Unit 3
HTML5 and Features

(8 Marks)

CTEVT Diploma in Computer Engineering
Subject : Web Technology
Shree Sarwajanik Secondary Technical School
Prepared By: Er. Abinash Adhikari

1. Introduction

- HTML5 builds upon the foundational structure of HTML but adds modern features to support multimedia, graphics, and application-like functionalities.
- It aims to reduce reliance on third-party plugins (e.g., Flash) by natively supporting audio, video, and interactive graphics.
- Key Goals of HTML5:
  - Enhanced support for multimedia (audio, video, and graphics).
  - Semantic tags to improve content readability for both developers and search engines.
  - Device-friendly features for better performance on mobile devices.
  - Offline capabilities using features like localStorage and cache.

2. Difference between HTML and HTML5

Feature	HTML (Pre-HTML5)	HTML5
Doctype	Complex and lengthy.	Simplified: <!DOCTYPE html>
Multimedia	Requires plugins (e.g., Flash).	Native support for <audio> and <video>.
Graphics	Not supported natively.	Introduced <canvas> and SVG support.
Semantic Tags	Lacked descriptive tags.	Introduced semantic tags like <header>, <footer>, <article>.
Forms	Basic input types.	New input types like email, date, range.
Offline Storage	Cookies only.	Added localStorage and sessionStorage.
Cross-Browser	Compatibility issues.	Improved standardization for modern browsers.

3. HTML5 New Semantics Elements

- Semantic elements provide meaning to the structure of a webpage, improving readability and SEO. Examples:
  - HEADER: Represents introductory content or navigation links for a section or page.
  - FOOTER: Represents the footer of a section or page, often containing contact info or links.
  - SECTION: Represents a thematic grouping of content, typically with a heading.
  - Example:

```
<header>
    <h1>Welcome to HTML5</h1>
</header>
<section>
    <h2>About HTML5</h2>
    <p>HTML5 introduces modern features for web development.</p>
</section>
<footer>
    <p>&copy; 2024 Your Company</p>
</footer>
```

4. HTML5 New Elements

4.1. Tables, Images, Colors, Canvas, Forms

A. Tables: Enhanced with better accessibility features using ARIA roles.
B. Images: Added support for vector graphics using SVG.
C. Colors: HTML5 allows input fields to accept color values through type="color".
D. Canvas: A drawing surface for rendering 2D graphics programmatically.
E. Forms: New input types like email, url, range, and attributes like required, pattern, and placeholder.

A. Tables (with ARIA Roles)

ARIA roles improve accessibility for users with assistive technologies by defining the purpose of elements in tables.

```
<table role="grid" aria-label="Student Data">
    <thead>
        <tr>
            <th role="columnheader">Name</th>
            <th role="columnheader">Age</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td role="gridcell">Alice</td>
            <td role="gridcell">22</td>
        </tr>
        <tr>
            <td role="gridcell">Bob</td>
            <td role="gridcell">24</td>
        </tr>
    </tbody>
</table>
```

Output:

Name	Age
Alice	22
Bob	24

B. Images (Using SVG)

SVG provides scalable and resolution-independent vector graphics directly in HTML.

Code:

```
<svg width="100" height="100">
    <rect width="100" height="100" style="fill:blue;" />
    <circle cx="50" cy="50" r="40" style="fill:white;" />
    <text x="50" y="55" font-size="18" fill="blue" text-anchor="middle">SVG</text>
</svg>
```

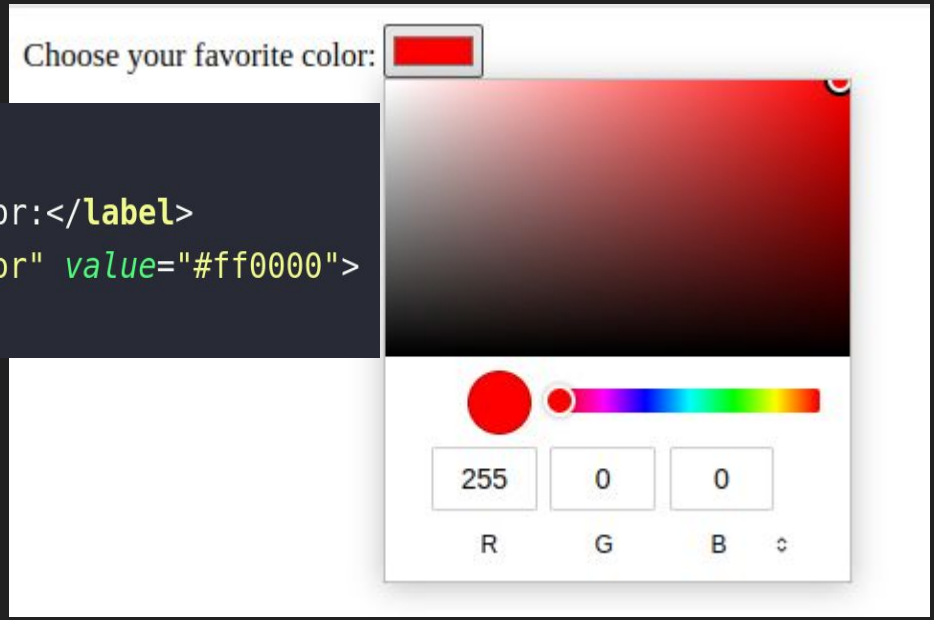
Output:

C. Colors (Input Type color)

HTML5 allows users to select a color through a color picker.

Code:

```
<form>
    <label for="favcolor">Choose your favorite color:</label>
    <input type="color" id="favcolor" name="favcolor" value="#ff0000">
</form>
```



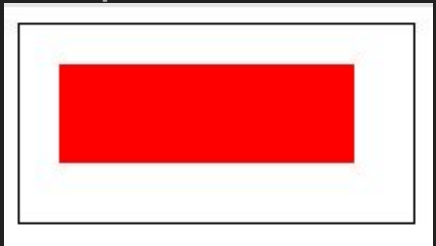
D. Canvas

The <canvas> element is used to create and draw 2D graphics programmatically.

Code:

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid black;"></canvas>
<script>
    const canvas = document.getElementById('myCanvas');
    const ctx = canvas.getContext('2d');
    ctx.fillStyle = 'red';
    ctx.fillRect(20, 20, 150, 50); // Draws a red rectangle
</script>
```

Output:



E. Forms (New Input Types)

HTML5 introduces input types such as email, url, and range, along with attributes like required, pattern, and placeholder.

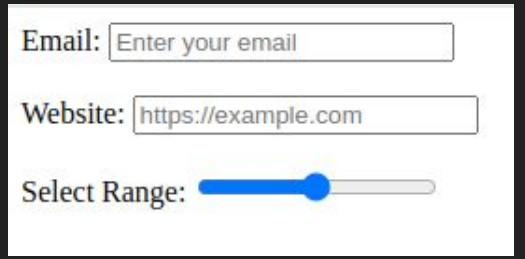
Code:

```
<form>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email" placeholder="Enter your email" required><br><br>

    <label for="website">Website:</label>
    <input type="url" id="website" name="website" placeholder="https://example.com"><br><br>

    <label for="range">Select Range:</label>
    <input type="range" id="range" name="range" min="1" max="100" value="50">
</form>
```

Output:



4.2. Interactive Elements

HTML5 introduces elements to create interactive content without plugins.

- <details>: Creates a collapsible content block.
- <summary>: Defines the heading of a <details> block.
- <dialog>: Represents a modal dialog box.

Code:

```
<details>
    <summary>More about HTML5</summary>
    <p>HTML5 provides better multimedia support and semantic structure.</p>
</details>
```

Output:

4.3. Graphics

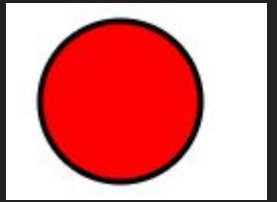
HTML5 supports advanced graphics through:

- <canvas>: For drawing 2D graphics.
- SVG (Scalable Vector Graphics): XML-based graphics format.

Code:

```
<svg width="100" height="100">
    <circle cx="50" cy="50" r="40" stroke="black" stroke-width="3" fill="red" />
</svg>
```

Output:



4.4. Multimedia

- HTML5 natively supports embedding audio and video.
  - Audio: The <audio> element allows embedding sound files with playback controls.
  - Video: The <video> element provides support for videos, including subtitles through <track>.
- Attributes:
  - Controls: Adds play, pause, volume buttons.
  - Autoplay: Automatically starts playback.
  - Loop: Repeats playback indefinitely.
  - Muted: Starts playback muted.

Code:

```
<video width="320" height="240" controls>
    <source src="example.mp4" type="video/mp4">
    Your browser does not support the video tag.
</video>
<br>
<br>
<br>
<audio controls>
    <source src="example.mp3" type="audio/mpeg">
    Your browser does not support the audio tag.
</audio>
```

Output: