Unit 3: HTML5

Prepared By:
Bipin Maharjan

Learning Objectives

- Create graphics using the HTML5 canvas element.
- Explore new form input types, including email, URL, date, time, and range.
- Master form validation using built-in attributes and placeholder text.
- Adhere to proper code indentation, formatting, and consistent naming conventions.

Contents

- HTML5 APIs
- HTML5 Forms
- Responsive Web Design
- HTML5
- Semantic Markup
- Best Practices and Optimization

HTML5

HTML5

- HTML5 is the latest major version of Hypertext Markup Language (HTML), used to structure and present content on the web.
- Incorporates all features from earlier versions of HTML
- Adds a diverse set of new tools for the web developer to use
- It is a complete platform for building modern web application that includes:
  - New HTML elements
  - Better multimedia support
  - Improved forms
  - Built-in APIs
  - Better semantics
  - Support for responsive and accessible web pages

Goals of HTML5

- Reduce the need for external plugins and scripts to show website content
- Improve the semantic definition (i.e. meaning and purpose) of page elements
- Make the rendering of web content universal and independent of the device being used.
- Handle web documents errors, in a better and more consistent fashion.

HTML5 Document Structure

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>HTML5 Example</title>
</head>
<body>
    <h1>Welcome to HTML5</h1>
    <p>This is a basic HTML5 document.</p>
</body>
</html>
```

Note:

- <!DOCTYPE html> tells the browser this is an HTML5 document
- No version numbers are required

Elements in HTML5

Tags	Description
<article>	Defines an article in a document
<aside>	Defines content aside from the page content
<dialog>	Defines a dialog box or window
<figcaption>	Defines a caption for a <figure> element
<footer>	Defines a footer for a document or section
<header>	Defines a header for a document or section
<main>	Defines the main content of a document
<mark>	Defines marked/highlighted text

Elements in HTML5

Tags	Description
<meter>	Defines a scalar measurement within a known range (a gauge)
<nav>	Defines navigation links
<progress>	Represents the progress of a task
<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time
<wbr>	Defines a possible line-break

New Features in HTML5

- Built-in audio and video support (without plugins)
- Vector graphics and animation
- Enhanced form controls and attributes
- The Canvas (a way to draw directly on a web page)
- Drag and Drop functionality
- Support for CSS3 (the newer and more powerful version of CSS)
- More advanced features for web developers, such as data storage and offline applications

Advantages of HTML5

- Cleaner markup / Improved code
  - Enable web designers to use cleaner, neater code. We can remove div tags and replace them with semantic HTML5 elements
- Elegant forms
  - Enables designer to use fancier forms. There will be different type of inputs, search and different fields for different purpose.
- Consistency
  - As websites will adopt the new HTML5 elements for consistency. This will make it much easier for designers and developers to immediately understand how a web page is created
- Supports rich media elements
  - HTML5 has an inbuilt capacity to play audio and video eliminating the plugin tags.
- Offline Application Cache
  - HTML5 offers an offline application cache facility which will load the page, the user has visited even if the user is temporarily offline. This feature will help the files to load much faster and reduces load on server.

Disadvantages of HTML5

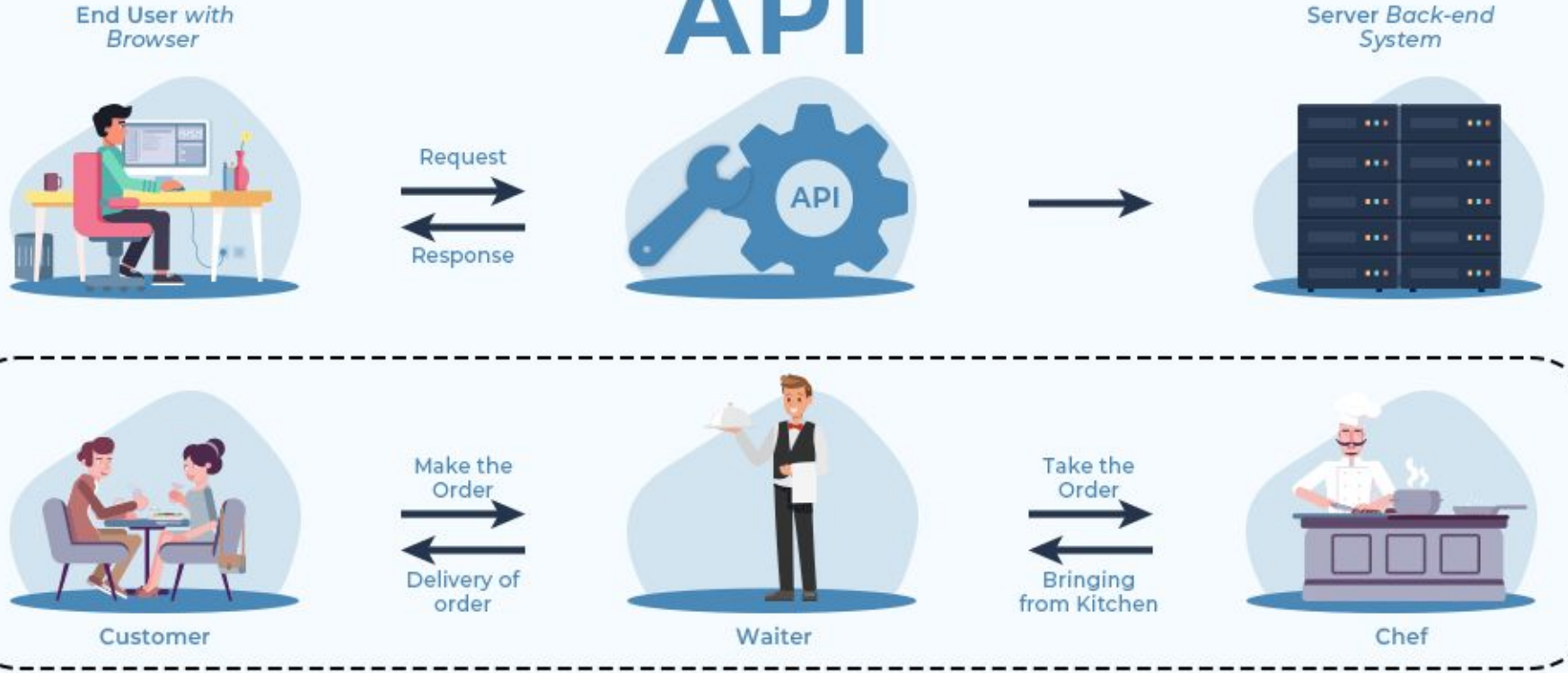
- Browser Inconsistencies
  - Older browsers may not fully support HTML5 features
- Mobile Compatibility
  - Some older mobile devices may struggle with certain HTML5 features
- Security Risks
  - New features like Web Storage can potentially expose user data if not handled properly
- Complexity
  - Advanced features like WebSockets and APIs can be challenging for beginners
- Lack of Support in Older Internet Explorer Versions
  - Some older versions of IE (like IE 8 and below) do not support HTML5 at all.

HTML5 APIs

What is API?

- APIs, or Application Programming Interfaces, are the invisible backbone of modern software development.
- They enable different applications, platforms, or systems to communicate and exchange data seamlessly, powering everything from your weather app to your favorite e-commerce checkout.
- Imagine you’re at a restaurant:
  - You (the client) place an order with the waiter.
  - The waiter (the API) conveys your request to the chef (the server).
  - The chef prepares the food (processes the request) and gives it to the waiter.
  - The waiter delivers it back to your table (response).

API


End User *with Browser*
Request
Response
API
Server *Back-end System*
Make the Order
Delivery of order
Take the Order
Bringing from Kitchen
Customer
Waiter
Chef

HTML5 API

- HTML5 APIs are built to make web development simple and easy.
- They offer ready-to-use tools for adding cool features like videos, graphics, and real-time updates to websites without needing complex code.

Main HTML5 APIs

- **Geolocation API**: This API is used to access the current location of a user (with latitude and longitude).
- **Drag and Drop API:** This API enables you to use drag-and-drop features in browsers.
- **Web Storage API**: This API has mechanisms to let browsers store key/value pairs (in a more intuitive way than cookies).
- **Web Workers API**: This API allows a JavaScript to run in the background, without affecting the performance of the page. Users can continue to do whatever they want: clicking, selecting things, etc., while the web worker runs in the background.
- **Server-Sent Events API**: This API allows a web page to automatically get updates from a server.
- **Canvas API**: This API lets you draw graphics, on the fly, via JavaScript.

Canvas API

```
<html>
  <body>
    <canvas id="myCanvas" width="200" height="200"></canvas>
    <script>
        const canvas = document.getElementById("myCanvas");
        const ctx = canvas.getContext("2d");
        // Set fill color to blue
        ctx.fillStyle = "blue";
        // Draw a filled rectangle
        ctx.fillRect(50, 50, 100, 100);
    </script>
</body>
</html>
```

HTML5 Forms

What is HTML5 Forms?

- Forms are used to collect user input and send data to a server.
- HTML5 significantly enhances forms by introducing:
  - New input types
  - Built-in validation
  - Better user experience
  - Less dependency on Javascript
- HTML5 forms are smarter, more user-friendly, and easier to code.

Structure of an HTML5 Form

```
<form action="#" method="post">
    <!-- form elements go here -->
</form>
```

Example of HTML5 Form

```
<form>
    <label>Name:</label>
    <input type="text"><br><br>

    <label>Email:</label>
    <input type="email"><br><br>

    <input type="submit">
</form>
```

HTML5 Input Types

EmailEnsures the user enters a valid email address.

```
<input type="email" placeholder="example@mail.com">
```

If the format is incorrect, the browser shows an error message.

URL: Accepts only valid website URLs.

```
<input type="url" placeholder="https://example.com">
```

Number: Accepts numeric values only.

```
<input type="number" min="1" max="100">
```

HTML5 Input Types

Date: Provides a date picker.

```
<input type="date">
```

Time: Allows selecting time

```
<input type="time">
```

Range: Creates a slider control.

```
<input type="range" min="0" max="10">
```

HTML5 Input Types

Color: Allows color selection.

```
<input type="color">
```

Search: Used for search fields

```
<input type="search">
```

Tel: Used for telephone numbers

```
<input type="tel" placeholder="98XXXXXXXX">
```

HTML5 Form Attributes (Validation)

Required: Makes a field mandatory.

```
<input type="text" required>
```

Placeholder: Displays hint text inside input fields.

```
<input type="text" placeholder="Enter your name">
```

Pattern: Uses regular expressions to validate input.

```
<input type="text" pattern="[A-Za-z]+" placeholder="Only letters">
```

HTML5 Form Attributes (Validation)

maxlength and minlength: Controls input length.

```
<input type="text" minlength="5" maxlength="10">
```

readonly and disabled

```
<input type="text" value="Nepal" readonly>
<input type="text" disabled>
```

Autofocus and Autocomplete

Autofocus: Automatically places cursor in the field.

```
<input type="text" autofocus>
```

Autocomplete: Controls browser auto-fill.

```
<input type="email" autocomplete="on">
```

Grouping Form Elements: fieldset and legend

Used to group related inputs.

```
<form>
    <fieldset>
        <legend>Personal Information</legend>
        <label>Name:</label>
        <input type="text"><br><br>
        <label>Age:</label>
        <input type="number">
    </fieldset>
</form>
```

Form Example

```
<form>
    <fieldset>
        <legend>Personal Information</legend>
        <label>Full Name:</label><br>
        <input type="text" placeholder="Enter full name" required autofocus><br><br>
        <label>Email Address:</label><br>
        <input type="email" placeholder="example@mail.com" required
autocomplete="on"><br><br>
        <label>Mobile Number:</label><br>
        <input type="tel" placeholder="98XXXXXXXX" pattern="[0-9]{10}" required><br><br>
        <label>Date of Birth:</label><br>
        <input type="date" required><br><br>
    </fieldset>

    <br>
```

```
  <fieldset>
     <legend>Academic Details</legend>
     <label>College Website:</label><br>
     <input type="url" placeholder="https://college.edu"><br><br>
     <label>Current Semester:</label><br>
     <input type="number" min="1" max="8" required><br><br>
     <label>Preferred Study Time:</label><br>
     <input type="time"><br><br>
     <label>Programming Skill Level:</label><br>
     <input type="range" min="1" max="5"><br>
     <small>1 = Beginner, 5 = Advanced</small><br><br>
  </fieldset>
  <br>
  <fieldset>
     <legend>System Information</legend>
     <label>Country:</label><br>
     <input type="text" value="Nepal" readonly><br><br>
     <label>Student ID:</label><br>
     <input type="text" value="Auto Generated" disabled><br><br>
  </fieldset>
  <br>
  <input type="submit" value="Register">
</form>
```

Student Registration Form

Personal Information

Full Name: Enter full name

Email Address: example@mail.com

Mobile Number: 98XXXXXXXX

Date of Birth: dd/mm/yyyy

Academic Details

College Website: https://college.edu

Current Semester:

Preferred Study Time: --:-- --

Programming Skill Level:

1 = Beginner, 5 = Advanced

System Information

Country: Nepal

Student ID: Auto Generated

Register

Responsive Web Design

HTML Responsive Web Design

- Responsive web design is about creating web pages that look good on all devices!
- Responsive Web Design is an approach to web development where a webpage adapts its layout and content to different screen sizes and devices.
  - Desktop computers
  - Laptops
  - Tablets
  - Mobile phones

Why is it important?

Without Responsive Web Design:

- Content may overflow on small screens
- Users need to zoom or scroll horizontally
- User experience becomes poor

Benefits of RWD

- One website for all devices
- Better usability
- Improved accessibility
- Better search engine ranking
- Reduced development and maintenance effort

Role of HTML5 in Responsive Web Design

While CSS mainly handles layout responsiveness, HTML5 contributes in structural and functional ways:

- Semantic elements for flexible structure
- Meta viewport for mobile scaling
- Responsive multimedia
- Form input optimization for mobile devices

Viewport Meta Tag

The viewport controls how a webpage is displayed on mobile devices.

Without Viewport

- Page is scaled down
- Text appears very small

With Viewport

- Page fits device width
- Content is readable

<meta name="viewport" content="width=device-width, initial-scale=1.0">

Responsive Images in HTML5

Responsive images are images that scale nicely to fit any browser size.

Using the width Property

If the CSS width property is set to 100%, the image will be responsive and scale up and down:

```
<img src="img_girl.jpg" style="width:100%;">
```

Using the max-width Property

If the max-width property is set to 100%, the image will scale down if it has to, but never scale up to be larger than its original size:

```
<img src="img_girl.jpg" style="max-width:100%;height:auto;">
```

Show Different Images Depending on Browser Width

- The HTML <picture> element allows you to define different images for different browser window sizes.
- Resize the browser window to see how the image below changes depending on the width:

```
<picture>
  <source srcset="img_smallflower.jpg" media="(max-width: 600px)">
  <source srcset="img_flowers.jpg" media="(max-width: 1500px)">
  <source srcset="flowers.jpg">
  <img src="img_smallflower.jpg" alt="Flowers">
</picture>
```

Responsive Multimedia (Audio and Video)

HTML5 multimedia elements automatically adjust better than older technologies.

```
<video controls>
    <source src="video.mp4" type="video/mp4">
</video>
```

Mobile-Friendly Forms (HTML5 Contribution)

Input Type	Mobile Benefit
`email`	Email keyboard
`number`	Numeric keypad
`tel`	Phone keypad
`date`	Date picker

Semantic Markup

What is Semantic Markup?

- Semantic Markup means using HTML elements that clearly describe the meaning and purpose of the content they contain.
- In simple terms:
  - Semantic tags tell what the content is
  - Non-semantic tags tell only how the content looks

Semantic Elements

Element	Meaning
`<header>`	Introductory content
`<nav>`	Navigation links
`<section>`	Related content
`<article>`	Independent content
`<aside>`	Side information
`<footer>`	Footer content

Example

<header>Header</header>

<nav>Menu</nav>

<section>Content</section>

<footer>Footer</footer>

Now, the structure is self-descriptive

<header>

- Represents introductory content.

```
<header>
    <h1>News Portal</h1>
    <p>Latest updates</p>
</header>
```

Can appear:

- At the top of the page
- Inside sections or articles

<nav>

- Contains navigation links.

```
<nav>
    <a href="#">Home</a>
    <a href="#">About</a>
    <a href="#">Contact</a>
</nav>
```

- Use only for main navigation, not every link.

<section>

- Groups related content.

```
<section>
  <h2>Courses</h2>
  <p>Web development courses.</p>
</section>
```

- Usually has a heading.

<article>

- Represents independent, reusable content.

```
<article>
    <h2>HTML5</h2>
    <p>HTML5 introduces semantic tags.</p>
</article>
```

Examples:

- Blog post
- News article
- Product card

<aside>

- Contains secondary or side content.

```
<aside>
    <p>Related links</p>
</aside>
```

Examples:

- Advertisements
- Author bio
- Related posts

<footer>

- Represents footer content.

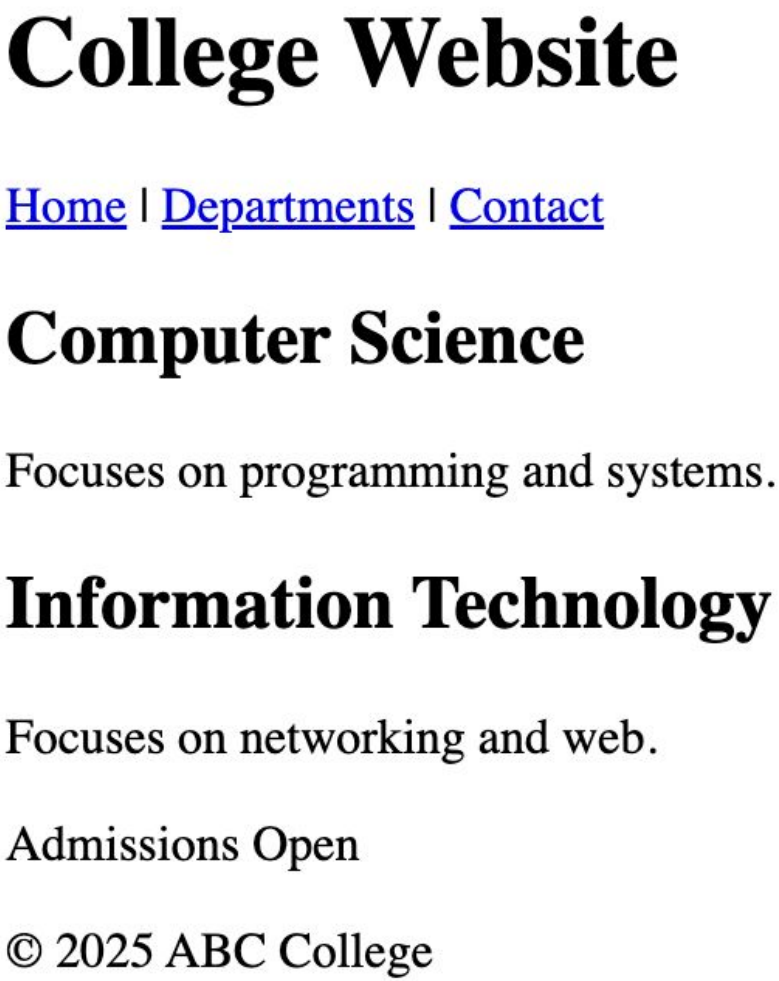
```
<footer>
    <p>© 2025 College Name</p>
</footer>
```

Can appear:

- At the bottom of the page
- Inside articles or sections

Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Semantic HTML5 Example</title>
</head>
<body>
<header>
    <h1>College Website</h1>
</header>
<nav>
    <a href="#">Home</a> |
    <a href="#">Departments</a> |
    <a href="#">Contact</a>
</nav>
<main>
    <section>
        <article>
            <h2>Computer Science</h2>
            <p>Focuses on programming and systems.</p>
        </article>
        <article>
            <h2>Information Technology</h2>
            <p>Focuses on networking and web.</p>
        </article>
    </section>
    <aside>
        <p>Admissions Open</p>
    </aside>
</main>
<footer>
    <p>© 2025 ABC College</p>
</footer>
</body>
</html>
```



Best Practices and Optimization

What Are Best Practices in HTML5?

- Best practices are recommended ways of writing HTML code so that it is:
  - Clean
  - Readable
  - Maintainable
  - Efficient
  - Accessible
  - Optimized for performance and search engines
- Optimization means improving the quality and efficiency of the webpage without changing its output.

Use Proper HTML5 Document Structure

Always start with a valid HTML5 structure.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>HTML5 Best Practices</title>
</head>
<body>
    <!-- Content goes here -->
</body>
</html>
```

Use Semantic HTML Wherever Possible

Bad Practice (Non-semantic)

```
<div>
    <div>Header</div>
    <div>Menu</div>
</div>
```

Good Practice (Semantic)

```
<header>Header</header>
<nav>Menu</nav>
```

Maintain Proper Code Indentation and Formatting

Poor Formatting

```
<section><article><h2>HTML5</h2><p>Topic</
p></article></section>
```

Good Formatting

```
<section>
    <article>
        <h2>HTML5</h2>
        <p>Topic</p>
    </article>
</section>
```

Use Meaningful and Consistent Naming

- IDs and Names Should Be:
  - Descriptive
  - Lowercase
  - Consistent

```
<form id="student-form">
    <input type="text" name="full_name">
</form>
```

Avoid: <input id="x1">

Optimize Images and Multimedia

Use alt Attribute for Images

<img src="college.jpg" alt="College building">

Why:

- Improves accessibility
- Helps search engines
- Displays text if image fails to load

Optimize Images and Multimedia

Use Correct Multimedia Tags

```
<video controls>
    <source src="lecture.mp4" type="video/mp4">
</video>
```

- Avoid outdated or unsupported elements.

Form Best Practices

Always Use <label>

<label for="email">Email:</label>

<input type="email" id="email">

Use Appropriate Input Types

<input type="email">

<input type="date">

Use Built-in Validation

<input type="text" required>

Accessibility Best Practices

Use Language Attribute

<html lang="en">

Use Headings in Order

<h1>Main Title</h1>

<h2>Sub Title</h2>

<h3>Section</h3>

Performance Optimization (HTML Level)

Minimize Unnecessary Elements

- Avoid excessive usage
- Keep structure simple

Avoid Inline JavaScript Where Possible

- HTML should focus on structure, not logic.

Mobile and Responsive Best Practices

Always Use Viewport Meta Tag

<meta name="viewport" content="width=device-width, initial-scale=1.0">

Use Mobile-Friendly Input Types

<input type="tel">

<input type="email">

Any Questions?