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

**Ans.** Some new tag added in html5 is

**<header>:** This tag is used to define the header of a document or a section. It typically contains headings, logos, navigation menus, etc.

**<footer>:** Similar to <header>, this tag defines the footer of a document or a section. It often contains copyright information, contact details, etc.

**<nav>:** Used to define a navigation menu. It typically contains links to other pages or sections within the website.

**<section>:** This tag defines a section within a document. It's often used to group related content together.

**<article>:** Used to define an independent piece of content, such as a blog post, a news article, or a forum post.

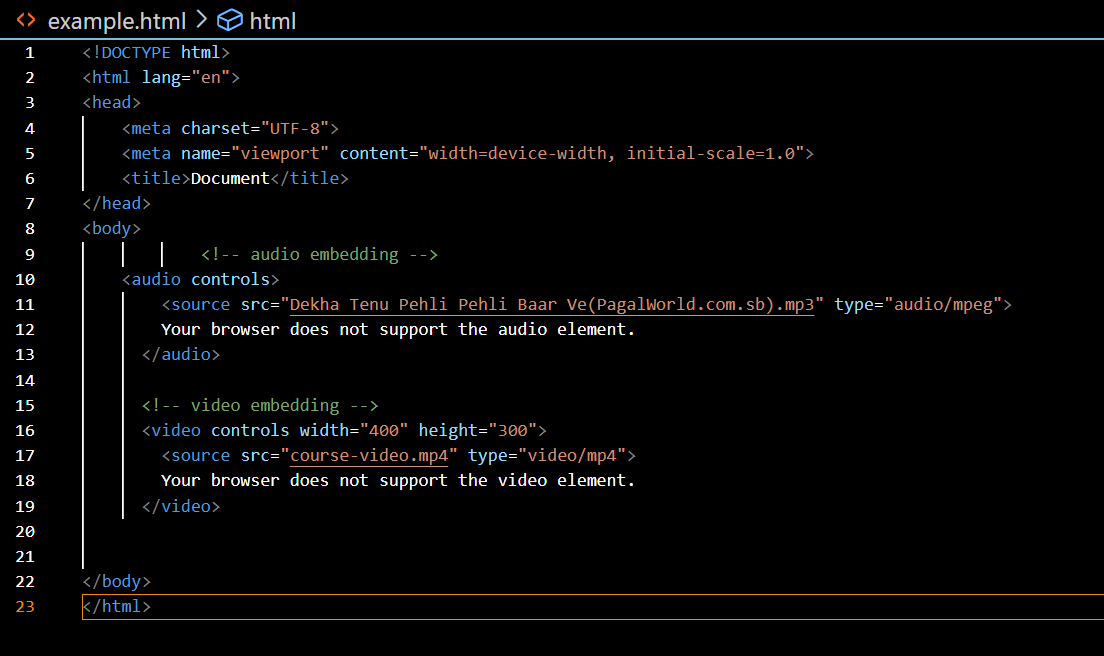
**<aside>:** This tag is used for content that is related to the main content but can be considered separate from it. It's often used for sidebars or supplementary information.

**<main>:** Defines the main content of a document. It should contain the primary content of the document, excluding headers, footers, and sidebars.

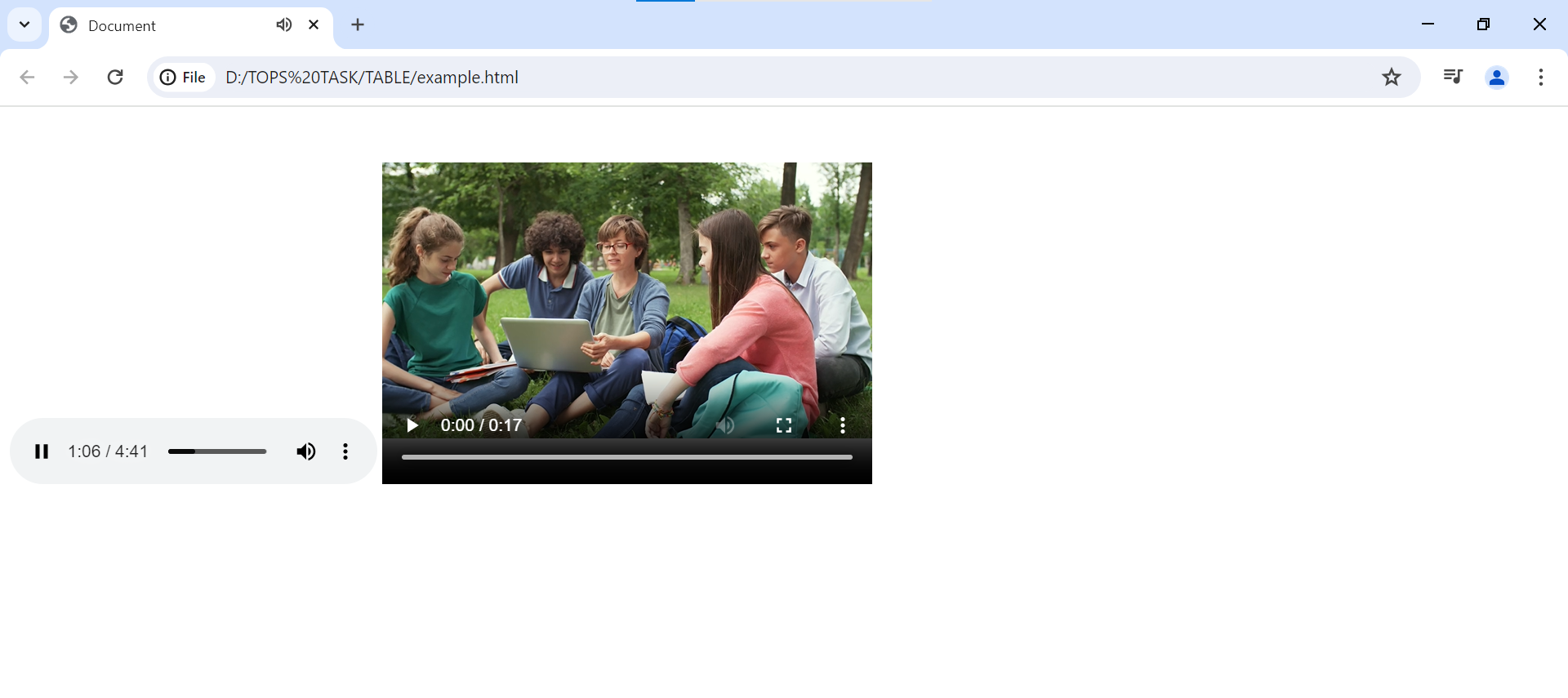
**Q-2** **How to embed audio and video in a webpage?**

**Ans.**

**Code :**

****

**Output :**



**Q-3. Semantic element in HTML5?**

**Ans.** Semantic elements in HTML5 are those that carry meaning beyond just formatting or appearance. They help to structure a web page in a meaningful and descriptive way, which is important for accessibility, search engine optimization (SEO), and overall code clarity. Some examples of semantic elements in HTML5 include:

**<header>:** Represents introductory content, typically a group of introductory or navigational aids.

**<nav>:** Represents a section of a page that links to other pages or to parts within the page.

**<main>:** Represents the main content of the <body> of a document or application.

**<article>:** Represents a self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable.

**<section>:** Represents a thematic grouping of content, typically with a heading.

**<aside>:** Represents a portion of a document whose content is only indirectly related to the main content.

**<footer>:** Represents a footer for its nearest <section>ing content or sectioning root element. A footer typically contains information about its section such as who wrote it, links to related documents, copyright data, and the like.

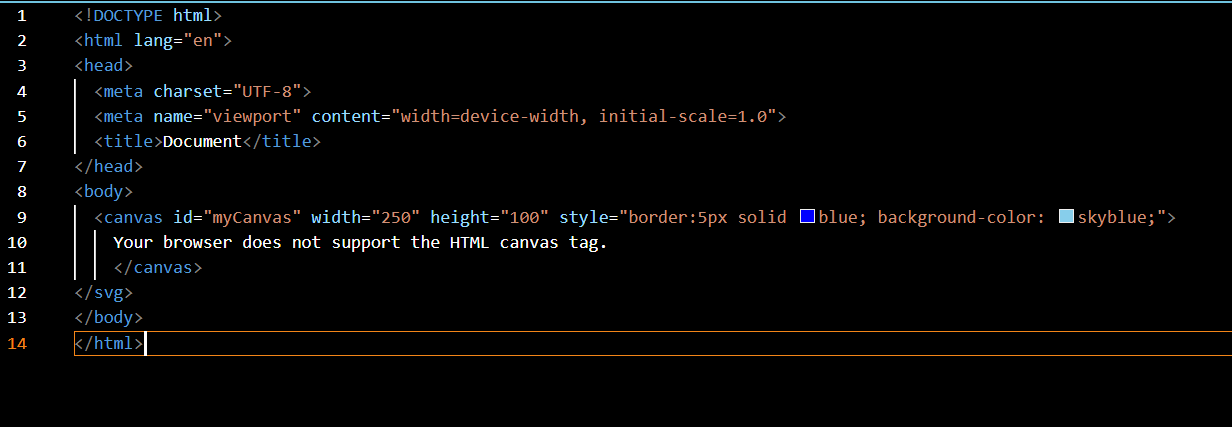
Using these semantic elements not only makes your HTML more readable but also helps search engines better understand the structure and content of your web pages, potentially improving your site's search engine ranking. Additionally, they enhance accessibility by providing clearer cues to screen readers and other assistive technologies about the purpose and structure of different parts of the page.

**Q-4.** Canvas and SVG tags

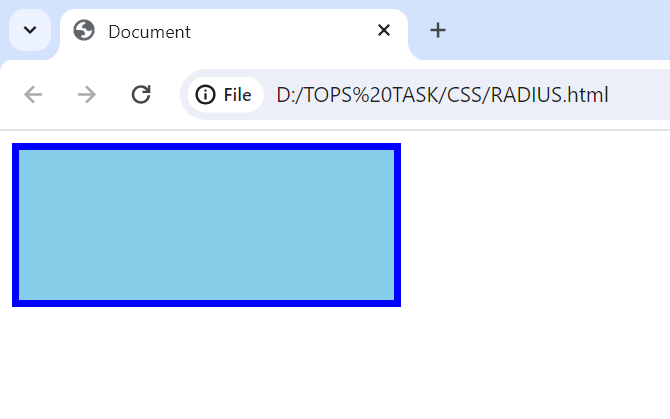
**Ans :** The <canvas> and <svg> tags are both used in HTML5 for creating graphics, but they have different approaches and use cases.

**<canvas>:** The <canvas> element is a drawing surface that allows you to use JavaScript to render graphics, animations, and interactive elements dynamically. It provides a bitmap-based rendering context, meaning that you manipulate pixels directly on the canvas. You can draw shapes, lines, text, images, and complex animations using JavaScript and the canvas API. The canvas is well-suited for dynamic and interactive graphics, such as games, data visualizations, and complex animations.

Example

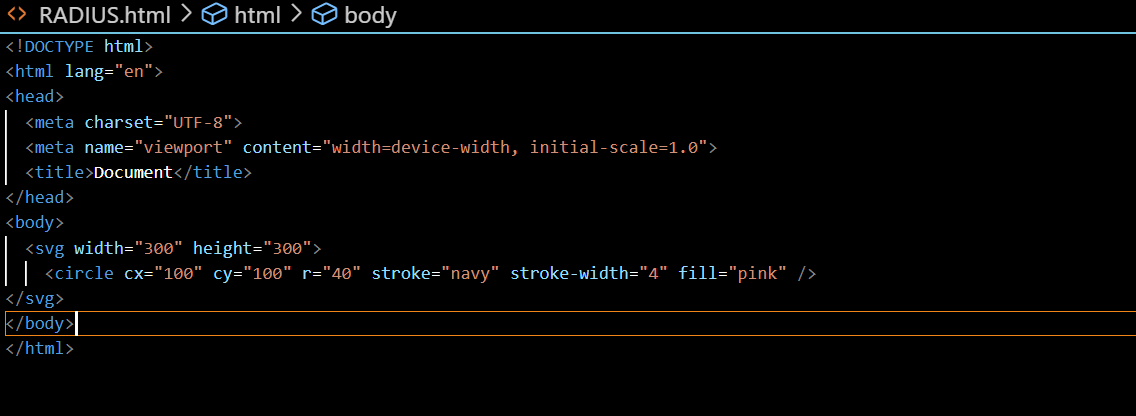


Output:



**<svg>:** The <svg> element is a vector graphics format that allows you to create scalable graphics using XML-based markup. Unlike the canvas, SVG provides a retained mode graphics model, meaning that the shapes and elements you draw are retained as objects in the DOM. SVG graphics are resolution-independent and can be scaled, rotated, and manipulated without losing quality. SVG is well-suited for static or dynamic graphics that require precise control over individual elements, such as diagrams, icons,

Example



Output:

