Lecture 2

HTML

HTML Syntax

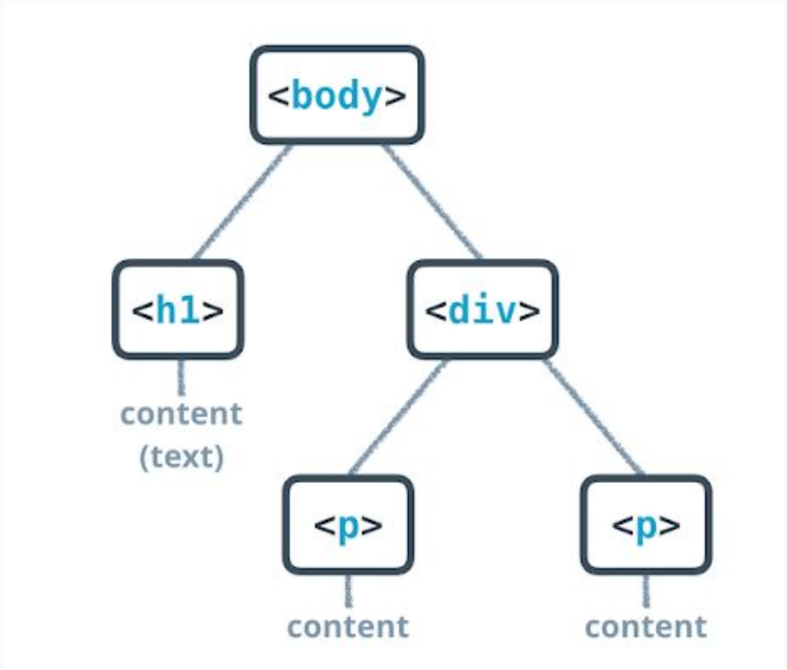
Topics

- **HTML Syntax**
- **Path/Links**
- **Forms**
  - **Input Elements**
  - **Select Element**
  - **Button Elements**
- **Global Attributes**
- **HTML 5**

HTML Syntax

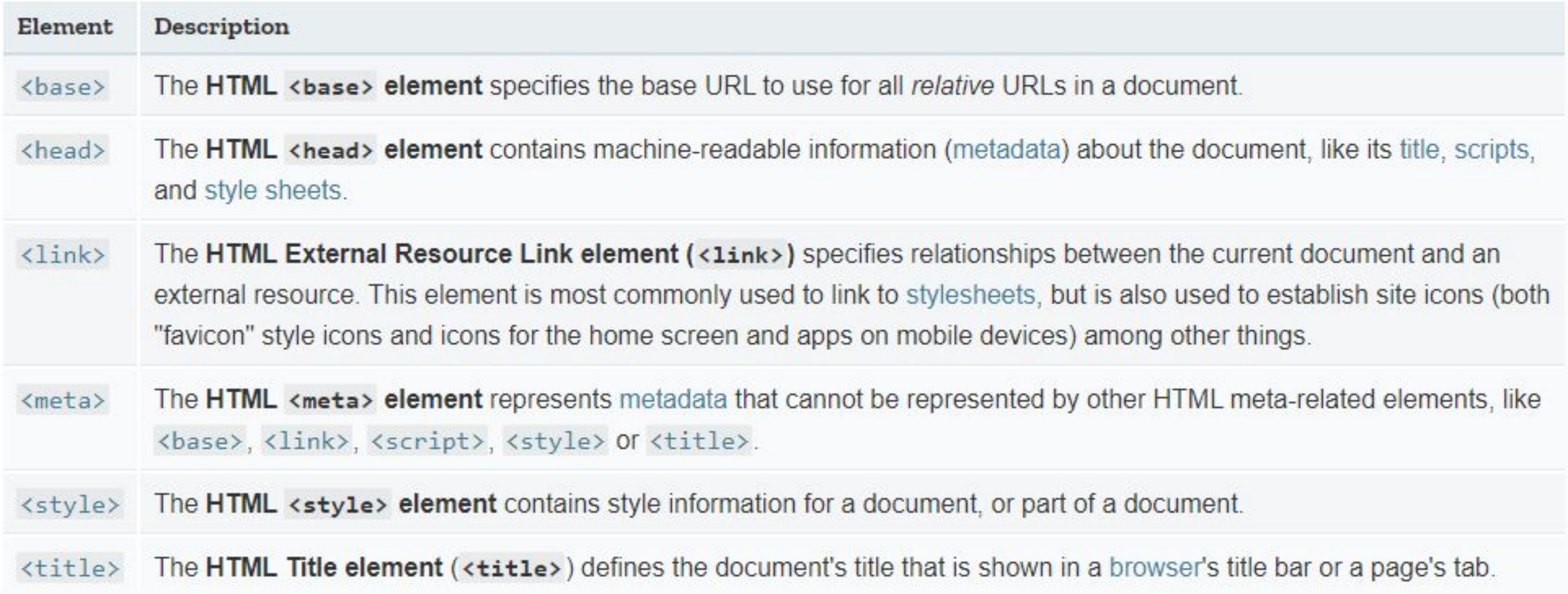
HTML Tree Recap

Each **HTML** document can actually be referred to as a document **tree**. We describe the elements in the **tree** like we would describe a family **tree**. There are ancestors, descendants, parents, children and siblings. It is important to understand the document **tree** because CSS selectors use the document **tree**.



HTML Document Recap

Document Metadata contains information about the page. This includes information about styles, scripts and data to help software (search engines, browsers, etc.) use and render the page.



Element	Description
`<base>`	The **HTML `<base>` element** specifies the base URL to use for all *relative* URLs in a document.
`<head>`	The **HTML `<head>` element** contains machine-readable information (metadata) about the document, like its title, scripts, and style sheets.
`<link>`	The **HTML External Resource Link element (`<link>`)** specifies relationships between the current document and an external resource. This element is most commonly used to link to stylesheets, but is also used to establish site icons (both "favicon" style icons and icons for the home screen and apps on mobile devices) among other things.
`<meta>`	The **HTML `<meta>` element** represents metadata that cannot be represented by other HTML meta-related elements, like `<base>`, `<link>`, `<script>`, `<style>` or `<title>`.
`<style>`	The **HTML `<style>` element** contains style information for a document, or part of a document.
`<title>`	The **HTML Title element (`<title>`)** defines the document's title that is shown in a browser's title bar or a page's tab.

HTML Tags

As you will realize when you build websites, you can write **HTML incorrectly** — by forgetting a closing tag, for example — and the browser will try to correct your mistake. Sometimes the browser will guess correctly and your website will look fine. But most of the time, it will be wrong and strange bugs will start appearing.

This is a good time to double check that you're creating elements correctly because you'll be writing more complicated HTML soon!

You create all html elements: <tag>content</tag>.

Remember every start <tag>, needs to closed with end </tag>

Header Tags

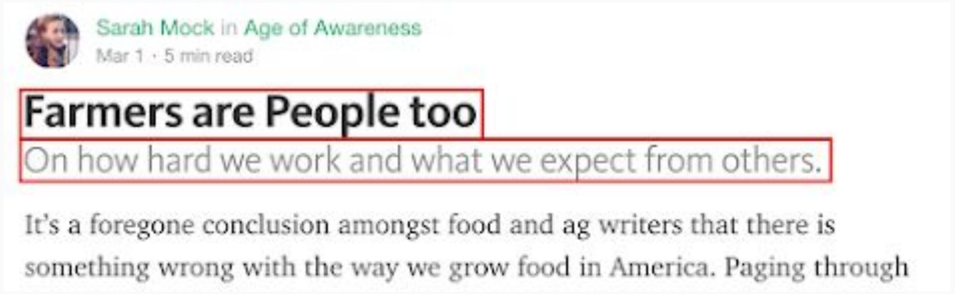
Websites have different ways to show you, the user, what is important. Take for example…

Big Important Words!

This style of text **Big Important Words!** is called a **header**. This should look familiar because almost every website uses headers.

Main Headings and Subheadings

It's also pretty common to see more than one kind of header being used. Here's an example of two headers on Medium.



Sarah Mock in Age of Awareness
Mar 1 · 5 min read

Farmers are People too

On how hard we work and what we expect from others.

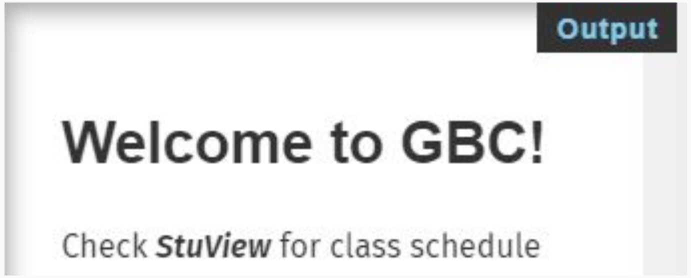
It's a foregone conclusion amongst food and ag writers that there is something wrong with the way we grow food in America. Paging through

<header> element

The **HTML <header> element** represents introductory content, typically a group of introductory or navigational aids. It may contain some heading elements but also a logo, a search form, an author name, and other elements.

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/header

```
<header>
    <h1>Welcome to GBC!</h1>
</header>

<main>
    <p>Check <em>StuView</em> for class schedule</p>
</main>
```


Output

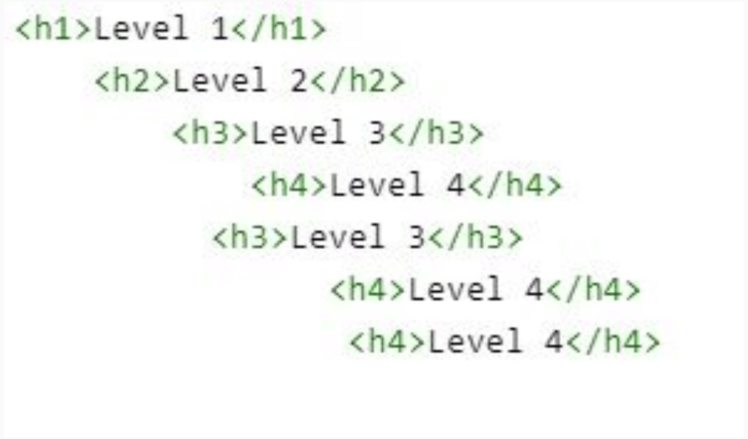
Welcome to GBC!

Check ***StuView*** for class schedule

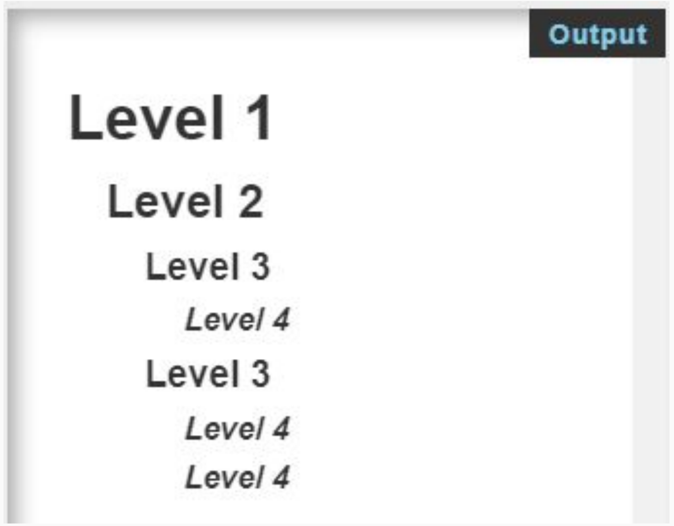
<h1-h6> elements

The **HTML <h1>–<h6> elements** represent six levels of section headings. <h1> is the highest section level and <h6> is the lowest.

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Heading_Elements



```
<h1>Level 1</h1>
    <h2>Level 2</h2>
        <h3>Level 3</h3>
            <h4>Level 4</h4>
          <h3>Level 3</h3>
                <h4>Level 4</h4>
                 <h4>Level 4</h4>
```



Output

Level 1

Level 2

Level 3

Level 4

Level 3

Level 4

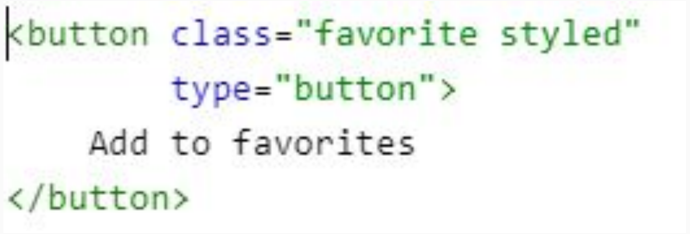
Level 4

<button> element

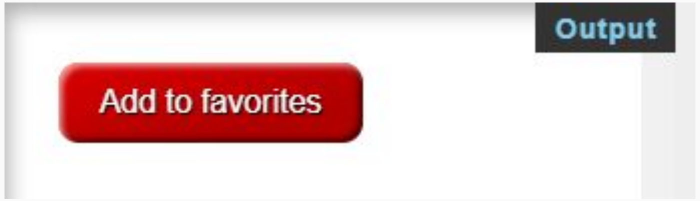
The **HTML <button> element** represents a clickable button, used to submit forms or anywhere in a document for accessible, standard button functionality. By default, HTML buttons are presented in a style resembling the platform the user agent runs on, but you can change buttons' appearance with CSS.

Buttons may look different based on browser and operating system.

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/button



```
<button class="favorite styled"
        type="button">
    Add to favorites
</button>
```

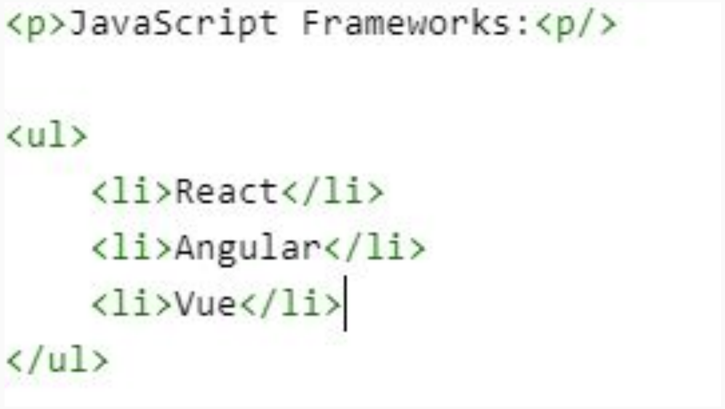


<ul> element - unordered lists

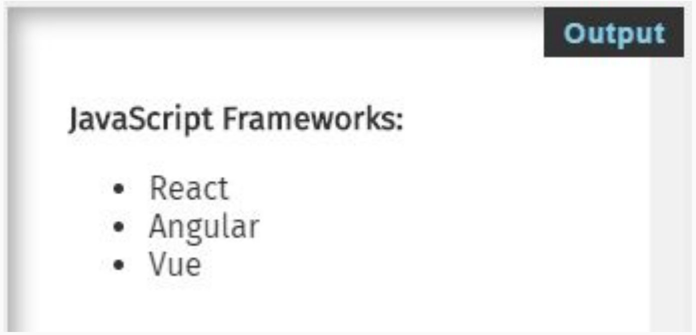
The **HTML <ul> element** represents an unordered list of items, typically rendered as a bulleted list.

The **HTML `<li>` element** is used to represent an item in a list. It must be contained in a parent element: an ordered list (`<ol>`), an unordered list (`<ul>`), or a menu (`<menu>`). In menus and unordered lists, list items are usually displayed using bullet points. In ordered lists, they are usually displayed with an ascending counter on the left, such as a number or letter.

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul
https://developer.mozilla.org/en-US/docs/Web/HTML/Element/li



```
<p>JavaScript Frameworks:<p/>

<ul>
    <li>React</li>
    <li>Angular</li>
    <li>Vue</li>
</ul>
```

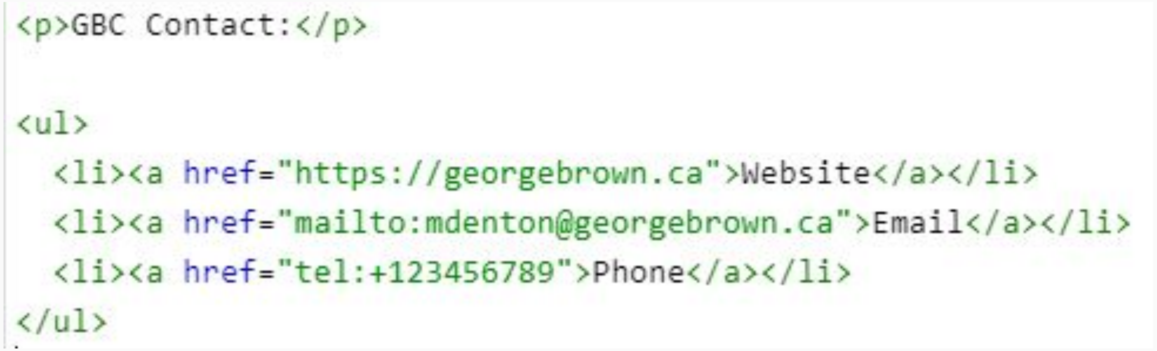


```
Output

JavaScript Frameworks:

  • React
  • Angular
  • Vue
```

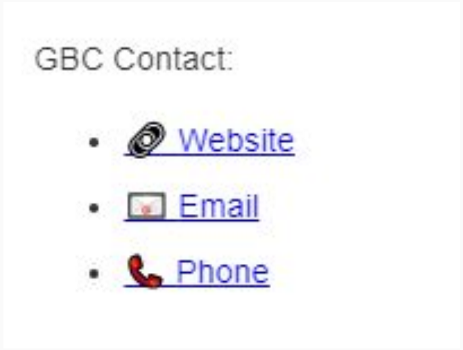
<a> hyperlinks

The power of the web is that pages can lead to other pages. When you click on a link on a web page, it takes you to another page. This link is called a **hyperlink**.

The **HTML <a> element** (or *anchor* element), with its `href` attribute, creates a hyperlink to web pages, files, email addresses, locations in the same page, or anything else a URL can address. Content within each `<a>` **should** indicate the link's destination.:



```
<p>GBC Contact:</p>

<ul>
  <li><a href="https://georgebrown.ca">Website</a></li>
  <li><a href="mailto:mdenton@georgebrown.ca">Email</a></li>
  <li><a href="tel:+123456789">Phone</a></li>
</ul>
```



GBC Contact:

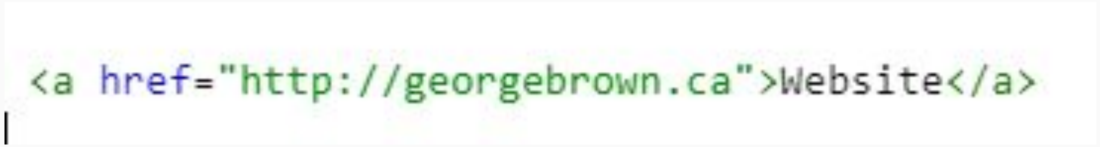
- Website
- Email
- Phone

<a> hyperlinks cont..

Inside the opening `a` tag there is `href`, which stands for "reference." This is called an **attribute**. Attributes like `href` describe the properties of HTML elements. In this case, the `href` attribute is the target URL that the link will open. The content inside the anchor element is the text that users see displayed on the page.

This is the format that you must use when you make hyperlinks! Note:

- There is a space between `a` and `href`.
- There are no spaces around the `=`.
- The website has two `"` around it.
- There are no spaces between the `href` attribute and the `>` of the opening tag.



```
<a href="http://georgebrown.ca">Website</a>
```

<img> image elements

- HTML supports various multimedia resources such as images, audio, and video.
- The **HTML `<img>` element** embeds an image into the document.
- The source attribute, `src`, is like the `href` of a link - it is the URL of the image you want to display. For now, your images will need to be hosted online, which means that the URL will need to start with `http://` or `https://`.

  https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img

```
<img class="fit-picture"
     src="/media/cc0-images/grapefruit-slice-332-332.jpg"
     alt="Grapefruit slice atop a pile of other slices">
```

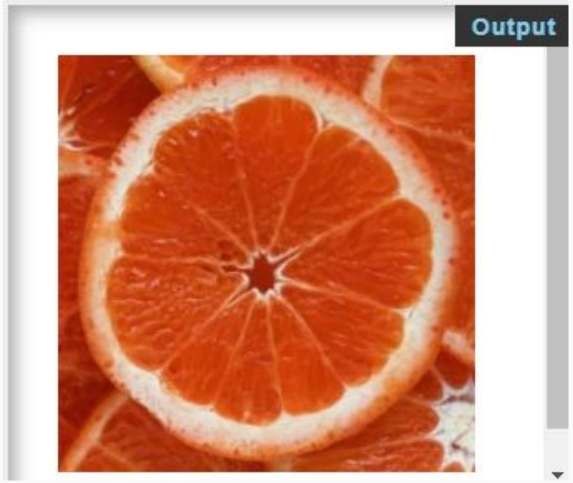
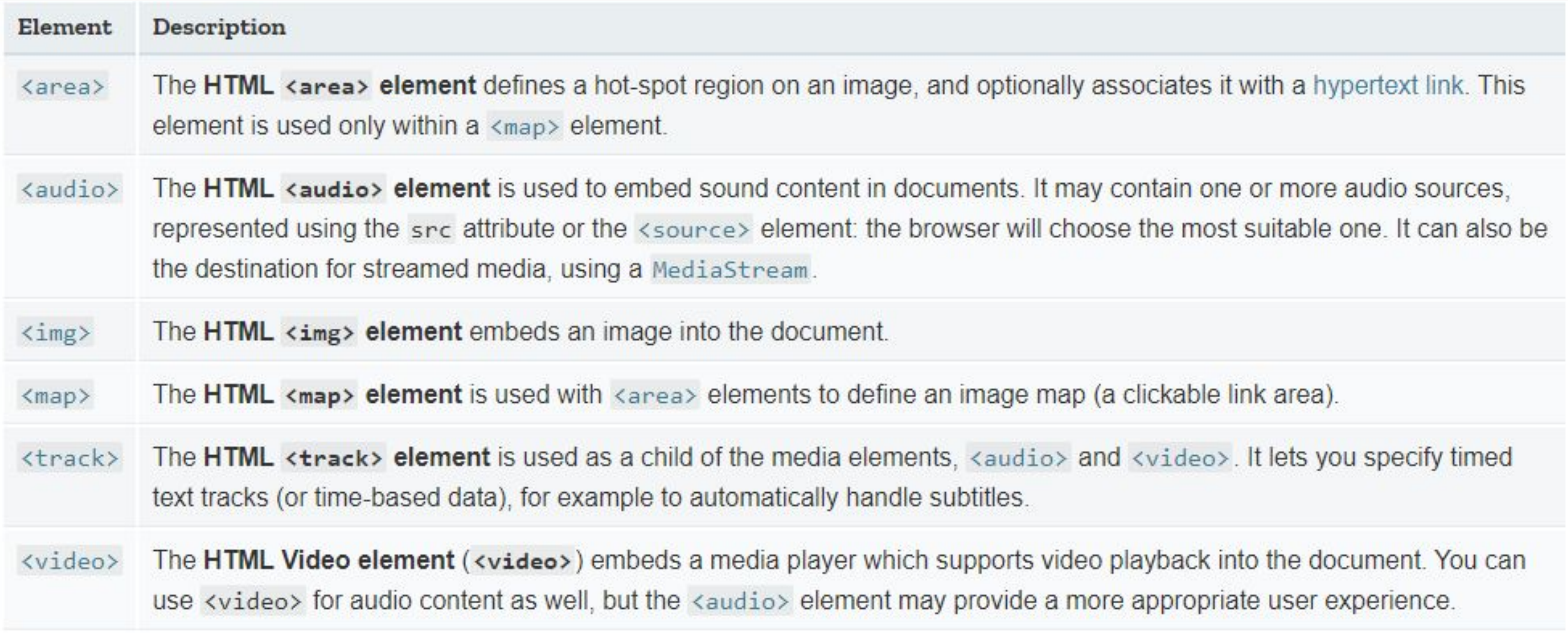


Image and Multimedia

- HTML supports various multimedia resources such as images, audio, and video.



Element	Description
`<area>`	The **HTML `<area>` element** defines a hot-spot region on an image, and optionally associates it with a hypertext link. This element is used only within a `<map>` element.
`<audio>`	The **HTML `<audio>` element** is used to embed sound content in documents. It may contain one or more audio sources, represented using the `src` attribute or the `<source>` element: the browser will choose the most suitable one. It can also be the destination for streamed media, using a `MediaStream`.
`<img>`	The **HTML `<img>` element** embeds an image into the document.
`<map>`	The **HTML `<map>` element** is used with `<area>` elements to define an image map (a clickable link area).
`<track>`	The **HTML `<track>` element** is used as a child of the media elements, `<audio>` and `<video>`. It lets you specify timed text tracks (or time-based data), for example to automatically handle subtitles.
`<video>`	The **HTML Video element** (**`<video>`**) embeds a media player which supports video playback into the document. You can use `<video>` for audio content as well, but the `<audio>` element may provide a more appropriate user experience.

Paths/Links

Paths

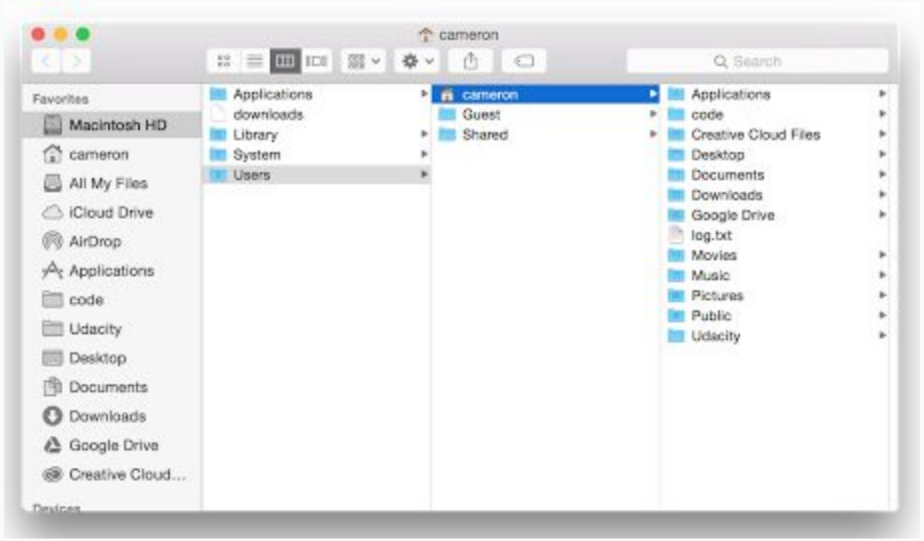
- In order to display a local image, you need to be able to write a **path**.
- If there is a file called `index.html` in a directory and there is another directory called `example/` in the same directory, you can access any files in `example/` from `index.html` with the URL (path) `example/filename.html`, e.g. `<a href="example/filename.html">Example Path</a>` .

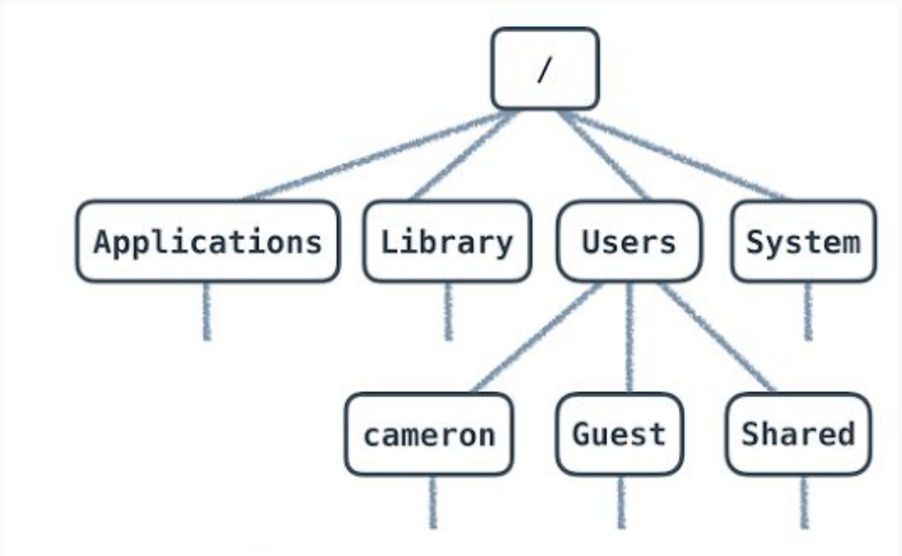
Local Paths

- Computers have folders (also called "directories"). Operating systems like Windows, Mac and Linux organize *all* of your files into a tree of directories called a **file system**. There's a top-most directory, often called the **root**, that contains all of the other directories. Within the root, there are files and directories. Within those directories are more files and more directories. And within those directories are even more files and directories, and so on.

Paths cont..

- Compare this part of the file system on a computer to a tree diagram showing the same directory structure:





Tree diagram of directory structure

Absolute vs Relative Path/Links

- One of the more tricky and confusing things about HTML is linking to other pages and sites, especially when absolute and relative paths come into play.

```
<a href="linkhere.html">Click Me</a>
```

Relative Paths

- index.html
- /graphics/image.png
- /help/articles/how-do-i-set-up-a-webpage.html

Absolute Paths

- http://www.mysite.com
- http://www.mysite.com/graphics/image.png
- http://www.mysite.com/help/articles/how-do-i-set-up-a-webpage.html

Absolute vs Relative Path/Links cont..

- The first difference you'll notice between the two different types of links is that **absolute paths** *always* include the domain name of the website, including **http://www.**,
- **Relative links** only point to a file or a file path. When a user clicks a relative link, the browser takes them to that location on the current site.

```
<a href="../about.html>Learn more about my Website.</a>
```

- When the browser sees **../** in front of the filename, it looks in the folder above the current folder. You can use this as many times as you need to. You can also tell the browser to look in a subfolder of the directory above the current one.

Absolute Paths

- Absolute paths provide the complete website address where you want the user to go. An absolute link would look like this:

  ```
  <a href="http://www.coffeecupshop.com">Click here to visit CoffeeCup Shop.</a>
  ```

- You *must* use absolute paths when linking to another Website, but you can also use absolute paths within your own website. This practice is generally frowned upon, though. Relative links make it easy to do things like change your domain name without having to go through all your HTML pages, hunting down links and changing the names.

- As an added bonus, they force you to keep your site structure neat and organized, which is always a good idea.

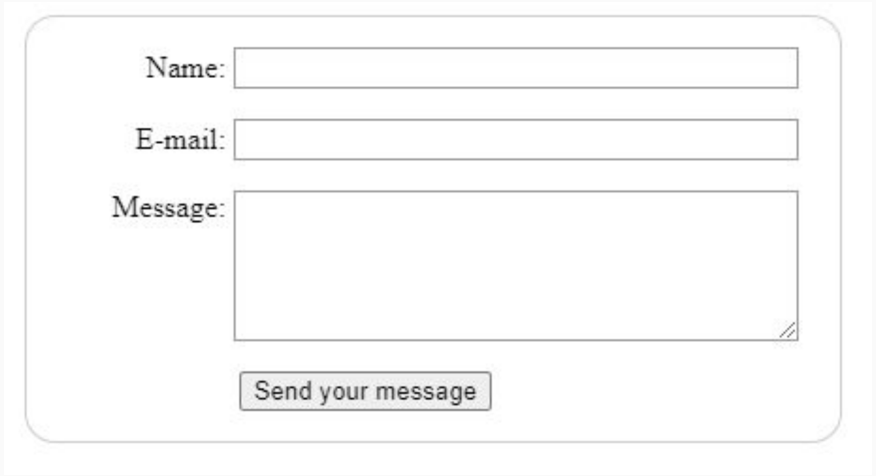
Forms

Web Forms

- HTML provides a number of elements which can be used together to create forms which the user can fill out and submit to the Web site or application. There's a great deal of further information about this available in the HTML forms guide.

- Mastering forms however requires more than just HTML knowledge — you also need to learn some specific techniques to style form controls, and some scripting knowledge is required to handle things like validation and creating custom form controls.

- Interactive web forms on the internet are powered by CSS and JavaScript.

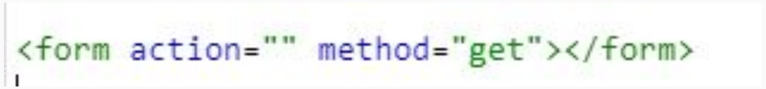


<form> element

- The **<form>** element formally defines a form and attributes that determine the form's behavior. Each time you want to create an HTML form, you must start it by using this element, nesting all the contents inside.
https://developer.mozilla.org/en-US/docs/Web/HTML/Element/form

- Many assistive technologies and browser plugins can discover **<form>** elements and implement special hooks to make them easier to use.

- The **HTML `<form>` element** represents a document section containing interactive controls for submitting information.



```
<form action="" method="get"></form>
```

- There are many different form controls, or widgets that can be nested within the <form> element and provide functionality and different options to collection user data.

- The following are some example of form controls: **`<form>`**, **`<fieldset>`**, **`<legend>`**, **`<textarea>`**, **`<label>`**, **`<button>`**, and **`<input>`**.

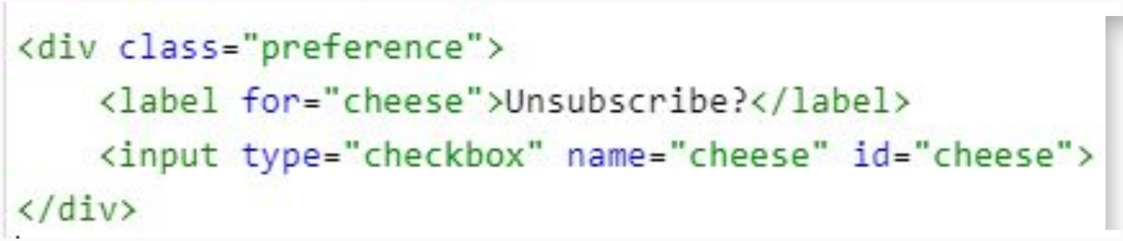
<label> element

- The **HTML `<label>` element** represents a caption for an item in a user interface.

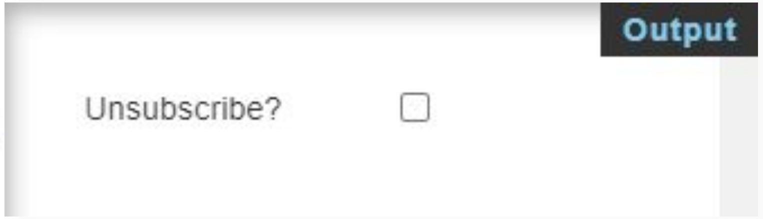
Associating a `<label>` with an `<input>` element offers some major advantages:

- The label text is not only visually associated with its corresponding text input; it is programmatically associated with it too. This means that, for example, a screenreader will read out the label when the user is focused on the form input, making it easier for an assistive technology user to understand what data should be entered.

- You can click the associated label to focus/activate the input, as well as the input itself. This increased hit area provides an advantage to anyone trying to activate the input, including those using a touch-screen device.

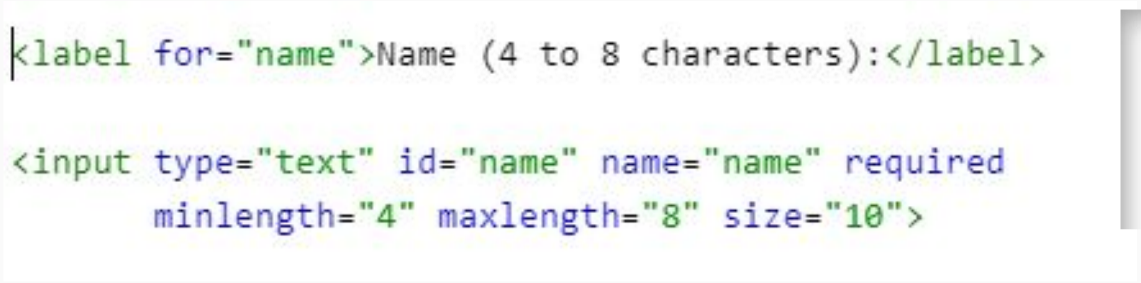


```
<div class="preference">
    <label for="cheese">Unsubscribe?</label>
    <input type="checkbox" name="cheese" id="cheese">
</div>
```

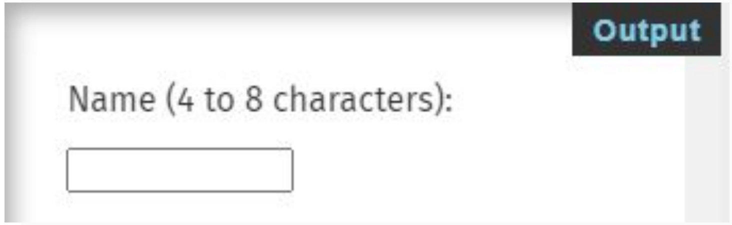


<input> element

- The **HTML `<input>` element** is used to create interactive controls for web-based forms in order to accept data from the user; a wide variety of types of input data and control widgets are available, depending on the device and user agent.
- The `<input>` element is one of the most powerful and complex in all of HTML due to the sheer number of combinations of input types and attributes.



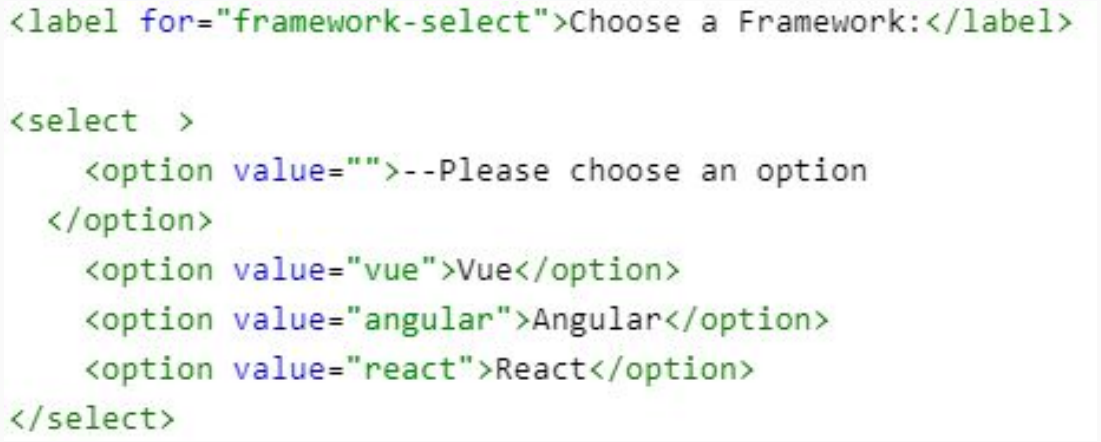
```
<label for="name">Name (4 to 8 characters):</label>

<input type="text" id="name" name="name" required
       minlength="4" maxlength="8" size="10">
```



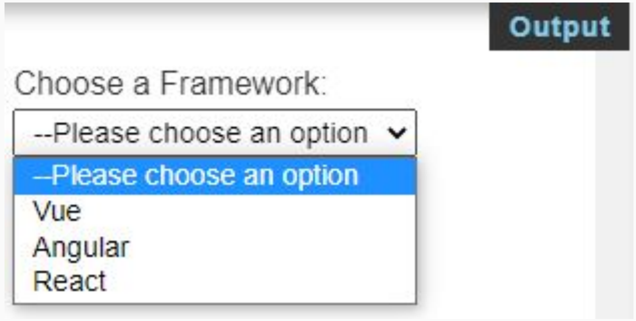
<select> element

- The **HTML `<input>` element** is used to create interactive controls for web-based forms in order to accept data from the user; a wide variety of types of input data and control widgets are available, depending on the device and user agent.

- Each `<option>` element should have a `value` attribute containing the data value to submit to the server when that option is selected. If no `value` attribute is included, the value defaults to the text contained inside the element. You can include a `selected` attribute on an `<option>` element to make it selected by default when the page first loads.



```
<label for="framework-select">Choose a Framework:</label>

<select  >
    <option value="">--Please choose an option
  </option>
    <option value="vue">Vue</option>
    <option value="angular">Angular</option>
    <option value="react">React</option>
</select>
```



Output

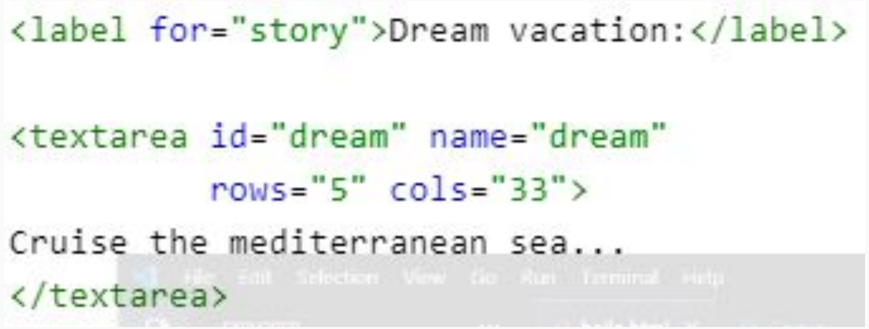
Choose a Framework:

```
--Please choose an option ⌄
--Please choose an option
Vue
Angular
React
```

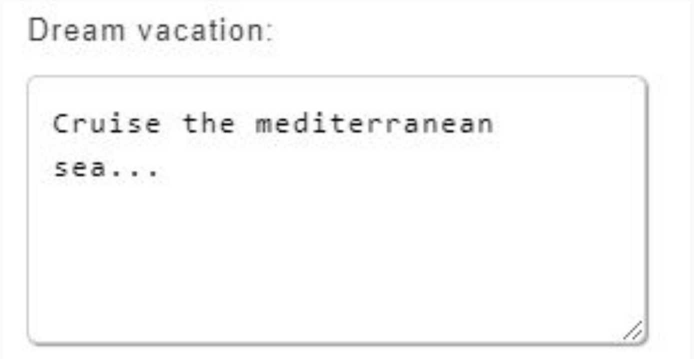
<textarea> element

- The **HTML `<textarea>` element** represents a multi-line plain-text editing control, useful when you want to allow users to enter a sizeable amount of free-form text, for example a comment on a review or feedback form.

  The above example demonstrates a number of features of `<textarea>`:
- An `id` attribute to allow the `<textarea>` to be associated with a `<label>` element for accessibility purposes
- A `name` attribute to set the name of the associated data point submitted to the server when the form is submitted.
- `rows` and `cols` attributes to allow you to specify an exact size for the `<textarea>` to take. Setting these is a good idea for consistency, as browser defaults can differ.
- Default content entered between the opening and closing tags. `<textarea>` does not support the `value` attribute.



```
<label for="story">Dream vacation:</label>

<textarea id="dream" name="dream"
          rows="5" cols="33">
Cruise the mediterranean sea...
</textarea>
```



Dream vacation:

```
Cruise the mediterranean
sea...
```

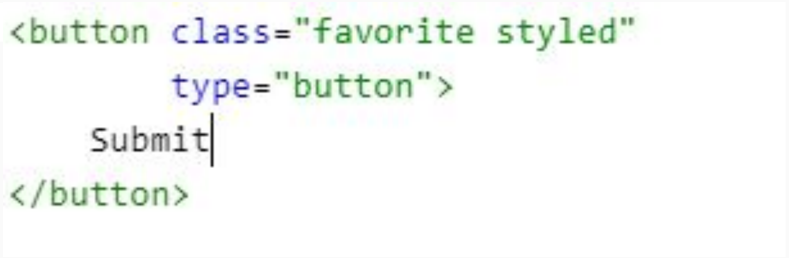
<button> element

- The **HTML `<button>` element** represents a clickable button, used to submit forms or anywhere in a document for accessible, standard button functionality. By default, HTML buttons are presented in a style resembling the platform the user agent runs on, but you can change buttons’ appearance with CSS.

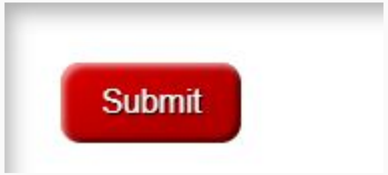
- Attributes

- **`disabled`**
  This Boolean attribute prevents the user from interacting with the button: it cannot be pressed or focused.

  Firefox, unlike other browsers, persists the dynamic disabled state of a `<button>` across page loads. Use the `autocomplete` attribute to control this feature.

- 

```
<button class="favorite styled"
        type="button">
    Submit
</button>
```



Global Attributes

Global Attributes

- **Global attributes** are attributes common to all HTML elements; they can be used on all elements, though they may have no effect on some elements.

- Global attributes may be specified on all HTML elements, *even those not specified in the standard*. That means that any non-standard elements must still permit these attributes, even though using those elements means that the document is no longer HTML5-compliant. For example, HTML5-compliant browsers hide content marked as `<foo hidden>...</foo>`, even though `<foo>` is not a valid HTML element.

- **List of global attributes**
- https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes

- **`class`**
- A space-separated list of the classes of the element. Classes allows CSS and JavaScript to select and access specific elements via the class selectors or functions like the method `Document.getElementsByClassName()`.
-

HTML 5

What is HTML 5?

- HTML5 is the new standard for HTML
- The previous version of HTML was - HTML 4.01, came in 1999.
- HTML5 is designed to deliver almost everything you want to do online without requiring additional plugins.
- It does everything from animations to apps, music to movies, and can be used to build complicated applications that run in your browser
- HTML5 is cross-platform (it does not care what device you are using tablet, smartphone, notebook or SmartTV)

Differences Between HTML4 & HTML5

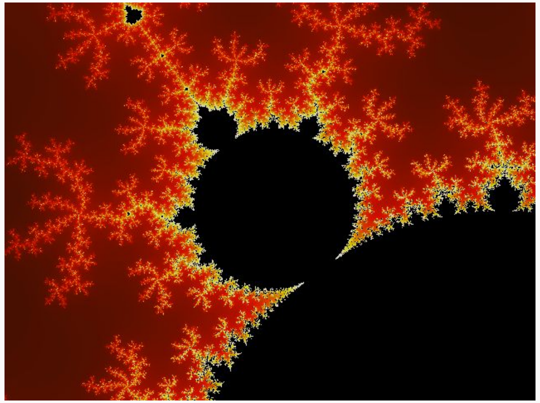
1. HTML5 is a work in progress
2. Simplified Syntax
3. The new **<canvas>** Element for 2D drawings
4. New content-specific elements like **<article>, <header>,<footer>,<nav>,<section>**
5. New **<menu>** and **<figure>** Elements
6. New **<audio>** and **<video>** Elements
7. New form controls like calendar, date, time, email, url, search
8. No more **<frame>, <center>, <big>**, and **<b>, <font>**
9. Support for local storage

<canvas> element

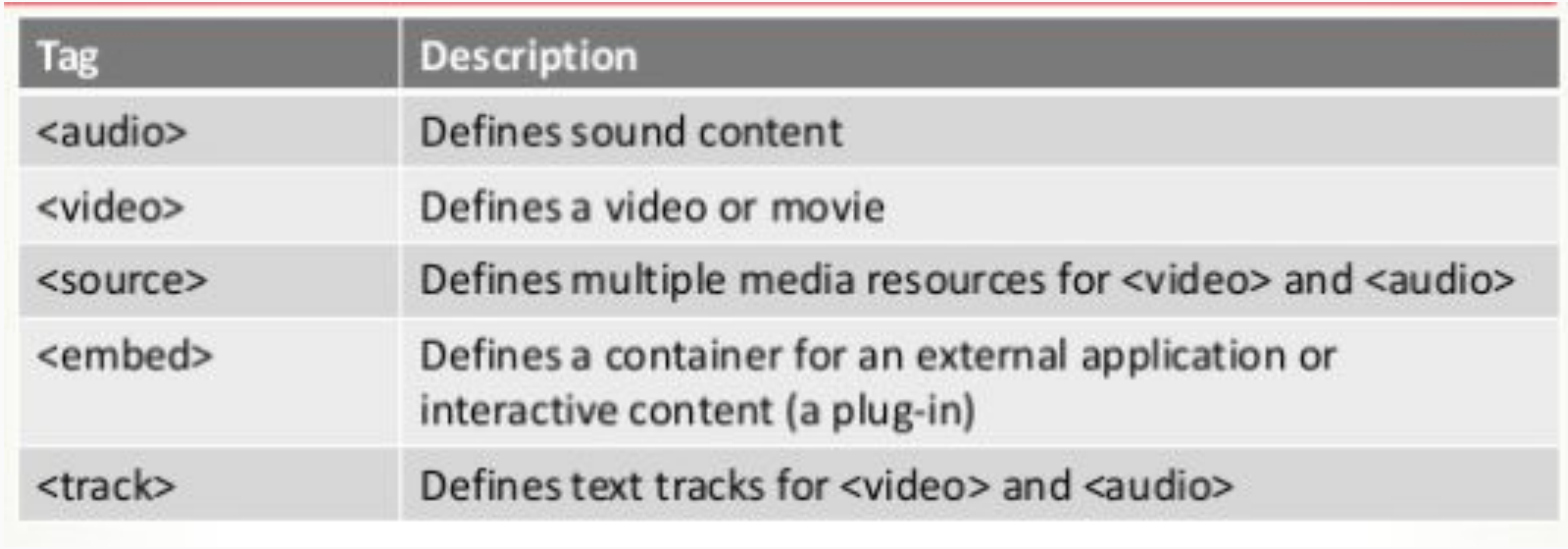
Rendering complex graphs or designs to the Web has always been a challenge that has typically been solved by using images, server-side processes or plugins such as Silverlight or Flash.

Although drawing charts with straight lines has never been a problem (with some creative CSS), rendering different types of shapes and gradients in the browser such as ellipses, bezier curves and other custom shapes has always been a problem. With the addition of the HTML 5 canvas in the latest version of all major browsers, a lot can be done with only JavaScript and HTML tags now.

So what is the canvas tag? Put simply, it's a way to render pixels on the client-side using JavaScript. This includes rendering text, images, shapes, linear and radial gradients, performing animations, plus more. Unlike Scalable Vector Graphics (SVG) which is vector based (and also available in many browsers now), the canvas is pixel-based.



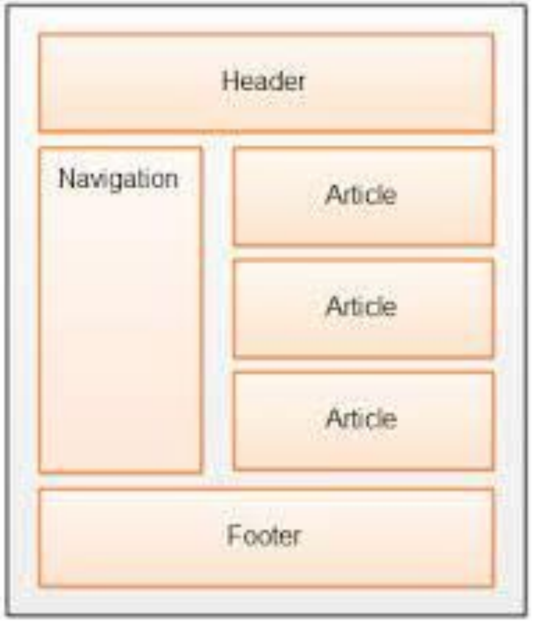
New <media> Elements



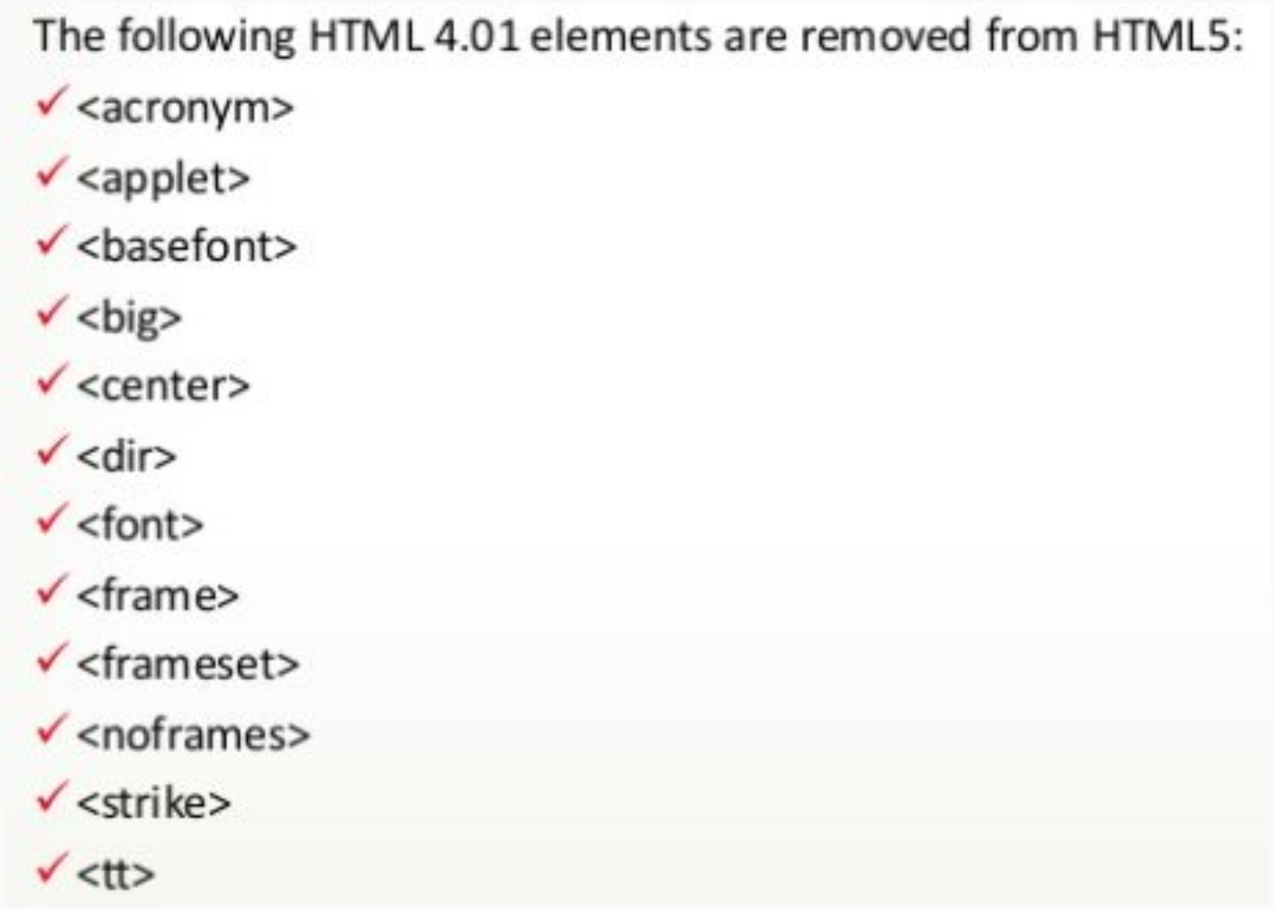
Tag	Description
<audio>	Defines sound content
<video>	Defines a video or movie
<source>	Defines multiple media resources for <video> and <audio>
<embed>	Defines a container for an external application or interactive content (a plug-in)
<track>	Defines text tracks for <video> and <audio>

New Semantic/Structure Elements

Tag	Description
<article>	Defines an article
<aside>	Defines content aside from the page content
<bdi>	Isolates a part of text that might be formatted in a different direction from other text outside it
<command>	Defines a command button that a user can invoke
<details>	Defines additional details that the user can view or hide
<dialog>	Defines a dialog box or window
<summary>	Defines a visible heading for a <details> element



Removed (Depreciated) Elements



The following HTML 4.01 elements are removed from HTML5:

✓ <acronym>

✓ <applet>

✓ <basefont>

✓ <big>

✓ <center>

✓ <dir>

✓ <font>

✓ <frame>

✓ <frameset>

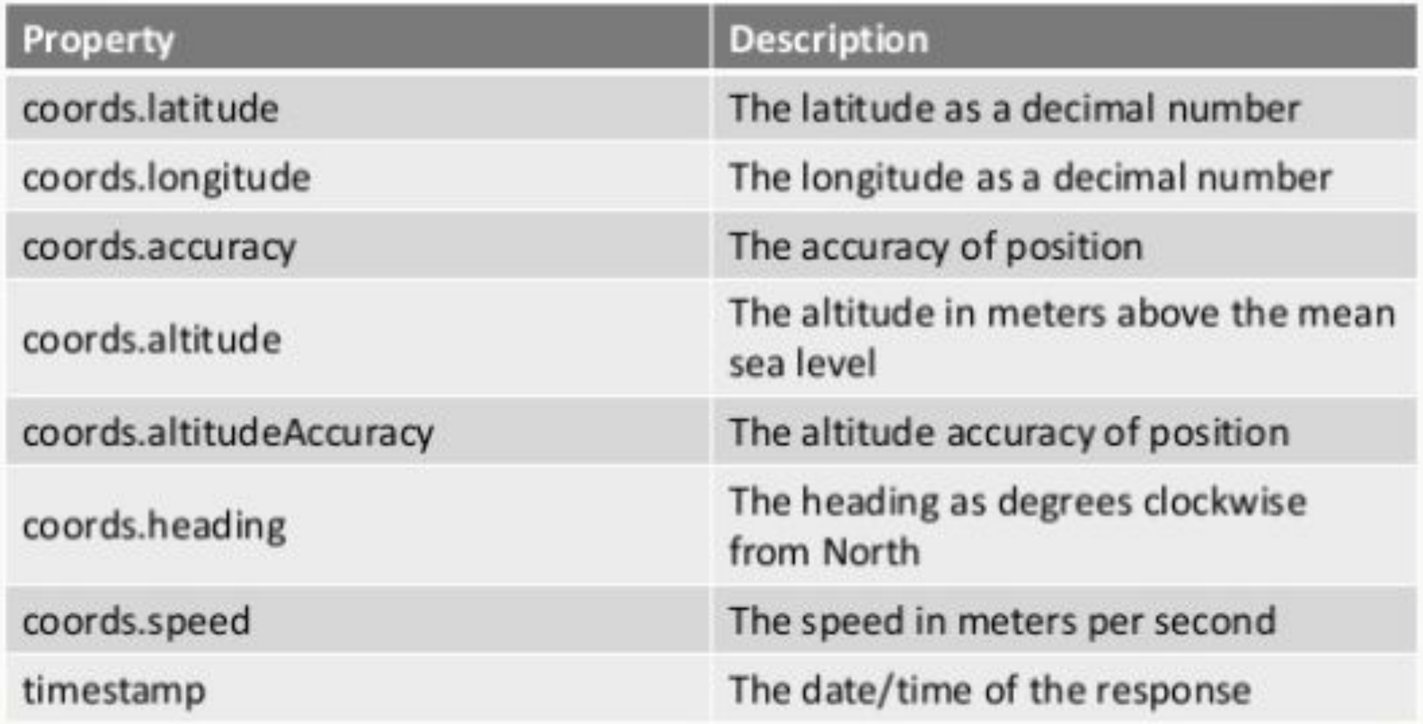
✓ <noframes>

✓ <strike>

✓ <tt>

HTML Geolocation

- The HTML5 Geolocation API is used to get the geographical position of a user
- Since this can compromise user privacy, the position is not available unless the user approves it



Property	Description
coords.latitude	The latitude as a decimal number
coords.longitude	The longitude as a decimal number
coords.accuracy	The accuracy of position
coords.altitude	The altitude in meters above the mean sea level
coords.altitudeAccuracy	The altitude accuracy of position
coords.heading	The heading as degrees clockwise from North
coords.speed	The speed in meters per second
timestamp	The date/time of the response

HTML5 Web Storage

- With HTML5, web pages can store data locally within the user's browser
- Earlier, this was done with cookies. However, Web Storage is more secure and faster. The data is not included with every server request, but used
- ONLY when asked for. It is also possible to store large amounts of data, without affecting the website's performance.
- The data is stored in key/value pairs, and a web page can only access data stored by itself.

HTML5 Web Storage cont..

- There are two new objects for storing data on the client:
  localStorage - stores data with no expiration date
  sessionStorage - stores data with one session

- The **sessionStorage** object is equal to the **localStorage** object, except that it stores the data for only one session. The data is deleted when the user closes the browser window.