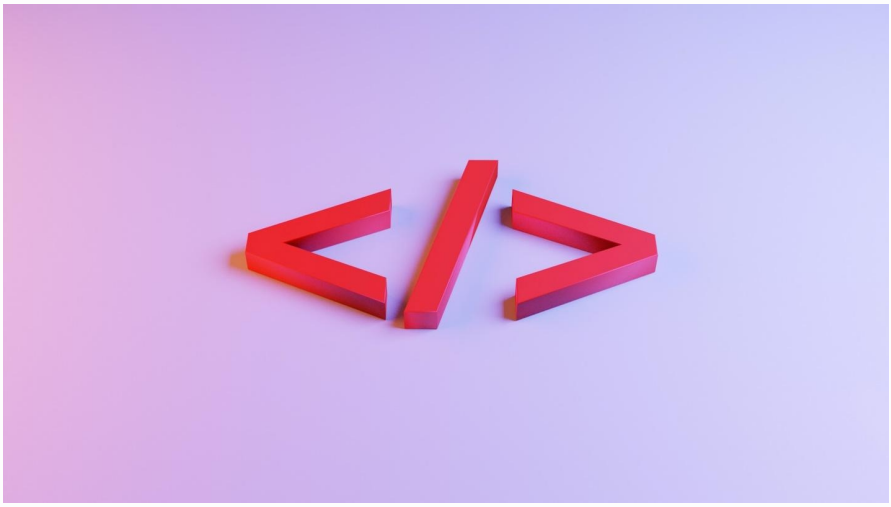
Intermediate / Advanced HTML



Goals
In this lesson you will learn about:
1. The structural components of a website
2. The different HTML tags
3. HTML symbols and file paths

Introduction

This lesson is intended to help you further your understanding of how HTML code is used to create different elements of a website.

HTML is the language that allows you to create web pages. It is what tells a browser how to display your content, and it is what tells you how to write code that works on all browsers.

Review: When you are learning HTML, there are two things you need to know:

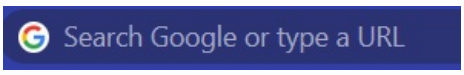
1. How can I make different elements on my page?
2. How do I make sure my pages look good in every browser ad on every device?

As you recall, HTML is a markup language that is used to create web pages. It allows you to create different elements and add content to your website. As you embark on your journey into web development, it is essential to have fundamental knowledge of how web pages are created. The topics covered in this lesson will help you understand the difference between HTML tables, lists, and iframes, to name a few.

As we discuss different elements of a webpage, we will highlight key areas for you to explore and demonstrate your understanding of how everything is pieced together.

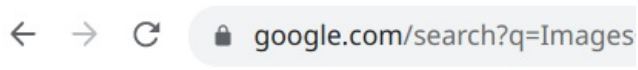
Images & Favicon

What is an HTML Favicon Image?



search bar

An HTML favicon image is a small icon that is displayed in the URL browser or tabs to show the web site identity. Each website has the option to include the icon at the beginning of the URL. Please note that specific browsers such as Google Chrome have replaced the favicon icons to now include the HyperText Transfer Protocol Secure (HTTPS) locked icon to determine if a website is encrypted or not.



Exercise 1:

An HTML favicon image should be saved in the same folder on your web server where the website's files are stored. To add a favicon on your page you should to use `<link>` element to your "index.html" file, after the `<title>` element, like this:

```
<link rel="icon" type="image/x-icon" href="/images/favicon.ico">
```

Note that `.ico` is the file type, though seach engines can also use `.png.` and other file types as favicons. In the code example, you can see that an image titled `html_icon.png` is being used as the favicon, which you can see if you launch the window as web page. You want your favicons to be small squares. Try replacing it with a your own choice of favicon!

Image Maps `<map>` `<area>`

What are HTML Image Maps?

HTML Image Maps are images that have clickable areas. The HTML tags that we will use to define the elements of the HTML image map are:

`<img>` Embed image tag for a website

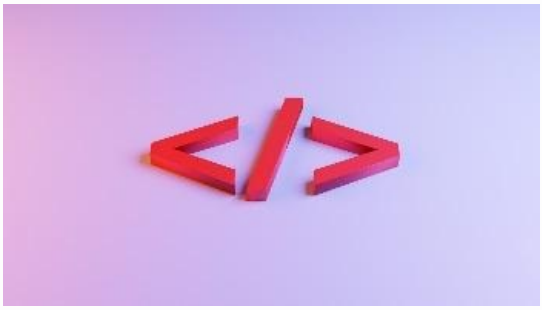
`<map>` Opening tag for the image map

`<area>` The area within the image map

Each area in an HTML image map is defined by an `<area>` element, which has 5 attributes: shape, alt, href (the file), coords (where on the page), and target (where it takes you). A shape attribute determines whether or not an area is active (i.e., it can be clicked). If you want to make multiple areas active, you must use a different value for each shape attribute. Possible values are circle, rect, or poly (which allows you to create any shape you want). An alt attribute specifies alternative text that displays if a user cannot see the image.

For example: “Click here to go home” would be used if a blind user had turned off images on their browser and attempted to click on your image map. An href attribute specifies where users will go when they click on an active area; this could be another page within your site or another website altogether (depending on how it’s coded).

Example 1



closing tag

```
<!DOCTYPE html>
<html>
<body>

<p>Click on the area close to the symbol </p>

<img src="myImage.png" usemap="#image_map" >
<map name="image_map">
   <area alt="symbol" shape="rect" coords="130,87,440,223"
href="https://unsplash.com/photos/_t-l5FFH8VA"  >
</map>


</body>
</html>
```

⚠ You may have to open the webpage in a new browser tab in order for the link to work.

Exercise 2:

Select a website you go to frequently and download an image from a web page then write the HTML code to create an image map for that image.
Hint: You can use this online tool to help you to generate the coordinates.

Tables

Tables

HTML Tables are used to display data in rows and columns for easy reading. In this section, we will show you how to create a table using HTML tags and attributes.

`<table>`

The `<table>` tag defines a table on a webpage. The `<table>` tag can be used with other HTML tags (such as `<tr>` or `<td>`) to create a full table layout.

Table Cells `<td>`

Cells are defined by a `<td>` tag. The cell content must be between the initial part and the final part of the tag as shown in the example below

```
<td>cell content</td>
```

Table Rows `<tr>`

The `<tr>` tag is used inside of a table to define rows within that table. Each row in the table should contain one or more cells that contain text or images (or both). The `<tr>` tag must be used in conjunction with either another `<tr>` tag or one or more `<td>` tags. If no other tags are used, then the browser will automatically start a new row when it encounters an empty cell within the row (i.e., an empty cell without any content inside it). These tags can be used to define the table structure, rows, and columns.

Border-collapse

The `border-collapse` CSS property specifies whether the borders between table cells should be collapsed into a single border or separated as distinct borders. When set to `separate`, each cell has its own border, which can create a visible border between adjacent cells. When set to `collapse`, the borders between adjacent cells are merged into a single border, which can create a more seamless and streamlined look for the table. The default value is `separate`.

border-collapse

Width

The `width` attribute sets the width in pixels or percentage of the containing block element's width that should be taken up by.

Padding

The `padding` is the empty space between the cell header and whatever is written within the cell. It can be adjusted by changing the font size or by using the `padding` property, which allows you to add or remove space around an element.
Colspan
The `colspan` is how you make a cell merge across more than one column. To do this, you put the number of columns that you want to merge into the `colspan` attribute.

Rowspan

The `rowspan` is how you make a cell merge across more than one row. To do this, you put the number of rows that you want to merge into the `rowspan` attribute.

Example 2

Suppose we have a table as shown below:

Cell 1	Cell 2	Cell 3
Cell 4	Cell 5	Cell 6
Cell 7	Cell 8	Cell 9

In html, it looks like:

```
<!DOCTYPE html>
<html>
   <head>
       <title>Table example</title>
   </head>
   <body>
       <table border="1">
          <tr>
             <td>Cell 1</td>
             <td>Cell 2</td>
             <td>Cell 3</td>
         </tr>
         <tr>
             <td>Cell 4</td>
             <td>Cell 5</td>
             <td>Cell 6</td>
         </tr>
         <tr>
           <td>Cell 7</td>
           <td>Cell 8</td>
           <td>Cell 9</td>
       </tr>
       </table>
   </body>
</html>
```

Say you want to merge cell 5 with cell 6, The code for this would look like:

```
<!DOCTYPE html>
<html>
  <head>
      <title>Table example</title>
  </head>
  <body>
      <table border="1">
         <tr>
            <td>Cell 1</td>
            <td>Cell 2</td>
            <td>Cell 3</td>
        </tr>
        <tr>
            <td>Cell 4</td>
            <td colspan="2">Cell 5 and
               6
            </td>
        </tr>
        <tr>
          <td>Cell 7</td>
          <td>Cell 8</td>
          <td>Cell 9</td>
      </tr>
      </table>
  </body>
</html>
```

And the result can be seen as follows:

Cell 1	Cell 2	Cell 3
Cell 4	Cell 5 and 6	
Cell 7	Cell 8	Cell 9

Exercise 3

Use the `rowspan` attribute to merge cells in the first column. Final table should look like this:

Unified Cell	Cell 2	Cell 3
	Cell 5	Cell 6
	Cell 8	Cell 9

Ordered and Unordered Lists

Ordered and Unordered Lists

HTML lists are a great way to organize content on your web page. They can be used to create unordered or ordered lists, and they have a variety of options for formatting the text within each list item.

HTML lists are created using HTML tags. There are three different types of HTML lists: ordered lists, unordered lists, and definition lists. Let's look at each type in more detail so you can decide which one will work best for your website!

`<ol>` - An **ordered** list includes the number of each line
`<ul>` - An **unordered** list does not include a number for each line
`<li>` - The list tag per line

Definition lists are created using the `<dl>`, `<dt>`, and `<dd>` tags, and are used to define a list of terms and their corresponding definitions.

Example 3

Let's look at an example of an **ordered** and **unordered** list:

```
<!DOCTYPE html>
<html>
  <head>
     <title>Days of the Week</title>
  </head>
  <body>
       <h4>Days of the Week</h4>
       <p>Ordered List</p>
       <ol>
           <li>Monday</li>
           <li>Tuesday</li>
           <li>Wednesday</li>
           <li>Thursday</li>
           <li>Friday</li>
           <li>Saturday</li>
           <li>Sunday</li>
       </ol>
        <p>Unordered List</p>
       <ul>
           <li>Monday</li>
           <li>Tuesday</li>
           <li>Wednesday</li>
           <li>Thursday</li>
           <li>Friday</li>
           <li>Saturday</li>
           <li>Sunday</li>
       </ul>
</body>
```

ordered and unordered lists

File Paths

In our previous HTML lessons, we looked at the `<img>` image tag and set the `src` attribute to the path of the file containing the image.

```
<img src="./image.jpg">
```

Now we will learn about additional HTML file paths to reference where we would like our files to be located within our web page.

An HTML file path is a way to describe the location of your website's files. File paths are used to link to external resources which include images, videos, and URLs. It is also possible to link style sheets (CSS) and scripts to execute some functionality in JavaScript language. These topics will be explored in detail later in the course.

Absolute paths

An **absolute** file path is when the full path is given to the file. Points to the exact location where a resource is stored.

```
<img src="https://es.wikipedia.org/wiki/Archivo:Amazon_logo.svg">
```

Relative paths

A **relative** file path does not include a domain name in the URL and is only relative to the root directory in which it is contained.

Example 1

```
<img src="./images/logo.jpg">
```

Notice the `./`, it makes reference to the current directory. In this case, the `./` indicates that the `images` folder is located in the same directory as the HTML file, and the `logo.jpg` file is located within the `images` folder. The `./` makes it clear that the file path is relative to the current directory.

Example 2

```
<img src="../images/logo.jpg">
```

`../` is another relative file path that points to a directory one level **above** the current directory. In the example, the file path `../images/logo.jpg` points to a file called `logo.jpg` located in the `images` folder, which is one level **above** the current directory.

Symbols

Symbols

HTML offers several symbols such as mathematical symbols, Greek letters, punctuation, and emojis that you can use in your web pages. Some symbols aren't listed on standard keyboards and are but might be important to your website, like the copyright symbol.

Therefore, these symbols are represented by an entity reference or an entity name. The two types of entities available are decimal and hexadecimal references. Decimal references are the most used; however, both types of entities can be used within HTML documents.

🔍 Since there are several symbols to choose from, we do not expect you to memorize all of them. You can reference this resource.

Example 4

Note how to reference the copyright and trademark symbols and dollar and percentage signs

```
<html>
   <body>
       <p>Copyright Symbol  &copy;</p>
       <p>Trademark &trade;</p>
       <p>Dollar &dollar;</p>
       <p>Percentage &percnt;</p>
   </body>
</html>
```

Copyright Symbol ©

Trademark ™

Dollar $

Percentage %

Iframes `<iframe>`

An HTML iframe is used to display content within a web page. This may include an image, video, or animation. Iframes are a type of inline frame, which means that the iframe does not have its own window or tab; instead, it runs in the canvas of the parent page.

To include an iframe into your web page, there are 4 attributes that you must include: the src attribute specifies the URL of the web page to be displayed in the iframe, width and height determine the size of the iframe, and title provides a title for the web page. You may also include CSS style attributes (e.g., style), we will discuss CSS in more detail later.

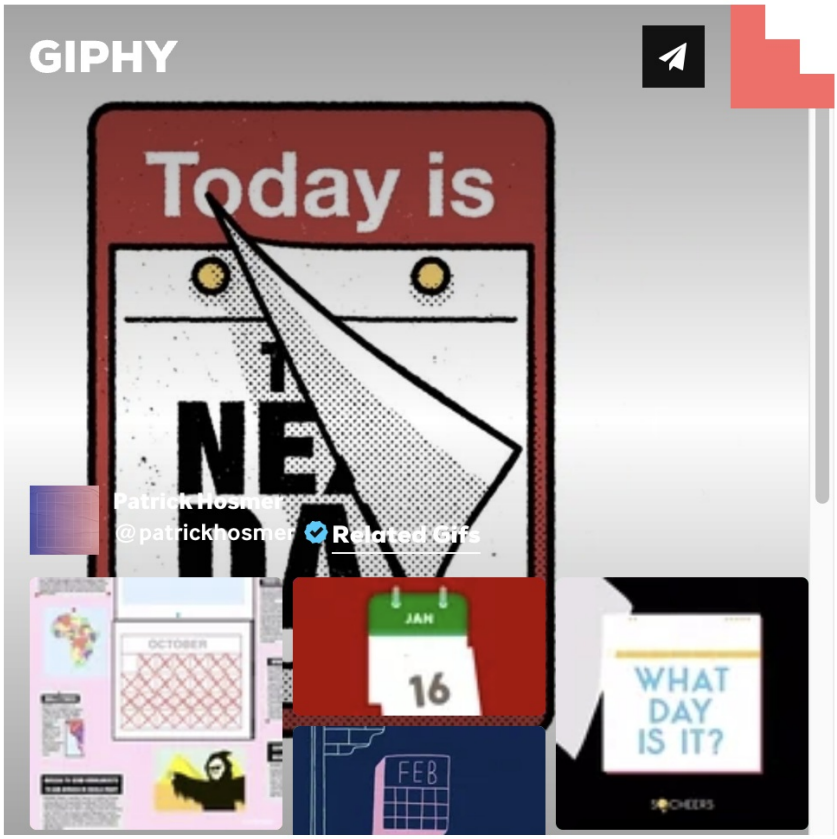
Example 5

```
<!DOCTYPE html>
<html>

<head>
    <title>Example Iframe</title>
</head>

<body>
    <iframe src="https://giphy.com/embed/QC7a49iu2NEwHCqp6b"
width="480" height="480" frameBorder="0"
        class="giphy-embed" allowFullScreen></iframe>
    <p>
        <a href="https://giphy.com/gifs/calendar-groundhog-day-
patrick-hosmer-QC7a49iu2NEwHCqp6b" aria-label="gif">via
            GIPHY</a>
    </p>
</body>
```

It looks like this:

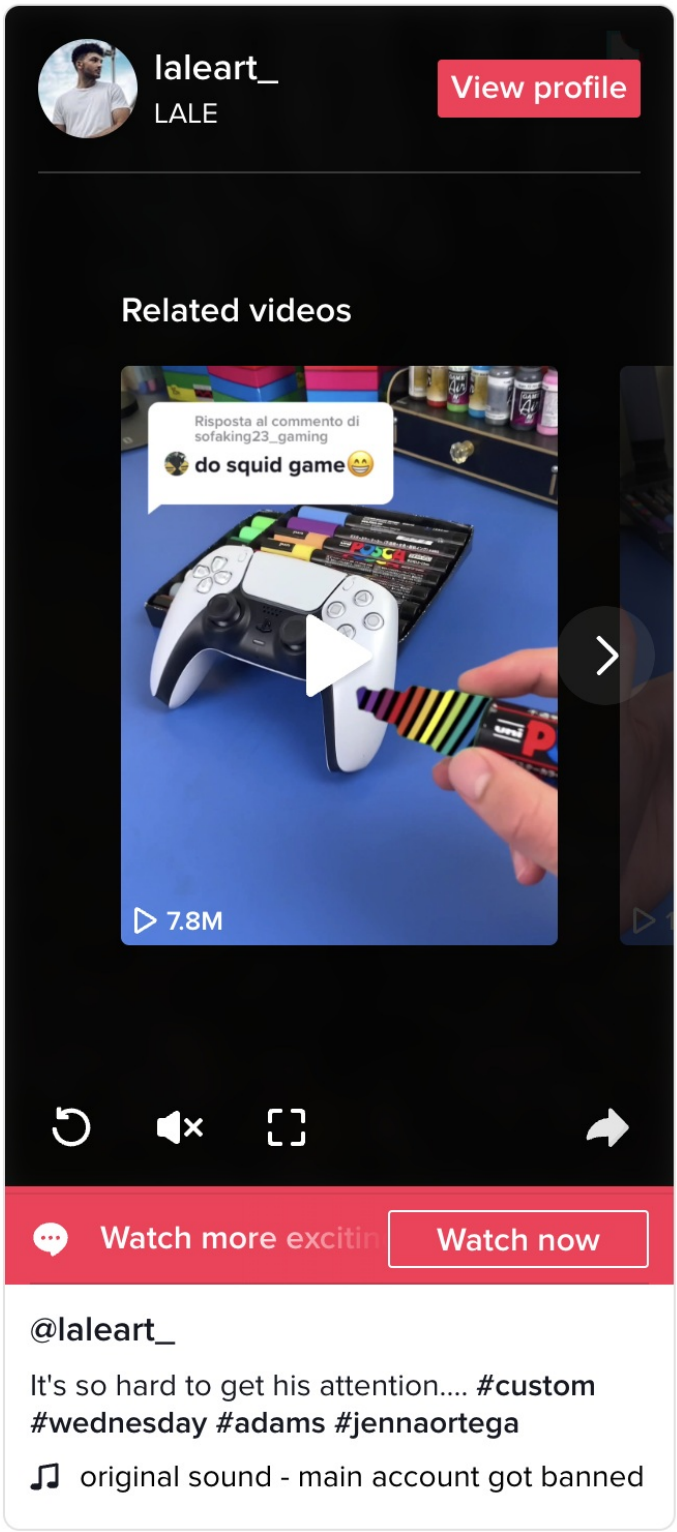


via GIPHY

Exercise 4

Create an iframe with your favorite website, video, song or image.
Note: Some websites have a button for sharing content, you must choose “Embed” and copy and paste in your code.

The sharing button is the curved arrow at bottom right of the video in this image:



Blocks

Blocks `<div>`

HTML block-level elements is a way to separate sections of text within your HTML code. Block-level elements are those that are separated by physical spaces. These elements include paragraphs, headers and subheadings, lists, and tables. These elements include `<div>`, `<h1>`, `<p>`, `<ul>` and many more. The `<div>` element is typically paired with other attributes such as style.

Inline elements are usually used to add emphasis to text by making it bold, italicized, or colored. Examples of inline elements include `<strong>` and `<em>`. Inline elements can appear within sentences and do not have to appear on a new line of their own. This is different from block-level elements, which are usually separated from other content.

Example 6

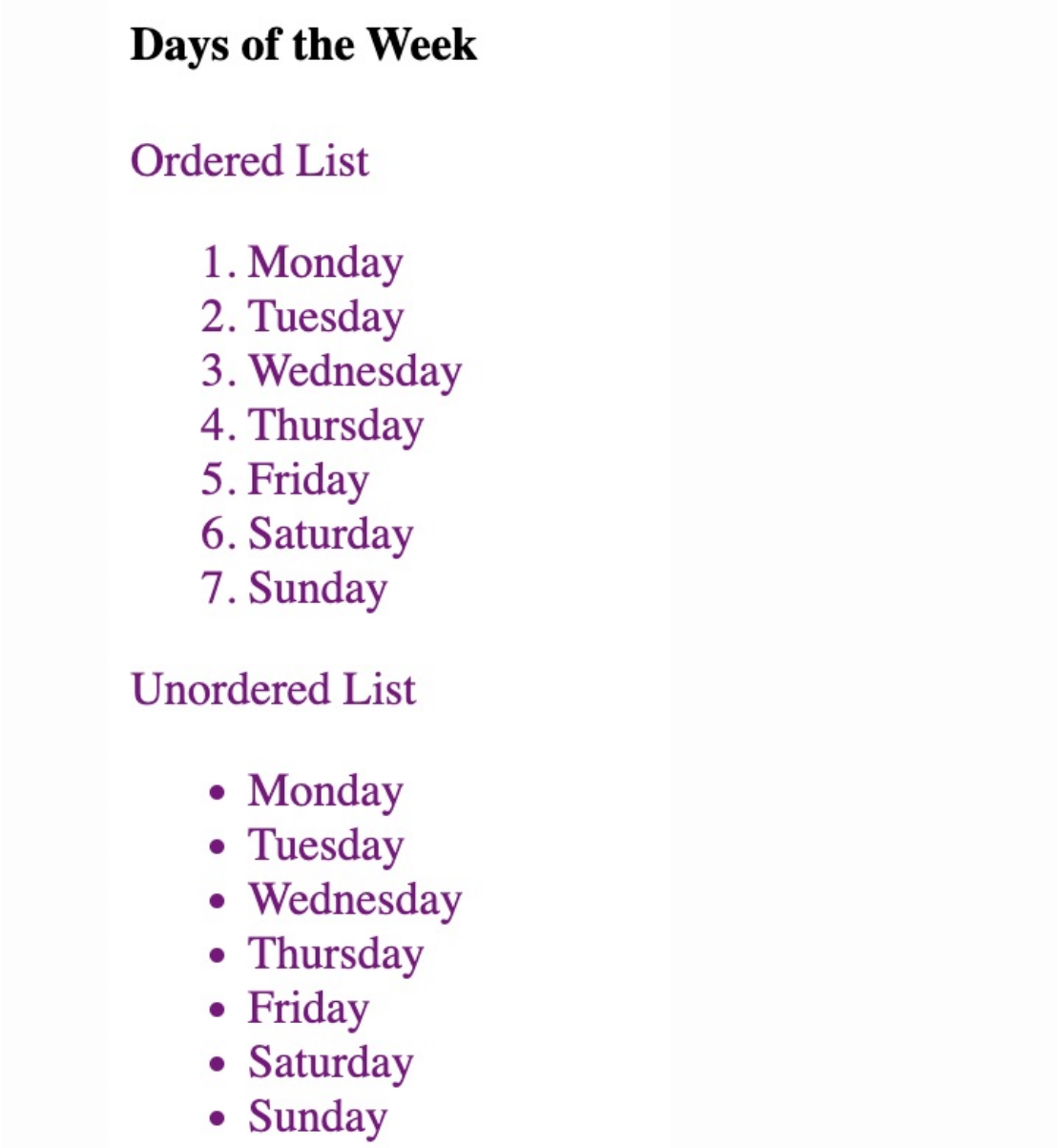
In our example below you will see that we've included the style attribute with the `<div>` element to change the color of the font for the ordered list.

```
<!DOCTYPE html>
<html>

<head>
    <title>Days of the Week</title>
</head>

<body>
    <h4>Days of the Week</h4>
    <div style="color:purple">
        <p>Ordered List</p>
        <ol>
            <li>Monday</li>
            <li>Tuesday</li>
            <li>Wednesday</li>
            <li>Thursday</li>
            <li>Friday</li>
            <li>Saturday</li>
            <li>Sunday</li>
        </ol>
        <p>Unordered List</p>
        <ul>
            <li>Monday</li>
            <li>Tuesday</li>
            <li>Wednesday</li>
            <li>Thursday</li>
            <li>Friday</li>
            <li>Saturday</li>
            <li>Sunday</li>
        </ul>
    </div>
</body>
```

It look like this:



Conclusion & Takeaways

Conclusion & Takeaways

HTML tags are used to add elements to a web page. They can be used to add headings, images, paragraphs, lists and more.
The `<map>` tag is used to add a map area in a web page. It defines an area on the page that will be clickable by the user and links it to another page or file on the site.
The `<area>` tag specifies an area of interest on the map for linking purposes.
The `<iframe>` tag defines an inline frame in which content from another source can be displayed within your own web page's content area.

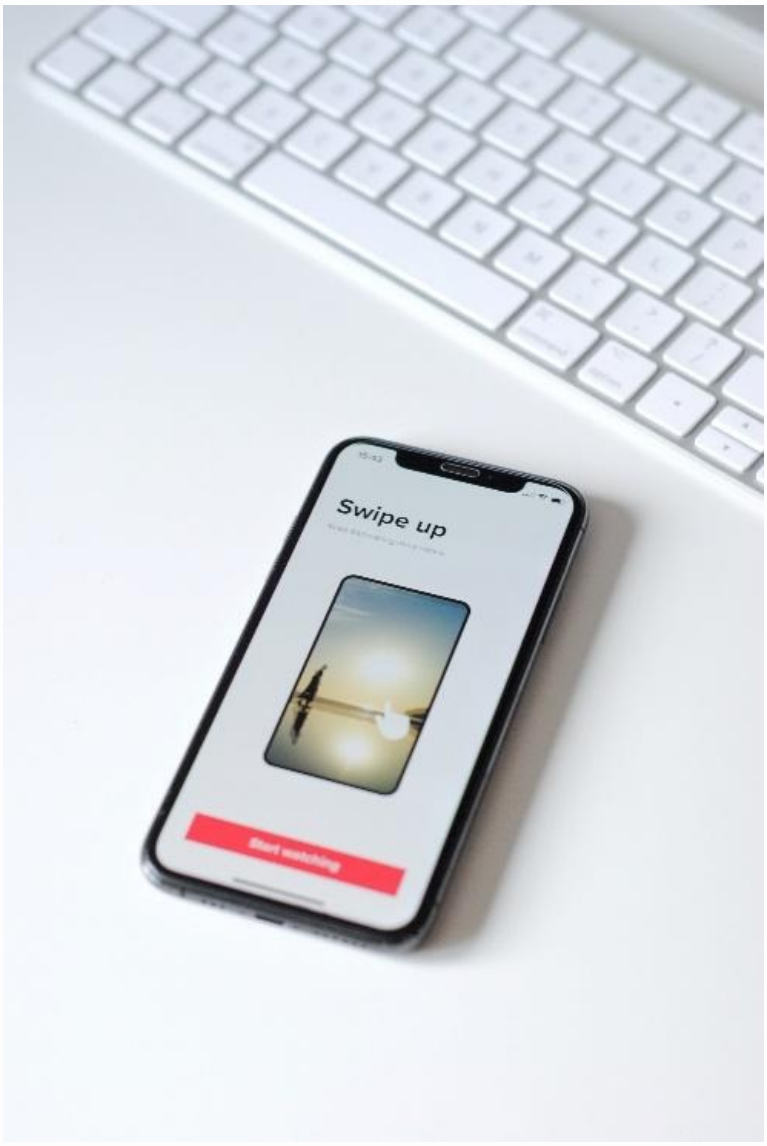
Suggestion: Create your own cheat sheet document with all of the tags you're familiar with!

Attribution

Source: Unsplash
(n.d.). Beautiful Free Images & Pictures. Unsplash. https://unsplash.com/

HTML Media

HTML Media



cover image

Goals

In this lesson you will learn about:
1. The definition of HTML multimedia
2. The different types of audio and video file formats
3. How to create embedded media

Introduction

This lesson is intended to help you further your understanding of how HTML code is used to create different types of embedded media formats for your web page.

Most websites use some type of multimedia to engage the user, such as Youtube or Spotify or some other type of media content.

What is multimedia?

Multimedia is any combination of digital media. The types of multimedia can be from images, videos, music, or animations. For example, it can be a document with an embedded video, or it can be a video on a webpage that plays automatically when you open it. Multimedia is commonly used on the web, in advertisements and presentations as well as on other electronic devices.

Multimedia formats and extensions

Multimedia elements (e.g., audio files and videos) are stored in media files. Multimedia files have formats and different extensions to help you identify them. Multimedia formats include: .midi, .wma, .aac, .wav, .mp3, .mpeg or .mpg, .avi, .wmv, .mov, .mp4

Multimedia Type	Extension	Notes
Audio	.wav	
Audio	.mp3	
Audio	.mp4	.mp4 can be used for audio and video
Video	.avi	
Video	.mov	
Video	.mp4	.mp4 can be used for audio and video

Video

Video `<video>` : MPEG, AVI, WMV, QuickTime, MP4

The `<video>` HTML element embeds in your page a media player for video playback such as a movie clip or other video streams. It can have one or more sources (src).

🔍 Check out this HTML Video Resource for more information and practice.

Example 1

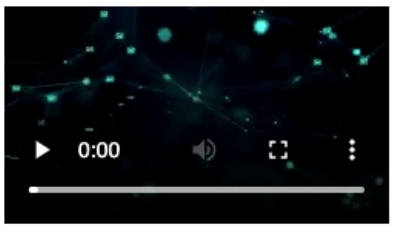
In the example below, we had a video in several multimedia format in a folder local:

```
<!DOCTYPE html>
<html>

<head>
   <title>Exercise Multimedia - Video</title>
</head>

<body>
   <video controls width="250">
       <source src="example_video.mp4" type="video/mp4">
       <source src="example_video.webm" type="video/webm">
       <p>
           Your browser doesn't support HTML video. Here is a
           <a href="example_video.mp4">link to the video</a>
instead.
         </p>
     </video>
     <p>Stock footage provided by Pressmaster, downloaded from
videvo.net</p>
</body>
```

It looks like this:



Stock footage provided by Pressmaster, downloaded from videvo.net

🔍 When it comes to video content, you can search for stock footage websites like Pexels, Pixabay, and Videvo.

⚠ It's important to always check the licensing terms for each resource to ensure that you are using it legally.

Exercise 1

Modify the previous code with a video of yours and open it in the browser.

Audio

Audio `<audio>`: MIDI, WMA, AAC, WAV, MP3

The `<audio>` HTML element embeds in your page an audio player such as music or other audio streams. It can have one or more sources (src).

🔍 Check out this HTML Audio Resource for more information and practice.

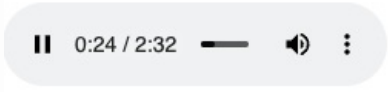
Example 2

In the example below, we have an audio in a folder local:
>

```
<!DOCTYPE html>
<html>

<head>
  <title>Exercise Multimedia - Audio</title>
</head>

<body>
  <audio controls width="250">
      <source src="Hair-Trigger-WST011601.mp3" type="audio/mp3">
      <p>
          Your browser doesn't support HTML audio. Here is a
           <a href="Hair-Trigger-WST011601.mp3">link to the
audio</a> instead.
        </p>
    </audio>
    <p>Stock footage provided by Pressmaster, downloaded from
videvo.net</p>
</body>
```

> It looks like this:
>
> 
>
>
> Stock footage provided by Pressmaster, downloaded from videvo.net
>
> 🔍 There are many resources for copyright-free audio that you can use on your website, such as: YouTube Audio Library and Free Music Archive.

⚠ It’s important to always check the licensing terms for each resource to ensure that you are using it legally.

Exercise 2

Modify the previous code with your favorite song and open it in the browser.

Conclusion & Takeaways

Conclusion & Takeaways

HTML multimedia formats are used to create web pages. The format of the HTML multimedia file is determined by the type of content it contains and how it will be displayed in a browser.

The most common HTML multimedia file is the image, which can be used to add photographs or illustrations to your site. There are also video files that allow you to embed videos from YouTube or other sources into your site. Audio files provide another way to add sound effects, music, and other audio clips to your pages.

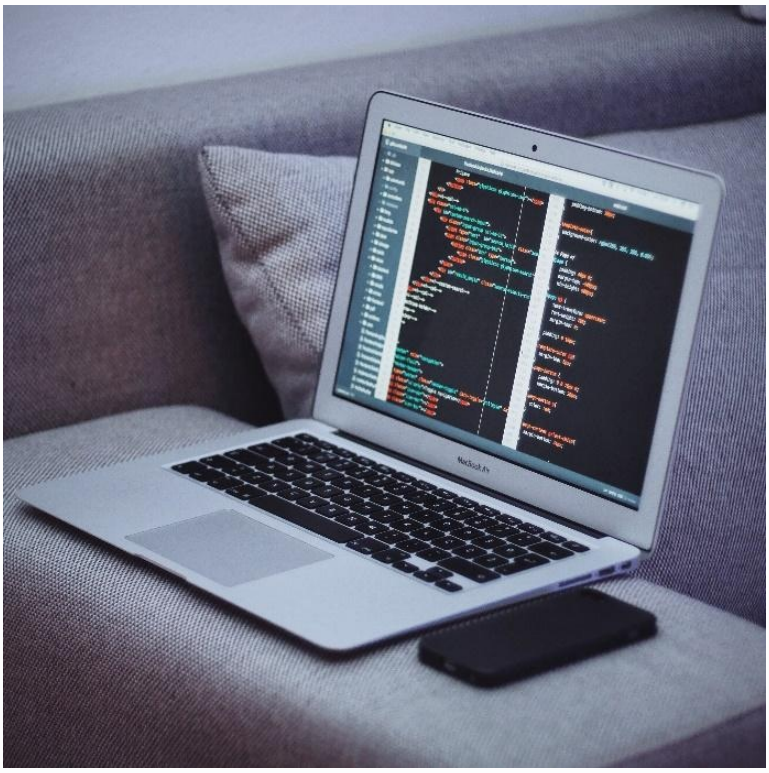
All three types of multimedia files can be used on their own or combined together in a single page for more complex presentations.

Further Investigation:

> Source: Unsplash
> - HTML video Tag. (n.d.). [Video]. https://www.w3schools.com/tags/tag_video.asp
> - : The Video Embed element - HTML: HyperText Markup Language | MDN. (2022, November 11). [Video]. https://developer.mozilla.org/en-US/docs/Web/HTML/Element/video
> - HTML audio tag. (n.d.). [Video]. https://www.w3schools.com/tags/tag_audio.asp
> - : The Embed Audio element - HTML: HyperText Markup Language | MDN. (2022, October 31). [Video]. https://developer.mozilla.org/en-US/docs/Web/HTML/Element/audio

HTML Review & Additional Tags

HTML Review & Additional Tags



computer

Goals

In this lesson you will:

1. review HTML tags
2. View additional HTML resources

Introduction

Let's take a look at all of the HTML tags and elements that we've learned throughout this lesson:

Images + Favicon, Image Maps, Backgrounds

Images + Favicon:

```
<link rel="icon" type="image/x-icon" href="/images/favicon.ico">
```

Image Maps:

```
<img src="image.jpg" usemap="#image_map" >
<map name="image_map">
   <area alt="symbol" shape="rect" coords="130,87,440,223"
href="https://unsplash.com/photos/_t-l5FFH8VA"  >
</map>
```

> **Tables: Definition, cells `<td>`, rows `<tr>`, headers `<th>`, borders, size, padding, spacing, colspan, rowspan**
>
> Syntax table:
> `<table></table>`

Add a row to table:
`<tr></tr>`

Add a cell to table:
`<td></td>`

Concat cells:
`<td colspan="Number of cells"></td>`

Concat rows:
`<td rowspan="Number of rows"></td>`

Lists:

Ordered:

```
<ol>
    <li>item</li>
</ol>
```

> Unordered:

```
<ul>
   <li>item</li>
</ul>
```

> **HTML File Paths: web pages, Images, style sheets, JavaScripts, absolute/relative**
>
> Relative:

```
<img src="./images/image.jpg">
```
>
Absolute:

```
<img
src="https://es.wikipedia.org/wiki/Archivo:Amazon_logo.svg">
>
```

HTML Symbols: mathematical, greek letters, punctuation, emojis

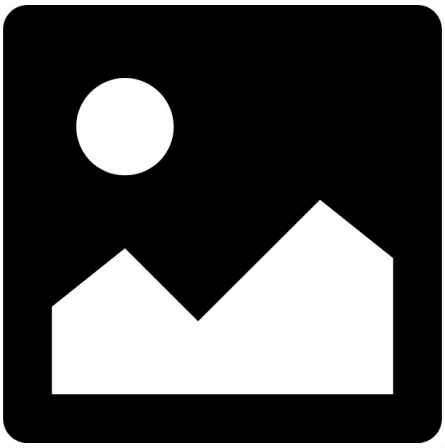
Full Math Reference
Full Greek Reference
Full Currency Reference
Full Arrows Reference
Full Symbols Reference
>
Blocks:
`<div></div>`
>
Iframes `<iframe>`

```
<iframe></iframe>
```
>
All attributes of iframe here.
>
HTML Media:
>
Video:
`<video></video>`
Video formats: MPEG, AVI, WMV, QuickTime, MP4 & Audio
>
Audio:
`<audio></audio>`
Audio formats: MIDI, WMA, AAC, WAV, MP3
>
As you continue to work through your activities and assignments, we want to provide you with some additional resources to help guide you through the cohort.

Supplement Video & HTML Tags Cheat Sheet

https://youtu.be/9lP8TcVJv-I

https://web.stanford.edu/group/csp/cs21/htmlcheatsheet.pdf
>
Exercise 1
Build a page like this:

Add an image of your pet or favorite animal	
Add your favorite song	
Add a video of your pet	

You may need to upload your image to the file tree in order to see it, and your path may not be the same as the above.

Tip: use a div in your code and add the attribute style with value `background-color: lightgray`

Note: you should include a brief description where you see `[alt text]`. This increases accessibility and shows you are aware of best practices.

Conclusion & Takeaways

Conclusion & Takeaways

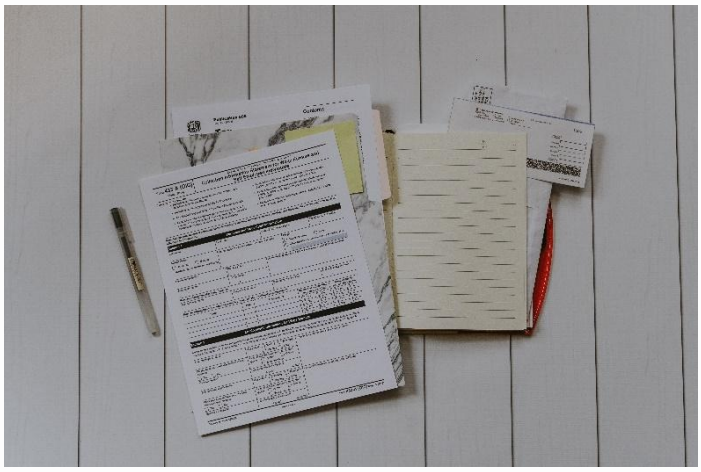
Embeded media can make your web page more appealing and interactive. Once we get to styling, you can really show of your skills!

Attribution

> Source:Unsplash
> - W3Schools Free Online Web Tutorials. (n.d.). [Video]. https://www.w3schools.com/default.asp

HTML Forms

HTML Forms



cover image

Definition `<form>` element

HTML forms `<form>` are used to allow users to enter data and submit it to a server. This input is usually collected from multiple fields, as well as required fields that must be filled out in order for the form to be submitted. When the user submits a form, an HTTP request is sent to a server and the response is parsed by JavaScript or PHP. This is how we receive and process data from a webpage.

Labels

The HTML `<label>` tag defines a label for an input control. The for attribute of the `<label>` must be equal to the id attribute of the related element, or an id value in a corresponding `<input>` element. A label can also be bound to an element by placing the element inside the `<label>` element.
Inputs `<input>` Text, radio, checkbox

The HTML `<input>` element is a special form field that you can use to accept user input like text or a number. It's used for creating text inputs (text boxes), checkboxes, radio buttons, submit buttons and other forms of input fields.
Example 1

```
<!DOCTYPE html>
<html>

<head>
    <title>Weekday</title>
</head>

<body>
    <h4>What Day Is It?</h4>
    <form action="/example_action.html">
        <label for="wday">Day: </label>
        <input type="text" id="wday" name="wday"><br><br>
        <input type="submit" value="Submit">
    </form>
</body>

</html>
```

In the previous example, we add the `<form>` HTML element inside the body and at the same time add a `<label>`, two `<input>` type “text” and “submit”, inside of the form element. The `<input>` type “text” receives a string and the `<input>` type “submit” activates the action indicated in the property “action” of the form element. The goal of the forms, most of the time is input data and save it in a database.

Note: the line `<form action="/example_action.html">` is a placeholder. This is backend related command, and we’ll see how this functions more when we study javascript. For now, just be sure you have a placeholder for the form action, knowing that eventually we’ll have to tell the form what to do with the data.

It looks like this:

Exercise 1

Create a form where you can input of the next information with an input type submit with value “Save”:
- Name (text)
- Last name (text)
- Age (number)
- Birthday (date)
- Gender (list, use `<select>` element documentation)
Note: use the types of input indicated for every field.

It looks like this:

Note: The `<select>` element will be seen in the next lesson.

Radio and Checkbox

Radio and Checkbox

The radio input type allows the user to select only one option from a group of options. This control is always presented in a group of radio buttons, where only one radio button can be selected at a time.
Whereas the checkbox input type is represented by a square box that is shown as ticked (checked) when the user selects the input. You can use checkboxes in forms to check or uncheck options. When you select an option from a list of options, it might be useful to be able to save your selection somewhere permanently; so that it can be used again in another part of the form or even on another page.

Example 2

```
<!DOCTYPE html>
<html>

<body>
    <form action="/example_action.html">
        <h1>What day is it?</h1>
        <p>Radio Input Type</p>
        <input type="radio" id="sunday" name="day"
value="Sunday">
        <label for="sunday">Sunday</label><br>
        <input type="radio" id="monday" name="day"
value="Monday">
        <label for="monday">Monday</label><br>
        <input type="radio" id="tuesday" name="day"
value="Tuesday">
        <label for="tuesday">Tuesday</label><br>
        <input type="radio" id="wednesday" name="day"
value="Wednesday">
        <label for="wednesday">Wednesday</label><br>
        <input type="radio" id="thursday" name="day"
value="Thursday">
        <label for="thursday">Thursday</label><br>
        <input type="radio" id="friday" name="day"
value="Friday">
        <label for="friday">Friday</label><br>
        <input type="radio" id="saturday" name="day"
value="Saturday">
```

```
        <label for="saturday">Saturday</label>
        <br>

        <Checkbox Input Type>
            <p>Checkbox Input Type</p>
            <input type="checkbox" id="sunday" name="day"
value="Sunday">
            <label for="sunday">Sunday</label><br>
            <input type="checkbox" id="monday" name="day"
value="Monday">
            <label for="monday">Monday</label><br>
            <input type="checkbox" id="tuesday" name="day"
value="Tuesday">
            <label for="tuesday">Tuesday</label><br>
            <input type="checkbox" id="wednesday" name="day"
value="Wednesday">
            <label for="wednesday">Wednesday</label><br>
            <input type="checkbox" id="thursday" name="day"
value="Thursday">
            <label for="thursday">Thursday</label><br>
            <input type="checkbox" id="friday" name="day"
value="Friday">
            <label for="friday">Friday</label><br>
            <input type="checkbox" id="saturday" name="day"
value="Saturday">
            <label for="saturday">Saturday</label><br><br>
            <input type="submit" value="Submit">
    </form>
</body>

</html>
```

Exercise 2

Update the exercise 1:
1. Change the field Gender for a Radio button
2. Create the field Hobbies with checkboxes (options: Sing, Dance, Read, Travel)

form example

Buttons

Buttons

The button tag is commonly used to define a button that represents a defined action. The button tag can also be used in conjunction with attributes such as type (e.g., submit), value, and class.
Example 3

```
<!DOCTYPE html>
<html>
  <head>
    <h1>Button Example</h1>
  </head>
  <body>
    <form>
      <input type="button"
onclick="window.location.href='https://www.correlation-
one.com/';" value="Click Example" />
    </form>
  </body>
</html>
```

It looks like this:

Dropdown menus

With `<select>` element you can create a dropdown list. Most time is used in forms or menus, the goal of this element is to have a list type field where you should choose a unique option between two or more options. For example, Gender has two options and you should choose one option.

As the other elements of a form you need indicate a name attribute to reference the form data after the form is submitted, if you omit this, no data from the drop-down list will be submitted.

Another important element is the id, it is needed to associate the drop-down list with a label.
Example 4
Let’s see in action in a HTML code:
>
In the example below has the field Gender as a type list:

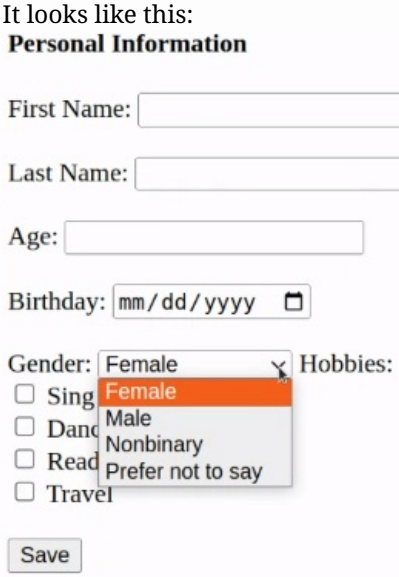
```
<!DOCTYPE html>
<html>

<head>
    <title>Form</title>
</head>

<body>
    <h4>Personal Information</h4>
    <form action="/example_action.html">
        <label for="fName"> First Name: </label>
        <input type="text" id="fName" name="fName"><br><br>
        <label for="fLastName">Last Name: </label>
        <input type="text" id="fLastName" name="fLastName"><br>
<br>
        <label for="fage">Age: </label>
        <input type="number" id="fage" name="fage"><br><br>
        <label for="fbirthday">Birthday: </label>
        <input type="date" id="fbirthday" name="fbirthday"><br>
<br>
        <label for="fgender">Gender: </label>
        <select name="fgender" id="fgender">
            <option value="Female">Female</option>
            <option value="Male">Male</option>
            <option value="Nonbinary">Nonbinary</option>
            <option value="Prefer not to say">Prefer not to
say</option>
        </select>
        <label for="fhobbies">Hobbies: </label><br>
        <input type="checkbox" id="sing" name="fhobbies"
value="Sing">
        <label for="sing">Sing</label><br>
        <input type="checkbox" id="dance" name="fhobbies"
value="Dance">
        <label for="dance">Dance</label><br>
        <input type="checkbox" id="read" name="fhobbies"
value="Read">
        <label for="read">Read</label><br>
        <input type="checkbox" id="travel" name="fhobbies"
value="Travel">
        <label for="travel">Travel</label><br><br>
        <input type="submit" value="Save">
    </form>
</body>

</html>
```

It looks like this:



Form Attributes: Action, Target, Methods, GET & POST

Earlier in the lesson, we briefly introduced form attributes, but we want to provide you with some additional context. The form action attribute defines the action to be performed when the form is submitted.

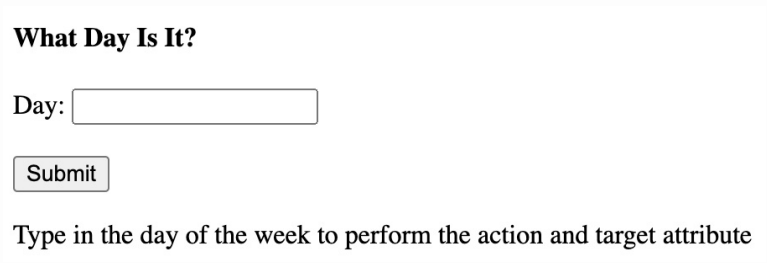
Typically, the form data is sent to a file on the server when the user clicks on the submit button. The target attribute specifies where to display the response that is received after submitting the form.

The method attribute specifies which HTTP (hypertext transfer protocol) method will be used when submitting form data. There are two possible values for this argument: GET and POST. Generally, GET is used to request data from a resource, while POST is used to send data to create (or update) a resource. The form data can be sent as URL variables (with method="get") or as HTTP post transaction (with method="post"). For the GET method, the values will be shown in the browser unlike the POST method once a user submits the form. This means the POST method is more secure, but it also means refreshing or hitting the back button can wipe the form.

When a user submits a form on your web page the request is handled by a server-side script on our server. Most languages that run on the web have this kind of capability, and here is an example of how we use PHP to handle the data from our form. In PHP, we tell the server to store the data in a variable called $my_form['my_first_name']—in other words, with the key "my_first_name". We will fill in this information later when we look at the scripts that handle these variables.

Example 5

```
<!DOCTYPE html>
<html>
   <head>
      <title>Weekday</title>
   </head>
   <body>
         <h4>What Day Is It?</h4>
<form action="/example/page.html" target="_blank">
  <label for="wday">Day: </label>
  <input type="text" id="fname" name="fname"><br><br>
  <input type="submit" value="Submit"> <br>
  <p>Type in the day of the week to perform the action and
target attribute</p>
</form>
</body>
```

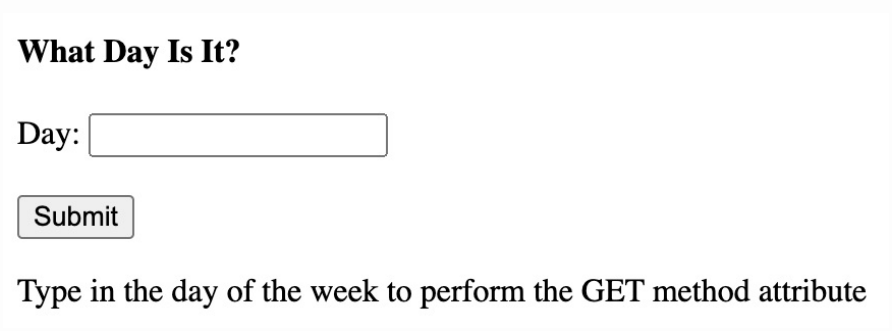


submit button

For exercises 6 & 7, you may need to launch the code in your browser to see the difference. Which one puts your answer directly in the search bar when you press
“Submit”?

Example 6

```
<!DOCTYPE html>
<html>
   <head>
    <title>Weekday</title>
   </head>
   <body>
        <h4>What Day Is It?</h4>
<form action="/example_action.html" target="_blank"
method="get">
  <label for="wday">Day: </label>
  <input type="text" id="weekday" name="weekday"><br><br>
  <input type="submit" value="Submit"> <br>
  <p>Type in the day of the week to perform the GET method
attribute</p>
</form>
</body>
```



>

Example 7

```
<!DOCTYPE html>
<html>
   <head>
      <title>Weekday</title>
   </head>
   <body>
         <h4>What Day Is It?</h4>
<form action="/example_action.html" target="_blank"
method="post">
  <label for="wday">Day: </label>
  <input type="text" id="weekday" name="weekday"><br><br>
  <input type="submit" value="Submit"> <br>
  <p>Type in the day of the week to perform the POST method
attribute</p>
</form>
</body>
```

> **What Day Is It?**
>
> Day:
>
> Submit
>
> Type in the day of the week to perform the POST method attribute

>

Exercise 3

Conclusion & Takeaways

Conclusion & Takeaways

HTML forms allow us to create different elements and attributes on a web page. There are many different types of HTML forms that you can use, but the most common are text fields, checkboxes, radio buttons, drop-down menus and submit buttons.
>

Design a Form

Design a Form

Goals
In this exercise you will:
1. Create an HTML form using

tags
1. Define a list of HTML elements and attributes
>
Introduction

Now that we've reviewed examples of HTML forms and elements, let's create your very first HTML form! We will render the form with some input fields (field sets are not required), a textarea element, and a button. The button will trigger an onclick event handler which will be used to submit the data to the server. We'll also discuss form processing in detail later on in the lesson. You can use our sample code from previous lessons to help get you started.

Instructions

- Choose a list of questions for your form
- Select the type of elements (e.g., checkbox, radio)
- Include graphical elements (e.g., background image, icon buttons)

You can use these questions, or make your own.

- What is your genre of music?
- Do you have a pet? Yes or no
- What is his name?
- What is your favorite sport? Choose from options

Use the GET method.

Create your form!

Code method POST

Code method POST:

> End result:
>
> **Questions**
>
 unplash	What is your favorite sport?	Soccer
 unplash	Do you have a pet?	○ Yes ○ No
	What is his name?	
 unplash	What is your genre of music?	☐ Rock ☐ Electronic Music ☐ Pop ☐ Country ☐ Other
>
> Save

Make edits to your code to use the POST method.

Conclusion & Takeaways

Conclusion & Takeaways

An HTML form can be created using these HTML tags and elements:

- `<form>`
- `<label>`
- `<action>`
- `<input>`
  target attribute
  method attribute

An HTML form can be created using these HTML tags and elements: `<form>`, `<label>`, `<input>`, `<select>` and `<textarea>`. The target attribute specifies where a submitted form will be posted. The method attribute specifies the method that should be used to send the form data to the server. Different types of data can be sent using different methods.

Attribution

Source: Unsplash

Web Accessibility

Web Accessibility

>

Goals

In this lesson/discussion you will:

1. Learn about web accessibility and inclusive design principles
2. The different types of disabilities that user may exhibit when interacting with web pages
3. The laws surrounding web accessibility

Introduction

Web accessibility is the practice of making websites available to as many people as possible. This includes making sure that content is readable by people with disabilities, such as color blindness or visual impairments.

Accessibility is important for HTML web pages because it allows a wider audience to access the information and services on your website. If your site is not accessible, you are losing out on potential customers and clients who may be unable to use your site, which could damage your reputation and potentially impact your bottom line.

Exercise: What kinds of accessibility accommodations do you think we should keep in mind when building websites?

Web Accessibility and Disabilities

Web Accessibility

There are many different types of disabilities users may exhibit when interacting with a web page. Some people have trouble reading the text on the page, while others have trouble entering information into the form fields. Some people may have trouble seeing the entire page at once, while others may have trouble using a mouse or keyboard. Some people even have trouble navigating through pages that are too busy or too slow to load.

It is important to know about these disabilities so that you can make your website more accessible for everyone.

Types of Disabilities when interacting with a web page

When a user is navigating a web page, they may experience one or more of the following types of disabilities:

- Auditory: This can be anything from mild hearing loss to complete deafness.
- Cognitive: This includes cognitive and learning difficulties as well as mental health conditions such as anxiety, depression, or autism spectrum disorder.
- Neurological: This includes spinal cord injuries and mobility issues that affect motor skills and dexterity.
- Physical: This includes any physical impairment that affects movement or mobility, including arthritis and Parkinson's disease.
- Speech: This is any speech impairment that affects verbal communication skills, including stuttering and lisps.
- Visual: This is any visual impairment that affects sightedness or peripheral vision (such as macular degeneration).

Exercise 1:

Laws Requiring Web Accessibility

Laws Requiring Web Accessibility

Section 508 is a federal law that requires all electronic and information technology developed or procured by the Federal Government to be accessible to people with disabilities. This includes websites, mobile apps, and software programs used by agencies to conduct their business and/or provide services to the public.

WCAG 2.0 is a technical standard for developers who want to build accessible websites that meet the needs of individuals with disabilities. It's often referred to as "Web Content Accessibility Guidelines."
Exercise 2

Inclusive Design Principles

Inclusive Design Principles

Inclusive design is a set of principles that are used to create user interfaces that are accessible to people with disabilities and that meet the needs of all users. These principles are:

- Provide comparable experience: If you offer an alternative for one type of content, you should also provide an alternative for another type of content (e.g., if you have a video, you should also have a transcript).
- Consider the situation: Make sure your site is usable regardless of the user's situation (e.g., whether they're using assistive technology or not).
- Be consistent: Any change in the interface should be consistent across the whole site (e.g., don't make one button green and another red).
- Give control: Give users control over their experience by allowing them to customize it as much as possible (e.g., allow them to change font size).
- Offer choice: Offer choices whenever possible so users can customize their experience even further (e.g., allow users to choose between two different color schemes).
- Prioritize content: Content should always be accessible above anything else on your site, including advertising or branding (e.g., put text before images.

Aria-Label

The HTML aria-label attribute defines a string value that labels an interactive element. The aria-label attribute does not provide any visual indication of the label's contents; it is for accessibility purposes only.

The aria-label attribute should be used to add descriptive text to an element that will be read by assistive devices such as screen readers. This text should describe what the element represents, such as "Search form".

Here is an example of the code:

```
<a href="https://giphy.com/gifs/calendar-groundhog-day-patrick-hosmer-QC7a49iu2NEwHCqp6b" aria-label="gif">via GIPHY</a></p>
```

Conclusion & Takeaways

Conclusion & Takeaways

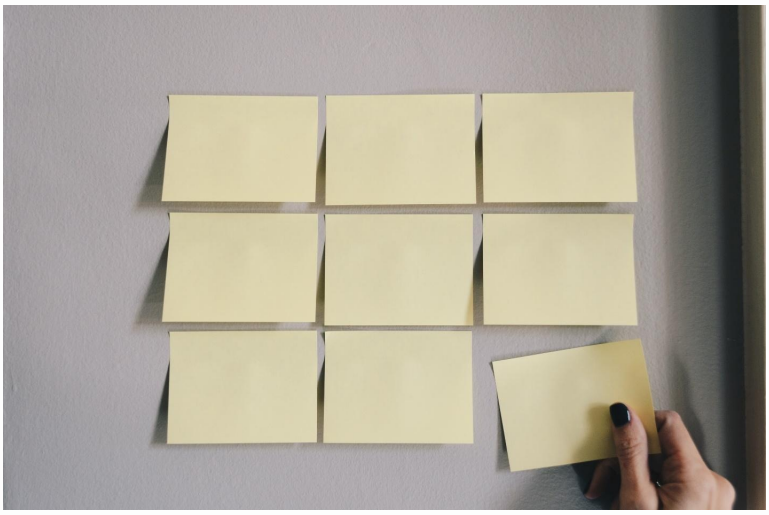
Adhering to the type of disabilities users may have when interacting with a web page will help you become a better developer. There are also laws in place that require web accessibility such as Section 508 and WCAG 2.0.

Attribution

Section508.gov. (n.d.). Retrieved November 21, 2022, from https://www.section508.gov/develop/applicability-conformance

Advanced HTML Review

Advanced HTML Review



cover image

Goals

In this lesson/discussion you will:
- Review HTML tags and elements

Introduction

HTML has a lot of features, but there are a few things you should know. Here is a quick review of some of the most advanced HTML tags and their uses.

Images: Images are one of the most important parts of your website, and HTML makes it easier than ever to add images to your pages. They can be used for logos, pictures, and even videos.

Favicons: A favicon is a small icon that appears next to the title of your website in browsers like Firefox and Chrome. They're useful because they let users know right away which page, they're on and make it easy for them to find their way back there later.

Tables: Tables are great tools for organizing content on your page—you can use them to display data in columns and rows, or just as an organizational tool for other elements on the page.

File paths: File paths allow you to specify where files will be located on your server when they're uploaded so that they'll load properly on your website or app. You can include them directly in URLs as well as in CSS code or `<script>` tags (the latter two options being especially helpful).

Iframe: is a nested HTML document that is placed inside another document or web page. It can be used to embed content from another site, but it can also be used to display a portion of your own website in a separate window. An iframe can be used to display any kind of content, including text, images, video and even other HTML documents. The main thing you need to know about iframes is that they are isolated from the document that contains them. This means that JavaScript code contained inside an iframe will not run on its parent web page unless you explicitly tell it to do so.

Forms: HTML forms `<form>` are used to allow users to enter data and submit it to a server. This input is usually collected from multiple fields, as well as required fields that must be filled out in order for the form to be submitted. When the user submits a form, an HTTP request is sent to a server and the response is parsed by JavaScript or PHP. This is how we receive and process data from a webpage.

```
<form action="/example_action.html">
        <label for="wday">Day: </label>
        <input type="text" id="fname" name="fname"><br><br>
        <input type="submit" value="Submit">
    </form>
```

> Some HTML elements used inside of form:
> Labels:
> Syntax `<label>``</label>`
> Example: `<label for="wday">Day: </label>`

Input text:
Syntax `<input type="text"></input>`
Example: `<input type="text" id="fname" name="fname"><br><br>`

Input submit:
Syntax `<input type="submit"></input>`
Example:`<input type="submit" value="Submit">`

Radio button:
Syntax `<input type="radio"></input>`
Example:

```
<input type="radio" id="sunday" name="day" value="Sunday">
        <label for="sunday">Sunday</label>
```

Button:
Syntax `<input type="button"></input>`
Example:`<input type="button" onclick="window.location.href='https://www.correlation-one.com/';" value="Click Example" />`

Dropdown:
Syntax `<select></select>`
Example:

```
<select name="fgender" id="fgender">
            <option value="Female">Female</option>
            <option value="Male">Male</option>
        </select>
```

For know more about of forms, please click here

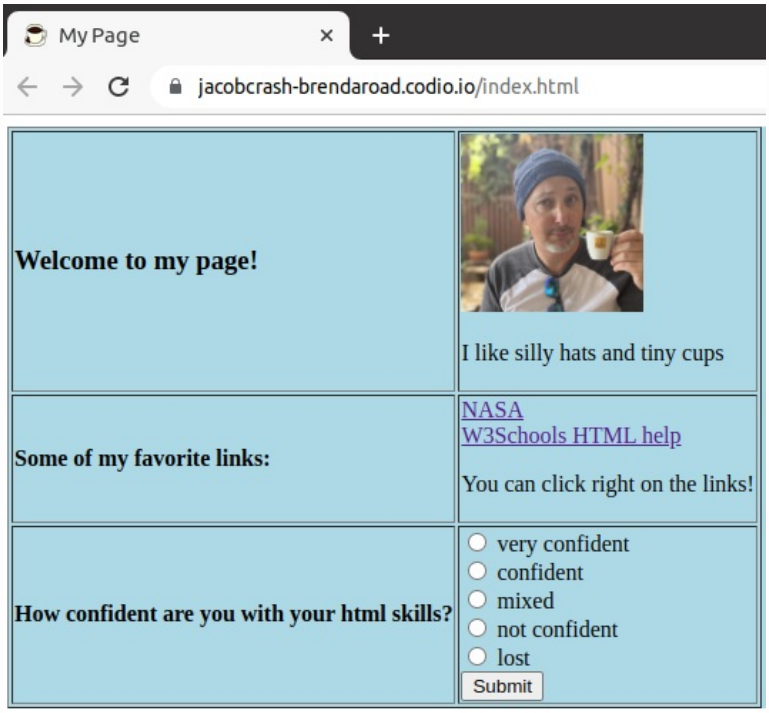
The HTML aria-label attribute defines a string value that labels an interactive element. The aria-label attribute does not provide any visual indication of the label's contents; it is for accessibility purposes only.

Here is an example of the code:
`<a href="https://giphy.com/gifs/calendar-groundhog-day-patrick-hosmer-QC7a49iu2NEwHCqp6b" aria-label="gif">via GIPHY</a></p>`
Exercise

Practice

Practice what we've learned

Recreate a webpage using this format:



page example

The page should be about yourself. To upload your own image, go to `File/Upload` and select the image. The path to the image will be the name of the image, such as `<img src="image1.jpg" alt="My image" width="128">`.

Be sure to include the following:
- Choose a background color. This one is light blue
- Add an image. Be sure to include alt text!
- Choose a favicon. This one is a picture of a coffee cup.
- Have the content embedded in a table with 3 rows and 2 columns.
- Include two links.
- Include a form input, either radio buttons or check boxes.

Feel free to ask questions! This is a great way to check your knowledge. When you've completed it, submit your code.

Conclusion & Takeaways

Conclusion & Takeaways

> You should be comfortable with:

- tags
- favicons
- tables
- links
- images
- forms

Remember to consider accessibility:

Adhering to the type of disabilities users may have when interacting with a web page will help you become a better developer. There are also laws in place that require web accessibility such as Section 508 and WCAG 2.0.

Attribution

Source: Unsplash

Recreate an HTML webpage

Advanced HTML & Webpage Exercise

Goals
In this exercise, you will:
1. Review some advanced html tools and tags
2. Recreate a web page utilizing advanced HTML elements

> ### Introduction
>
> There are more html tags, tools, and tricks than we can possibly cover. We'll see a few more of them here, and then in the exercise, you will recreate a web page using some of these advanced html elements.

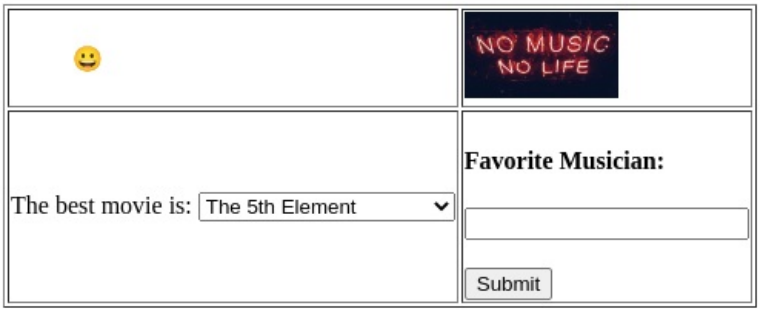
Meta
The `<meta>` element. This is another tag that doesn't need to be closed. It is also not displayed on the web page. It should be in the `<head>` of your page, and it is useful for browsers for how to render the page, or for search engines looking for keywords. When you use the `!` trick and prepopulate the basic html content, you'll see that meta adds default information on charset (utf-8 is unicode, the universal alphabet), http-equiv (browser support), and name (info for search engines).
Often people will add keywords for search engines, like this: `<meta name="keywords" content="html, resume, web development">`
You can also use meta to store author information, or general description information.

Sections
Some HTML elements are useful for breaking down components you're going to defining the header `<header>`, an area for navigation links `<nav>`, break the content into sections using `<section>`, an area defined for a self-contained piece of information like an article `<article>`, and a footer area `<footer>`.

Emojis
Emojis are similar to entities, and require that they be written like this, `😀`, which is the smile emoji 😀. Changing the last two digits will give you other options, and you can find lists of emoji codes.

Image sizes
Images are uploaded as they are, unless you are uploading them to be favicons. In order to resize the image, you can change both height and width. You really only need to adjust one, as it will maintain the aspect

ratio and change the other. Example: `<img src="my_image.jpg" alt="My image" width="128">`. The 128 tells it how many pixels to make it. This can be useful maintaining an image size instead of using the full size, or resizing to match window size.
>
Instructions
>
Recreate the web page below utilize the advanced HTML elements. Be sure to include what you see here as well as: keywords in a meta tag, a few movie examples in the dropdown in the lower left box, and a resized image. Spacing is tricky in html, so we use `<blockquote> [emoji code] </blockquote>` to indent it here even though blockquote is usually used for text. Pick your own favorite emoji instead of the one in the page here. Remember, we start with a table to give us the general layout.



page to copy

Post your code in the text editor.

Conclusion & Takeaways

Conclusion & Takeaways

There are different HTML elements that are necessary to create a professional-looking web page. These include headings, paragraphs, lists, tables, and images. We also learned how to use these elements together to create a more complex web page design.

Personal Website: Create an HTML Landing Page

Personal Website: Create an HTML Landing Page

computer with code

Goals

In this exercise, you will:
1. Define the message for your personal website
2. Choose the HTML elements
3. Create an HTML landing page

Introduction

In this exercise, you will create a landing page for your HTML website. Landing pages are single-page websites that serve as the home page of your website. They are typically used to collect email addresses or other contact information from visitors who want more information about your product or service.

The goal of this exercise is to build an HTML landing page that includes:

- A header and footer with standard elements such as navigation, branding, and search functionality
- Content area with text and images that explain what you do and how you do it
- A call-to-action button that encourages visitors to sign up for something (like a request for information where they submit a

question and their contact information.

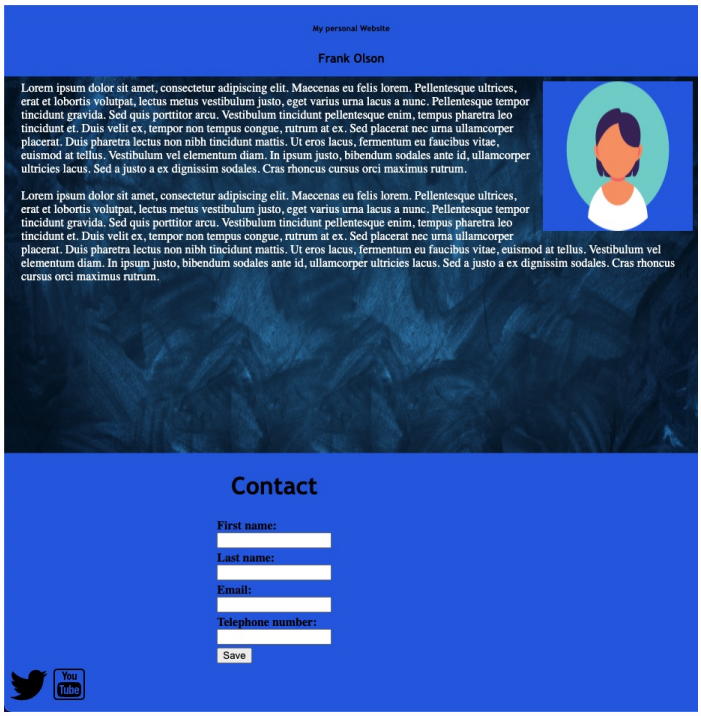
Instructions

- Create an HTML landing page
- Utilize the advanced HTML elements

You'll need to do the following:
1. Design a header for your landing page. This should include an image and text that clearly tell visitors what they can find on the site.
2. Create an introduction to your site, which should provide details about the purpose and mission of your site. Include some information about yourself and why you've created it in this section as well.
3. Design an About Me section that includes more detailed information about yourself and why you created this website. This should include links to other websites where people can find out more about you if they'd like additional information (such as LinkedIn).
4. Write out at least two paragraphs of text explaining what each section of your site contains (for example, if there's a blog or news feed on the site). You don't need to write out every single piece of content—just enough so that someone visiting the site gets an idea of what it looks like without having to click around too much first!
5. Write a blog post about your journey to become a web developer! Summarize what you've learned so far, and what you've found most interesting.
6. A call-to-action button that encourages visitors to sign up for something (like an email newsletter) or make a purchase

Post your code in the text editor below

An example of a landing page with the tools that we've learned until these moment:



example page

Once you've built your page, submit your code.

Conclusion & Takeaways

Conclusion & Takeaways

> The goal of this exercise is to provide a brief overview of the features and content of your site, so that anyone who visits it will immediately know what to expect.

You've made excellent progress, improving on your initial "About Me" page from last week and creating something that's on its way to being your professional portfolio!

Remember that this takes lots of practice. Keep all of the code you write in a text editor or your own codepen.io page, or somewhere else you can easily access it. Try to recreate some of these materials from memory. Getting good at this takes time and practice.
>

"Experience is the name everyone gives to their mistakes." – Oscar Wilde

"Always code as if the [person] who ends up maintaining your code will be a violent psychopath who knows where you live" — John Woods

> It should also be noted that everyone who has experience coding has, at some point, read some code and said "What idiot wrote this?" only to find, later, that they wrote it themselves a couple years ago.

Attribution

> Source: Unsplash