

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

**Ans:**

- 1) article
- 2) audio
- 3) canvas
- 4) command
- 5) datalist
- 6) details
- 7) embed
- 8) figcaption
- 9) figure
- 10) footer
- 11) header
- 12) hgroup
- 13) keygen
- 14) mark
- 15) math
- 16) meter
- 17) nav
- 18) output
- 19) progress
- 20) rp
- 21) rt
- 22) ruby
- 23) section
- 24) source
- 25) summary
- 26) svg
- 27) time
- 28) track
- 29) vide
- 30) wbr

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

**Ans:** The HTML5 <audio> and <video> tags make it simple to add media to a website. You need to set src attribute to identify the media source and include a controls attribute so the user can play and pause the media.

Video- <video with = "300" height = "200" controls autoplay>

```
<source src = "/html5/foo.ogg" type = "video/ogg"/>
<source src = "/html5/foo.mp4" type = "video/mp4"/>
</video>
```

Audio- <audio controls autoplay>

```
<source src = "/html5/foo.ogg" type = "audio/ogg"/>
<source src = "/html5/foo.mp4" type = "audio/wav"/>
</audio>
```

## 3. Semantic elements in HTML5?

**Ans:** Semantic elements: Semantic elements have meaningful name which tells about type of content. For example header, footer, table, ...etc. HTML5 introduces many semantic elements as mentioned below which make the code easier to write and understand for the developer as well as instructs the browser on how to treat them.

- 1) Article
- 2) Aside
- 3) Details
- 4) Figcaption
- 5) Figure
- 6) Footer
- 7) Header
- 8) Main
- 9) Mark
- 10) Nav
- 11) Section

## 4. Canvas and SVG tags

Ans: SVG: the scalable vector graphics (SVG) is an XML-based image format that is used to define two-dimensional vector-based graphics for the web.

-syntax - <svg id="svgelem" height="200" >

<circle id="greencircle" cx="60"

cy="60" r=50" fill="green" />

</svg>

Canvas- The HTML element is used to draw graphics on the fly, via scripting (usually JavaScript). The element is only a container for graphics.

Syntax- <canvas id="newCanvas" width= "100" height="100"

Style= " border: 1px solid #000000">

</canvas>

<script>

Var c = document.getElementById('newCanvas');

Var ctx = c.getContext('2d');

Ctx.fillStyle = '#7cce2b';

Ctx.fillRect(0,0,100,100);

</script>