ASSIGNMENT

[DATE: 17-01-19 HTML ASSIGNMENT](#_2wkrm0z9hsjz)

[What are the different charsets in HTML?](#_f7oxwnoeo7ud)

[What is the behaviour if self closing tags are used with end tags?(i.e)<img/> vs <img></img>](#_l9qxkrby8qr8)

[What are the different types of ordered list and unordered list in html?](#_pa8l5sfjtuxv)

[How to include an image of your choice as a bullet in unordered list?](#_xoorm3x9sf4i)

[How to design a table with a single border?](#_dwyvaz938ols)

[What is the use of rowspan and columnspan?](#_2gm6rv3kdwhi)

[How to restrict a particular file type in an image select?](#_auwraf7ktzp0)

[How to restrict input numbers or define a range in input element?](#_2gflyvnokam)

[How to include a youtube video in the document?](#_7ipp2lbc648s)

[Which tag is used to get the location of user?](#_9wx9egabgtoe)

[Why do we need multipart?](#_hfd29cvlgzm3)

[HTTP request and response format](#_uw05pu5eaw5t)

[Difference between http and https](#_oi4l2vc1a7gf)

[Use of AJAX](#_9s8j8fo9tyrl)

[How to identify if a browser supports JavaScript or not?](#_rayu5g97r45)

[How to identify if a browser supports HTML5 or not?](#_8s4n15z3wnud)

[Differences in usage of <strong> and <em> with examples.](#_r20jjyy569fb)

[What are the different coloring Standards in html?](#_5q104jo18iod)

[How to semantically include references in html?](#_2krmn2fyaadm)

[How to embed and open a pdf document in html?](#_7fru7eebxo0v)

[Difference between object and embed tag.](#_s4lppjtr82v8)

[DATE: 18-01-19 CSS ASSIGNMENT](#_4kz9a4ilshz2)

[What is the use of em?](#_o82onhjxjpc8)

[How to strikethrough a text in CSS?](#_lmkfnpyvjt3k)

[Divide a page into three vertical columns.](#_3uafkh60nb7o)

## 

## DATE: 17-01-19 HTML ASSIGNMENT

### What are the different charsets in HTML?

Different character sets in HTML are:

* **ASCII** was the first **character encoding standard** (also called character set).
* **ASCII defined 128** different alphanumeric characters that could be used on the internet: **numbers (0-9), English letters (A-Z), and some special characters like ! $ + - ( ) @ < > .**
* **ANSI** (Windows-1252) was the original Windows character set, with support for **256 different character codes.**
* **ISO-8859-1** was the **default character set for HTML 4** and supports **256 different character codes.**
* Because ANSI and ISO-8859-1 were so limited, HTML 4 also supported UTF-8.
* **UTF-8 (Unicode)** covers almost all of the characters and symbols and is **default character encoding for HTML5**.
* To display an HTML page correctly, a web browser must know the character set used in the page and this information is provided by <meta> tag.

**<meta charset = “UTF-8”>** is used in HTML5

### What is the behaviour if self closing tags are used with end tags?(i.e)<img/> vs <img></img>

<!DOCTYPE html>

<meta charset="UTF-8">

<html>

<body>

<h2>Sample</h2>

<img src="https://images.pexels.com/photos/112460/pexels-photo-112460.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500" alt="car" width="500" height="333"/>

</body>

</html>

Having or not having an end tag for a self closing tag is optional since it does not cause any change in the code.but closing the tag by having a / at the end of the opening tag is a good practice.

Eg.<img src=”path of the image”**/**>

Some of the self closing tags are:

<area /> <base /> <br /> <col /> <embed /> <hr /> <img /> <input /> <link /> <meta /> <param /> <source /> <track /> <wbr />

### What are the different types of ordered list and unordered list in html?

ORDERED LIST is same as UNORDERED LIST but is normally used when the order of displaying contents plays an importance.

|  |  |
| --- | --- |
| ORDERED LIST | UNORDERED LIST |
| Starts with <ol> and ends with </ol> | Starts with <ul> and ends with </ul> |
| TYPES:  <ol type="1">  <ol type="a">  <ol type="A">  <ol type="I">  <ol type="i"> | TYPES:  <ul type ="disc">  <ul type ="circle">  <ul type ="square">  <ul type ="none"> |
| DEFAULT:  1 | DEFAULT:  disc |

### How to include an image of your choice as a bullet in unordered list?

<html>

<head>

<style>

**ul {**

**list-style-type: none;**

**}**

**ul li:before{**

**content: '';**

**display: inline-block;**

**height: 10px;**

**width: 10px;**

**background-size: 15px;**

**background-image: url("https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQLoH66lOSKA5ihx5\_nIvPKsrs4JoORYnh7VX\_shDZnrqKDtKxG");**

**background-repeat: no-repeat;**

**background-position: center center;**

**margin-right: 5px;**

**}**

</style>

</head>

<body>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Coca Cola</li>

</ul>

</body>

</html>

### 

### How to design a table with a single border?

<!DOCTYPE html>

<html>

<head>

<style>

**table {**

**border-collapse: collapse;**

**}**

**table,td,th{**

**border: 1px solid black;**

**}**

</style>

</head>

<body>

<table>

<tr>

<th>fname</th>

<th>lname</th>

</tr>

<tr>

<td>sruthi</td>

<td>anu</td>

</tr>

</table>

</body>

</html>

### What is the use of rowspan and columnspan?

ROWSPAN :

It allows you to merge specific number of rows

<td rowspan="*number*">

number specifies the number of rows a cell should span.

<td rowspan = “2”> will span 2 rows including current row.

<td rowspan = ”0”> will span all the rows from current row till the end of the table.

COLSPAN :

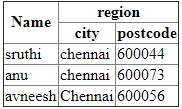
It allows you to merge specific number of columns

<td colspan="*number*">

number specifies the number of columns a cell should span.

<td colspan = “2”> will span 2 columns including current column.

Example:



<!DOCTYPE html>

<html>

<head>

<style>

table{

border-collapse:collapse;

}

</style>

</head>

<body>

<table border=1px>

<tr>

<th rowspan="2">Name</th>

<th colspan="2"> region</th>

</tr>

<tr>

<th> city</th>

<th>postcode</th>

</tr>

<tr>

<td>sruthi</td>

<td>chennai</td>

<td>600044</td>

</tr>

<tr>

<td>anu</td>

<td>chennai</td>

<td>600073</td>

</tr>

<tr>

<td>avneesh</td>

<td>Chennai</td>

<td>600056</td>

</tr>

</table>

</body>

</html>

### How to restrict a particular file type in an image select?

The for attribute of the <label> tag should be equal to the id attribute of the related element to bind them together.

<!DOCTYPE html>

<html>

<body>

<form>

<label for="jpg">JPG</label>

**<input type="file" id=jpg accept=".jpg" />** <br>

<label for="docx">DOCS</label>

**<input type="file" id ="docs" accept=".docx" />** <br>

<label for="pdf">PDF</label>

**<input type="file" id ="pdf" accept=".pdf" />** <br>

<label for="gif">GIF</label>

**<input type="file" id="gif" accept=".gif" />** <br>

<label for="png">PNG</label>

**<input type="file" id="png" accept=".png" />** <br>

</form>

</body>

</html>

### 

### How to restrict input numbers or define a range in input element?

<!DOCTYPE html>

<html>

<body>

<form action="https://www.w3.org/">

**<input type="number" min="1" max="10" required="true"/>**

<input type="submit">

</form>

</body>

</html>

### How to include a youtube video in the document?

Youtube video can be linked into html document using <iframe> or <object> tag.

Click the share button in the video and click on embed option and use the url that appears.

<!DOCTYPE html>

<html>

<body>

**<iframe width="560" height="315" src="https://www.youtube.com/embed/P\_WJJnfPVHo" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>**

**<object width="560" height="315" data="https://www.youtube.com/embed/P\_WJJnfPVHo" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></object>**

</body>

</html>

### 

### Which tag is used to get the location of user?

* Check if Geolocation is supported
* If supported, run the getCurrentPosition() method. If not, display a message to the user
* If the getCurrentPosition() method is successful, it returns a coordinates object to the function specified in the parameter (showPosition)
* The showPosition() function outputs the Latitude and Longitude

<!DOCTYPE html>

<html>

<body>

<p>Click the button to get your coordinates.</p>

<button onclick="**getLocation()**">Try It</button>

<p id="demo"></p>

<script>

var x = document.getElementById("demo");

**function getLocation() {**

**if (navigator.geolocation) {**

**navigator.geolocation.getCurrentPosition(showPosition);**

**} else {**

**x.innerHTML = "Geolocation is not supported by this browser.";**

**}**

**}**

**function showPosition(position) {**

**x.innerHTML = "Latitude: " + position.coords.latitude +**

**"<br>Longitude: " + position.coords.longitude;**

**}**

</script>

</body>

</html>

### 

### Why do we need multipart?

When you make a POST request, you have to encode the data that forms the body of the request in some way.HTML forms provide three methods of encoding.<form enctype="*value*"> The value can be any of the below:

* application/x-www-form-urlencoded (the default)
* multipart/form-data
* text/plain

The enctype attribute specifies how the form-data should be encoded when submitting it to the server.

**Note:** The enctype attribute can be used only if method="post".

<!DOCTYPE html>

<html>

<body>

**<form action="https://www.w3.org/" method="post" enctype="multipart/form-data">**

First name: <input type="text" name="fname" required="true"/><br>

Last name: <input type="text" name="lname"><br>

<input type="submit" value="Submit" required="true"/>

</form>

</body>

</html>

### 

### HTTP request and response format

A simple HTTP request message from a client computer consists of the following components:

* A request line to get a required resource, for example a request GET /content/page1.html is requesting a resource called /content/page1.html from the server.
* Headers (Example – Accept-Language: EN).
* An empty line.
* A message body which is optional.

A simple HTTP response from the server contains the following components:

* HTTP Status Code (For example HTTP/1.1 301 Moved Permanently, means the requested resource was permanently moved and redirecting to some other resource).
* Headers (Example – Content-Type: html)
* An empty line.
* A message body which is optional.

All the lines in the client request and server response should end with a carriage return and line feed. The empty line should only have carriage return and line feed without any spaces.

EXAMPLE:

* **HTTP REQUEST:**
* **Request URL:**
* https://www.google.com/search?q=what+is+http&oq=what+is+http&aqs=chrome..69i57j69i60j0l4.17642j0j7&sourceid=chrome&ie=UTF-8
* **Request Method:**
* GET
* **Status Code:**
* 200 (from ServiceWorker)
* **Referrer Policy:**
* no-referrer-when-downgrade
* **HTTP RESPONSE:**
  + **alt-svc:**
  + quic=":443"; ma=2592000; v="44,43,39,35"
  + **cache-control:**
  + private, max-age=0
  + **content-encoding:**
  + br
  + **content-type:**
  + text/html; charset=UTF-8
  + **date:**
  + Sat, 19 Jan 2019 17:07:45 GMT
  + **expires:**
  + -1
  + **server:**
  + gws
  + **status:**
  + 200
  + **strict-transport-security:**
  + max-age=31536000
  + **x-event-id:**
  + YVlDXN-VMsqw9QPVq4m4CA
  + **x-frame-options:**
  + SAMEORIGIN
  + **x-xss-protection:**
  + 1; mode=block
* **Request Headers**
  + **Provisional headers are shown**
  + **DNT:**
  + 1
  + **Upgrade-Insecure-Requests:**
  + 1
  + **User-Agent:**
  + Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Mobile Safari/537.36

### Difference between http and https

|  |  |
| --- | --- |
| **HTTP** | **HTTPS** |
| URL begins with http:// | URL begins with https:// |
| Uses port no 80 | Uses port no 443 |
| Unsecure since encryption is absent  Does not require any certificate | Secured since encryption is present  Requires SSL certificate |
| Works at application layer | Works at transport layer |

* Hyper Text Transfer Protocol Secure (HTTPS) is the secure version of HTTP, the protocol over which data is sent between your browser and the website that you are connected to. The 'S' at the end of HTTPS stands for 'Secure'. It means all communications between your browser and the website are encrypted. HTTPS is often used to protect highly confidential online transactions like online banking and online shopping order forms.
* Http does not require any validation where Https include validation like domain verification, organization verification or legal document validation

**DRAWBACKS OF HTTP:**

As an “**application layer protocol”**, HTTP remains **focused on presenting the information**, but cares less about the way this information travels from one place to another. Unfortunately, this means that HTTP **can be intercepted and potentially altered**, making both the information and the information receiver (that’s you) vulnerable.

**HTTPS:**

Without HTTPS, any data you enter into the site (such as your username/password, credit card or bank details, any other form submission data, etc.) will be sent as plaintext and therefore susceptible to interception or eavesdropping. For this reason HTTPS is used.

With a HTTPS connection, all communications are securely encrypted. This means that even if somebody managed to break into the connection, they would not be able decrypt any of the data which passes between you and the website.

**WORKING OF HTTPS:**

HTTPS pages typically use one of two secure protocols to encrypt communications - SSL (Secured Socket layer) or TLS (Transport Layer Security). Both the TLS and SSL protocols use what is known as an **'asymmetric' Public Key Infrastructure (PKI)** system. An asymmetric system uses two 'keys' to encrypt communications, **a 'public' key and a 'private' key.** Anything encrypted with the public key can only be decrypted by the private key and vice-versa.In addition to encrypting the data transmitted between the server and your browser, TLS also authenticates the server you are connecting to and protects that transmitted data from tampering.

**HTTPS CERTIFICATE:**

When you request a HTTPS connection to a webpage, the website will initially send its SSL certificate to your browser. This certificate contains the public key needed to begin the secure session. Based on this initial exchange, your browser and the website then initiate the 'SSL handshake'. The SSL handshake involves the generation of shared secrets to establish a uniquely secure connection between yourself and the website.When a trusted SSL Digital Certificate is used during a HTTPS connection, users will see a padlock icon in the browser address bar. When an **Extended Validation Certificate** is installed on a web site, the **address bar will turn green.**

### Use of AJAX

AJAX is an acronym for **Asynchronous JavaScript and XML**.AJAX allows you to send and receive data asynchronously without reloading the web page. So it is fast.

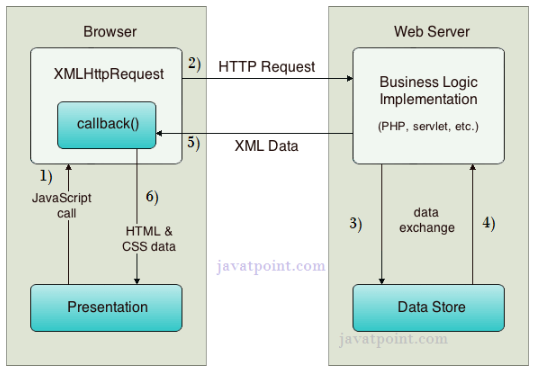
AJAX allows you to send only important information to the server not the entire page. So only valuable data from the client side is routed to the server side. It makes your application interactive and faster.

Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.

With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen.

**STEPS OF AJAX OPERATION:**

* A client event occurs.
* An XMLHttpRequest object is created.
* The XMLHttpRequest object is configured.
* The XMLHttpRequest object makes an asynchronous request to the Webserver.
* The Webserver returns the result containing XML document.
* The XMLHttpRequest object calls the callback() function and processes the result.
* The HTML DOM is updated.



### How to identify if a browser supports JavaScript or not?

The <noscript> tag defines an alternate content for users that have disabled scripts in their browser or have a browser that doesn't support script.

This element can be used in both <head> or <body>.The content inside the <noscript> element will be displayed if scripts are not supported, or are disabled in the user's browser.

<!DOCTYPE html>

<html>

<body>

<script>

document.write("Hello World!")

</script>

**<noscript>Sorry, your browser does not support JavaScript!</noscript>**

<p>A browser without support for JavaScript will show the text inside the noscript element.</p>

</body>

</html>

### 

### How to identify if a browser supports HTML5 or not?

Almost all the browsers are compatible with html5 features but they are not completely compatible in the sense certain features aren’t supported by browsers.For example:

IE supports Email input type in html5 but not date input type.

This can be tested by javascript code.

<!DOCTYPE html>

<html>

<head>

<title>html5 compatibility</title>

</head>

<body>

<p> Test for html5 Compatibility</p>

<script>

function IsInputTypeSupported(typeName) {

// Create element

var input = document.createElement("input");

// attempt to set the specified type

input.setAttribute("type", typeName);

// If the "type" property equals "text"

// then that input type is not supported

// by the browser

var val = (input.type !== "text");

// Delete "input" variable to

// clear up its resources

delete input;

// Return the detected value

return val;

}

if (!IsInputTypeSupported("date")) {

// Do something special

alert("date input is not supported");

}

if (!IsInputTypeSupported("email")) {

// Do something special

alert("email input is not supported");

}

</script>

<noscript> js not supported</noscript>

</body>

</html>

### Differences in usage of <strong> and <em> with examples.

**<strong> vs <bold>**

The **<b>** tag is for "offset text conventionally styled in bold" but without conveying any extra emphasis or importance".**<strong>** represents a span of text with strong importance." There is semantic meaning of importance here. In fact, a **<strong>** tag within another **<strong>** tag has even more importance.

**<b> is a style** - we know what "bold" is **supposed to look like.<strong>** however **is an indication of how something should be understood**.

<strong> is semantic—it describes the text it surrounds (e.g., *"this text should be stronger than the rest of the text you've displayed"*) as opposed to describing *how* the text it surrounds *should be displayed* .

**<em> vs <i>**

Just like **<strong>** and **<b>** we have changes to **<em>** and **<i>**. The [**<i>**](http://dev.w3.org/html5/markup/i.html) tag is for "text conventionally styled in italic". There is no semantic meaning. [**<em>**](http://dev.w3.org/html5/markup/em.html) "represents a span of text with emphatic stress". Like **<strong>**, **<em>** is a tag with semantic meaning and that matters when nested.

The <i> element is more appropriately used for presentational purposes

Emphasis <em> can be used to change the meaning of a sentence depending on where the emphasis is placed, transforming a general statements of fact to an emotional response or assertion.

Using the em element, this sentence can imply different meanings:

* ***Cats* are cute animals.**How one would add to a conversation about cute animals.
* **Cats *are* cute animals.**How one would respond to someone saying that cats aren’t cute animals.
* **Cats are *cute* animals.**How one would argue with someone who said that cats are ugly animals.
* **Cats are cute *animals*.**How one would correct someone suggesting that cats are cute vegetable.

### What are the different coloring Standards in html?

Colors may be specified as an **RGB triplet** or in **hexadecimal format (a *hex triplet*)**. They may also be specified according to their **common English names** in some cases.

|  |  |  |
| --- | --- | --- |
| COLOR | **HEX** | **RGB** |
| **GREEN** | **#008000** | **RGB(0, 128, 0)** |

|  |  |  |
| --- | --- | --- |
| **YELLOW** | **#FFFF00** | **RGB(255, 255, 0)** |

### How to semantically include references in html?

The **HTML Citation element** (**<cite>**) is used to describe a reference to a cited creative work, and must include either the title or author or the URL of that work.

The <cite> tag defines the title of a work (e.g. a book, a song, a movie, a TV show, a painting, a sculpture, etc.). Default value for <cite> tag is italics.

<!DOCTYPE html>

<html>

<body>

<p><cite><a href="https://codepen.io/pen/">click here</a></cite> to get into code pen.</p>

</body>

</html>

### How to embed and open a pdf document in html?

<!DOCTYPE html>

<html>

<head>

</head>

<body>

**<embed src="https://www.w3.org/TR/1998/REC-html40-19980424/html40.pdf" type="application/pdf" width="100%" height="600px" />**

**<object data="https://www.w3.org/TR/1998/REC-html40-19980424/html40.pdf" type="application/pdf" width="100%" height="600px"/>**

</body>

</html>

### Difference between object and embed tag.

Both object and embed tags are used for adding multimedia components like audio,video,files etc.

But if you want your **page to be backward compatible with html 4, then use <object>**. if you **just want Html5, use <embed>**.

The HTML <embed> tag defines a container to load external content.

The <object> tag defines an embedded object within an HTML document. Use this element to embed multimedia (like audio, video, Java applets, ActiveX, PDF, and Flash) in your web pages.

You can also use the <object> tag to embed another webpage into your HTML document.

You can use the <param> tag to pass parameters to plugins that have been embedded with the <object> tag.

**<embed src="https://www.w3.org/TR/1998/REC-html40-19980424/html40.pdf" type="application/pdf" width="100%" height="600px" />**

**<object data="https://www.w3.org/TR/1998/REC-html40-19980424/html40.pdf" type="application/pdf" width="100%" height="600px"/>**

## DATE: 18-01-19 CSS ASSIGNMENT

### What is the use of em?

CSS has several different units for expressing a length.

There are two types of length units: absolute and relative.

**em** comes under relative length.

**Relative length** units specify a length relative to another length property. (‘2’em means ‘2’ times the size of the current font).

The default font-size in browser is **16px**. The default size of **1em** is **16px**. (font-size:**16px**;).

### How to strikethrough a text in CSS?

<!DOCTYPE html>

<html>

<head>

<style>

**h1 {**

**text-decoration: line-through;**

**}**

</style>

</head>

<body>

<h1>Strike through a text in css</h1>

</body>

</html>

### Divide a page into three vertical columns.

**USING TABLE TAG:**

<!DOCTYPE html>

<html>

<head>

<style>

.one{

background-color: cyan;

}

.two{

background-color: yellow;

}

.three{

background-color: violet;

}

</style>

</head>

<body>

<table border="1" width="100%" height="100%">

<tr>

<td class="one" rowspan="0"></td>

<td class="two" rowspan="0"></td>

<td class="three" rowspan="0"></td>

</tr>

</table>

</body>

</html>

**USING DIV ELEMENT:**

<!DOCTYPE html>

<html>

<head>

<style>

div.b {

display: inline-block;

width: 30%;

height: 300px;

padding: 5px;

border: 1px solid blue;

background-color: yellow;

}

</style>

</head>

<body>

<h1>The display Property</h1>

<div class ="b"></div><div class ="b"></div><div class ="b"></div>

</body>

</html>