Web Basics

Technology, HTML, and CSS

HTTP

- The Internet works primarily through the network protocol known as HyperText Transport Protocol
  - Invented by Sir Tim Berners-Lee (KBE, OM, Turing Award Recipient 2016) in 1989
- Defines how your computer asks for and recieves data from a server, and vice-versa
  - The browser handles this, but you could request it manually using `telnet`
- Other modern protocols exist, the most commonly seen is HTTPS, which is just a more secure HTTP

```
telnet google.com 80
Trying 172.217.9.206...
Connected to google.com.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.0 200 OK
Date: Wed, 18 Oct 2017 17:47:11 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
Server: gws
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Set-Cookie: 1P_JAR=2017-10-18-17; expires=Wed, 25-Oct-2017 17:47:11 GMT; path=/;
 domain=.google.com

Set-Cookie: NID=114=eBuvPeznELacI04cEvZN4bITyJWjchMh7IeLTctwGNzxw8C6P02hMSzR_7TG
W9YBLnERNGvpb3KGVOVSAHhoDCGu9BSy--gFRyqygPNLF65GcWAo2kZke6-8CW-N7dD0; expires=Th
u, 19-Apr-2018 17:47:11 GMT; path=/; domain=.google.com; HttpOnly
Accept-Ranges: none
Vary: Accept-Encoding

<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en"
><head><meta content="Search the world's information, including webpages, image
s, videos and more. Google has many special features to help you find exactly wh
at you're looking for." name="description"><meta content="noodp" name="robots"><
meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content=
"/images/branding/googleg/1x/googleg_standard_color_128dp.png" itemprop="image">
<title>Google</title><script>(function(){window.google={kEI:'n5PnWeLxIIHmmAHF2a-
wDw',kEXPI:'201806,1352821,1352960,1353383,1353606,1354277,1354401,1354916,13552
17,1355324,1355735,1355801,1355820,1355892,3700289,3700439,3700440,3700476,40298
15,4031109,4043492,4045841,4048346,4063965,4072775,4076999,4081038,4081164,40921
```

http://google.com

Parts of a Web App

- Structural - The content and how it is structured
- Presentation - The appearance of the content
- Interactive - Actions that manipulate either the structure or appearance layers

Main Web Languages

- HTML (HyperText Markup Language - defines the content and structure of the page
- CSS (Cascading Style Sheets) - defines the appearance of the HTML content
- JavaScript - runs in the browser (typically) and allows interaction with the content

HTML

- Also invented by Tim Berners-Lee
  - The first website is http://info.cern.ch/hypertext/WWW/TheProject.html
- Meant to simply format and link to text documents
- Originally based off a more generic markup language, SGML (Standard Generalized Markup Language)
- Standardized by the World Wide Web Consortium (W3C)
  - Directed by Tim Berners-Lee

HTML Today

- In the late 90s and early 2000s, the W3C developed the next standard of HTML, based on XML which was more strict that SGML
    - This lead to the release of XHTML in 2000
    - The community as a whole pushed back somewhat on the strictness imposed by XML
- Outside of the W3C, the Web Hypertext Application Technology Working Group (WHATWG) began working on a more direct successor to HTML
    - Accepted by W3C as HTML 5 in 2007 as a working draft
    - Released as a reccomendation in 2014

How HTML is displayed in a Browser

- The W3C defines the HTML standards
- It is up to the programmers of the browsers to conform to these standards
  - There is no major penalty if they don't
- Compare current desktop browsers at https://html5test.com/compare/browser/chrome-63/firefox-57/edge-16/safari-11/ie-11.html

HTML Tags

- An HTML tag describes the meaning of the content it holds
- Comes in one of two forms

```html
<tag>Content</tag>
  <tag> <!-- also written as <tag/> -->
```

- Tags may have additional attributes, which are defined in the opening tag

```html
<tag att1=val1 att2=val2>content</tag>
<tag boolean1=boolean1 boolean2>content</tag>
```

The basic HTML Document

- The basic HTML document consists of three major parts, with an optional (but highly recomended fourth tag)
    - `<html>` - Contains all the HTML on the page
    - `<head>` - Contains a lot of meta-data about the page as well as information about styling and JavaScript
    - `<body>` - Contains the content displayed by the web-browser
    - <!DOCTYPE html> - which is optional, but lets the browser know you are using HTML

In [ ]:

```
%%html
<!DOCTYPE html>
<html>
    <head>
    </head>
    <body>
        A very simple page
    </body>
</html>
```

The HTML Tag

- In general the HTML tag is just used to enclose the `<body>` and `<head>` tags
- Several useful attributes can be placed on the HTML tag itself
    - The dir attribute informs the browser which way the text is meant to be displayed, ie "ltr" or "rtl"
    - The lang attribute signifies which language the text in the document is, primarily

In [ ]:

```
%%html

<!DOCTYPE html>
<html lang="en" dir="rtl">
    <head>
    </head>
    <body>
        A very simple page
    </body>
</html>
```

Common Tags found in Head

- The `<head>` tag contains many pieces of information that help search engines as well as being the most common place other files are "included"
- Common tags
  - `<title>` - Defines the title of the page, commonly displayed at the top of a window or tab
  - `<meta>` - Used to define numerous different attributes about the page, such as viewport, character-set, and description
  - `<link>` - Used to link style sheets
  - `<script>` - Used to include JavaScript, or write it in the document itself

In [ ]:

```
%%html
<!DOCTYPE html>
<head>
<title>A simple document</title>
<meta charset="utf-8">
</head>
<body>
Simple
</body>
```

Inline vs Block Elements

- The tags in between and define elements of the page
- The elements can either be known as **block** or **inline** elements, which is based on their display properties
    - Block elements are placed on their own line on the page by default, nothing before or after them
    - Inline elements are displayed on the same line as other inline elements
- This property can be overridden using CSS, it is just a default

Paragraph and Heading Tags

- The paragraph tag, `<p>`, sets off a block of text as a paragraph
  - Usually causes a small space before and after the paragraph
  - Can be used to group other logical text, like the an address or a byline
- There are 6 header tags, `<h[1-6]>`,
  - Denote different levels of importance
  - Usually denoted visually by font-size
  - `<h1>` is the highest, should only be one on the page

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>
    <body>
        <p>This is a paragraph</p><p>This is too is runs longer than a This is too
 is runs longer than This is too is runs longer This is too is runs longerThis is
 too is runs longer than a   than a than a  a  line is </p>
        <p>    What

        happens now?</p>
    </body>
</html>
```

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>
    <body>
        <h1>H1 is good for the title of your page</h1>
        <h2>H2 is less important, good for subtitles, or sections</h2>
        <h3>H3 might be a subsection</h3>
        <h4>H4 is often the size of paragraph text</h4>
        <p>Paragraph text for reference</p>
        <h5>H5 is not very common</h5>
        <h6>H6 is even less common</h6>
    </body>
</html>
```

Pre-formatted Text and Code

- To maintain everything as typed, including spaces and blank lines, the `<pre>` tag is used
  - Most often used for code
  - Another use case given in the HTML5 specification is ASCII art
- To denote a block of computer code, use the `<code>` tag
  - No special formal way to denote the language the code is in
  - But HTML5 sepcification specfically says to use the `class` attribute with a value of `language-X`

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>
    <body>
        <p>              _______
         ..-'`       ````---.
       .'          ___ .'````.'SS'.
      /        ..-SS####'.  /SSHH##'.
     |       .'SSSHHHH##|/#/#HH#H####'.
    /      .'SSHHHHH####/||#/: \SHH#####\
   /      /SSHHHHH#####/!||;`___|SSHH###\
-..__    /SSSHHH######.         \SSSHH###\
`.'-.''--._SHHH#####.'           '.SH####/
  '. ``'-  '/SH####`/_             `|H##/
  | '.     /SSHH###|`'==.       .=='/\H|
  |   `'-.|SHHHH##/\__\/        /\//|~|/
  |    |S#|/HHH##/             |``  |
  |    \H' |H#.'`              \    |
  |        ''`|               _     /
  |          /H\          .----    /
  |         |H#/'.           `    /
  |          \| | '..            /
  |    ^~DLF   /|    ''..______.'
   \          //\__    _..-. |
    \         ||   ````     \ |_
     \    _.-|               \| |_
     _\_.-'   `'''''-.        |   `--.
 ''``    \            `''-;    \ /
          \      .-'|     ````.' -
          |    .'  `--'''''-.. |/
          |  .'               \|
          |.'
          </p>
```

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>
    <body>
        <pre>              _______
         ..-'`       ````---.
       .'          ___ .'````.'SS'.
      /        ..-SS####'.  /SSHH##'.
     |       .'SSSHHHH##|/#/#HH#H####'.
    /      .'SSHHHHH####/||#/: \SHH#####\
   /      /SSHHHHH#####/!||;`___|SSHH###\
-..__    /SSSHHH######.         \SSSHH###\
`.'-.''--._SHHH#####.'           '.SH####/
  '. ``'-  '/SH####`/_             `|H##/
  | '.     /SSHH###|`'==.       .=='/\H|
  |   `'-.|SHHHH##/\__\/        /\//|~|/
  |    |S#|/HHH##/             |``  |
  |    \H' |H#.'`              \    |
  |        ''`|               -     /
  |          /H\          .----    /
  |         |H#/'.           `    /
  |          \| | '..            /
  |    ^~DLF   /|    ''..______.'
   \          //\__    _..-. |
    \         ||   ````     \ |_
     \    _.-|               \| |_
     _\_.-'   `'''''-.        |   `--.
 ''``    \            `''-;    \ /
          \      .-'|     ````.' -
          |    .'  `--'''''-.. |/
          |  .'               \|
          |.'
          </pre>
```

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>
    <body>
        <pre>
        <code class="language-bash">
#!/bin/bash
    echo "Hello Word"
        </code>
        </pre>
    </body>
</html>
```

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>
    <body>
        <pre>
<samp>linuxserver2.cs.umbc.edu[105]</samp><kbd>pwd</kbd>
<samp>/home/csee1/bwilk1/www/331</samp>
        </pre>
    </body>
</html>
```

Lists

- HTML provides 3 types of lists, which, according to the spec, should never occur inside a `<p>` tag
- Ordered list `<ol>`
  - Represents information that is important to present in that order, like directions
  - The browser usually displays these as numbers, or letters if nested
  - List elements denoted using the `<li>` tag

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>

    <body>
    <p>How to make grilled cheese :</p>
<ol>
<li>Butter two pieces of bread on one side</li>
<li>Place them on a griddle, butter side down</li>
<li>Put cheese on top of one</li>
<li>When cheese begins to melt, place
one slice of bread on top of the other, butter sides out</li>
<li>Grill until golden brown</li>
</ol>
    </body>
</html>
```

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>

    <body>
    <p>The top grossing Broadway shows of all time are :</p>
<ol reversed="reversed">
<li>Chicago (1996 Revival)</li>
<li>Mamma Mia!</li>
<li>The Phantom of the Opera</li>
<li>Wicked</li>
<li>The Lion King</li>
</ol>
    </body>
</html>
```

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>
    <body>
        <p>So far we have talked about :</p>

<ol>
<li>Paragraphs</li>
<li>Headers</li>
<li>Pre-formatted text</li>
<li>Lists
    <ol>
        <li>Ordered Lists</li>
        <li>Unordered Lists</li>
        <li>Description Lists</li>
    </ol>
</li>
</ol>
    </body>
</html>
```

Lists

- HTML provides 3 types of lists, which, according to the spec, should never occur inside a `<p>` tag
- Unordered list `<ul>`
  - Represents a collection of information whose relative ordering is unimportant
  - Usually displayed using bullets
  - List elements denoted using the `<li>` tag
- Description list `<dl>`
  - Consists of key value pairs, specified in the `<dt>` and `<dd>` tags

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>

    <body>
    <p>Some famous sports teams are: </p>
<ul>
<li>Real Madrid</li>
<li>The Yankees</li>
<li>The Lakers</li>
<li>The Patriots</li>
<li>Ohio State Buckeyes</li>

</ul>
    </body>
</html>
```

```
In [ ]: %%html
<!DOCTYPE html>
<html lang="en">
    <head></head>

    <body>
     <p>Places people live: </p>
<ul>
<li>Maryland
    <ul>
        <li>Baltimore</li>
        <li>Frederick</li>
        <li>Gaithersburg</li>
        <li>Columbia</li>
    </ul>
    </li>
<li>Pennsylvania
 <ul>
        <li>Philadelphia</li>
        <li>Pittsburgh</li>
    </ul>
</li>
<li>Virginia</li>
<li>Washington, DC</li>

</ul>
    </body>
</html>
```

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>
    <body>
<ul>
<li>Real Madrid</li>
<li>The Yankees</li>
<li>The Lakers</li>
<li>The Patriots</li>
<li>Ohio State Buckeyes</li>

</ul>
    </body>
</html>
```

Images

- Images are included by using the `<img>` tag, which has no closing tag
    - The location of the image is specified by setting the `src` attribute to the URL of the image
    - HTML5 requires the `alt` attribute as well, which should describe the image
        - This especially important for screen readers, and other accessability technology
    - The `title` attribute is optional, and is meant to be a short bit of text about the image

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>
    <body>
<img src="https://upload.wikimedia.org/wikipedia/commons/d/d0/Chesapeakebayretriever01-l.jpg"
alt="A cheasapeake bay retriever standing on some cleared earth,
holding a tennis ball in it's mouth">
    </body>
</html>
```

Common Image Formats

- GIF - mostly used for animation, support less colors than other
- JPG - good for pictures, no transparency
- PNG - good for all types of graphics, especially useful due to transparency support
- SVG - meant for drawings rather than photographs, becoming more widely supported
    - Is a markup language describing shapes and vectors and positioning, etc.
    - HTML5 allows it to be defined in the page itself

Links

- The `<a>` tag, for anchor, is used to provide links to external pages, or to another location on the same page
- The `href` attribute determines where the link goes to
- The `target` attribute determines how the link opens
  - In a new window, tab, etc.
- The `<a>` tag can contain either images or text inside of it

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>
    <body>
        <a href="umbc.edu">UMBC (doesn't work)</a>
        <a href="http://umbc.edu">UMBC</a>
        <a href="Lecture13.ipynb" target="_blank">Tuesday's lecture</a>
        <a href="mailto:bwilk1@umb.edu">Email Me</a>
    </body>
</html>
```

Text Formatting

- Text formatting should primarily be done using CSS, but some text decoration also carries meaning, and so is acceptable to be encoded in HTML
  - `<em>` Emphasizes text, usually displayed as italics, but should not be used ONLY for that purpose
  - `<strong>` Emphasizes text even more strongly, usually displayed at bold text, should not be used only for that purpose
  - `<sup>` and `<sub>` indicate super- and subscripts respectively
  - `<del>` is presented as a strike through, but has a meaning of deleted text

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>
    <body>
        <p>This text might have
        <em>emphasis, like a rising intonation</em>
        when read aloud</p>
        <p>This text has even <strong>more
        emphasis, like a sterm talking to,
        or yelling</strong></p>
        <p>Superscripts are good for dates,
        like October 19<sup>th</sup></p>
        <p>Subscripts are good for simple
        math expressions, like x<sub>1</sub></p>
        <p>Only use delete to keep track
        of <del>are</del> changes, or make
        a change obvious</p>
    </body>
</html>
```

Tables

- Tables should only be used to represent tabular data
    - Early in the web, they were used for layout, don't do this!
- The entire table is enclosed in the `<table>` tag
    - Each row is a `<tr>` tag
        - Each cell is a `<td>` tag
    - The table rows can be optionally grouped using
        - `<thead>`
        - `<tbody>`
        - `<tfoot>`

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>
    <body>
        <table>
            <tr>
                <td>Name</td>
                <td>Date of Birth</td>
            </tr>
            <tr>
                <td>George Washington</td>
                <td>February 22, 1732</td>

            </tr>
            <tr>
                <td>John Adams</td>
                <td>October 30, 1735</td>
            </tr>
            <tr>
                <td>Thomas Jefferson</td>
                <td>April 13, 1743</td>
            </tr>
        </table>
    </body>
</html>
```

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>
    <body>
        <table>
            <thead>
                <tr>
                    <th>Name</th>
                    <th>Date of Birth</th>
                </tr>
            </thead>
            <tbody>
            <tr>
                <td>George Washington</td>
                <td>February 22, 1732</td>
            </tr>
            <tr>
                <td>John Adams</td>
                <td>October 30, 1735</td>
            </tr>
            <tr>
                <td>Thomas Jefferson</td>
                <td>April 13, 1743</td>
            </tr>
            </tbody>
        </table>
    </body>
</html>
```

Forms

- Forms are used all over the web to collect data, and provide results
- The various parts of a form are all wrapped up in the `<form>` tag
    - The `action` attribute indicates where the form information should be sent
    - The `method` attribute indicates how it should be sent
        - `GET` puts the values in the URL
        - `POST` puts the values in the HTTP header

Input

- Most types of form input are indicated using the `input` tag
  - The `type` attribute indicates the type of input
    - `text`
    - `password`
    - `radio`
    - `checkbox`
    - `submit`
  - The `name` attribute is what is used when submitting the data to form a key-value pair

Other Form Elements

- Other common tags in a form are
    - `label` - which is used to link a label to a input field
    - `textarea` which creates a larger text box than just a single line
    - `select` creates drop down option menu
        - Each possible value goes in an `option` tag

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>
    <body>
       <form action="./" method="GET">
            <label for="user_name">User Name:
            <input type="text" name="user_name" id="user_name"/></label>

            <label for="email">Email:</label>
            <input type="email" name="mail" id="email"/>

            <label for="on"><input type="radio" name="on_off" id="on"/>ON</label>
            <label for="off"><input type="radio" name="on_off" id="off"/>OFF</label>

            <label for="check1"><input type="checkbox" name="checks[]" id="check1">Option 1</label>
            <label for="check2"><input type="checkbox" name="checks[]" id="check2">Option 2</label>
            <label for="check3"><input type="checkbox" name="checks[]" id="check3">Option 3</label>
            <label for="check4"><input type="checkbox" name="checks[]" id="check4">Option 4</label>

            <label for"year">Select year:</label>
            <select name="year" id="year">
                <option>Freshman</option>
                <option>Sophmore</option>
                <option>Junior</option>
                <option>Senior</option>
            </select>

            <label for="essay">Write your essay here:</label>
            <textarea name="essay" id="essay" rows=5 cols=80>Default</textarea>
```

Divs and Spans

- Other than paragraphs, there was no common way to group elements
- The `div` tag is used to group elements at a block level, and commonly holds many elements, like `p` tags and `ol` tags
- The `span` tag is an inline tag often used to mark up sections of text that need to be styled a certain way
- Both `div` and `span` have no greater meaning then group these things together, and other new tags should be used when appropriate

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>
    <body>
       <form action="./" method="GET">
            <div>
                <label for="user_name">User Name:</label>
                <input type="text" name="user_name" id="user_name"/>
            </div>
            <div>
            <label for="email">Email:</label>
            <input type="email" name="mail" id="email"/>
            </div>
            <div>
            <label for="on"><input type="radio" name="on_off" id="on"/>ON</label>
            <label for="off"><input type="radio" name="on_off" id="off"/>OFF</labe
l>
            </div>
            <div>
                <label for="check1"><input type="checkbox" name="checks[]" id="che
ck1">Option 1</label>
                <label for="check2"><input type="checkbox" name="checks[]" id="che
ck2">Option 2</label>
                <label for="check3"><input type="checkbox" name="checks[]" id="che
ck3">Option 3</label>
                <label for="check4"><input type="checkbox" name="checks[]" id="che
ck4">Option 4</label>
            </div>
            <div>
            <label for"year">Select year:</label>
            <select name="year" id="year">
                <option>Freshman</option>
                <option>Sophmore</option>
                <option>Junior</option>
```

HTML Character Entities

- Like most languages, special characters need to be escaped
- The special characters in HTML are things like <, >, &, etc.
- They are escaped using the general structure of `&CHAR;`
    - CHAR can either be a character name or a numeric code

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>
    <body>
       <p> Some common HTML entities are </p>
       <table>
            <thead>
                <tr>
                    <th>Character Entity</th>
                    <th>Result</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td><pre>&amp;lt;</pre></td>
                    <td>&lt;</td>
                </tr>
                <tr>
                    <td><pre>&amp;gt;</pre></td>
                    <td>&gt;</td>
                </tr>
                <tr>
                    <td><pre>&amp;amp;</pre></td>
                    <td>&amp;</td>
                </tr>
            </tbody>
       </table>
        <p>A full table can be found at
        <a href="https://dev.w3.org/html5/html-author/charref">The official W3C re
ference site</a></p>
    </body>
</html>
```

Article and Section

- In HTML5, more meaningingful grouping tags were introduced, `article` and `section`
- An article is the main focus of the page, and should be relatively unique to that webpage
- A section denotes a group of elements that are related thematically
  - Can be inside an `article`
  - Can have multiple `article`'s in them

In [ ]:

```
%%html
<!DOCTYPE html>
<html lang="en">
    <head></head>
    <body>
           <article>
                phthis<h1>This might be the title of the article</h1>
                <section>
                    <h2>This is my first main section</h2>
                    <p>I will write some text</p>
                    <p>And even more text</p>
                </section>
                <section>
                    <h2>The next section</h2>
                    <p>This is the next part of my article</p>
                </section>
        </article>
    </body>
</html>
```

Other Common HTML5 Tags

- HTML5 also contains semantic mark-up for common elements of complex websites
    - `header` and `footer` are the type of information that might be repeated on every page of a site
    - An `aside` is not always related to the main article, or could be meant to provide extra information, like a glossary
    - Recognizing that navigation is a meaningful part of websites, the `nav` element is used to group navigation elements, `a` tags or otherwise

Audio and Video

- Before HTML5, external libraries like Flash were needed to display multimedia
- In HTML5, both native audio and video is supported
    - The file formats supported very across browsers
    - The location of the media is defined by the `src` attribute
    - To display controls, the `controls` attribute must be present

In [ ]:

```
%%html
<audio controls src="https://upload.wikimedia.org/wikipedia/commons/a/a4/Washington_Post.ogg"></audio>
<audio controls src="https://upload.wikimedia.org/wikipedia/commons/0/04/Pyotr_Ilyich_Tchaikovsky_-_1812_overture.ogg"></audio>
<audio controls src="https://upload.wikimedia.org/wikipedia/commons/0/04/Pyotr_Ilyich_Tchaikovsky_-_1812_overture.ogg#t=00:15:34"></audio>
```

In [ ]:

```
%%html
<video controls
src="https://upload.wikimedia.org/wikipedia/commons/1/10/Panoramic_view_of_the_Eiffel_Tower_taken_from_the_outside.ogv"></video>
```

In [ ]:

```
%%html
<video  controls
src="https://upload.wikimedia.org/wikipedia/commons/3/37/Front_loading_garbage_truck_loading_a_dumpster.webm"></video>
```

In [ ]:

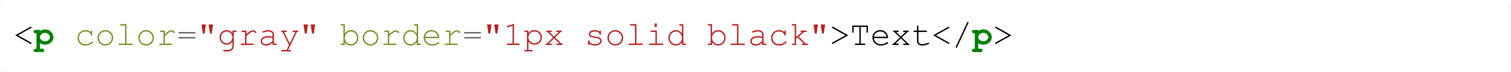
```
%%html
<video controls width="640" height="360"
src="https://upload.wikimedia.org/wikipedia/commons/3/37/Front_loading_garbage_truck_loading_a_dumpster.webm"></video>
```

Debugging HTML

- HTML is a remarkably flexible language
  - The W3C specification list how to parse HTML in erroneous conditions, so content is always displayed
- If the content is not being displayed how you expect, it can be difficult to find the missing end tag or other small typo
  - A good resource in these instances is an HTML validator, which will tell you how your HTML code is not meeting the specifications
  - W3C provides one located at https://validator.w3.org/

CSS and HTML

- CSS is short for cascading style sheets
  - Cascading refers to inheritance
- Prior to the development of CSS in 1996, the style of a website had to be controlled using attributes



```
<p color="gray" border="1px solid black">Text</p>
```

  - This deviates from HTMLs goal of only expressing content

CSS Rules

- A CSS Rule describes what styles to apply to which elements of the page
- A CSS Rule has three main parts
    - Selector
    - Properties
    - Values

```
selector {property1: value1; property: value2;}
```

CSS Location

- CSS can be written
  - In a separate document
  - Inside a `<style>` tag, which is usually in the `<head>` tag
  - Inside the style attribute of a tag
    - Avoid this

Selectors

- The elements to which a style is applied to are controlled by the selector, which can be
  - A tag name
  - An id
  - A class name
  - A psuedo-class
  - A specific nesting of tag names

In [ ]:

```
%%html
<!DOCTYPE html>
<html>
    <head>
        <style>
            p {background-color:gray}
        </style>
    </head>
    <body>
        <div class="main">
            <p>I am a paragraph inside a div! <span>I am a span inside a paragrap
h, inside a div</span></p>
        </div>
        <p id="alone" class="main">I am a paragraph not in a div</p>
        <p lang="es">Soy un párrafo en español</p>
    </body>
</html>
```

In [ ]:

```
%%html
<!DOCTYPE html>
<html>
    <head>
        <style>
            p#alone {border:3px solid black}
        </style>
    </head>
    <body>
        <div class="main">
            <p>I am a paragraph inside a div! <span>I am a span inside a paragrap
h, inside a div</span></p>
        </div>
        <p id="alone" class="main">I am a paragraph not in a div</p>
        <p lang="es">Soy un párrafo en español</p>
    </body>
</html>
```

In [ ]:

```
%%html
<!DOCTYPE html>
<html>
    <head>
        <style>
            .main {border:3px solid blue}
        </style>
    </head>
    <body>
        <div class="main">
            <p lang="en">I am a paragraph inside a div! <span>I am a span inside a paragraph, inside a div</span></p>
        </div>
        <p class="main">I am a paragraph not in a div</p>
        <p lang="es">Soy un párrafo en español</p>
    </body>
</html>
```

Psuedo-Classes

- Psuedo-classes are used to refine selectors to only match elements with certain properties or in certain states
- They are preceded by the colon character
  - :only-child
  - :lang()
  - :hover
  - :disabled
- A full list is available at MDN

In [ ]:

```
%%html
<!DOCTYPE html>
<html>
    <head>
        <style>
            :lang(es) {border:3px solid red}
        </style>
    </head>
    <body>
        <div class="main">
            <p>I am a paragraph inside a div! <span>I am a span inside a paragraph, inside a div</span></p>
        </div>
        <p id="alone" class="main">I am a paragraph not in a div</p>
        <p lang="es">Soy un párrafo en español</p>
    </body>
</html>
```

In [ ]:

```
%%html
<!DOCTYPE html>
<html>
    <head>
        <style>
            a:hover{border:3px dashed gray}
            input:disabled{background-color: white} /*Don't do this*/
            input:checked{width:2rem}
        </style>
    </head>
    <body>
        <a href="">This is a link to nowhere</a>
        <form>
            <div>
                <input type="text" disabled value="You can't change me"/>
            </div>
            <div>
                 <input type="text" value="You can change me"/>
            </div>
            <label ><input type="checkbox">I am a checkbox</label>
        </form>
    </body>
</html>
```

Psuedo-Elements

- Psuedo elements are similar to psuedo-classes, but they can be used to either add or change part of an elements content
  - Like having the first letter wrapped in a span, but with out all the effort
- While psuedo-elements have existed in some form since CSS 1, in CSS3, the syntax was changed to used double colons (::)
  - ::before
  - ::after
  - ::first-letter
  - ::first-line

In [ ]:

```
%%html
<!DOCTYPE html>
<html>
    <head>
        <style>
            p.psuedo-examples{background-color:white}
            p.psuedo-examples::first-letter{font-size:2rem;}
            p.psuedo-examples::first-line{font-weight:900;}
        </style>
    </head>
    <body>
       <p class="psuedo-examples">I am, a very very long paragraph. I am, a very very long paragraph.
        I am, a very very long paragraph. I am, a very very long paragraph. I am, a very very long paragraph.
        I am, a very very long paragraph. I am, a very very long paragraph. I am, a very very long paragraph.
        I am, a very very long paragraph. I am, a very very long paragraph. I am, a very very long paragraph.
        I am, a very very long paragraph. I am, a very very long paragraph. </p>

        <p class="psuedo-examples">I am, a very very long paragraph. I am, a very very long paragraph.
        I am, a very very long paragraph. I am, a very very long paragraph. I am, a very very long paragraph.
        I am, a very very long paragraph. I am, a very very long paragraph. I am, a very very long paragraph.
        I am, a very very long paragraph. I am, a very very long paragraph. I am, a very very long paragraph.
        I am, a very very long paragraph. I am, a very very long paragraph. </p>
    </body>
</html>
```

CSS Properties

- There are many more CSS properties than there are HTML elements
  - Some only have effects on certain elements
  - Some can be used almost anywhere
- Each property has a set of possible values
  - You'll notice some general themese
- Some properties are shortcut properties

```
div {border:3px solid black;}
  div {border-width:3px; border-style:solid; border-color:black;}
```

Basic Text Styling

- The following attributes are some properties used to style text in an element
  - color
  - font-family
  - font-size
  - font-weight
  - font-style
  - text-transform

In [ ]:

```
%%html
<!DOCTYPE html>
<html>
    <head>
        <style>
            div#text-examples{font-family:"Ubuntu",sans-serif;
                              font-size:2rem;
                              color:#444444;
                              font-weight:800;
                              font-style:italic;
                              text-transform:uppercase;
                              line-height:2.3rem;}
        </style>
    </head>
    <body>
       <div id="text-examples">
           Text Text Text Text Text Text Text Text Text Text Text Text Text Text
           Text Text Text Text  Text Text Text Text Text Text Text Text Text
       </div>
    </body>
</html>
```

Sizing Units on the Web

- There are many different units you can use to size fonts as well as any other element on the web
  - mm, cm, in - Generally avoid, unless you are styling for print
  - pt - Points
  - px - Pixels
  - em - 1em is the size of the capital M in the current element
  - rem - 1rem is the size of the capital M in the root element (HTML)
  - vh,vw - 100vh is the height of the viewport

In [ ]:

```
%%html
<!DOCTYPE html>
<html>
    <head>
        <style>
            p{line-height:1.25em;}
            p#ex1{font-size:20pt;}
            p#ex2{font-size:20px;}
            p#ex3{font-size:10mm;}
            p#ex4{font-size:2em;}
            p#ex5{font-size:5vh;}
            p#ex6{font-size:3vw;}
            p#ex7{font-size:2em;}
            p#ex8{font-size:2em;}
            p#ex9{font-size:2rem;}
            p#ex10{font-size:2rem;}
        </style>
    </head>
    <body>
        <p id="ex1">Example 1</p>
        <p id="ex2">Example 2</p>
        <p id="ex3">Example 3</p>
        <p id="ex4">Example 4</p>
        <p id="ex5">Example 5</p>
        <p id="ex6">Example 6</p>
        <div id="ex7">
            <p id="ex8">Example 8</p>
        </div>
        <div id="ex9">
            <p id="ex10">Example 10</p>
        </div>
    </body>
</html>
```

Positioning

- The positioning of elements is controlled through CSS, although some of these properties are less common nowadays
    - position - Changes the positioning system used top place an element
        - left
        - right
        - top
        - bottom
    - float - allows multiple block elements to be next to each other
    - display - changes block elements to inline, or other options

In [ ]:

```
%%html
<!DOCTYPE html>
<html>
    <head>
        <style>
           div#f1{float:left; width:50%;}
           div#f2{float:left; width:50%;}
           div#f3{float:right; width:50%;}
           div#f4{float:right; width:50%;}
           div#f5{float:left; width:25%;}
           div#f6{float:left; width:25%;}
           div#f7{float:right; width:30%;}
           div#f8{float:right; width:25%;}
        </style>
    </head>
    <body>
        <div id="f1">
            Float 1
        </div>
        <div id="f2">
            Float 2
        </div>
         <div id="f3">
            Float 3
        </div>
         <div id="f4">
            Float 4
        </div>
        <div id="f5">
            Float 5
        </div>
        <div id="f6">
            Float 6
        </div>
         <div id="f7">
```

The Box-Model

- The sizing of an element in CSS is based on the box model <img alt="The diagram of the box model, showing how padding and margin relate to size" src="https://mdn.mozillademos.org/files/13647/box-model-standard-small.png"/ style="width:50%"> Image from MDN

In [ ]:

```
%%html
<!DOCTYPE html>
<html>
    <head>
        <style>
           div#box1{margin:20px;background-color:red;}
           div#box2{margin:40px auto; width:50%; background-color:blue}
           div#box3{background-color:gray;padding:20px}
        </style>
    </head>
    <body>
       <div id='box1'>
        Box1
       </div>
       <div id='box2'>
        Box2
       </div>
       <div id='box3'>
        Box3
       </div>
    </body>
</html>
```