# HTML – Interview Questions

## What is HTML?

* 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 vs HTML 5

1. **Introduction and Development**
   1. **HTML:**
      1. Introduced in 1991 by Tim Berners-Lee.
      2. The foundational version of web markup.
      3. Developed over various versions (HTML 2.0, HTML 3.2, HTML 4.01).
   2. **HTML5:**
      1. Announced in 2008 by the Web Hypertext Application Technology Working Group (WHATWG).
      2. Officially recommended by W3C in 2014.
      3. Built on the foundation of HTML but adds new features and capabilities.
2. **Syntax and Structure**
   1. **HTML:**
      1. Basic structure: Doctype declaration, head, and body.
      2. Limited in terms of semantic tags and attributes.
   2. **HTML5:**
      1. Simplified and more flexible doctype declaration: <!DOCTYPE html>.
      2. Enhanced semantic structure with new tags like <header>, <footer>, <article>, <section>, and <nav>.
      3. Introduced new attributes and forms elements for better data handling and input types.
3. **Multimedia Support**
   1. **HTML:**
      1. Limited support for multimedia.
      2. Required third-party plugins like Flash and Silverlight for audio and video.
   2. **HTML5:**
      1. Native support for multimedia elements: <audio>, <video>.
      2. Provides better control and integration of multimedia content.
4. **Graphics and Animations**
   1. **HTML:**
      1. Relied on external plugins and technologies for graphics and animations.
   2. **HTML5:**
      1. Introduced the <canvas> element for drawing graphics using JavaScript.
      2. Integrated support for scalable vector graphics (SVG).
5. **APIs and Functionality**
   1. **HTML:**
      1. Limited API support; relied on external libraries and plugins.
   2. **HTML5:** Provides new APIs and enhanced functionalities:
      1. Geolocation API for location-based services.
      2. Drag-and-drop API for better user interactions.
      3. Web Storage API (localStorage and sessionStorage) for client-side data storage.
      4. Web Workers for background processing without affecting the user interface.
      5. WebSockets for real-time communication.
6. **Browser Compatibility**
   1. **HTML:**
      1. Supported by all browsers but may lack modern features.
   2. **HTML5:**
      1. Widely supported by modern browsers (Chrome, Firefox, Safari, Edge).
      2. Some features might not be fully supported by older browsers, requiring polyfills or fallbacks.
7. **Performance and Mobile Support**
   1. **HTML:**
      1. Less optimized for modern web applications and mobile devices.
   2. **HTML5:**
      1. Improved performance and responsiveness.
      2. Better suited for mobile devices with enhanced touch support and responsive design features.
8. **SEO and Accessibility**
   1. **HTML:**
      1. Basic support for SEO and accessibility.
      2. Limited use of semantic tags affects search engine ranking and accessibility.
   2. **HTML5:**
      1. Enhanced semantic elements improve SEO.
      2. Better support for accessibility features, making content more accessible to screen readers and assistive technologies.

## HTML vs XHTML

### What is XHTML?

* XHTML stands for E**X**tensible **H**yper**T**ext **M**arkup **L**anguage
* XHTML is a stricter, more XML-based version of HTML
* XHTML is HTML defined as an XML application
* XHTML is supported by all major browsers

### Why XHTML?

XML is a markup language where all documents must be marked up correctly (be "well-formed").

XHTML was developed to make HTML more extensible and flexible to work with other data formats (such as XML). In addition, browsers ignore errors in HTML pages, and try to display the website even if it has some errors in the markup. So XHTML comes with a much stricter error handling.

## HTML vs XML

## Structure of a simple in HTML 5 Document

HTML:

|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">  <html>  <head>  <title>Hello, World!</title>  </head>  <body>  <h1>Hello, World!</h1>  </body>  </html> |

HTML 5:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Hello, World!</title>  </head>  <body>  <h1>Hello, World!</h1>  </body>  </html> |

## HTML Elements / Tags

An HTML element is defined by a start tag, some content, and an end tag.

Here is a list of void elements (self-closing tags) in HTML5:

1. <area> - Defines a clickable area within an image map.
2. <base> - Specifies a base URL for relative URLs in a document.
3. <br> - Inserts a line break.
4. <col> - Specifies column properties for an HTML table column group.
5. <command> (deprecated) - Defines a command that a user can invoke.
6. <embed> - Embeds external content (such as multimedia) into the document.
7. <hr> - Creates a thematic break or horizontal line.
8. <img> - Embeds an image.
9. <input> - Defines an input control (like text fields, checkboxes, etc.).
10. <keygen> (deprecated) - Generates a key pair for form submission.
11. <link> - Defines relationships between the current document and external resources (usually for linking stylesheets).
12. <meta> - Defines metadata about an HTML document.
13. <param> - Defines parameters for plugins (like <object>).
14. <source> - Specifies multiple media resources for elements like <audio> and <video>.
15. <track> - Provides text tracks for <video> and <audio> elements.
16. <wbr> - Suggests an optional line break opportunity within text.

## HTML Attributes

* All HTML elements can have attributes
* Attributes provide additional information about elements
* Attributes are always specified in the start tag
* Attributes usually come in name/value pairs like: name="value"

## HTML Head Section

* The ‘<head>’ section of an HTML document is used to define metadata and links to external resources.
* It is not displayed directly on the webpage but provides crucial information for browsers and search engines.

Following are tags used inside `<head>` tag:

### The `<meta>` tag

* The <meta> tag defines metadata about an HTML document. Metadata is data (information) about data.
* Metadata is used by browsers (how to display content or reload page), search engines (keywords), and other web services.

|  |
| --- |
| <head>  <meta charset="UTF-8">  <meta name="description" content="A brief description of the page">  <meta name="keywords" content="HTML, CSS, JavaScript">  <meta name="author" content="Author Name">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  </head> |

* Attributes:
  + `charset`:
    - **Value:** character\_set
    - **Description:** Specifies the character encoding for the HTML document
  + `content`:
    - **Value:** text
    - **Description:** Specifies the value associated with the `http-equiv` or `name` attribute
  + `name`: Specifies a name for the metadata
    - `application-name`: Specifies the name of the Web application that the page represents
    - `author`: Specifies the name of the author of the document.
    - `description`: Specifies a description of the page. Search engines can pick up this description to show with the results of searches.
    - `generator`: Specifies one of the software packages used to generate the document (not used on hand-authored pages).
    - `keywords`: Specifies a comma-separated list of keywords - relevant to the page (Informs search engines what the page is about).
    - `viewport`: Controls the viewport (the user's visible area of a web page).
  + `http-equiv`: Provides an HTTP header for the information/value of the content attribute
    - `content-security-policy`: Specifies a content policy for the document.
    - `content-type`: Specifies the character encoding for the document.
    - `default-style`: Specified the preferred style sheet to use.
    - `refresh`: Defines a time interval for the document to refresh itself.

### The `<link>` tag

* The `<link>` tag defines the relationship between the current document and an external resource.
* It is most often used to link to external style sheets or to add a favicon to website.
* It is an empty element and contains only attributes.

|  |
| --- |
| <head>  <link rel="stylesheet" href="styles.css">  </head> |

* Attributes:
  + `crossorigin`: Specifies how the element handles cross-origin requests
    - Values:
      * anonymous
      * use-credentials
  + `href`: Specifies the location of the linked document
    - Value: URL
  + `hreflang`: Specifies the language of the text in the linked document
    - *Value: language\_code*
  + `media`: Specifies on what device the linked document will be displayed
    - Value: *media\_query*
  + `referrerpolicy`: Specifies which referrer to use when fetching the resource
    - `no-referrer`:
    - `no-referrer-when-downgrade`:
    - `origin`:
    - `origin-when-cross-origin`:
    - `unsafe-url`:
  + `rel`:
  + `sizes`:
  + `title`:
  + `type`:

### The `<title>` tag

### The `<script>` tag

### The `<base>` tag

### The `<style>` tag

## HTML Headings

## HTML Paragraphs

## HTML Text Formatting

## HTML Quotation and Citation Elements

## HTML Comments

## HTML Colours

## HTML Style and CSS

## HTML Links

## HTML Images

## HTML Tables

## HTML Lists

## HTML Block & Inline

## HTML Division

## HTML Class

## HTML Id

## HTML iFrame

## HTML JavaScript

## HTML Layout

## HTML Responsive

## HTML Computer Code

## HTML Semantics

## HTML Input

## HTML Forms

## HTML Graphics

### HTML Canvas

### HTML SVG

## HTML Media

### HTML Media

### HTML Video

### HTML Audio

### HTML Plug-ins

### HTML YoutTube

## HTML APIs

### HTML Geolocation

### HTML Drag/Drop

### HTML Web Storage

### HTML Web Workers

### HTML SSE

# CSS – Interview Questions

# Tailwind– Interview Questions

# JavaScript – Interview Questions

# React.js – Interview Questions

# React Router DOM – Interview Questions

# Redux Tool Kit – Interview Questions

# React Native – Interview Questions

# Tan Stack – Interview Questions

# Node.js – Interview Questions

# Express.js – Interview Questions

# MongoDB – Interview Questions

# Mongoose.js – Interview Questions

# Prisma ORM – Interview Questions

# SQL – Interview Questions

# PostGreSQL– Interview Questions