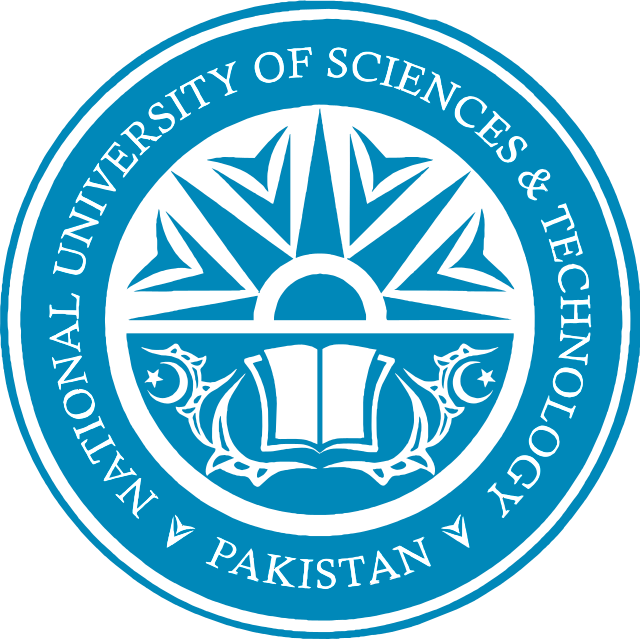


Lab Manual

# Web Engineering Fall 2020

****

|  |  |
| --- | --- |
| Instructor | Sir Qasir |
| Student Name | M Anas Hassaan |
| CMSID | 410555 |
| Department | Computer Science |
| Semester | 7th |

**Lesson Set 1**

**Introduction to VS Code And HTML**

|  |  |  |
| --- | --- | --- |
| **Purpose** | 1. | To get a basic awareness of VS Code |
|  | 2. | To understand HTML and why we are using it. |
|  | 3. | To learn the basics of an editor and create simple pages. |
|  | 4. | File Handling |
| **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 1.1 through 1.4 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 1** | | | |
| **Lab 1.1**  Installing VS Code | Pre-lab reading | 30 min | 5 |
| **Lab 1.2**  Lab Tasks | Awareness with VS code | - | 9 |

|  |  |
| --- | --- |
| **PRE-LAB READING ASSIGNMENT** | |
| **What Is HTML** | HTML is the standard markup language for creating Web pages.   * HTML stands for Hyper Text Markup Language * HTML describes the structure of Web pages using markup * HTML elements are the building blocks of HTML pages * HTML elements are represented by tags * HTML tags label pieces of content such as "heading", "paragraph", "table",... * Browsers do not display the HTML tags, but use them to render the content of the page   **An Example**  A paragraph is your content, Putting your content into an HTML tag to make it look like a paragraph is a structure  *<p>This is the content</p>* |
| **HTML Elements and Tags** | * An element is an individual component of HTML. * paragraph, image, header, ... * an element name indicates its purpose: p for paragraph A tag marks the beginning and the end of an HTML element Opening tag and Closing Tag   *<tagname>Stuff in the middle</tagname>*  HTML Tag Breakdown: *<p>……</p>* |
| **Anatomy of an HTML element** | **Container Element**   * An element that can contain other elements or content. * A paragraph *(<p>content</p>)* contains text. * Stand Alone Element * An element that cannot contain anything else *<br/>, <img/>*   **Attribute**   * Provides additional information about the HTML element * Class, ID, language, style, identity, source * Placed inside an opening tag, before the right angle bracket. * Value * Value is the value assigned to a given attribute. * Values must be contained inside quotation marks.   *<img src="my\_picture.jpg" />* |

|  |  |
| --- | --- |
| **HTML Page Structure** | **<!DOCTYPE>**  The <!DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly. It must only appear once, at the top of the page (before any HTML tags).  The *<!DoCtYpE>* declaration is not case sensitive.  The *<!DOCTYPE>* declaration for HTML5 is *<!DOCTYPE html>*  **<html>,<head>,<body>**  After <!doctype>, the page content must be contained between <html> tags. The head contains the title of the page & meta information about the page. Meta information is not visible to the user, but has many purposes, like providing information to search engines.  The body contains the actual content of the page. Everything that is contained in the body is visible to the user. (Some Exceptions!) |
| **Let's create our first HTML page** | open a new file in your text editor and copy this code in it:  *<!DOCTYPE html>*  save the document as  *<html>* myfirstpage.html and open it with  *<head>* a browser  *<title>Title of the page </title>*  *</head>*  *<body>*  *The page content here.*  *</body>*  *</html>* |
| **Nesting** | All elements "nest" inside one another Nesting is what happens when you put tags inside other containing tags.  **For example**, you would put the *<p>* inside of the *<body>* tags. The *<p>*  is now nested inside the *<body>*  *<body>*  *<p>The paragraph goes here.</p>*  *</body>*  Whichever element OPENS first CLOSES last |

|  |  |
| --- | --- |
| **Headings** | HTML headings are defined with the <h1> to <h6> tags. <h1> defines the most important heading. <h6> defines the least important heading.  *<h1>This is heading 1</h1>*  **This is heading 1**  **This is heading 2**  **This is heading 3 This is heading 4 This is heading 5**  **This is heading 6** |
| **Paragraphs** | HTML paragraphs are defined with the <p> tag:  *<p>This is a paragraph</p> <p>Paragraph 2</p>* |
| **The Poem Problem and Line Breaks** | This poem will be displayed in a single line:  *<p>*  *My Bonnie lies over the ocean.* The browser will remove any *My Bonnie lies over the sea.* extra spaces and extra lines *My Bonnie lies over the ocean.* when the page is displayed  *Oh, bring back my Bonnie to me.*  *</p>*  *<p>*  *My Bonnie lies over the ocean. <br>* The <br> element defines a line  *My Bonnie lies over the sea. <br>* break without starting a new  *My Bonnie lies over the ocean. <br>* paragraph.  *Oh, bring back my Bonnie to me.*  *</p>*  **Preformatted Text** |

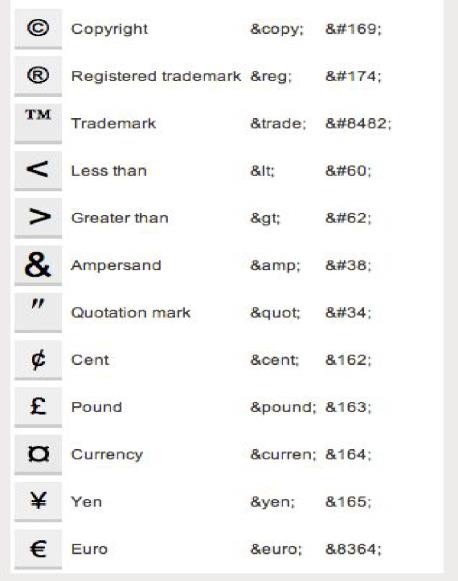
|  |  |
| --- | --- |
| This will be shown correctly with line breaks preserved. The HTML <pre> element defines preformatted text. The text inside a <pre> element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks. | *<pre>*  *My Bonnie lies over the ocean. My Bonnie lies over the sea.*  *My Bonnie lies over the ocean. Oh, bring back my Bonnie to me.*  *</pre>* |

# Formatted Text

|  |  |
| --- | --- |
| *<p>*  *Here is a paragraph with*  *<em>emphasized</em> text and*  *<strong>important</strong> text.*  *</p>*  *<p>*  *Here is another paragraph with*  *<i>Italic</i> text and <b>Bold</b> text.*  *</p>* | Here is a paragraph with *Emphasized* text and **Important** text.  Here is another paragraph with  *Italic* text and **Bold** text. |

|  |  |
| --- | --- |
| *<p>Here is a*  *<mark>highlighted</mark> text and this is <small>smaller</small> text.</p>* | Here is a Highlighted text and this is smaller text. |
| *<p>Here is <del>some deleted</del> text and this is*  *<ins>some inserted</ins> text.</p>* | Here is some deleted text and this is some inserted text |
| *<p>This is*  *<sup>superscripted</sup> text and this is <sub>subscripted</sub> text.</p>* | Here is superscripted text and this is  subscripted text |

**Character Codes**

**Many mathematical, technical, and currency symbols, are not present on a normal keyboard. To add such symbols to an HTML page, you can use an HTML entity name or code. One character that I use most is *&nbsp;*

More**:** [https://html.spec.whatwg.org/mul](https://html.spec.whatwg.org/multipage/syntax.html#named-character-references) [tipage/syntax.html#named-](https://html.spec.whatwg.org/multipage/syntax.html#named-character-references) [character-references](https://html.spec.whatwg.org/multipage/syntax.html#named-character-references)

# Links

HTML links are defined with the <a> tag The link's destination is specified in the href attribute. target attribute specifies where to open the linked document(\_self,\_blank,...)

*<a href="*[*http://www.google.com"*](http://www.google.com/) *target="\_blank">This is a link to Google!</a>*

[This is a link to Google!](https://www.google.com/)

# Images

HTML images are defined with the <img> tag. The source file (src), alternative text (alt), width, and height are provided as attributes.

*<img src="fsu-logo.jpg" alt="FSU Logo" width="104" height="142">*

|  |  |
| --- | --- |
| **Relative vs. Absolute paths for links & Images** | Relative paths change depending on the page the link is on   * Links within the same directory need no path information. "filename.jpg" * Sub directories are listed without preceding slashes. "img/filename.jpg" Absolute paths refer to a specific location of a file, including the domain * Typically used when pointing to a link that is not within your own domain. * [http://one.fsu.edu/alumni/image/community/clubs/FSU-Seal-](http://one.fsu.edu/alumni/image/community/clubs/FSU-Seal-full-color.jpg) [full-color.jpg](http://one.fsu.edu/alumni/image/community/clubs/FSU-Seal-full-color.jpg)   **Lists** *<ul>*  Unordered list (bullets)   * + List Item *<li>List Item</li>*   + Another List Item *<li>Another List Item</li>*   *</ul>*  Ordered list (sequence)   1. List Item *<ol>* 2. Another List Item *<li>List Item</li>*   *<li>Another List Item</li>*  *</ol>*  Lists are used everywhere and can be customized to look as you want.    **Tables**  Tables are a way to represent complex information in a grid format. Tables are made up of rows and columns. |

*<table>*

*<tr>*

*<th>Head</th>*

*<th>Head</th>*

*</tr>*

*<tr>*

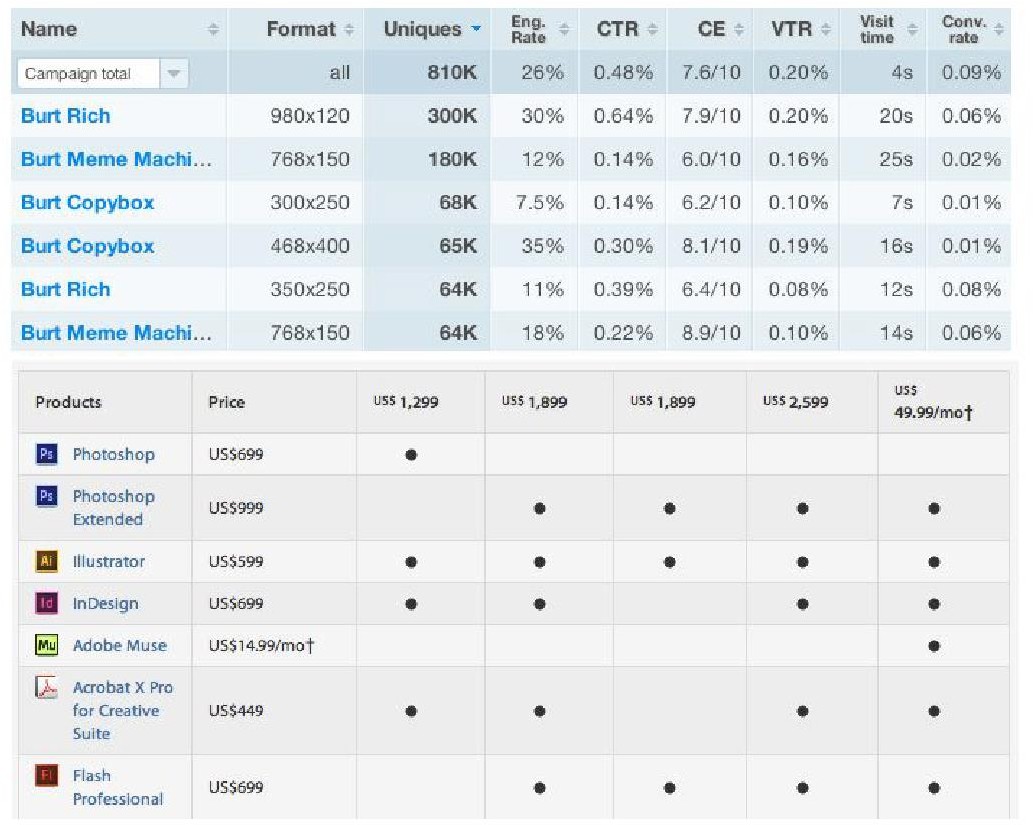
*<td>Data</td>*

*<td>Data</td>*

*</tr>*

*</table>*

Tables can be styled with CSS to add zebra striping or to highlight important rows/columns. Extra functionality can be added to tables like filtering or sorting rows and columns.



# Comments

You can add comments to your code that will not be seen by the browser, but only visible when viewing the code. Comments can be used to:

* organize your code into sections.
* 'comment out' large chunks of code to hide it from the browser.

*<!-- Beginning of header -->*

*<div id="header">Header Content </div>*

*<!-- End of header -->*

|  |  |
| --- | --- |
| **PRELAB WRITING ASSIGNMENT** | |
| **Fill in the blanks** | 1. **HTML** stands for **Hypertext Markup Language**. 2. The basic structure of an HTML document consists of **<head>** and **<body>**. 3. The **<a>** tag is used to define a hyperlink in HTML. 4. The **<img>** tag is used to add an image to an HTML document. 5. The **<table>** tag is used to define a table in HTML. |

|  |  |
| --- | --- |
| **Lab 1** |  |
| **Lab 1.1**  **Installing VS Code** | Visual Studio Code (a.k.a.VS Code) is a lightweight developer tool (or code editor).   1. Download and run Visual Studio Code installer from [http://code.visualstudio.com/Download.](http://code.visualstudio.com/Download) 2. In the following window, check    * “Add to PATH (available after restart)” and    * “Run Code after installation”.    * You can decide on the rest of the options according to your preferences. If Visual Studio Code is added to the PATH, you can type > code folder\_name (don’t type the > it represents the prompt) in the Command Prompt to launch Visual Studio Code.      1. To maximize your web development productivity, you should install Mohamed Abusaid’s HTML Snippets extension. You install the extension by launching VS Code, opening up the Command Palette (Control + Shift + P) and entering ext install html-snippets in it, then pressing ENTER. 2. This will give you access to rich language support for HTML markup in VS Code including full HTML tags, colorization, snippets, quick info, etc. 4. To benefit from the HTML Snippets extension, type part of a snippet, press ENTER, and   see the snippet unfold. Snippets are named as tag names, without braces. |

|  |  |
| --- | --- |
| **Lab 1.2** | **Lab Tasks** |

1. Create a webpage that includes an image of your favorite animal, a hyperlink to a website that provides information about that animal, and a list of interesting facts about the animal.

A grey cat with yellow eyes

AI-generated content may be incorrect.

<!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>

    <h1>Lab 01</h1>

    <h2>Favorite Animal</h2>

    <p>My favorite animal is the cat.</p>

    <img src="src/cat.jpeg" alt="A cute cat" width="300">

    <p>For more details visit <a href="https://en.wikipedia.org/wiki/Cat">details about cat</a>.</p>

</body>

</html>

1. Build a webpage that showcases your favorite recipes. Use a table to organize the recipes by meal type (breakfast, lunch, dinner, dessert) and include an image of each dish. Use headers to indicate the recipe names and paragraph tags to provide instructions and ingredient lists.

A screenshot of a computer

AI-generated content may be incorrect.

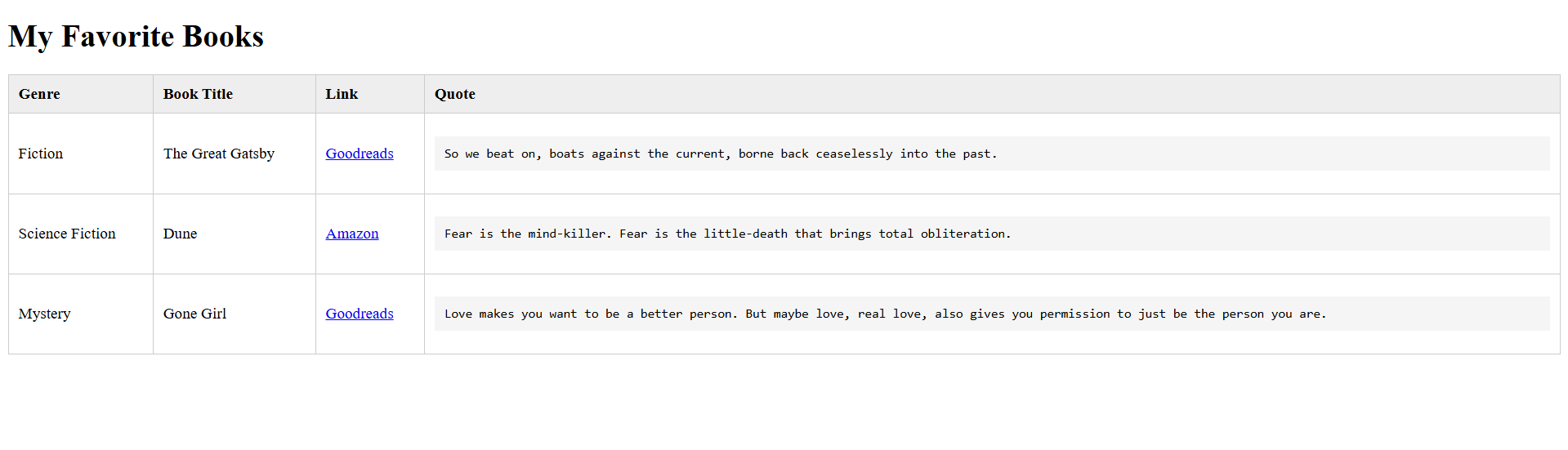
A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

1. Develop a webpage that includes a list of your favorite books. Use hyperlinks to link to the book's page on Amazon or Goodreads and use a table to organize the books by genre. Include preformatted text to display quotes from each book.



A screen shot of a computer

AI-generated content may be incorrect.

A computer screen with text on it

AI-generated content may be incorrect.