Day4 Assignments (13/10/2023)

1. Find the details below:

**a. What is differences between <head>, <h1> and <header> tags?**

Ans:

**<head> tag**

The head tag is used for holding Meta information, title, links, etc. and is not displayed on the page. The header tag is used within the body of the website and can be used multiple times if required, e.g. to determine the top of an article.

**<h1> tag**

**Heading Tag Hierarchy Example**

**<h1>heading 1 - Most important</h1>**

**<h2>heading 2 - Second most important</h2>**

**<h3>heading 3 - Third most important</h3>**

**<h4>heading 4 - Fourth most important</h4>**

**<h5>heading 5 - Fifth most important</h5>**

**<h6>heading 6 - Least important</h5>**

Headings are considered extra relevant by search engines, so it's a good idea to spend some time optimizing the headers on your page. This includes making sure the most relevant keywords appear in your headers. Headings are also important for your visitors, to help them quickly scan a page and see its structure. Typically headings arranged from the most to the least important, with the most important acting as the largest when visually displayed.

The <header> element represents a container for introductory content or a set of navigational links.

A <header> element typically contains:

* one or more heading elements (<h1> - <h6>)
* logo or icon
* authorship information

**b. How to validate the textbox valid email id in HTML4? How to perform email validation in JavaScript?**

[JavaScript is an open-source](https://www.simplilearn.com/tutorials/javascript-tutorial/introduction-to-javascript) programming language. It is designed for creating web-centric applications. It is lightweight and interpreted, which makes it much faster than other languages. JavaScript is used to create beautifully designed web applications.

Email validation in JavaScript is an important part of the user experience of a web application. Validation assists in inputting forms and makes sure that only valid information is passed to the server from the client-side of the application.

What is Validation?

Validation is commonly defined as the process of checking the values inputted by the user. It plays an important role in web applications and enhances the overall user experience. We can validate email, password, date, mobile numbers, and many other fields.

JavaScript is used to validate the form data on the client-side of a web application, and this speeds up the validation process because of faster data processing when compared with server-side validation.

Now that we know a bit about validation, let’s go ahead and implement email validation in JavaScript.

How to validate email using JavaScript?

Email validation is a critical part of validating an [HTML form.](https://www.simplilearn.com/tutorials/html-tutorial/html-form-tag) An email is a string or a subset of ASCII characters separated into two parts by @ symbol. The first part contains personal information while the other contains the domain name at which the email is registered.

The personal information part can contain the following ASCII characters:

Uppercase and lowercase letters (A-Z and a-z)

Numeric characters (0-9)

Special characters - ! # $ % & ' \* + - / = ? ^ \_ ` { | } ~

Period, dot, or full stop (.) with the condition that it cannot be the first or last letter of the email and cannot repeat one after another.

The domain name contains:

Letters

Digits

Hyphens

Dots

**c. What are cookies in web applications?**

A cookie is information that a website puts on a user's computer. Cookies store limited information from a web browser session on a given website that can then be retrieved in the future. They are also sometimes referred to as browser cookies, web cookies or internet cookies.

Cookies can be accessed by the browser user, the site a user is on or by a third party that might use the information for different purposes. Common use cases for cookies include session management, personalization and tracking.

Cookies first appeared in 1994 as part of the Netscape Navigator web browser. They helped the browser understand if a user had already visited a given website. Netscape developer Lou Montulli invented the initial cookie implementation. He was granted U.S. Patent No. 5,774,670A, with the description, "Persistent client state in a hypertext transfer protocol based client-server system."

**Types of cookies**

There are multiple types of cookies that run in modern web browsers. Different types of cookies have specific use cases to enable certain capabilities.

**HTTP cookies.** This is the overall category of computer cookies used with modern web browsers to enable specific capabilities. All the cookies in this list -- except for flash cookies -- are forms of HTTP cookies.

**Session cookies.** A session cookie is only persistent while the user is navigating or visiting a given website.

**Persistent cookies.** Also sometimes referred to as permanent cookies, these persist for a configurable length of time or until a certain date that is set by the web server.

**First-party cookies.** Also known as SameSite cookies, the cookie and information it contains is restricted to the same site on which it was set.

**Third-party cookies.** These cookies are not restricted to the initial site where the cookie was created. [Third-party cookies](https://www.techtarget.com/whatis/definition/third-party-cookie) enable entities other than the original site to access them for user tracking and personalization purposes.

**Zombie cookies.** This refers to a type of cookie that persists, even after the user attempts to delete it.

**Flash cookies.** These are not browser or HTTP cookies but, rather, a specific type of cookie that works with [Adobe Flash](https://www.techtarget.com/whatis/definition/Flash). With the decline in the use of Flash, these cookies are no longer widely used.

**Secure cookies.** These are first- and third-party cookies that can only be sent over encrypted HTTPS connections.

**2. Create a web page to find out total amount based on the unit price and qty using JavaScript.**

**3. How to handle multiple submit buttons in HTML Forms?**

Every HTML form deals with the server-side with an action attribute. The HTML action Attribute is used to specify where the form data is to be sent to the server after submission of the form. As the destination of our data is stored in the action attribute each and every button will lead us to the same location. To overcome this difficulty we have to use the [formaction attribute](https://www.geeksforgeeks.org/html-formaction-attribute/) of HTML input and buttons.

**Approach:**The formaction attribute is used to specify where to send the data of the form. After submission of the form, the formaction attribute is called. The form data is to be sent to the server after the submission of the form. It overrides the feature of the action attribute of a <form> element. We are going to  implement our problem using this ‘formaction’ attribute

Let’s learn the steps of performing multiple actions with multiple buttons in a single HTML form:

* Create a form with method ‘post’ and set the value of the action attribute to a default URL where you want to send the form data.
* Create the input fields inside the as per your concern.
* Create a button with type submit. This button will trigger the default action attribute of the form and send our input data to that URL.
* Create another button with type submit. Also add a ‘**formaction**‘ attribute to this button and give it the value of the secondary URL where you want to send the form-data when this button is clicked.
* The formaction attribute will override the action attribute of the form and send the data to your desired location.

**Syntax :**

<form action="/DEFAULT\_URL" method="post">

<!-- Input fields here -->

<!-- This button will post to the

/DEFAULT\_URL of the form-->

<button type="submit">BUTTON 1</button>

<!-- This button will post to the custom

URL of the formaction attribute-->

<button type="submit" formaction="/URL2">

BUTTON 2

</button>

</form>

**4. Can we submit the input fields that are submitted outside the form tag?**

Form fields do not necessarily have to appear in a <form> tag. You can put them anywhere in a page. Such fields cannot be submitted (only a form as a whole can), but when responding to input with JavaScript, we often do not want to submit our fields normally anyway.

**5. What is datalist tag in HTML5?**

The <datalist> tag is used to provide autocomplete feature in the HTML files. It can be used with an input tag so that users can easily fill the data in the forms using select the data.  
**Syntax:**

<datalist> ... </datalist>

<!DOCTYPE html>

<**html**>

<**body**>

    <**form** action="">

        <**label**>Your Cars Name: </**label**>

        <**input** list="cars">

        <!--datalist Tag starts here -->

        <**datalist** id="cars">

            <**option** value="BMW"/>

            <**option** value="Bentley"/>

            <**option** value="Mercedes"/>

            <**option** value="Audi"/>

            <**option** value="Volkswagen"/>

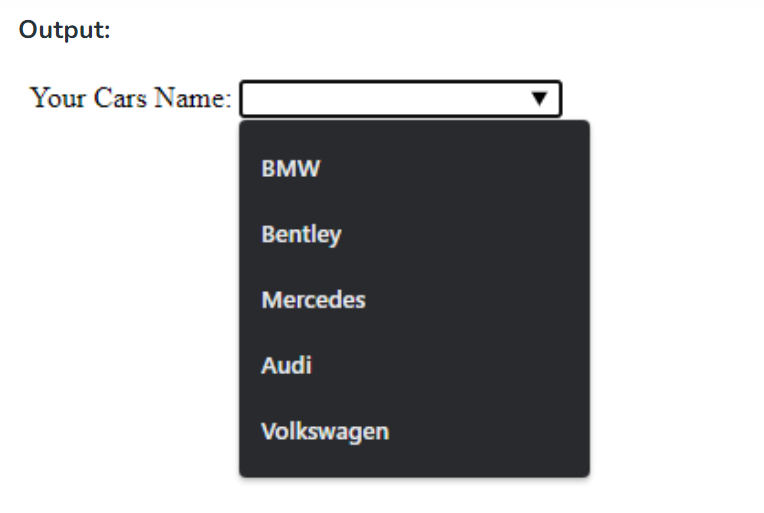
        </**datalist**>

        <!--datalist Tag ends here -->

    </**form**>

</**body**>

</**html**>



**6. Create a home page using HTML5 semantic tags: header, footer, nav, section and article.**

**a. Involve the mark to highlight some content**

**b. Use details and summary tag to show/hide some content.**