HTML LANGUAGE

MS18911336

ANUSHYA ARUMUGAMPILLAI

HTML History

- The first document Web created by Tim Berners-Lee published in 1991 with the name HTML Tags, was the hypertext system to share the documents.
- The last official HTML specification was published on December 24, 1999 and is named HTML 4.01. It's a revision and update of the HTML 4.0 version, so it doesn't include significant news.
- For this reason, in 2004, the companies Apple, Mozilla and Opera showed their concern about the lack of interest of the W3C in HTML and decided to organize in a new association called WHATWG (Web Hypertext Application Technology Working Group).
- The primary formal document with the outline of HTML was published in 1991 under the name HTML Tags (HTML tags) and may still be consulted online today as a computer relic.
- Although significant progress was made (at this time the labels for images, tables and forms were defined) none of the two standard proposals, called HTML and HTML +, managed to become the official standard.
- HTML 4.0 was published on April 24, 1998 (being a corrected version of the primary publication of December 18, 1997) and maybe a great leap from previous versions.
- As of 1996, the HTML standards are published by another standard organization called W3C (World Wide Web Consortium).
- The first official proposal to convert HTML into a standard was made in 1993 by the IETF (Internet Engineering Task Force).
- Since the publication of HTML 4.01, the standardization activity of HTML stopped and therefore the W3C focused on the development of the XHTML standard.
- The origin of HTML dates back to 1980, when the physicist Tim Berners-Lee, a worker at CERN (European Organization for Nuclear Research) proposed a new "hypertext" system for sharing documents.
- The HTML 3.2 version was published on January 14, 1997 and it was the first HTML recommendation published by the W3C.

HTML What Is the Origin of The Hypertext Markup Language

- HTML is the acronym of Hypertext Markup Language, they're mostly used on the Web, a language based on labels, which allows displaying Web pages.
- HTML is a standard Standardized Markup Language application, a system that permits you to define structured documents and markup languages to represent those same documents.
- The term HTML is used to refer both things, the type of document and the language of marks.
- HTML stands for Hypertext Markup Language or hypertext markup language, the most widely used language on the Web, it's a simple language based on tags, which allows to display the information of Web pages on the user's monitor through browsers.

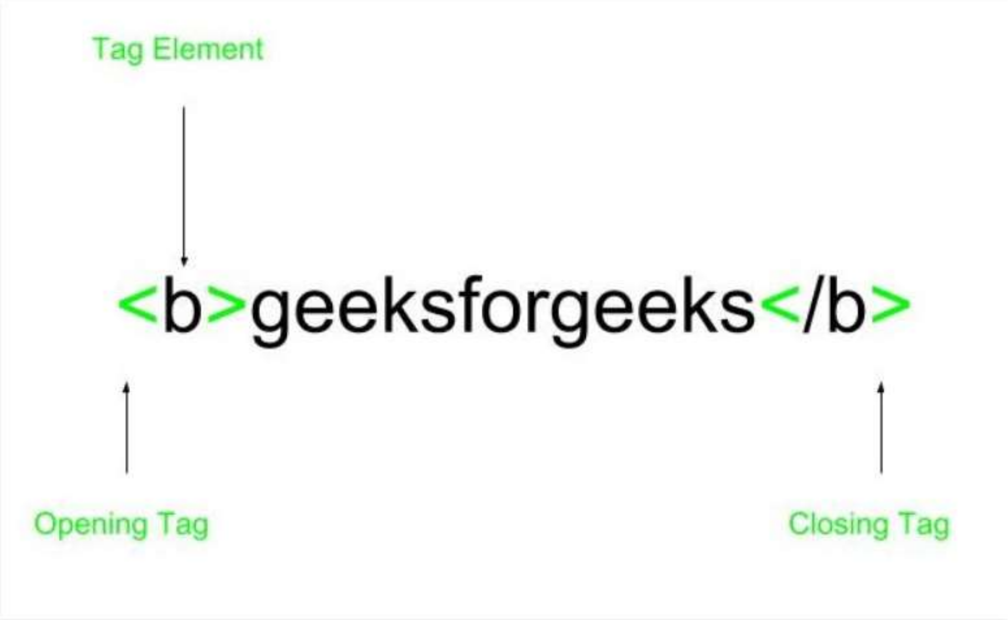
Introduction of HTML

HTML stands for Hyper Text Markup Language. This is used to design web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. Markup language is used to describe the text document within tag which defines the structure of web pages. This language is used to make notes for the computer so that a machine can understand it and manipulate text accordingly. Most of markup (e.g. HTML) languages are human readable. Language make use of tags to define what manipulation has to be done on the text.

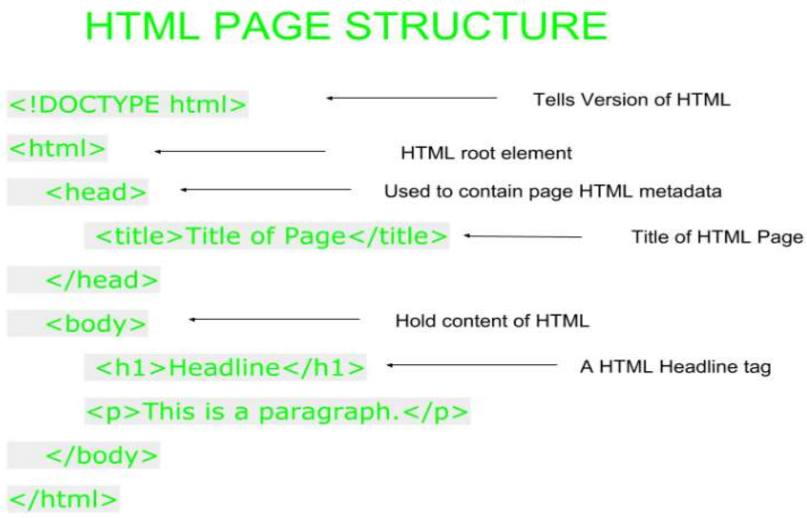
HTML is a markup language which is used by the browser to manipulate text, images and other content to be displayed it in required format. HTML was created by Tim Berners-Lee in 1991. The first version of HTML was HTML 1.0 but the first standard version which was published in 1999 was HTML 2.0.

HTML VERSION	YEAR
HTML 1.0	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML 5	2014

Elements and Tag: HTML uses predefined tags and elements that tells the browser about content display property. If a tag is not closed then the browser applies that effect till end of page.



HTML page structure: The Basic structure of the HTML page is given below. It contains some of the elements like head, title, body, … etc. The blocks of web pages are built by the use of these elements.



```
<!DOCTYPE html>
<html>
   <head>
      <title>Title of Page</title>
   </head>
   <body>
      <h1>Headline</h1>
      <p>This is a paragraph.</p>
   </body>
</html>
```

Advantages:

- HTML is used to build a website.
- It is supported by all browsers.
- This can be integrated with other languages like CSS, JavaScript etc.

Disadvantages:

- HTML can create only static webpages but for dynamic web page other languages have to be used.
- Large amount of code has to be written to create a simple web page.
- Security feature is not good.

The reason for this disapproval is that there other more efficient ways.

- However, a tag is deprecated does not mean you should not use it, much less. In fact, in the document, I speak of some of them very common today and in the future, it.
- Is important that you understand the scope of the word "disapproved" in this context.
- The tags not disappear disapproved or browsers will no longer recognize them in long time. The use of these tags does not represent, in any way, an imminent danger to your pages.
- It is simply a reference. Although it is important that you know thoroughly, and use properly, cascading style sheets.
- It will be also missing some of the attributes of many tags, in the first part. This is because those attributes aren't utilized in basic HTML, but in dynamic HTML and, more specifically, in scripting languages.
- There are some HTML tags, that are not standardized by the Consortium, that only work in a specific browser (Explorer, Firefox, etc.)
- To write a code in HTML, we will need a plain text editor, that is, that does not introduce unwanted characters in the code. For example, the Windows notebook (except, of course, that you use another operating system).
- Later on, when you are more familiar with the plain text editing, you can choose other more powerful and flexible editors, but at first, the notebook is very comfortable.
- HTML really is not a programming language, in the strict sense of the word, since by its very nature, it does not implement some basic functions of any programming language, such as, for instance, access to database, graphic editing functions, etc.

However, it is essential to enter the design of web pages.

- HTML isn't sensitive to the utilization of uppercase and lowercase letters. It will work an equivalent if we type the tags and their parameters (also called attributes) in uppercase, lowercase or combination of both. Of course, this rule has an exception.
- As a general rule, we will get used to writing the tags and their attributes (as well as their values) in lowercase. The reason for this is that, after studying HTML and DHTML, we can study PHP and this Yes requires the use of lowercase letters so we better get used to it.
- In any case we will mention about HTML because the language par excellence for the creation of web pages. Nowadays, all the pages that we discover within the network also implement other technologies, such as CSS, JavaScript, Jquery, AJAX, JSON, PHP, ASP, etc.
- However, absolutely all have a base of HTML on which, in turn, those other technologies are implemented.
- Therefore, if we would like to dedicate ourselves to the design and creation of web pages, either to create our own page or to work in the sector, a deep knowledge of HTML is essential.

HTML INJECTION

HTML injection is an attack that is almost like Cross-site Scripting (XSS). While within the XSS vulnerability the attacker can inject and execute Javascript code, the HTML injection attack only allows the injection of certain HTML tags. When the application doesn't properly control the user supplied data then the attacker can provide valid HTML code by providing via a parameter value, and inject their own content into the page. This attack is typically utilized in conjunction with some form of social engineering, because the attack is exploiting a code-based vulnerability and a user's trust.

HTML 5

- HTML5 is the HTML update, the language in which the web is created.
- HTML5 is also a marketing term for grouping new web application development technologies: HTML5, CSS3 and new Javascript capabilities.
- The previous and most used version of HTML, HTML4, lacks the necessary features for the creation of modern applications based on a browser.
- The strong use of Javascript has helped to improve this, thanks to frameworks such as jQuery, 1 jQuery UI2, Sproutcore3, among others.

What is HTML5?

- HTML5 was officially published in October 2014 by the World Wide Web Consortium (W3C) although adoption began earlier by forward thinking organizations and developers who wanted to utilize its futuristic functionalities.
- HTML5 (HyperText Markup Language 5) is the markup language used for structuring and presenting content on the World Wide Web. HTML5 is made up of three different kinds of code: HTML, which gives websites their structure; Cascading Style Sheets (CSS), which give websites their presentation attributes; and JavaScript, which powers a majority of the functions that we are used to on modern websites.
- HTML5 boasts support for the latest multimedia and it is easily readable by the humans while consistently understood by computers, browsers and parsers. HTML5 includes features that are designed for low-powered devices which makes it ideal for hybrid mobile applications.
- Setting HTML5 apart from Flash is the fact that HTML5 on its own cannot be used for animation or interactivity since it needs CSS3 or JavaScript to accomplish either.

Why was HTML5 created?

- While HTML4 was able to deliver the functionalities needed in the earlier stages of widespread internet adoption, as web applications progressed further and further in their functionalities, the demand increased for a markup language that was able to deliver more than simply the static pages that HTML4 was capable of.
- HTML4 that depends on the heavy use of plugins to provide website users with any functionalities which went beyond simple text and images, that something we now take for granted.
- The use of such a multitude of plugins can cause problems for websites which were viewed on multiple browsers and devices but still some functionalities may be lost if plugins weren't supported.
- HTML5 has introduced the following elements which have transformed websites into the rich, immersive experiences that they are today which allow once static websites to become application platforms.
- Special Flash has been used to replace HTML to develop web apps that exceed the abilities of a browser:
  - Audio
  - Video
  - Webcams
  - Microphones
  - Binary data
  - Vector animations
  - Complex interface components
  - Among many other things
- Now HTML5 is able to do this without the need of plugins and with a great compatibility between browsers.

New Tags HTML5

- HTML4 and HTML5 tags are 100% compatible between each other
- All the code you have in normal HTML will continue to work without problems in HTML5.
- To start using HTML5 all you have to do is place this DOCTYPE before the <html>: tag

Top 5 HTML5 Vulnerabilities Each Organization Must Know

1) PostMessage Vulnerabilities

As a robust HTML5 API, postMessage makes it easier for developers to facilitate cross-origin communication between web applications. The developers can use postMessage to make web pages communicate and exchange data regardless of their port, protocol, and hostnames. But developers often forget to implement Same-Origin Policy (SOP) while using postMessage. When the same origin policy is not implemented while using the HTML5 API, the web application becomes vulnerable to cross-site scripting attacks. The developers can easily prevent cross-site scripting attacks by validating the origin or source of message received by the application through the postMessage method. The developers even need to clearly specify the sender or receiver of the message exchange through the postMessage API.

2) CORS Vulnerabilities

When the developers do not implement a Same-Origin Policy, the HTML5 application becomes vulnerable to cross-origin resource sharing (CORS) issues. CORS keeps cross-domain requests secure by implementing three headers. But the developers must implement the headers properly to eliminate CORS vulnerabilities. They can easily prevent CORS attacks by setting value of the Access-Control-Allow-Credentials header as true. When the value of the header is set as true, the application will use and exchange cookies. The developers also need to use Access-Control-Allow-Origin header to specify the origin of web pages for making requests to the server and receiving responses from the web server.

3) Web Storage Vulnerabilities

The local or offline storage feature of HTML5 makes it easier for developers to store user data in a client-side database. They can even enhance the web application’s performance and user experience by storing the frequently accessed data in the client-side database. Despite improving user experience, the local storage feature makes the HTML5 application vulnerable to cross-site scripting (XSS) attacks. As the cyber-criminal can access the client-side database through JavaScript code, they can easily inject malicious code or scripts as input through text fields. While enabling the local Storage feature, the developer must validate each text field. Also, they need to ensure that sensitive user data are not stored in the client-side database as they can be accessed by other users. They can even consider not storing any data in the database which can be stored or transmitted as plain text.

4) WebSockets Vulnerabilities

Many web developers nowadays use WebSockets to facilitate enable interactive and persistent communication sessions between the web server and browsers. During HTML5 application development, developers use WebSockets to facilitate real-time communication between the server and the client. But the WebSockets protocol lacks built-in authorization and authentication mechanism. Hence, the bidirectional client-server communication facilitated by WebSockets makes the web application vulnerable to Cross-Site WebSocket Hijacking (CSWSH) vulnerability. There are always chances that a user may access multiple WebSockets-enabled web applications on the same browser. If the users accessed any malicious site, it can control the user session through WebSockets.

5) Middleware Vulnerabilities

HTML5 makes it easier for developers to build cross-platform web applications. But the technology relies on a middleware framework to communicate with the underlying operating system in its native language. The middleware has the capability to accept both data and script as input, and execute the scripts automatically. Hence, the middleware framework makes the web application vulnerable to malicious code injection and cross-site scripting (XSS) attacks. While developing a HTML5 application, the developers must explore ways to prevent the middleware from accepting and executing malicious scripts. They must keep the data and script separated, restrict permissions to untrusted code, and prevent the middleware from accepting input from insecure sources.

On the whole, HTML5, like other technologies, has its own shortcomings and vulnerabilities. These HTML5 vulnerabilities make it easier for cyber criminals to execute targeted malware attacks and access corporate data. An organization must conduct proper web application testing continuously in varying user environments to combat these HTML5 vulnerabilities. Also, it must ask developers to overcome HTML5 vulnerabilities by implementing security best practices and referring to HTML5 security cheat sheet.