements. (identify selecto property and value).

DESCRIPTION:

Inline CSS

- Styles are applied directly to individual HTML elements using the style attribute.
- Example: a heading can have its color and alignment defined inside the tag itself.
- Advantage: Quick and easy for small changes.
- Disadvantage: Not reusable, mixes design with content, harder to maintain.

Internal CSS

- Styles are written inside a <style> tag within the <head> section of the HTML document.
- These styles apply only to that particular web page.
- Advantage: Better separation of content and style than inline, good for single-page sites.
- Disadvantage: Cannot be reused across multiple pages.

External CSS

- Styles are stored in a separate .css file and linked to the HTML document using the <link> tag.
- Allows multiple web pages to share the same stylesheet.
- Advantage: Clean separation of structure (HTML) and design (CSS), reusable, easier maintenance, consistent styling across pages.
- Disadvantage: Requires an extra file (won't work if CSS file is missing or link breaks).

In short:

- Inline = one-time quick style
- Internal = page-level style
- External = project-wide reusable style

Program:

1) Inline Function

```
<html>
<body>
<h2>24B11CS171</h2>
<h1 style="color:blue;text-align:center;bgcolor:black;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
</body>
</html>
```

OUTPUT:

2) Internal function

```
<html>
<head>
<style>
body {
        background-color:yellow;
}
</style>
</head>
<body>
<h2>24B11CS171</h2>
<h1 style="color:blue;text-align:center;"><strong>This is a heading</strong></h1>
<p style="color:red;font-size:30px;">This is a paragraph.</p>
</body>
```

</html>

OUTPUT:

24B11CS171

This is a heading

This is a paragraph.