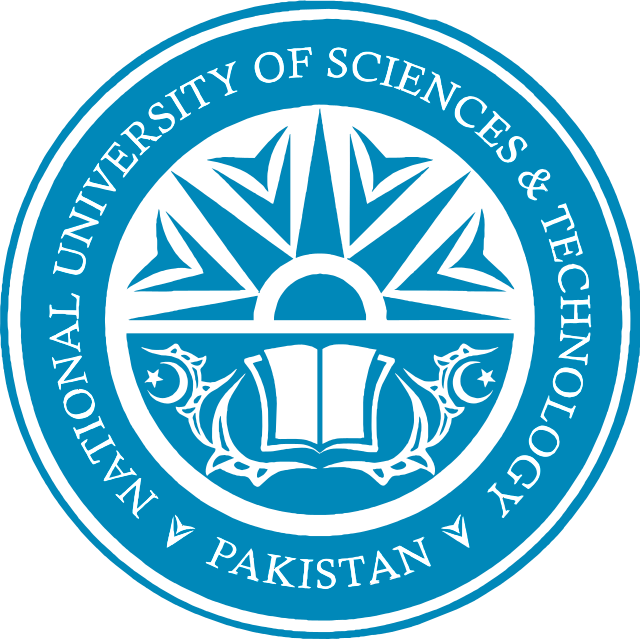


Lab # 02

**Web Engineering Fall 2020**

****

|  |  |
| --- | --- |
| Instructor | Sir Qasie |
| Student Name | Muhammad Anas |
| CMSID | 410555 |
| Department | Computer Science |
| Semester | 7th |

**Lesson Set 2**

**Advanced Hyper Text Markup Language**

|  |  |  |
| --- | --- | --- |
| **Purpose** | 1. | To get basic awareness of advanced tags |
|  | 2. | To understand more tags and why we are using them. |
|  | 3. | To create simple pages and use the tags. |
| **Procedure** | 1. | Students should read the Pre-lab Reading assignment before coming to lab. |
|  | 2.  3. | Students should complete the Pre-lab Writing assignment before coming to lab.  In the lab, students should complete Labs 2.1 through 2.4 in sequence. Your |
|  | 4. | instructor will give further instructions as to grading and completion of the lab. Students should complete the set of lab tasks before the next lab and get |
|  |  | them checked by their lab instructor. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Contents** | **Pre-requisites** | **Completion Time** | **Page Number** |
| Pre-lab Reading Assignment | - | 20 min | 3 |
| Pre-lab Writing Assignment | Pre-lab Reading | 10 min | 4 |
| **Lab 2** | | | |
| **Lab 2.1**  Installing VS Code | Pre-lab reading | 30 min | 5 |
| **Lab 2.2**  Lab Tasks | Awareness with VS code | - | 9 |

|  |  |
| --- | --- |
| **PRE-LAB READING ASSIGNMENT** | |
| **Advance HTML** | Advanced HTML" is a term that can refer to a number of different concepts and techniques beyond the basics of HTML. And HTML5 is the latest version of HTML and includes a number of advanced features such as video and audio elements, canvas elements for graphics, semantic tags like <header>, <footer>, and <nav>, and more. Knowing how to use these features can be considered advanced HTML. |
| **Time Tag** | <time> tag: The <time> tag is used to define a date and/or time. It is a semantic tag that helps search engines and assistive technologies understand the meaning of the content. Here's an example:  *<p>The first moon landing occurred on <time datetime="1969-07- 20T20:17:40Z">July 20, 1969</time>.</p>*  In this example, the <time> tag is used to indicate the date and time of the first moon landing. The datetime attribute is used to specify the date and time in ISO 8601 format. |
| **Mark Tag** | <mark> tag: The <mark> tag is used to highlight text. It is often used to indicate search terms or important information. Here's an example:  *<p>The most important rule of web development is to always <mark>test your code thoroughly</mark>.</p>*  In this example, the <mark> tag is used to highlight the phrase "test your code thoroughly" to draw attention to its importance |
| **Conditional Comments** | Conditional comments in HTML are a way to provide specific instructions or content to certain versions of Internet Explorer. This is often used to work around bugs or compatibility issues with older versions of the browser.  Here is an example of how to use a conditional comment in HTML:  *<!--[if IE]>*  *<p>This content is only visible in Internet Explorer.</p>*  *<endif]-->*  In this example, the content inside the conditional comment will only be displayed in Internet Explorer. The comment starts with *<!--[if IE]>* and ends with *<![endif]-->.* The if IE statement is a conditional expression that checks if the browser is Internet Explorer. You can also use other conditional expressions to target specific versions of Internet Explorer, such as *if lt IE 9* to target versions older than Internet Explorer 9.  You can put any HTML content inside the conditional comment tags, such as specific stylesheets or scripts, or even different versions of the |

|  |  |
| --- | --- |
|  | page content itself. However, note that conditional comments are no longer supported in newer versions of Internet Explorer, and should be used with caution. It's often better to use modern web development techniques like feature detection and graceful degradation to ensure compatibility across a wider range of browsers. |
| **Header Tag** | <header> tag: The <header> tag is used to define the header section of a document or a section of a web page. Here's an example:  *<header>*  *<h1>Welcome to My Website</h1>*  *<nav>*  *<ul>*  *<li><a href="#">Home</a></li>*  *<li><a href="#">About</a></li>*  *<li><a href="#">Contact</a></li>*  *</ul>*  *</nav>*  *</header>*  In this example, the <header> tag is used to contain the title of the website and a navigation menu. The <nav> tag is used to define the navigation section and contains an unordered list of links. |
| **Footer Tag** | <footer> tag: The <footer> tag is used to define the footer section of a document or a section of a web page. Here's an example:  *<footer>*  *<p>&copy; 2023 My Company. All rights reserved.</p>*  *</footer>*  In this example, the <footer> tag is used to contain a copyright notice. |
| **Navigation Tag** | <nav> tag: The <nav> tag is used to define a section of a document that contains navigation links. Here's an example:  *<nav>*  *<ul>*  *<li><a href="#">Home</a></li>*  *<li><a href="#">About</a></li>*  *<li><a href="#">Contact</a></li>*  *</ul>*  *</nav>*  In this example, the <nav> tag is used to contain an unordered list of navigation links. |

|  |  |
| --- | --- |
| **Accessible Links** | Accessible links in HTML are important for users with disabilities who rely on assistive technology to navigate websites. Here is an example of how to create an accessible link in HTML:  *<a href="#" title="Learn more about accessibility">Accessibility Information</a>*  To make the link even more accessible, you can also include a tabindex attribute to control the order in which links are navigated with the keyboard:  *<a href="#" title="Learn more about accessibility" tabindex="0">Accessibility Information</a>*  In this example, the *tabindex* attribute is set to "0", which means that the link will be included in the normal tab order of the page. This can be useful for users who navigate with the keyboard rather than a mouse.  Finally, you can also include text that describes the purpose of the link:  *<a href="#" title="Learn more about accessibility" tabindex="0">*  *<span aria-hidden="true">Accessibility Information</span>*  *<span class="sr-only">Learn more about accessibility</span>*  *</a>*  In this example, two <span> tags are used to provide text that is visible to sighted users and text that is accessible to screen reader users. The first <span> tag contains the visible text, and the aria-hidden attribute is set to "true" to indicate that this text should be hidden from screen readers. The second <span> tag contains the text that should be read by screen readers, and has a class of "sr-only" to visually hide the text from sighted users. This technique is often used to provide accessible text for links that have images or icons as their visible content. |
| **Accessible Form** | Accessible forms are important for users with disabilities who rely on assistive technology to interact with web content. Here is an example  of how to create an accessible form in HTML: |

|  |  |
| --- | --- |
|  | *<form>*  *<label for="name">Name:</label>*  *<input type="text" id="name" name="name" aria-required="true" required>*  *<label for="email">Email:</label>*  *<input type="email" id="email" name="email" aria-required="true" required>*  *<label for="message">Message:</label>*  *<textarea id="message" name="message" aria-required="true" required></textarea>*  *<button type="submit">Submit</button>*  *</form>*  In this example, the <form> tag is used to contain a set of form controls. Each form control has a <label> tag associated with it using the for attribute. For attribute value must match the id attribute of the form control it is associated with. This helps screen reader users understand the relationship between the label and the form control. Each form control also has an aria-required attribute and a required attribute. The aria-required attribute tells screen reader users that the form control is required, and the required attribute tells the browser to prevent form submission if the user hasn't provided a value for the required field.  Finally, the <button> tag is used to create a submit button for the form. The text on the button should be clear and descriptive and should indicate what action will be taken when the button is clicked.  By following these guidelines, you can create accessible forms that are usable by a wide range of users, including those with disabilities. |
| **Video Tag** | To embed a video on a web page, use the <video> element:  *<video src="video.mp4" controls>*  *Your browser does not support the video tag.*  *</video>*  In this example, the src attribute specifies the URL of the video file, and the controls attribute adds a set of playback controls to the video player. If the browser does not support the <video> element, the text between the opening and closing tags will be displayed as a fallback message. |
| **Audio Tag** | To embed an audio file on a web page, use the <audio> element: |

|  |  |
| --- | --- |
|  | *<audio src="audio.mp3" controls>*  *Your browser does not support the audio tag.*  *</audio>*  In this example, the src attribute specifies the URL of the audio file, and the controls attribute adds a set of playback controls to the audio player. If the browser does not support the <audio> element, the text between the opening and closing tags will be displayed as a fallback message. |
| **Canvas Tag** | The <canvas> element can be used to draw graphics and animations on a web page using JavaScript. Here is an example:  *<canvas id="myCanvas" width="200" height="200"></canvas>*  *<script>*  *var canvas = document.getElementById("myCanvas"); var ctx = canvas.getContext("2d");*  *ctx.fillStyle = "red"; ctx.fillRect(0, 0, 200, 200);*  *</script>*  In this example, the <canvas> element creates a canvas with a width and height of 200 pixels. The <script> element contains JavaScript code that uses the canvas context (ctx) to draw a red rectangle on the  canvas. |

|  |  |
| --- | --- |
| **PRELAB WRITING ASSIGNMENT** | |
| **Fill in the blanks** | 1. The <mark> element is used to highlight **text** on a web page. 2. **Conditional Comments** can be used to include or exclude code based on the browser being used to view the web page. 3. The <header>, <footer>, and <nav> elements are examples of   **Semantic** HTML elements that help organize web page content.   1. The <a> element is used to create **hyperlink** to other web pages or resources. 2. The <canvas> and <video> elements are examples of   **Multimedia** HTML elements that can add interactivity and rich media content to web pages. |

|  |  |
| --- | --- |
| **Lab 2.2** | **Lab Tasks** |

1. Open a VS Code, Sublime Text, or other IDE.
2. Create a new file and save it with yourName.html extension (e.g. Moiz.html).
3. Begin by creating the basic HTML structure with the <html>, <head>, and <body> tags.
4. In the <head> section, add a <title> tag with the title of your web page.
5. Inside the <body> section, create a <header> tag to hold the top portion of your page, including your website logo or name, navigation menu, and any other introductory information.
6. Next, add a <nav> tag to hold the links to the various pages of your website.
7. Create the main content of your page using headings, paragraphs, and other HTML elements as necessary.
8. Use the <mark> tag to highlight important text.
9. Include a <time> tag to display a date or time stamp.
10. Add a <canvas> tag to create a graphics area where you can draw graphics or animations.
11. Use <audio> and <video> tags to add multimedia elements to your page.
12. Include a <form> tag to create a form that visitors can use to submit information.
13. Use conditional comments to target specific versions of Internet Explorer, if necessary.
14. Finally, create a <footer> tag to hold the bottom portion of your page, including copyright information, links to social media accounts, or any other relevant information.

![A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect. A screen shot of a computer code

AI-generated content may be incorrect.