

W3Schools HTML Review: Samson Ko

Layout	1
LISTS	2
LINKS & IMAGES	3
CSS TOOLTIPS	4
<DIV> ELEMENTS	5
HTML <head> ELEMENT	6
FORMS AND INPUTS	7
Input types	8
Input Attributes	8
BUTTON	9
HTML5	10
HTML4 vs HTML5	10
Multimedia	11
Geolocation	12

Layout

<html>
<head>

<style> is used for declaring the css elements

<body>
<h1>, <h2> is used for heading
<p> is used for paragraphs
 is for images. Can be used with src is the source of the image.

 is used for breaks between texts
<hr> creates a horizontal line as a thematic break

Always remember to put end tags

<a> is for links. For example, is used for links
<html lang = "en-US"> </html> This is used for declaring the language for search engines
<p title> title is used for the tooltip that appears when you put your mouse over it

<meta> refers to its own data
<meta charset = "UTF-8"> to the character set used by the doc

You can use the *style* element within the tag name attribute. For example:

<body style = "property:value"> </style>
<body style = "background-color:powderblue;">

<!-- → is used for comments

<table> is used for tables. LOL

<tr> is used for table ROW and <th> is used for table HEADER and <td> is used for table CELL/DATA. You can use the CSS border property to edit border size and color.

Table, th, td {

border: 1px solid black;

border-collapse : collapse; (this collapses the borders to one border)

Text-align: center, left, right, etc (This is for aligning text)

}

<caption> is used for captions.. Must be inserted right after the table tag

LISTS

 is unordered list, is ordered list

 can be used to create a navigation menu, with many 's to make the menu items.

For example:

```
<style>
```

```
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
    overflow: hidden;  
    background-color: #333333;  
}
```

```
li {  
    float: left;  
}
```

```
li a {  
    display: block;  
    color: white;  
    text-align: center;  
    padding: 16px;  
    text-decoration: none;  
}
```

```
li a:hover {  
    background-color: #111111;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<ul>
```

```
<li><a href="#home">Home</a></li>
```

```
<li><a href="#news">News</a></li>
```

```
<li><a href="#contact">Contact</a></li>
```

```
<li><a href="#about">About</a></li>
```

```
</ul>
```

LINKS & IMAGES

HTML links are hyperlinks.

Local links are used without the <https://www>

`` will link to a page within the html folder on the current website.

`` will open the link in a new tab or browser window. `target = "_self"` will open the link in the same window/tab as it was clicked (This is default). `Target = "_parent"` will open the link in the parent frame. `"_top"` will open the link in a full body of the window.

`` will open straight to the chapter 4 id.

Images can be link also.

```
<a href = "url">  
    <img src = "img url">  
</a>
```

Image floating helps put the image on a certain side of a text. Put it under the `<style>` attribute within an `` attribute.

Image maps are super cool.

Put links within an image, all separately.

Use

```
  
<map name="workmap">  
    <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">  
    <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">  
    <area shape="circle" coords="337,300,44" alt="Cup of coffee" href="coffee.htm">  
</map>
```

```
<body style = "background-image:url ('img url')"></body>
```

`<picture>` attribute is used for more flexibility within declaring images. For example, when the browser is resized, the whole img can actually change. Shows different images for different devices.

[Try it Yourself »](#)

CSS TOOLTIPS

Attribute: hover is used with the mouse of hovering over the text or link

Attribute: active is used when the link or text is activated.

Border: 0 gets rid of the border surrounding an image

It is better to put the width and height attributes within the *style* attribute. This cleans up the code also.

<DIV> ELEMENTS

<div> is a block-level element

 cover the whole width as necessary

Class attribute can cover multiple taglines. For example:

```
<style>
```

```
.city {
```

```
    background-color: tomato;
```

```
    color: white;
```

```
    padding: 10px;
```

```
}
```

```
</style>
```

```
<h2 class="city">London</h2>
```

```
<p>London is the capital of England.</p>
```

```
<h2 class="city">Paris</h2>
```

```
<p>Paris is the capital of France.</p>
```

```
<h2 class="city">Tokyo</h2>
```

```
<p>Tokyo is the capital of Japan.</p>
```

The class will cover all the elements of that class. Javascript will use classes to get the elements by the class name. For example:

```
<script>
function myFunction() {
  var x = document.getElementsByClassName("city");
  for (var i = 0; i < x.length; i++) {
    x[i].style.display = "none";
  }
}
</script>
```

Javascript is all under the <script>
You can use the same class for different taglines.

```
<h2 class="city">Paris</h2>
<p class="city">Paris is the capital of France</p>
```

Instead of classes, you can use the *id* attribute, which is used uniquely within an HTML element. This is unique within the HTML document. It is referenced in the <style> by a hashtag. If id = "myHeader", then <style> would show as #myHeader {}.

ID is used for one unique id that belongs to a single element, while a class name can be used for multiple elements.

<iframe src = "url"> is used to show web pages within a webpage.

HTML <head> ELEMENT

The <head> element is used for METADATA! Placed between the <html> tag and <body> tag. Not displayed. Tags that go within a head element can include: <title>, <style>, <meta>, <link>, <script>, and <base>.

<title> gives the website the title, shows on the browser tab.

<style> is for CSS

<link> is for importing external style sheets.

<meta> is used for metadata, including <meta name = "description" content = "content">

<meta name = "author" content = "John Doe">

It can also be used to refresh a document every number of seconds.

<meta http-equiv= "refresh" content = "30">

Or to get keywords for search engines to find it:

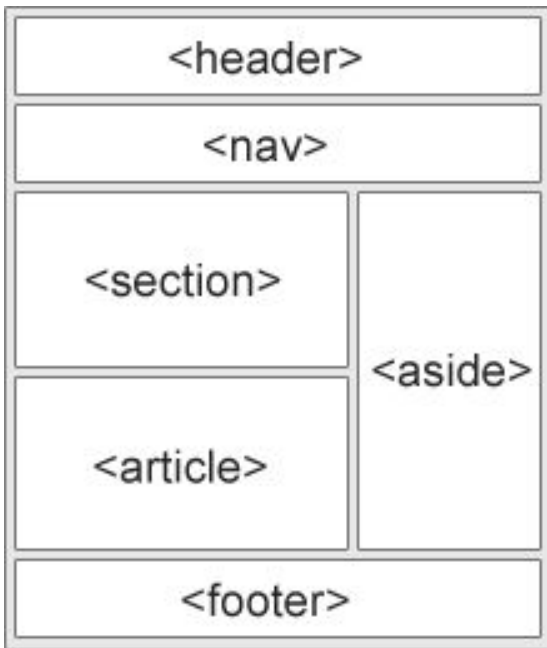
<meta name = "keywords" content = "HTML, CSS, XML, JavaScript">

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

The viewport element tells the browser how to control page's dimensions.

<base> gives default address or target for all links on page

- <header> - Defines a header for a document or a section
- <nav> - Defines a container for navigation links
- <section> - Defines a section in a document
- <article> - Defines an independent self-contained article
- <aside> - Defines content aside from the content (like a sidebar)
- <footer> - Defines a footer for a document or a section
- <details> - Defines additional details
- <summary> - Defines a heading for the <details> element



RESPONSIVE WEB DESIGN

Set ``

GOLDEN RATIO

FORMS AND INPUTS

`<form>` is used for collecting user input.

`<input>` element is used for certain type of input. For example:

`<input type = "text">` is for text field

`<input type = "radio">` is used for radio button, selecting one of many choices.

`<input type = "submit">` is used for submitting a form

`<form action = "url link">` When the user submits the form, it goes to the new url link

`<form method = "post,get,etc">` is used to declare an action of submitting a form. Get is used for public data, while post if private data.

Group the form data and inputs with `<fieldset>` element. `<legend>` element goes inside the `<fieldset>`, specifying the caption, or the detail of what the form is receiving from the user.

Within a form, `<input name = "name">` tell what is the name of the form, the info it is receiving, while `<input value = "Jackie">` specifies the default value that is inside the textbox or button. `<option selected> selectedValue </option>` get the specific value as default.

Input types

`<input type = "password">` is used for a masked set of text shown as asterisks or circles covering the user input.

`<input type = "reset">` creates a button for resetting the values to default values.

`<input type = "checkbox">` creates a checkbox of values.

`<input type = "color">` creates a color input field.

`<input type = "date">` creates an input field for dates.

You can use `<input type = "date" min = "someDate" max = "someDate">` to specify the boundaries.(Use for age restriction!)

`<input type = "datetime-local">` is for a date and time without time zones

`<input type = "email">` is self-explanatory

Input types: month, number, range (creates a scrollable number line), search (behaves like a regular text), tel (telephone number), time, url, week (selecting a week and a year).

Input Attributes

Input type specifies the type of input.

Input readonly prohibits the user from typing. Input disabled specifies that the input field is disabled. Size is for the size of the input field. Input maxlength is for the character limit.

Autocomplete = "on,off" is for auto filling the text boxes. Form is used when there are multiple forms to submit.

<select> is a drop down list. <option> is the option elements within the <select> element. The first value is default. The <select multiple> attribute allows for multiple selecting, <select size = "someNumber"> allows for the number of visible values.

<textarea name = "someName" rows = "someNumber" cols = "someNumber">
This is for a certain text area size.

Types of Forms

<forms> attribute is when there are a multiple of <input> elements inside.

<input formaction = "target url"> specifies where the processed input will go to.

<input formenctype = "target url"> specifies that the data is encoded when moving to the specified url.

Formmethod is for the type of input being transferred, could be a 'get' or 'post'

Formnovalidate is for forms being submitted with being validated.

Formtarget is used when you want to specify where the response will show. For example, formtarget = "_blank" will create a new window with the response.

Height and width can be used with the input buttons, images, and more.

<input pattern = "[A-Za-z]{3}"> This input will not allow the user to submit data unless the input is the same *pattern* of the given pattern.

<input placeholder = "someText"> will have a light text that helps the user input data. It can say "first name" in light text to show the reader that they have to type their first name.

<input type = "text" required> the required instructs the user that the text input field must be filled out before submitting the form.

<input type = "number" step = "3"> the step gives users a choice for intervals

BUTTON

```
<button type = "button" onclick = "alert('hello')"> Click ME! </button>
```

Clickable button, onclick is the state of the button.

<datalist> is for a given set of options. Basically options but for a text box.

<output> element is for results of a calculation.

HTML5

HTML5 allows for creation for totally new elements. For example:

```
<script>document.createElement("myHero")</script>
```

```
<style>
```

```
myHero {
```

```
    display: block;
```

```
    background-color: #dddddd;
```

```
    padding: 50px;
```

```
    font-size: 30px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>A Heading</h1>
```

```
<myHero>My Hero Element</myHero>
```

In this way, javascript creates the new element.

HTML4 vs HTML5

```
<div id="header">    <header>
<div id="menu">      <nav>
<div id="content">   <section>
<div class="article"> <article>
<div id="footer">    <footer>
```

How to display a map on a webpage

Add the Google Maps API.

```
<script src="https://maps.googleapis.com/maps/api/js?callback=myMap"></script>
```

```
<div id="map" style="width:400px;height:400px">
```

```
function myMap() {
```

```
    var mapOptions = {
```

```
        center: new google.maps.LatLng(51.5, -0.12),
```

```
        zoom: 10,
```

```
        mapTypeId: google.maps.MapTypeId.HYBRID
```

```
    }
```

```
    var map = new google.maps.Map(document.getElementById("map"), mapOptions);
```

```
}
```

Multimedia

`<video controls autoplay>` the controls adds video controls like play, pause, and volume on to the video. Autoplay makes the video start right as the webpage is loaded. `</video>`

`<object>` The object element creates a data field that you can add almost anything too. It uses PDF readers, flash players, etc `</object>`

`<embed>` does a similar thing

Easiest way to play videos is use a Youtube video. Use `<iframe>` to do this.

```
<iframe width="420" height="315"
src="https://www.youtube.com/embed/tgbNymZ7vqY?autoplay=1">
</iframe>
```

The `autoplay=1` will start the video as soon as the website is loaded. This must be within the link.

In order to play an entire playlist, Use a comma to play the list of videos to play.

```
<iframe width="420" height="315"
src="https://www.youtube.com/embed/tgbNymZ7vqY?playlist=tgbNymZ7vqY&loop=1">
</iframe>
```

The `loop=1` within the link plays the video forever. Loops forever, while the 0 default value will play the video only once.

```
<iframe width="420" height="315"
src="https://www.youtube.com/embed/tgbNymZ7vqY?controls=0">
</iframe>
```

The `control=0` will not allow the user to control the video, only gives the user a pause function. The default is that they have the control.

Try to use `<iframes>` the most!

Geolocation

Use this link to learn about geolocation:

https://www.w3schools.com/html/html5_geolocation.asp

Other HTML API's:

https://www.w3schools.com/html/html5_draganddrop.asp

