Web Designing Assignment

**Module (HTML5) -3**

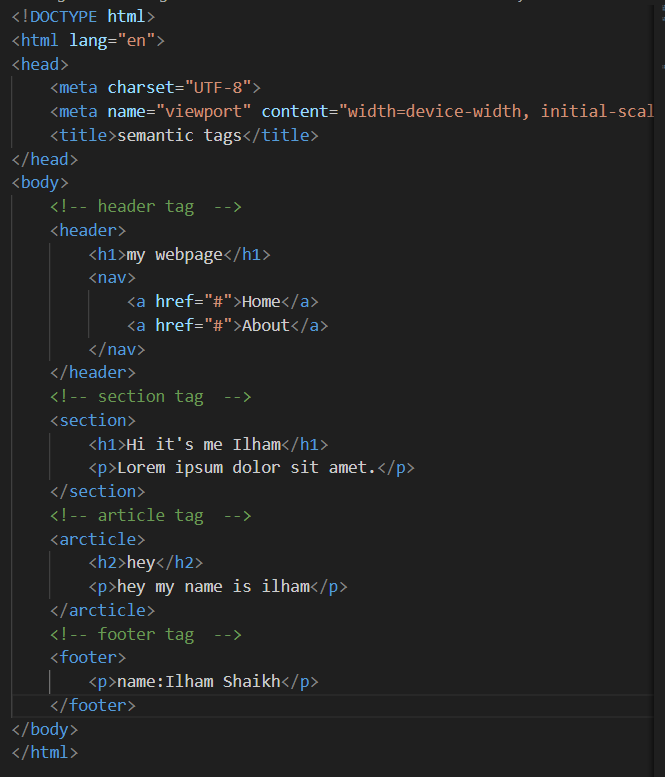
**1.What are the new tags added in HTML5?**

**Ans- HTML5 introduced a bunch of new tags that can be broadly categorized into three main areas.**

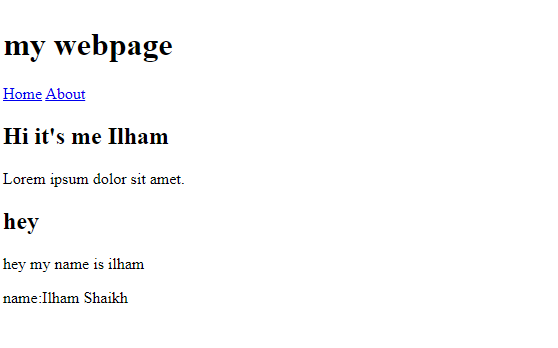
**(1) Semantic tags: These tags describe the meaning of a section of content, making the HTML more readable and understandable for both developers and search engines. ( <article> , <header> , <footer> , <nav> , <section>).**

1. **Form input types: New input types were added to the <input>tag, providing more specific input fields. ( date , email , search , tel , url ).**
2. **Media tags: These tags allow for embedding multimedia content directly into webpages. ( <audio> , <video> , <source>).**

**Example:**

****

**Output:**

****

**2.How to embed audio and video in a webpage?**

Ans- Embedding audio and video in a webpage is straightforward using HTML5.

**Embedding Audio:**

**1. The <audio> tag:** This is the core element for embedding audio on your webpage. It functions similarly to the **<img>** tag.

**2. The src attribute:** This attribute specifies the location of your audio file. the path to the audio file goes within the quotes after **src.**

**3. Controls:** By default, browsers don’t display controls for audio playback. To add controls like play, pause and volume. You can use the controls attribute within the **<audio>** tag.

**Embedding Video:**

The process for embedding video is very similar to audio.

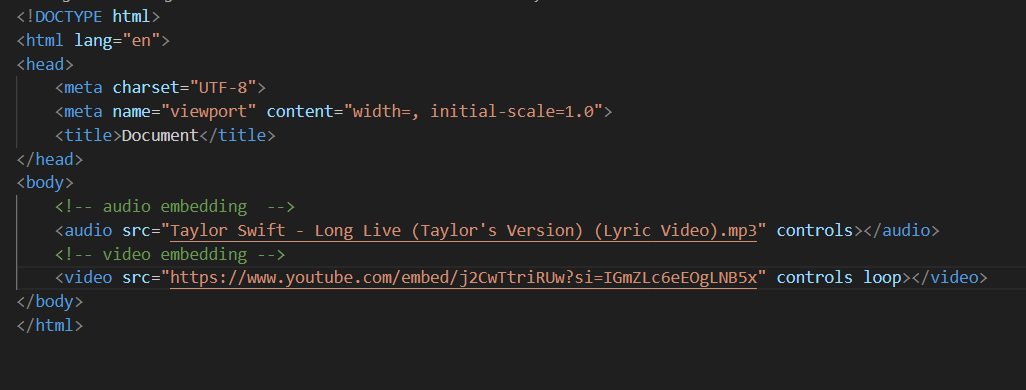
**1. The <video> tag:** This tag is used to embed video content on your webpage.

**2. The src attribute:** Similar to audio, this attribute specifies the location of your video file.

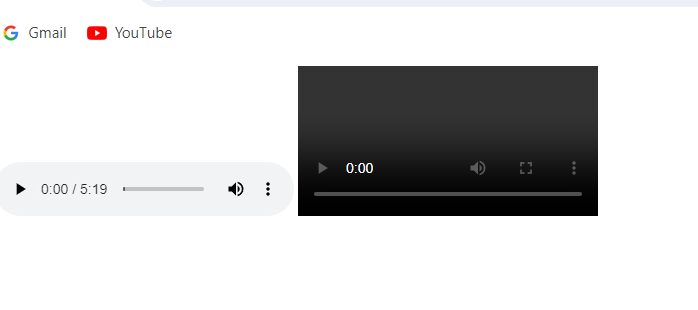
**3. Controls:** you can add control using the controls attribute within the **<video>** tag.

**4. Poster image:** You can specify a poster image to be displayed before the video starts playing using the poster attribute within the **<video>** tag.

**Example:**

****

**Output:**

****

**3. Semantic elements in HTML5?**

Ans- Semantic elements have names that clearly convey their meaning. Making the code more understandable for both developers and machines like search engines.

**Semantic elements in HTML5:**

**~ <header> :** Represents the header section of a webpage, containing logos , navigation and introductory content.

**~ <nav> :** Defines a section containing navigation links.

**~ <section> :** Represent a standalone section within a webpage, often used to group related content.

**~ <article> :** Defines an independent , reusable piece of content, like a blog post or news article.

**~ <aside> :** Represents content that’s indirectly related to the main content of the page, often for sidebars.

**~ <footer> :** Defines the footer section of a webpage. Containing copyright information, contact details, and other supplementary content.

**~ <main> :** Represent the main content of the webpage.

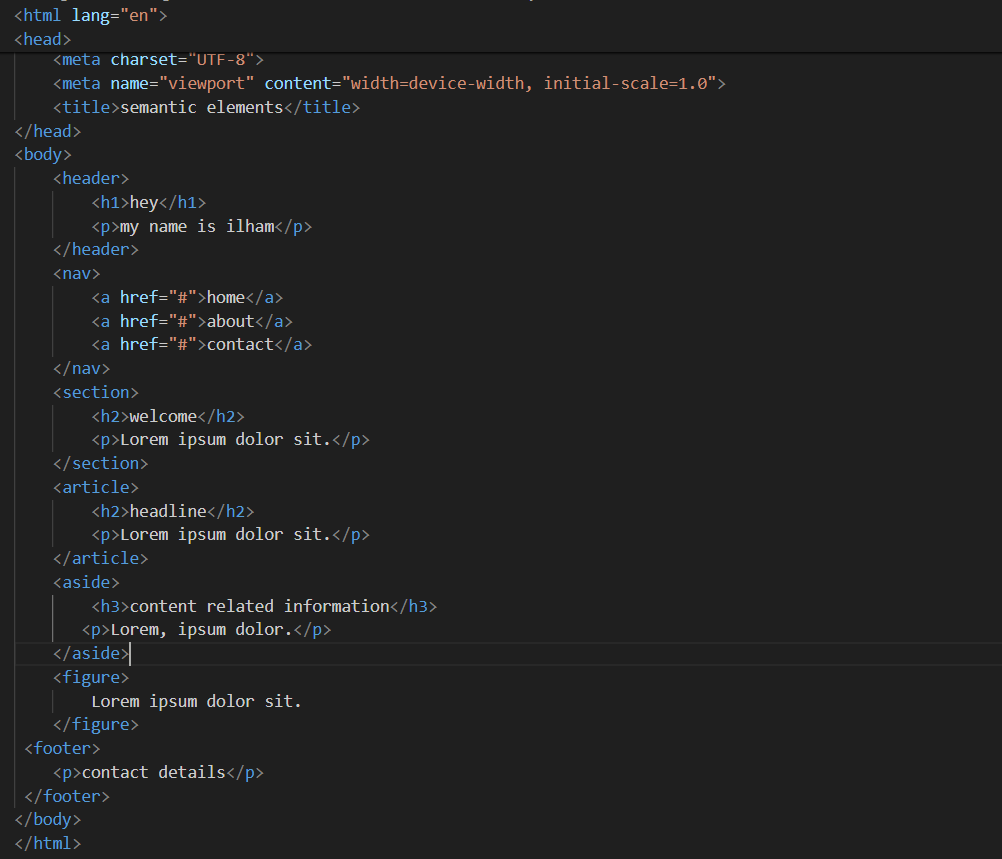
**~ <figure> :** Groups an image or other visual content with its caption.

**~ <figcaption> :** Provides a caption for a figure element.

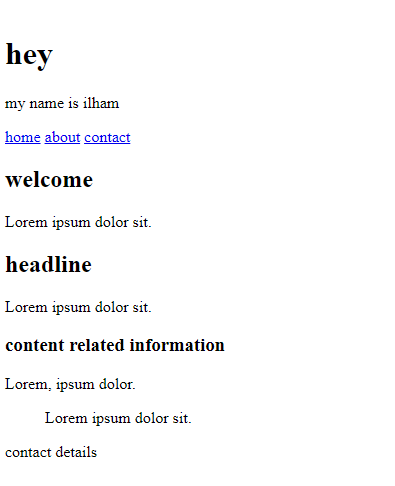
**~ <details> :** Creates a disclosure widget, allowing users to expand or collapse additional content.

**~ <summary>** : Specifies a summary or heading for a <details> element.

**Example:**



**Output:**

****

**4. Canvas and SVG tags**

Ans- Canvas and SVG are closely related to HTML.

**Canvas:** Canvas refers to the <canvas> HTML element. It acts like a designated area on your webpage where you can draw graphics using JavaScript. Imagine it as a blank canvas where you use code to paint your picture.

**How it Works:** You won’t see anything initially on the webpage when you just include the <canvas> element.

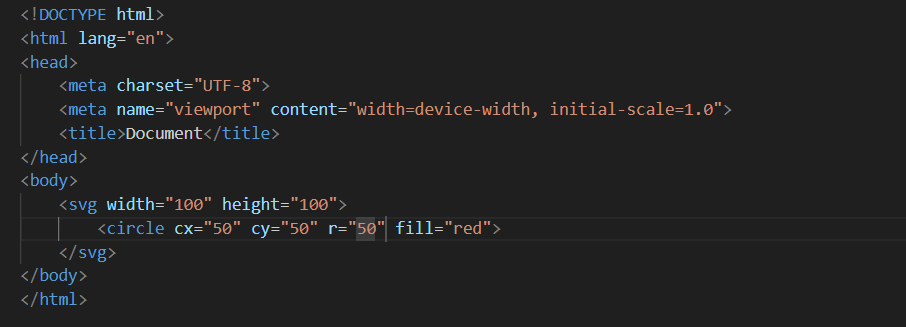
**SVG:** SVG stands for Scalable Vector Graphics. It’s a separate file format (.svg) used for creating vector graphics. While not an HTML tag itself, you use the <img> tag to reference and display SVG images on your webpage.

**How it Works:** Unlike canvas that relies on JavaScript for rendering, SVG code defines the image It uses shapes , paths and text elements written in XML format. This makes SVG graphics lightweight and infinitely scalable without losing quality.

**~ Canvas:** Ideal for dynamic and interactive graphics where you need to draw using code

**~ SVG:** Perfect for logos, icons, illustration, and other scalable graphics where quality and accessibility are important.

**Example:**

****

**Output:**

****