1. Scenario or instructional?
2. Talk to the student or 3rd party
3. Videos?
4. Citations or SME text?
5. Preferred Text editor? Atom or notepad++
6. Will you be using CodePen? Can it be embedded in the LMS?
7. Are you using GitHub for versioning?
8. Where are lists include?
9. Are the majority of students using a MAC or a PC?
10. Since graphics are likely are there preferred colors for the graphics?
11. Should learners work through an example situation first and download the code and then work on their own examples?
12. Are discussion boards to be used?

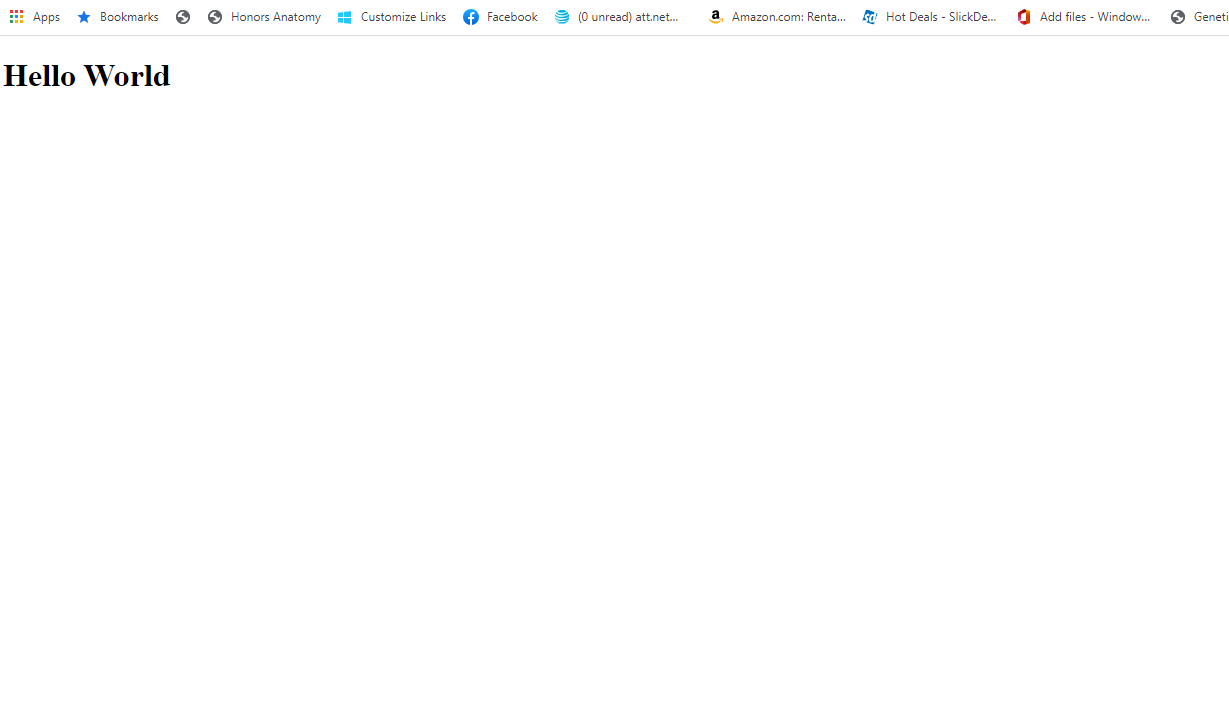
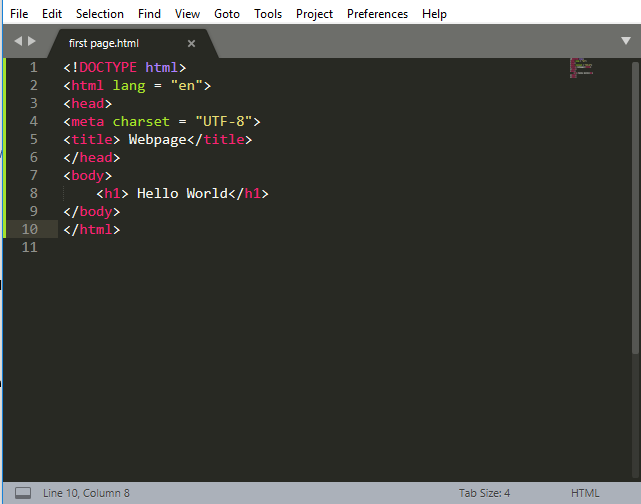
## The graduate creates web pages using a graphic user interface (GUI) editor as well as basic HTML5 and CSS 3 elements. Demonstrate knowledge required to create a web page.

Adobe Dreamweaver, Microsoft Expression Web, Mozilla Seamonkey, Google Web Designer, Blue Griffon

# Module 1

## The student creates a Web page using the most current version of HTML.

## Document web page and website creation.

1. Naming folder and file. Never use spaces and be consistent about capitalization, uses dashes in your file names. Sloppy file management is the biggest problem. Not saving the file. Not refreshing the file. Make sure your quotes are going the right way.
2. Editor: Notepad or Sublime: Windows, TexEdit, Text Wrangler or Sublime. Textedit make sure it is in plain text, make sure you save your files in UTF-8, check your file extension, when you copy and paste you can get some strange characters if that happens retype by hand.
3. Steps to get started
   1. Open your editor
   2. Select Save or Save as and name your file, after you create a folder to save it in
   3. Add Doctype, head and body tags
   4. Save file
   5. Open in Browser
   6. 
   7. Three parts of any well-formed document. The document object model. The basis of HTML 5 is that all new features should be based on HTML, CSS the DOM and JavaScript. The DOM provides a common tree-like structure that all pages should follow.
      1. DocType: Version of HTML that you will be using and the markup language, always at the top line <!DOCTYPE html>
      2. HTML tag: encloses all markup for the entire page including the language tag
      3. Head: metadata: what the people do not see, has the <meta> tag (defines the character set used in the HTML document. It specifies the Unicode character set. <mata name includes keywords, author or description, ,the <link> tag <link rel="stylesheet" type="text/css" href="theme.css"/> which references the CSS3 file, the <title> tag and the <script> tag
      4. Body: displayable content

## Define Extensible Markup Language (XML). (identify features and appropriate use)

XML stands for extensible markup language. XML documents are used to transfer data from one place to another on the internet. It is a language that is used to describe data elements on a webpage. It enhances the structure and navigation of the page, not to format it. Tags are added to a document to provide extra information. They describe the data. HTML tags tells a browser how to display information and give meaning to the reader. HTML tags have a fixed meaning that is understandable by browsers and are used for display. XML tags are different based on the application and are used to describe documents and data. W3C governs the development of XML. When using XML, the uniform resource Identifier (URI) is preferred over the standard html term Uniform resource locator (URL). <https://www.w3.org/MarkUp/> https://www.w3.org/TR/2014/REC-html5-20141028/single-page.html

Tag rules:

* Tags are enclosed in angle brackets
* They come in pairs with a start-tag and an end-tag
* Tags must be properly nested
* If a tag does not have an end tag it must be terminated with a / such as <br/>
* Tags are case sensitive
* XML in any case must not be part of the tag.
* Tags may not contain an < or a &
* Tags follow Java naming conventions, except some characters such as a single colon is permitted
* Documents must have a single root tag.
* XML like JAVA uses Unicode the most common one is UTF-8
* An XML parser can check to see if a document is well-formed. A DOM creates a parse tree and requires a tree traversal.
* XML files are trees- with a parent node, any number of children and children are ordered and may have siblings

## Analyze XML's features in conjunction with their appropriate use.

All tags must have an opening tag and a closing tag, but sometimes tags are self-closing and do both. (so they do not always have to have both)

<h1> Hello World </h1> opening tag, text and then closing tag

<img src="hello.jpeg"/> <img is the opening /> self closes and src is the attribute for an image as is href for a link.

Display is one of the most important attributes of an element. The two most common are block and display.

* **Block** can take a width and height It has Newline inserted before and after
  + Headings are block tags and have both syntax and semantics <h1> Headings have semantics they are used to delineate importance
  + Paragraphs are block, but should online contain inline elements <p> </p> Never put a paragraph or a heading inside of a paragraph.
  + Divs are blocks and are generic sections that are larger than a paragraph <div> </div> Divs have no semantic meaning
* **Inline** cannot take a width and height and uses only as much space as is needed.

**Class attribute:** applies special properties to groups of elements

**Id-** specifies a unique ID to a specific element on the page.

**Style:** lets you apply special style- avoid this one

**Accesskey:** a short cut key to activate an element

**Tabindex:** the order elements will come into focus using the tab key. The parts of the page with the highest importance can be given the lowest tab number to signal their importance.

Special entities:

If you want: Then use:

< &lt;

> &gt;

Copyright &copy;

Blank space &nbsp;

Cent sign &cent;

& &apm;

<div> is a block-level element that is used to identify large groupings of content

<span> an inline element to identify smaller groupings of text within a block level element.

<br> is a line break in the content, self-closing tags

Attributes are always specified in the start tag.

Generic tags: <p>, <div>

Semantic tags: <header>,<nav>,<footer>,<figure>

Container tags <article>, <aside>, <section>, <main>

<hr> horizontal rule puts a line across the page

<address> lets screen readers know that this is the address

<blockquote> has a cite attribute The <blockquote> tag indents a block of text. As its name suggests, it is often used to specify quotes within a Webpage. The <blockquote> tag has a cite attribute that specifies the source of the quote. Do not use <blockquote> tags within <p> tags, and do not use <h1> tags within <blockquote> tags. Doing so will prevent your code from validating.

<details> with <summary> does not run in Firefox.

## Write comments within HTML code.

HTML and CSS give us the ability to leave comments with our code. Any content wrapped in comments will not be displayed on the webpage. Comments keep the file organized and set reminders. Comments are especially useful when multiple people are working on the same field. HTML comments start with <!—and end with -->

When creating markup pages, you can use comments to:

"Comment out" code to see how a page will appear without a particular markup element.

Inform others about important elements in the code you are creating.

Remind yourself why you inserted a particular piece of code.

Insert programming code, such as JavaScript.

## Apply HTML elements and tags to format paragraphs and text.

### Headings

Headings are block-level elements, and they come in six different rankings, <h1> through <h6>. Headings help to quickly break up content and establish hierarchy, and they are key identifiers for users reading a page. They also help search engines to index and determine the content on a page.

Headings should be used in an order that is relevant to the content of a page. The primary heading of a page or section should be marked up with an <h1> element, and subsequent headings should use <h2>, <h3>, <h4>, <h5>, and <h6> elements as necessary. Headings are used for semantic reasons not to change the size of the text.

Headings are often followed by supporting paragraphs. Paragraphs are defined using the <p> block-level element. Paragraphs can appear one after the other, adding information to a page as desired. Here is an example of how to set up paragraphs.

To make text bold and place a strong importance on it, we’ll use the <strong> inline-level element. There are two elements that will bold text for us: the <strong> and <b> elements. It is important to understand the semantic difference between the two.

### Text

The <strong> element is semantically used to give strong importance to text, and is thus the most popular option for bolding text. The <b> element, on the other hand, semantically means to stylistically offset text, which isn’t always the best choice for text deserving prominent attention. We have to gauge the significance of the text we wish to set as bold and to choose an element accordingly.

To italicize text, thereby placing emphasis on it, we’ll use the <em> inline-level element. As with the elements for bold text, there are two different elements that will italicize text, each with a slightly different semantic meaning.

The <em> element is used semantically to place a stressed emphasis on text; it is thus the most popular option for italicizing text. The other option, the <i> element, is used semantically to convey text in an alternative voice or tone, almost as if it were placed in quotation marks. Again, we will need to gauge the significance of the text we want to italicize and choose an element accordingly.

### Lists

Ordered lists and unordered lists

An unordered list is simply a list of related items whose order does not matter. Creating an unordered list in HTML is accomplished using the unordered list block-level element, <ul>. Each item within an unordered list is individually marked up using the list item element, <li>.

The ordered list element, <ol>, works very much like the unordered list element; individual list items are created in the same manner. The main difference between an ordered list and an unordered list is that with an ordered list, the order in which items are presented is important.

## Identify common website navigation issues.

Ensure web page and website accessibility.

## Create basic web pages using the most current version of Hypertext Markup Language (HTML).

1. HTML 1 was created in 1990 as a way to electronically connect documents via hyperlinks. Lists of scientist research papers, ability to link pages together. HTML works across any platform. Tim Berners-Lee
2. 1993: Mosaic was the first graphical browser- images Challengers were Netscape and Internet Explorer who each wanted to create proprietary tags.
3. 2005: HTML to create the content and CSS to style it.
4. HTML 2: Developed by the Internet Engineering Task Force to include stylized text and tables
5. 1996: CSS 1
6. 1997: HTML 3.2 Developed by W3C and included specific features for browsers
7. 1997: HTML 4: A move back to normalizing the pages across platforms
8. 1998: CSS 2
9. 1999: HTML 4.01 Introduced different document types
10. HTML 5: Back to HTML plus multimedia and stylized tags
    1. New features based on HTML, CSS, The DON and JavaScript.
    2. Reduce needs for external plugins such as Flash
    3. More markup to replace scripting
    4. Device independent

Start with <!Doctype html>

Then the language <html lang="en">

Then the head section: <head>

With the charset <meta charset="UTF-8">

The title on the browser tab <title>This is the tile</title>

If you want to add a favicon or an image in the browser tab it also must go in the head section

<link rel="icon" type="image/png" href="imgs/logo.png"/> you are linking to the image and you can only use special icon images or .png

Alt text: provides contextual meaning to non-text content so it can be read by screen readers, displayed in place of images if images are not downloaded and it provides semantic meaning for search engines which can help your SEO. Always think about the meaning behind the alt text. If the image is not pertinent or necessary to understand the content or message on the page then use decorate or empty quots. Do not use a space in your quotes as the screen reader may read the space. Good alt text is accurate, succinct, not redundant and does not state the obvious such as this is a picture of or image of. If information is already on the page, do not include the same information in the alt text. Don't skip the alt text though it is OK to have empty quotes or to write decorative image.

The <header> element, is a group of introductory or navigational aids. It can include the title and the navigational aid to move from page to page. Do not confuse the <header> with the <head> section which is where the metadata is stored and is not accessible to the user. Also do not confuse it with the heading tags which are used to delineate the most important information on the page. The header tag does not affect the placement on the page. It provides context to those viewing the page who are nonvisual.

The <nav> element identifies a section of major navigational links on a page. The <nav> element should be reserved for primary navigation sections only, it is the section of the page that links to other pages or sections within a page. It is often found in the header

<body>

<nav id="navbar"> <!-- Navigation Bar -->

<header>Canvas Reference Sheet</header>

<ul> (unorder list)

<a href="#Introduction" ><li>Introduction</li></a>

<a href="#Files" ><li>Files</li></a>

<a href="#Quizzes" ><li>Quizzes</li></a>

<a href="#Videos" ><li>Videos</li></a>

</ul>

</nav>

The <article> element is used to identify a section of independent, self-contained content that may be independently distributed or reused. We’ll often use the <article> element to mark up blog posts, newspaper articles, user-submitted content, and the like.

The <section> element is used to identify a thematic grouping of content, which generally, but not always, includes a heading. The grouping of content within the <section> element may be generic in nature, but it’s useful to identify all of the content as related.

Both the <article> and <section> elements contribute to a document’s structure and help to outline a document. If the content is being grouped solely for styling purposes and doesn’t provide value to the outline of a document, use the <div> element.

If the content adds to the document outline and it can be independently redistributed or syndicated, use the <article> element.

If the content adds to the document outline and represents a thematic group of content, use the <section> element.

The <aside> element holds content, such as sidebars, inserts, or brief explanations, that is tangentially related to the content surrounding it. When used within an <article> element, for example, the <aside> element may identify content related to the author of the art icle.

The <footer> element identifies the closing or end of a page, article, section, or other segment of a page. Generally, the <footer> element is found at the bottom of its parent. And includes information such as copyright data, related documents and links to social media. The footer tag does not put the footer at the bottom of the page.

<figure> has more semantic meaning than <img> As it can include captions and multiple multi-media resources.

<figure>

<img src="The sky.jpeg" alt="the sky is blue">

<figcaption>A beautiful sunset</figcaption>

</figure>

# Module 2

The student defines and applies essential aspects of the (CSS3) Cascading Style Sheets version 3 standard.

Apply industry-standard design and color principles to web pages.

Define the Cascading Style Sheets (CSS) standard.

Apply basic aspects of CSS.

# Module 3

The student creates basic HTML that accepts user input.

Create a basic HTML form that accepts user input.

Define the document object model (DOM).

Analyze the relationship between document object model (DOM) and Dynamic HTML (DHTML)

# Module 4

The student describes the differences between creating a website using a text editor and a WYSIWYG editor.

Analyze ways in which a web browser can become an application delivery platform.

Create basic web pages using graphic user interface (GUI) based HTML editing software.

Apply third-party applications to a web page.

# Module 5

### The student adds tables, hyperlinks, images, audio, and video and the associated graphical formatting to HTML files.

### Apply HTML hyperlinks for text, images, local files, and remote sites with both internal and external links.

### Images

For the <img> element to work, a src attribute and value must be included to specify the source of the image. The src attribute value is a URL, typically relative to the server where a website is hosted.

Images are inline <img src ="Picture.jpeg" alt="A picture of a girl"/> however, they can be changed with css to float, display and box model properties. Block positioning makes the image have its own line with the content positioned above and below the image. Sometimes displaying an image as inline or block, or perhaps even inline-block, isn’t ideal. We may want the image to appear on the left or right side of its containing element, while all of the other content wraps around the image as necessary. To do this, we use the float property with a value of either left or right.

Many files types are widely supported including JPEG, GIF and PNG, SVG and BMP are additional options. The file extensions must be included. Every image must be downloaded, so size is a factor, and every image requires an HTTP request. When you link to an image the browser displays the image as big or small as the file which is rarely what you want so you can either change the size of the image in the fie or use width and height attributes. Editors such as Paint can be used to change a file, but they do change the size of the image permanently and it only works on local files.

<figure>

<img src="The sky.jpeg"

width="500px" alt="the sky is blue">

<figcaption>Set image size</figcaption>

</figure>

If you only set the width that is OK because the browser will automatically set the height to keep it proportional to the width

<figure>

<img src="The sky.jpeg"

width="50%" alt="the sky is blue">

<figcaption>Relative image size</figcaption>

</figure>

When you set it at a percentage it will remain at that percentage of the page size. So, if the browser window gets bigger or smaller than the image size will proportionally grow and shrink to match.

Font awesome: If you use font awesome then the link goes in the head section of your webpage. <script src="https://kit.fontawesome.com/e1d691fac8.js" crossorigin="anonymous"></script> (each person has their own)

Example <i class="fas fa-camera"></i> <!-- this icon's 1) style prefix == fas and 2) icon name == camera -->

<i class="fas fa-camera"></i> <!-- using an <i> element to reference the icon -->

<span class="fas fa-camera"></span> <!-- using a <span> element to reference the icon -->

If you use an icon as a link to a webpage it is not accessible so inside of the <a> tag right after the link add aria-label="and the name of the link"

### Hyperlinks

Hyperlinks are what make the web a web by linking pages together. Links are called anchor links and thus the <a> tag. Links are created with the anchor element, <a>. Anchor elements are container tags that encompass the text or image (or both) to be used as the link. The href attribute is used to specify the link's hypertext reference, or the target of the link. You can specify a fully qualified URL or a relative URL reference for any file, page or site.

<a href="https:leannposton.com">Leann's website</a>

Includes both the link to the site and content that tells the browser what the site is and is what the user actually clicks on.

**Absolute links:** has the fully formed URL in the href section (protocol, host and the document). There is the opening tag, then the where to go on click, then the click text and finally the closing tag. Absolute used in someone else's page

<a href="https:leannposton.com">Leann's website</a>

**Relative links:** relative links can be to second pages in documents that are stored in the same folder. Only the file name is added. Can also use the ID tag. Use on your own website so when you move pages you do not need to change links. Never include file structure or a reference that is local to your machine. The clickable component can be an image. Always test your links. Make sure all links have an informative name. All images must have alt text. Embedded links are links within the same page. Use developer tools and then View Page source to see how developers are creating their pages and links.

<a href="page2.html">Second page</a>

Internal links <a href="#history">History section</a>

Graphical links

When target equals self which is the default you will open in the same tab. When tab equal blank it will open in a new tab or window.

<a href="http://leannposton.com/" target="\_blank">Leann Poston</a>

<body id="top">

...

<a href="#top">Back to top</a>

...

</body>

### Design HTML tables to present information in an organized way.

You should only use tables if you want to display data. Sketch your layout for your table before you get started. Decide on the number of rows and columns. Do any rows or columns need to span multiple rows or columns.

Table heading: don't use TD elements and bold them. Use the semantic head <th>….</th>

<table> the container tag

<tr>…</tr> the rows

<td>…</td> the columns (think of as cells instead of columns.)

<table>

<tr><td>One</td><td>Two</td><td>Three</td><td>Four</td><td>Five</td></tr>

</table>

Use row span and column span to combine multiple rows and columns inside of the td tags add rowspan="\_\_"

Can add border ="\_\_" inside of the table tag

Captions: use the <caption> tag inside of the table

### Apply graphics and audio to HTML files.

Video element <video>

The video tag uses a src attribute or embedded <source>

Common attributes

* Height, width
* Autoplay
* Loop
* Controls Autoplay, controls and loops are boolean and are true or false no need to add + anything
* Text inside a <video>…</video> is displayed if the browser cannot support the software.

Audio element, audio>

* Audio tag uses asrc attribute to link to the audio file which is typically .mp3 or .wav
* Common attributes include autoplay, control and loop, buffered, muted and volume.
* <audio src="jazz.ogg"></audio>

You can set both audio and video elements to play clips by adding to the src attribute

* .ext#t=5,25 play the audio from 5 to 25 seconds
* .ext#t=,39 play from beginning to 39 seconds
* .ext#t=,1:38:45 start at 1:38 and play for 45 seconds
* .ext#t=42 start at 42 seconds and play from there

Make sure to provide links to plugins

Include text description and closed captioning

Multimedia should enhance your content, not eb a distraction

### Create images and audio for HTML files.

# Unit 2

# Module 6

The graduate develops a plan for creating and maintaining a website that addresses specific business needs while maintaining industry and ethical standards.

"ES1: The student describes the copyright and ethical issues involved when creating Web pages.

ES2: The student defines e-commerce, as well as other related technologies and concepts, necessary to develop a secure, useful interface (i.e., storefront).

ES5: The student identifies the essential issues in planning, developing, and maintaining a Web site."

Identify common steps in the website planning and development project cycle.

Identify essential aspects involved with developing and maintaining a website including project management, testing, and legal issues.

Analyze the relationships between electronic commerce (e-commerce) and related technologies and concepts necessary to develop a secure, useful interface.

Analyze website performance issues.

Test website performance.

Discuss potential copyright and ethical issues to resolve during the web site planning process.

Plan oral presentations of your website, during and after site development.

# Module 7

"ES3: The student identifies languages commonly used to provide database connectivity to Websites.

ES4: The student identifies the technologies for enhancing the user's Web experience.

ES7: The student identifies common strategies for managing an end user's experience and improving site creativity."

Identify technologies for enhancing a user's web experience, including programming languages and multimedia technologies.

List languages commonly used to provide database connectivity to websites.

Identify common strategies for managing an end-user's experience and improving site creativity.

# Module 8

The student describes how to design, develop, and publish web pages to industry standards.

Analyze the advantages and disadvantages of running your own web server versus using a service provider.

Apply industry standards to guide the design process of a web page.

# Videos:

<https://www.youtube.com/watch?v=VANORrzKX50> The true story of the Internet: Browser Wars

<https://www.youtube.com/watch?v=-c8iWUh3Sv4&app=desktop>: Joseph Hardin: NCSA Mosaic

# Web Links

<http://html5test.com/index.html> Looks at how well your browser supports most HTML5 tags

<http://www.html5acessibility.com/>

<http://validator.w3.org> : Used to validate website code. Don't worry about warnings. You can copy the code directly into the validator.

<https://chrome.google.com/webstore/detail/validity/bbicmjjbohdfglopkidebfccilipgeif>

https://www.toptal.com/designers/colorfilter

<https://learn.shayhowe.com/html-css/>

<https://www.wpkube.com/html5-cheat-sheet/> A cheat sheet for HTML5 and 4

<https://www.w3.org/WAI/tutorials/images/decision-tree/> This is a decision tree for alt text from web accessibility tutorials.

<https://www.w3.org/>

<http://dev.w3.org/html5/spec/Overview.html>

<http://www.ada.gov/>

<http://corporate.findlaw.com/law-library/applying-the-ada-to-the-internet-a-web-accessibility-standard.html>

<http://www.usability.gov/>

<https://www.yokoco.com/ada-website-compliance-in-2018/>

<https://www.w3.org/TR/2006/WD-WCAG20-20060427/appendixB.html>

<https://www.w3.org/TR/ATAG20/>

<https://section508.gov/content/technology-accessibility-playbook> Consider having learners evaluate the accessibility of a webpage. Microsoft Accessibility Insights

Wireframing software: Gliffy, Mockingbird and HotGlo

<https://www.whatiscopyright.org/>

<http://bluefish.openoffice.nl/>

http://www.w3schools.com/tags/tag\_pre.asp

http://blog.teamtreehouse.com/beginners-guide-to-responsive-web-design

Abbreviations:

World Wide Web Consortium (W3C); HTML and its evolution

Internet Corporation for Assigned numbers and Names (ICANN)

Internet Engineering Task Force (IETF) how different networks should work together.

W3C: Accessibility

HTML: Hypertext markup language not the same as programming languages, use tags to markup documents