Task-2

# HTML Task

1. What is HTML? Give basic structure of the HTML page.

* HTML stands for Hyper Text Markup Language
* HTML is the standard markup language for creating Web pages
* HTML describes the structure of a Web page
* HTML consists of a series of elements
* HTML elements tell the browser how to display the content
* HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

Basic Structure

</html>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <title>Document</title>

</head>

<body>

</body>

</html>

* The <!DOCTYPE html> declaration defines that this document is an HTML5 document
* The <html> element is the root element of an HTML page
* The <head> element contains meta information about the HTML page
* The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
* The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

1. Difference between inline and block level element.

Every HTML element has a default display value, depending on what type of element it is.

There are two display values: block and inline.

## Block-level Elements

A block-level element always starts on a new line.

A block-level element always takes up the full width available (stretches out to the left and right as far as it can).

A block level element has a top and a bottom margin, whereas an inline element does not.

Example: <ul>, <div>, <article>, <h1>…<h6>, <header>, <footer>, etc

## Inline Elements

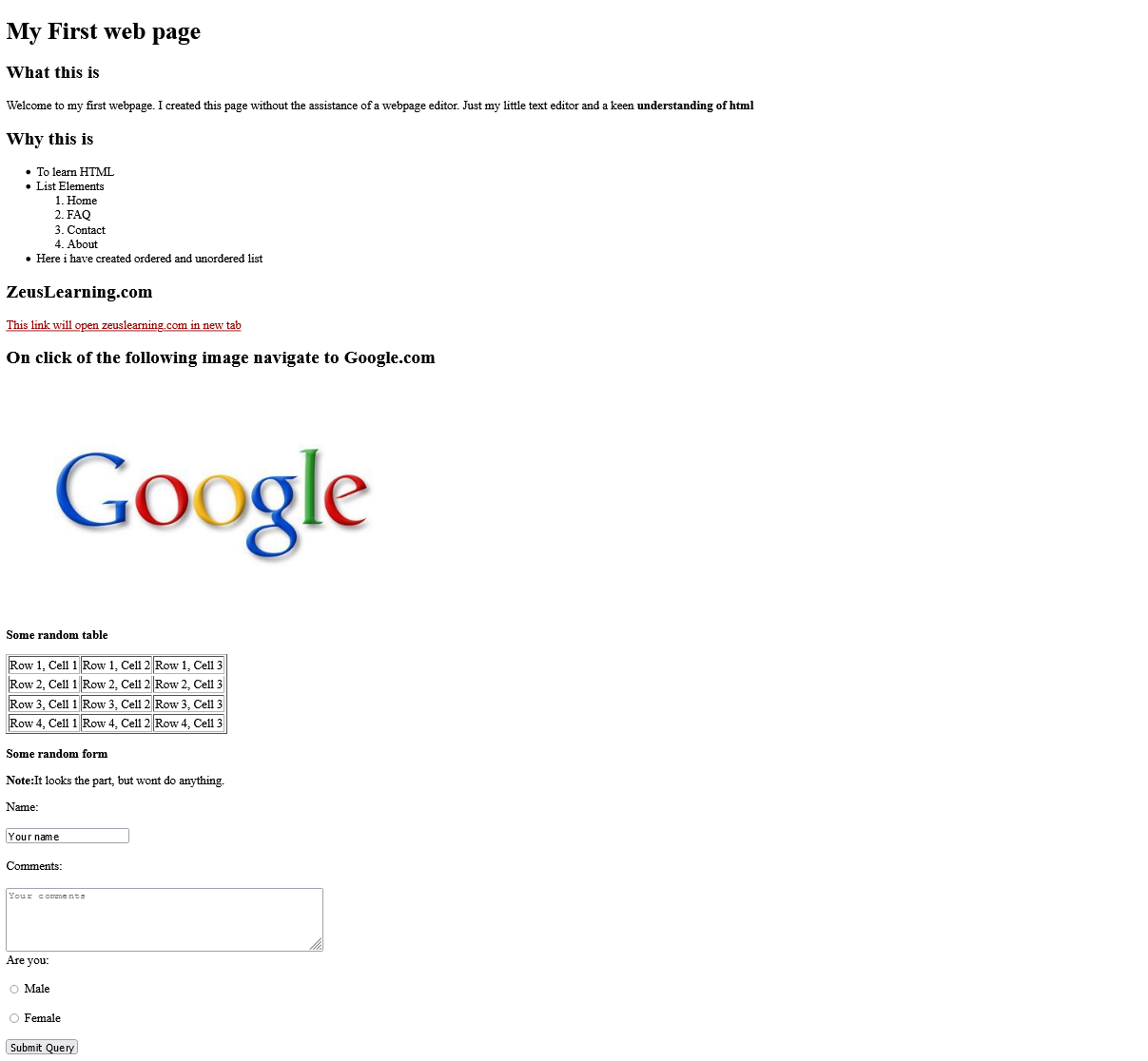
An inline element does not start on a new line.

An inline element only takes up as much width as necessary.

An inline element cannot contain a block-level element.

Example: <a>, <span >, <script >, <br >, <img>, etc

1. Create an HTML page “index.htm” with following content



# CSS Task

1. Explain the different ways in which CSS can be applied to HTML, what is the preferred way and why.

There are three ways of inserting a style sheet:

1. External CSS

* With an external style sheet, you can change the look of an entire website by changing just one file!
* Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

Example: The Following code to be written in external.css file.

body {

background-color: linen;

}

1. Internal CSS

* An internal style sheet may be used if one single HTML page has a unique style.
* The internal style is defined inside the <style> element, inside the head section.

Example:

<Head><style>

body {

background-color: linen;

}</style></Head>

1. Inline CSS

* An inline style may be used to apply a unique style for a single element.
* To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

Example: <h1 style="color:blue;text-align:center;">This is a heading</h1>

1. What are different CSS selectors, with example explain Element, Class and Id selectors.

CSS selectors are patterns used to select the element(s) you want to style.

**Basic Selectors**

* **Universal selector**

Selects all elements. Optionally, it may be restricted to a specific namespace or to all namespaces.

Syntax: \* ns|\* \*|\*

Example: \* will match all the elements of the document.

* **Type selector**

Selects all elements that have the given node name.

Syntax: elementname

Example: input will match any <input> element.

* **Class selector**

Selects all elements that have the given class attribute.

Syntax: .classname

Example: .index will match any element that has a class of "index".

* **ID selector**

Selects an element based on the value of its id attribute. There should be only one element with a given ID in a document.

Syntax: #idname

Example: #toc will match the element that has the ID "toc".

* **Attribute selector**

Selects all elements that have the given attribute.

Syntax: [attr] [attr=value] [attr~=value] [attr|=value] [attr^=value] [attr$=value] [attr\*=value]

Example: [autoplay] will match all elements that have the autoplay attribute set (to any value).

**Grouping Selectors**

* **Selector list**

The , is a grouping method, it selects all the matching nodes.

Syntax: A, B

Example: div, span will match both <span> and <div> elements.

**Combinators**

* **Descendant combinator**

The (space) combinator selects nodes that are descendants of the first element.

Syntax: A B

Example: div span will match all <span> elements that are inside a <div> element.

* **Child combinator**

The > combinator selects nodes that are direct children of the first element.

Syntax: A > B

Example: ul > li will match all <li> elements that are nested directly inside a <ul> element.

* **General sibling combinator**

The ~ combinator selects siblings. This means that the second element follows the first (though not necessarily immediately), and both share the same parent.

Syntax: A ~ B

Example: p ~ span will match all <span> elements that follow a <p>, immediately or not.

* **Adjacent sibling combinator**

The + combinator selects adjacent siblings. This means that the second element directly follows the first, and both share the same parent.

Syntax: A + B

Example: h2 + p will match all <p> elements that directly follow an <h2>.

**Pseudo**

* **Pseudo classes**

The : pseudo allow the selection of elements based on state information that is not contained in the document tree.

Example: a:visited will match all <a> elements that have been visited by the user.

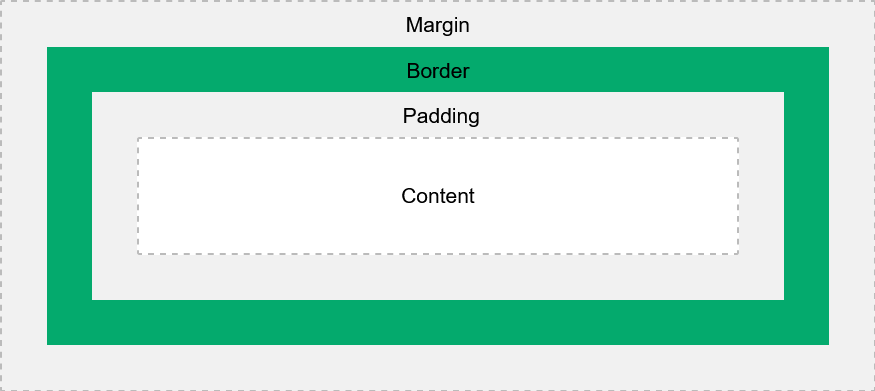
* **Pseudo elements**

The :: pseudo represent entities that are not included in HTML.

Example: p::first-line will match the first line of all <p> elements.

1. With the help of a diagram explain CSS Box Model.

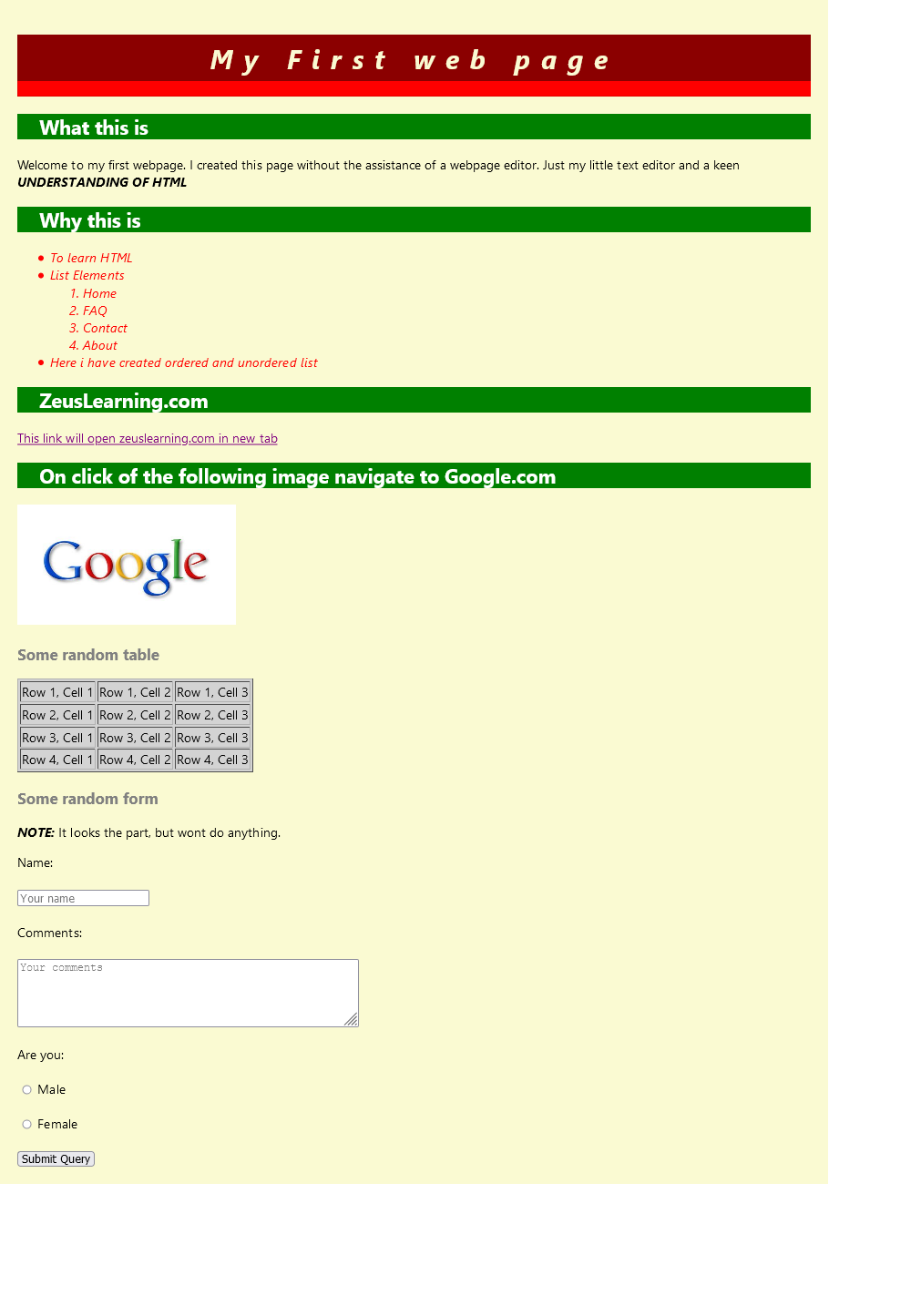
The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:



* Content - The content of the box, where text and images appear
* Padding - Clears an area around the content. The padding is transparent
* Border - A border that goes around the padding and content
* Margin - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

1. To HTML page create in the task 1.a, apply following CSS



# JavaScript Task

1. List down ways in which JavaScript command can be added to a webpage, what is the preferred way.

There are 2 Ways in which JavaScript can be added to a webpage which are as follow:

1. The <Script> Tag in Head or Body
2. The <Script> Tag in Head linking to external JavaScript File.

External JavaScript ways is the preferred way as it separates the concerns and makes it easier to read. Also, JS files can be cached to increase the speed.

External scripts are practical when the same code is used in many different web pages.

1. To the webpage created in the task 2.d, on click of “Submit” button call JavaScript function to validate:
   1. Name and comments fields are not empty.
   2. User has selected one entry from male/female radio button
   3. In case user has not entered data for name and comment field or not selected anything from male/female radio button – show alert message “All fields are compulsory” and set focus to the first filed that is empty.