HTML5 ASSIGNMENTS

1. what are the new tags added in HTML5?

HTML5 added a number of new tags, including:

- Semantic elements: These tags provide more meaning to the content of a web page, which makes it easier for browsers and other tools to understand and process the page. Semantic elements include:
  - `<article>`: Defines a self-contained piece of content, such as a news article or blog post.
  - `<aside>`: Defines content that is related to the main content of the page, but not essential to it, such as a sidebar or footnote.
  - `<details>`: Defines content that can be collapsed or expanded by the user.
  - `<figcaption>`: Provides a caption for a `<figure>` element.
  - `<figure>`: Defines a self-contained unit of content, such as an image, video, or quote.
  - `<footer>`: Defines the footer section of a web page.
  - `<header>`: Defines the header section of a web page.
  - `<main>`: Defines the main content area of a web page.
- Media elements: These tags provide new ways to embed and control multimedia content on a web page. Media elements include:
  - `<audio>`: Embeds an audio file on a web page.
  - `<canvas>`: Allows JavaScript to draw graphics on a web page.
  - `<video>`: Embeds a video file on a web page.
- Form elements: These tags provide new input types for forms. Form elements include:
  - `<datalist>`: Provides a list of suggestions for an `<input>` element.
  - `<email>`: Defines an `<input>` element that must contain a valid email address.
  - `<number>`: Defines an `<input>` element that must contain a valid number.
  - `<range>`: Defines an `<input>` element that allows the user to select a value from a range.

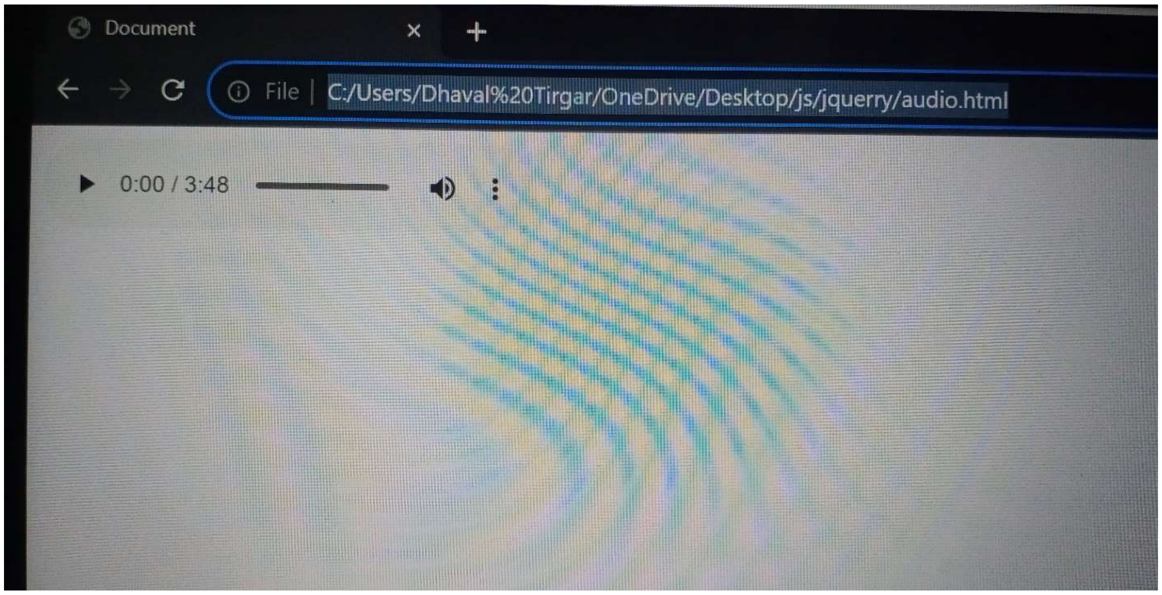
- `<search>`: Defines an `<input>` element that allows the user to enter a search term.
- `<tel>`: Defines an `<input>` element that must contain a valid phone number.
- `<url>`: Defines an `<input>` element that must contain a valid URL.

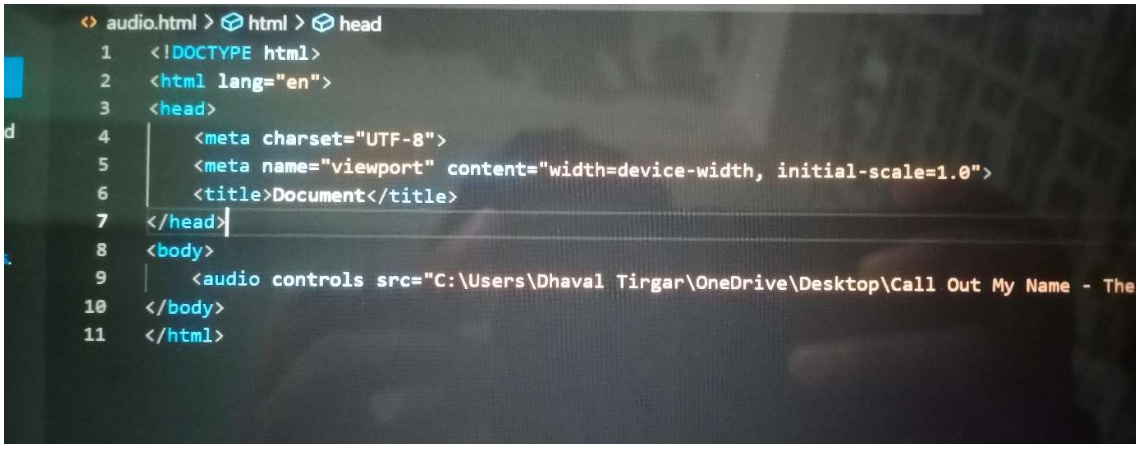
HTML5 also added a number of other new tags, such as `<dialog>`, `<keygen>`, and `<mark>`.

2. how to embed audio and video in a webpage?

To embed audio and video in a webpage, you can use the HTML5 `<audio>` and `<video>` elements, respectively. These elements allow you to specify the URL of the audio or video file that you want to embed, as well as a number of other attributes that control the playback of the file.

Here is an example of how to embed an audio file in a webpage:

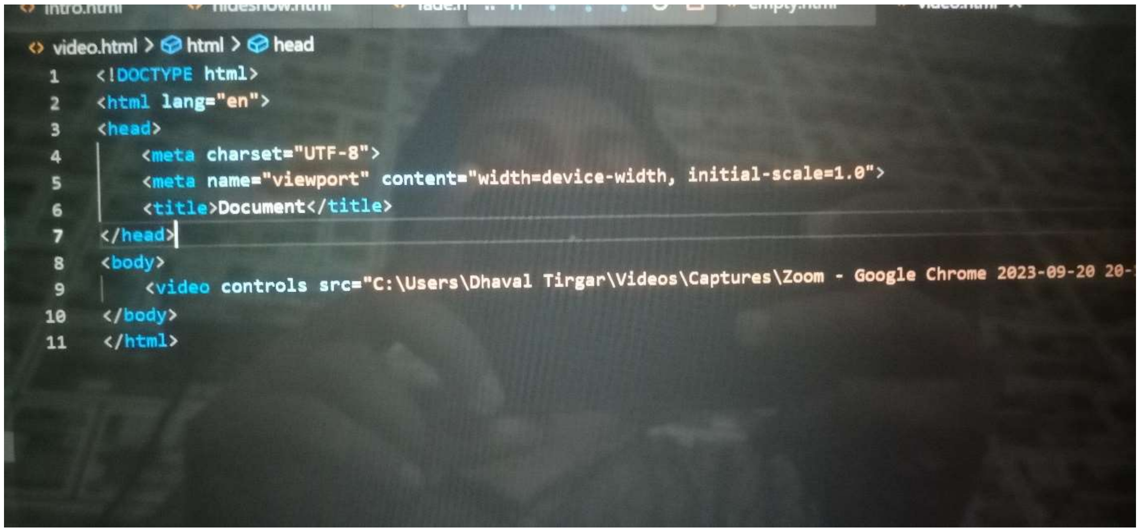

Document
File
C:/Users/Dhaval%20Tirgar/OneDrive/Desktop/js/jquerry/audio.html
0:00 / 3:48



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <audio controls src="C:\Users\Dhaval Tirgar\OneDrive\Desktop\Call Out My Name - The
</body>
</html>
```

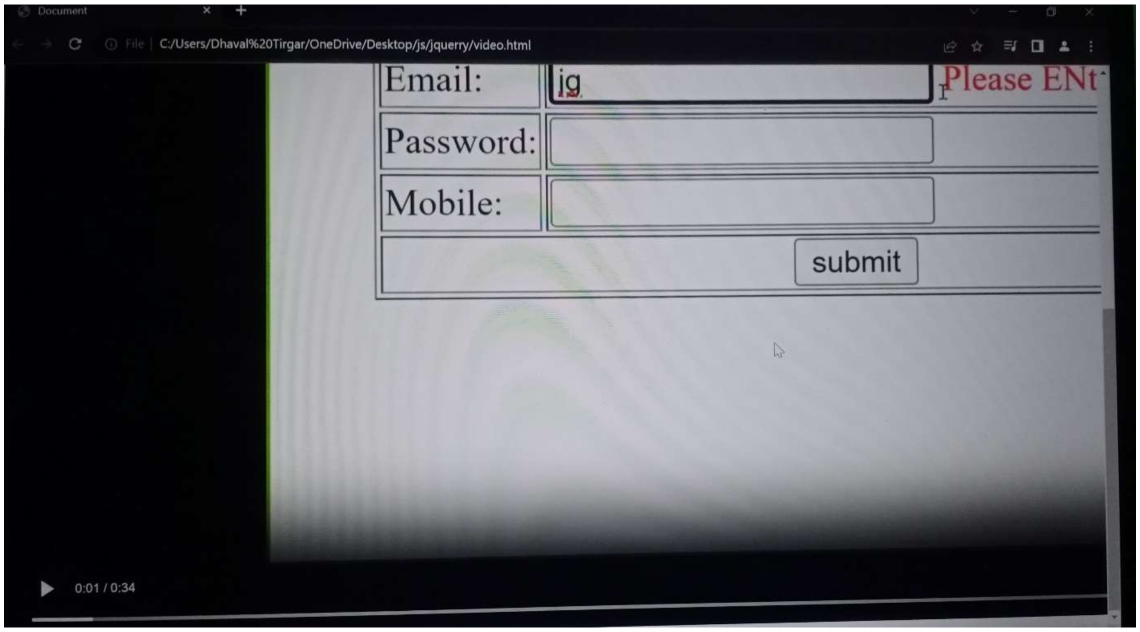
This code will embed an audio file called `audio.mp3` in the webpage, and provide the user with controls to play, pause, and stop the playback.

Here is an example of how to embed a video file in a webpage:



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <video controls src="C:\Users\Dhaval Tirgar\Videos\Captures\Zoom - Google Chrome 2023-09-20 20-
</body>
</html>
```

Output:-


Document
C:/Users/Dhaval%20Tirgar/OneDrive/Desktop/js/jquery/video.html
0:01 / 0:34

Email: jg Please ENt

Password:

Mobile:

submit

3 . semantic element in HTML5.

Semantic elements in HTML5 are elements that have a specific meaning and purpose. They are designed to make HTML code more readable and understandable for both humans and machines. Semantic elements can be used to structure the content of a web page in a way that makes it easier for browsers and other tools to process and interpret the page.

Here are some examples of semantic elements in HTML5:

`<article>`: Defines a self-contained piece of content, such as a news article or blog post.

`<aside>`: Defines content that is related to the main content of the page, but not essential to it, such as a sidebar or footnote.

`<details>`: Defines content that can be collapsed or expanded by the user.

`<figcaption>`: Provides a caption for a `<figure>` element.

`<figure>`: Defines a self-contained unit of content, such as an image, video, or quote.

`<footer>`: Defines the footer section of a web page.

`<header>`: Defines the header section of a web page.

`<main>`: Defines the main content area of a web page.

Semantic elements can also be used to provide information about the type of content that is contained within them. For example, the `<section>` element can be used to define a section of a web page that contains related content, such as a section for news articles or a section for product reviews.

Using semantic elements has a number of benefits, including:

It makes HTML code more readable and understandable for both humans and machines.

It makes it easier for browsers and other tools to process and interpret web pages.

It can improve the accessibility of web pages for users with disabilities.

It can help search engines to better understand the content of web pages and rank them more accurately.

Overall, semantic elements are a powerful tool that can help you to create more effective and accessible web pages.

4. Canvas and SVG tags.

The `<canvas>` and `<svg>` tags are both used to create graphics in HTML5, but they have different strengths and weaknesses.
Canvas
The `<canvas>` tag is used to draw graphics on the fly using JavaScript. It is a raster-based graphic, meaning that it is composed of pixels. This makes it well-suited for creating dynamic and interactive graphics, such as games and animations.

However, canvas graphics are not as scalable as SVG graphics. This means that they can become blurry or pixelated when they are scaled up or down.

SVG

The `<svg>` tag is used to create vector-based graphics. This means that the graphics are composed of shapes, rather than pixels. This makes them highly scalable and they will always look sharp, regardless of their size.

SVG graphics are also well-suited for creating complex graphics, such as logos and icons. However, they are not as well-suited for creating dynamic and interactive graphics as canvas graphics.

When to use which

Which tag you should use depends on your specific needs. If you need to create dynamic and interactive graphics, then canvas is the better choice. If you need to create scalable and high-quality graphics, then SVG is the better choice.

Here is a table that summarizes the key differences between canvas and SVG:

Feature	Canvas	SVG
Graphic type	Raster	Vector
Scalability	Poor	Excellent
Suitability for dynamic and interactive graphics	Good	Poor
Suitability for complex graphics	Poor	Good

Examples of when you might use canvas:

- Creating a game
- Creating an animation
- Creating a dynamic data visualization

Examples of when you might use SVG:

- Creating a logo
- Creating an icon
- Creating a complex illustration
- Creating a scalable graphic for a website or app