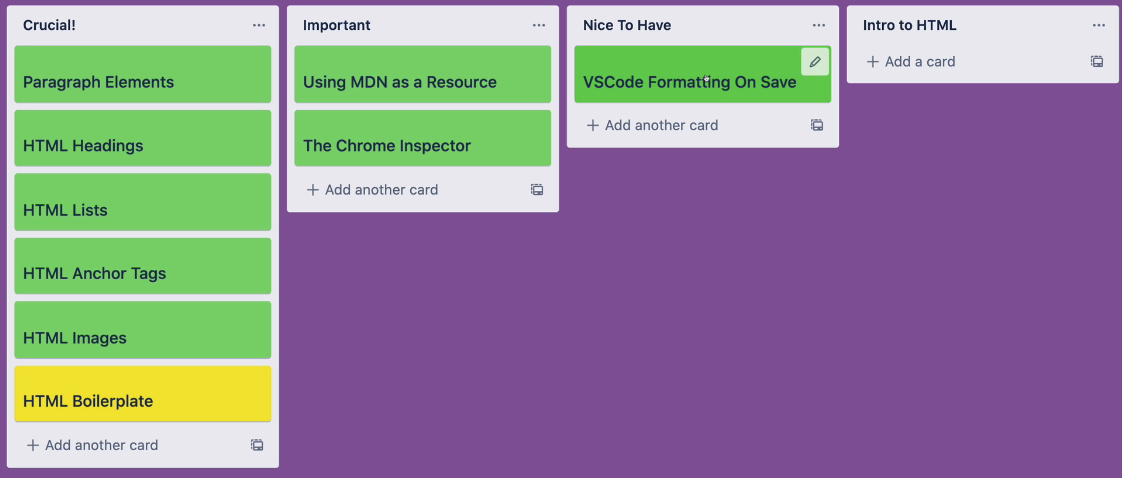
HTML - 1

**Introductory Topic**

**1.1 Objectives**



**1.1 HTML Boilerplate**

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="utf-8">

        <title>BoilerPlate</title>

        <link>

        <script></script>

    </head>

    <body>

        <p>Hello</p>

    </body>

​

</html>

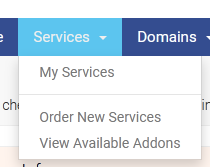
**1.2 Free Unlimited Web-Hosting**

* Domain Name and file Location: ***www.domain.com/path/file.html***

*Domain found* (or error - returned by Internet Provider) **>** *File location* (or error from web-host) **>** *Display*

* Web Hosting: Login and Register in *thecompletewebhosting*. Use *subdomain* from *Developer Island*.

<https://thecompletewebhosting.com/>

Services > order new services: 

Platinum Package

Free sub-domain: Use a *subdomain* from *Developer Island*. Enter a domain name, check the domain name (if it already exist).

To enter a *promo code* to turn the price to 0.

https://thecompletewebhosting.com/student.php

click "url"

**1.3 HTML Headings**

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="utf-8">

        <title>Headingse</title>

        <link>

        <script></script>

    </head>

    <body>

        <h1>Hello 1</h1>

        <h2>Hello 2</h2>

        <h3>Hello 3</h3>

        <h4>Hello 4</h4>

        <h5>Hello 5</h5>

        <h6>Hello 6</h6>

    </body>

​

</html>

**1.4 Paragraph Elements**

* Line Break: <br>

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="utf-8">

        <title>PraGraph elements</title>

        <link>

        <script></script>

    </head>

    <body>

        <p>Hello From paragraph</p>

        <p>Hello From other paragraph, follwoing is line-break <br> this is line break </p>

    </body>

​

</html>

**1.5 HTML Anchor Tags**

* Links: links are defined with the ***<a>*** tag, which stands for ***anchor***. And we used the ***href*** attribute to tell the browser where the link should take us. Link wont active until you use “***href***” attribute.

1. Online page
2. Offline page (local)
3. Same page to navigate

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="utf-8">

        <title>Links</title>

        <link>

        <script></script>

    </head>

    <body>

        <p id = pragraph\_1>Hello Do esse quem an nostrud, laboris irure irure mentitum ipsum. De enim

            reprehenderit, incurreret hic cillum offendit, nulla mentitum arbitror o quae

            occaecat do illum fore ita ita doctrina lorem  ,<br>

        <!--        Go to online html page -->

            <!--"title" is used for hover-Info-->

            <a href="#" title="A Lnk">link</a>

        </p>

        <!--        Go to other local html page -->

        <a href="html\_3\_pagrp\_elmn.html" title="Offline Page">ParaGrapg</a>

        <p> Hello Do esse quem an nostrud, laboris irure irure mentitum ipsum. De enim

            reprehenderit, incurreret hic cillum offendit, nulla mentitum arbitror o quae

            occaecat do illum fore ita ita doctrina loremHello Do esse quem an nostrud, laboris irure irure mentitum ipsum. De enim

            reprehenderit, incurreret hic cillum offendit, nulla mentitum arbitror o quae

            occaecat do illum fore ita ita doctrina loremHello Do esse quem an nostrud, laboris irure irure mentitum ipsum. De enim

            reprehenderit, incurreret hic cillum offendit, nulla mentitum arbitror o quae

            occaecat do illum fore ita ita doctrina lorem </p>

        <!-- move around same page/ use id: For a very big webpage to nevigate -->

        <a href="#pragraph\_1" title="Same Page">ParaGraph</a>

    </body>

​

</html>

* To open the link in a ***new tab***, we need to define the ***target*** attribute and then we choose the ***\_blank*** options.

<a href = "http://www.youtube.com" target="\_blank">Youtube</a> <br>

* Internal links: We can also link to any document inside our current project. For example, other webpages, or even images, or other documents, and those will be internal links. So let's just add a link to the HTML image we used before.

<a href = " wallpaper\_blog.png" target="\_blank">WallPaper</a> <br>

* <a> The Anchor element: 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. It is an Inline-Element. <a href="URL">Text</a>

<p>You can reach Michael at:</p>

​

<ul>

  <li><a href="https://example.com">Website</a></li>

  <li><a href="mailto:m.bluth@example.com">Email</a></li>

  <li><a href="tel:+123456789">Phone</a></li>

</ul>

* If **https://** protocol is not used then it will use file protocol.
* Link another page: I sometimes do want to link to another page that is not on **https://**. But be sure to give it a correct path or it should be on the same file directory of the current page source. **<a href="page2.html">GO TO PAGE-2</a>**

**1.5 HTML Images**

<img *src*="*./img/im2.jpg*"*,* *title*="*Logo 2*"*,* *alt*="*Logo 2*"*,* *width*="*200px*"*,* *height*="*300px*">

**1.6 Inline vs Block-Level elements**

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="utf-8">

        <title>Inline vs Block</title>

        <link>

        <script></script>

        <style>

            .highlight {

            background-color:#ee3;

            }

​

​

​

        </style>

    </head>

    <body>

        <!-- In this example, the <div> block-level element contains some text. Within that text is a <span> element, which is an inline element. -->

            <div>The following span is an <span class="highlight">inline element</span>;

            its background has been colored to display both the beginning and end of

            the inline element's influence.

            </div>

        <!-- A block-level element, such as <p>. The <p> element totally changes the layout of the text, splitting it into three segments: the text before the <p>, then the <p>'s text, and finally the text following the <p>. -->

            <div>The following paragraph is a <p class="highlight">block-level element;</p>

                its background has been colored to display both the beginning and end of

                the block-level element's influence.</div>

    </body>

​

</html>

**1.7 Iframe**

The ***<iframe>*** tag specifies an *inline frame*. An *inline frame* is used to *embed* another *document* within the *current HTML document*.

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="utf-8">

        <title>BoilerPlate</title>

        <link>

        <script></script>

    </head>

    <body>

        <iframe src="html\_3\_pagrp\_elmn.html" width="200px" height="150px"></iframe>

        <iframe src="html\_3\_pagrp\_elmn.html" width="200px" height="150px" frameborder = "1"></iframe>

        <!--Open a page inside an iframe, using "target" attribute -->

        <iframe name="my\_iframe"></iframe>

            <!--target="\_blank" for new tab-->

        <a href="html\_6\_inline\_vs\_block\_lvl.html" target="\_blank">Link open in new tab</a>

            <!--target="my\_iframe" to open in iframe-->

        <a href="html\_5\_images.html" target="my\_iframe">Link open inside iframe</a>

        <br>

            <!--iframe for youtube video: youtube -> share -> imbade -> copy the link-->

        <iframe width="560" height="315" src="https://www.youtube.com/embed/32v8ARqaBas" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

    </body>

​

</html>

**1.8 HTML Lists**

* Unordered-list

        <!--Unordered Lists-->

        <h1>Unordered Lists</h1>

        <ul>

            <li>Madrid</li>

            <li>Spain</li>

            <li>Iran</li>

            <li> London

                <ul>

                    <li>Cloudy</li>

                    <li>Cosmopoliton</li>

                </ul>

            </li>

        </ul>

* Ordered-list

        <ol>

            <li>Fee</li>

            <li>Fo</li>

            <li>Fum</li>

        </ol>

Exercise

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="utf-8">

        <title>Lists</title>

        <link>

        <script></script>

    </head>

    <body>

        <!--Unordered Lists-->

        <h1>Unordered Lists</h1>

        <ul>

            <li>Madrid</li>

            <li>Spain</li>

            <li>Iran</li>

            <li> London

                <ul>

                    <li>Cloudy</li>

                    <li>Cosmopoliton</li>

                </ul>

            </li>

        </ul>

            <!--Ordered Lists-->

        <h1>Ordered Lists using numbers</h1>

        <ol>

            <li>Fee</li>

            <li>Fo</li>

            <li>Fum</li>

        </ol>

        <h1>Ordered Lists using Capital Letters</h1>

        <ol type="A">

            <li>Fee</li>

            <li>Fo</li>

            <li>Fum</li>

        </ol>

        <h1>Ordered Lists using small Letters</h1>

        <ol type ="a">

            <li>Fee</li>

            <li>Fo</li>

            <li>Fum</li>

        </ol>

        <h1>Ordered Lists using 'Roman' Letters</h1>

        <ol type="I">

            <li>Fee</li>

            <li>Fo</li>

            <li>Fum</li>

        </ol>

            <!--Using roman style-->

        <ol type="i">

            <li>Introduction</li>

            <li>List of Greivances</li>

            <li>Conclusion</li>

        </ol>

        <h1>Ordered Lists Starting from "4" </h1>

            <!--Starting from specific number-->

        <ol start="4">

            <li>Speedwalk Stu</li>

            <li>Saunterin’ Sam</li>

            <li>Slowpoke Rodriguez</li>

        </ol>

        <h1>Things I've Learned</h1>

        <h2>Internet Basics</h2>

        <ol>

            <li>HTTP Requests</li>

            <li>IP Address</li>

            <li>Servers</li>

        </ol>

        <h2>HTML</h2>

        <ul>

            <li>Stands for <strong>Hyper Text Markup Language</strong></li>

            <li>Lots of tags

                <ul>

                    <li>Boilerplate

                        <ol>

                            <li>Doctype</li>

                            <li>HTML</li>

                            <li>Head

                                <ol><li>Title</li></ol>

                            </li>

                            <li>Body</li>

                        </ol>

                    </li>

                    <li>Headings</li>

                    <li>Paragraph</li>

                    <li><em>em</em></li>

                    <li><strong>strong</strong></li>

                </ul>

            </li>

        </ul>

    </body>

​

</html>

**1.9 Description-list**

* The ***<dl>*** tag defines a *description list*.
* The ***<dl>*** tag is used in conjunction with ***<dt>*** (defines *terms/names*) and ***<dd>*** (describes each *term/name*).

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="utf-8">

        <title>Description Lists</title>

        <link>

        <script></script>

    </head>

    <body>

        <dt>HTML</dt>

        <dd>-Stands for Hypertext Markup Language</dd>

        <dt>CSS</dt>

        <dd>-Stands for Cascadinh Style sheet</dd>

​

        <!--Proper Description lists-->

        <h2>Proper Description lists</h2>

        <dl>

              <dt>Coffee</dt>

              <dd>Black hot drink</dd>

              <dt>Milk</dt>

              <dd>White cold drink</dd>

        </dl>

    </body>

​

</html>

**1.10 HTML Tables**

* The ***<thead>*** tag is used to group header content in an HTML table. The ***<thead>*** element is used in conjunction with the ***<tbody>*** and ***<tfoot>*** elements to specify each part of a table (header, body, footer).
* Usage: Browsers can use these elements to enable scrolling of the table body independently of the header and footer. Also, when printing a large table that spans multiple pages, these elements can enable the table header and footer to be printed at the top and bottom of each page.
* Note: The ***<thead>*** element must have one or more ***<tr>*** tags inside.
* The ***<thead>*** tag must be used in the following context: As a ***child*** of a ***<table>*** element, after any ***<caption>*** and ***<colgroup>*** elements, and before any ***<tbody>, <tfoot>,*** and ***<tr>*** elements.
* Tip: The ***<thead>***, ***<tbody>***, and ***<tfoot>*** elements will not affect the layout of the table by default. However, you can use CSS to style these elements.
* <table> The Table element: The HTML **<table>** element represents tabular data — that is, information presented in a two-dimensional table comprised of rows and columns of cells containing data.
* **<tr>** defines a row of cells in a table. The row's cells can then be established using a mix of **<td>** (***data cell***) and **<th>** (***header cell***) elements.
* **<td>** defines a cell of a table that contains *data*.
* **<th>** defines a cell as *header* of a group of *table cells*.
* To visible margin use **border='1'** attribute
* **<thead>** element defines a set of rows defining the head of the columns of the table.
* **<tbody>** encapsulates a set of table rows (<tr> elements), indicating that they comprise the body of the table (<table>).

In HTML5 we use **<thead>** and **<tbody>**

**1.11 HTML entities**

HTNL doesn’t show some characters. For example: "**<**" and "**>**" or space "". To show those in HTML, we have to use some special character of HTML or **HTML entities**.

* Some characters are reserved in HTML. If you use the less than (<) or greater than (>) signs in your text, the browser might mix them with tags.
* Character entities are used to display reserved characters in HTML. A character entity looks like this:

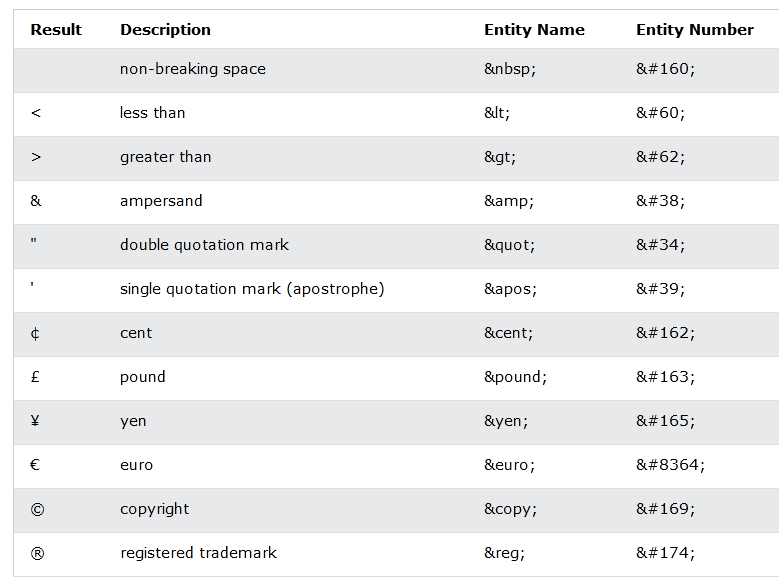
&entity\_name;

OR

&#entity\_number;

EG: To display a less than sign (**<**) we must write: **&lt;** or **&#60;**

**Some Useful HTML Character Entities**



* Advantage/Disadvantage of using an entity name:
* An entity name is easy to remember.
* Browsers may not support all entity names, but the support for entity numbers is good.
* To use space:

        <p>Hello &nbsp;&nbsp;&nbsp;&nbsp; There</p>

        <p>Hello &#160&#160&#160&#160&#160 There</p>

        <!--To view less-Than-->

        <p>Less &lt; Greater</p>

        <!--Copyright symbols-->

        <p>Less &copy; Greater</p>

**1.12 HTML Forms**

HTML forms are very important to build any kind of website. Fields that filled by users with their info and then send those info to a server.

* Create a simple Login form:

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="utf-8">

        <title>Simle Form</title>

        <link>

        <script></script>

    </head>

    <body>

        <h1>Simple Form</h1>

        <form>

            <label for="email">Email</label>

            <input type="text" name="email" placeholder="Email" size="30px" maxlength="100" id="email">

            <label for="password">Password</label>

            <input type="password" name="password" placeholder="Password" size="30px" maxlength="10" id="password">

            <input type="submit" value="login">

        </form>

    </body>

​

</html>

* Here id's in input tags are used to link the <label> tags.

More form inputs:

<!DOCTYPE html>

<html lang="en">

​

<head>

    <meta charset="utf-8">

    <title>Simle Form 2</title>

    <link>

    <script></script>

</head>

​

<body>

    <h1>Simple Form</h1>

    <form>

        <label for="email">Email</label>

        <input type="text" name="email" placeholder="Email" size="30px" maxlength="100" id="email">

        <label for="password">Password</label>

        <input type="password" name="password" placeholder="Password" size="30px" maxlength="10" id="password">

        <input type="submit" value="login">

        <br>

​

        <label for="address">Address: </label>

        <input type="text" name="address" placeholder="Address" id="address" sixe="20px" maxlength="200">

        <br>

        <label for="postcode">Postcode: </label>

        <input type="text" name="postcode" placeholder="Postcode" id="postcode" sixe="20px" maxlength="200">

        <br>

        <label for="city">City: </label>

        <input type="text" name="city" placeholder="City" id="city" sixe="20px" maxlength="200">

        <br>

​

        <p>Choose your shipping method</p>

        <select>

            <option value="Free super saver shipping">Free super saver shipping</option>

            <option value="Two dayr shipping">Two day shipping</option>

            <option value="One day shipping">One day shipping</option>

        </select>

​

        <p>Choose your payment method</p>

        <label>Paypal<input type="radio" name="Payment" value="Paypal"></label><br>

        <label>Debit Card<input type="radio" name="Payment" value="Debit Card"></label><br>

        <label>Credit Card<input type="radio" name="Payment" value="Credit Card"></label><br>

​

        <label for= "comments">Comments:</label>

        <textarea rows="4" cols="40" name="comments" id="comments"></textarea>

        <br>

        <label>Signup for Email updates<input type="checkbox" name="email updates" checked = "checked"></label><br>

        <label>Securely save my Payment details<input type="checkbox" name="Save Payment" checked = "checked"></label><br>

        <input type="submit" value="submit">

​

    </form>

</body>

​

</html>

​

**1.12 HTML text Decoration**

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="utf-8">

        <title>Text Decoration</title>

        <link>

        <script></script>

    </head>

    <body>

        <p>This word is <em>emphesized</em></p>

        <p>This <strong>word</strong> is Strongly <strong><em>emphesized</em></strong></p>

        <p>This <strong>word</strong> is niet Strongly <i>italic</i></p>

        <p>This <strong><strike>word</strike></strong> is crossed out or struck through</p>

        <p>Subscript-Superscript CO<sub>2</sub>...X<sup>2</sup></p>

    </body>

​

</html>

**1.14 HTML comments**

<!--this is a comment-->

​

<!DOCTYPE html>

<!--this is a comment-->

<html lang="en">

    <head>

        <meta charset="utf-8">

        <title>Text Decoration</title>

        <link>

        <script></script>

    </head>

    <!--this is a comment-->

    <body>

        <!--this is a comment-->

        <p>This word is <em>emphesized</em></p>

    </body>

​

</html>

Kalob Taulien

**1.15 HTML**

* HTML Elements,
* tags,
* boilerplate,
* Block element
* Attributes:
  + Key-Value pair.
* Line break
* Headers & Paragraphs,
* HTML entity
* Links,
  + web-link,
  + same page (id = "top"; href = "#top"),
* Images,
  + src,
  + border,
  + height,
  + width,
  + alt
* Formatting text:
  + strong,
  + em,
  + i,
  + u.
  + Combine tags.
  + Sub & sup
  + Code: to display codes
  + <pre>: to show formatted text.
  + ***style*** attribute: inline css. style = "color: red"
    - font-size
    - font-weight
    - color
    - background-color
    - padding
    - margin
* Html comments
* <div>; block, display: inline block
* Lists:
  + ul: type= square/disc/circle
  + li
  + ol: type= 1/a/i
  + <dl>: Definition/Description list, <dt>: Definition title, <dd>: Definition dsecriptin
* HTML-Tables: Style table element-text using CSS.
  + table, tr, td
  + cellspacing, cellpading
  + colspan, rowspan
* Forms: place following inputs inside <form>
  + Text box: <input type ="text">
  + Always use "name"
  + Password: <input type =" password">
  + Dropdown: <select>, <option>. Similar to unordered-lists.
  + Text-areas: <textarea>
  + Radio buttons: <input type ="radio">, to link use same name, different values & ids
  + Checkbox: <input type ="checkbox">, to link use same name, different ids (no values because 1/0)
  + Buttons: submit,
    - button inside form
      * <button type="button>
      * <input type="button>
      * <button type="reset>. Clear the form inputs
      * <input type="reset>. Clear the form inputs
      * <button type="submit>
      * <input type="submit>
* Inline frames/Iframes: Page inside page
  + <iframe src="">
  + Fareme border
  + Height, weight
  + Scrolling: yes, no, auto
* <!DOCTYPE>: Declaring the doctype
  + <head>
  + *<meta name="description", content = "this will describe the webpage">*: SEO
  + *<meta name="keywords", content = "learn, html, html5, css, css3">*: SEO
  + *<meta http-equiv="refresh", content = "5">* : refresh in 5 seconds
  + *<meta http-equiv="refresh", content = "0; URL=newpage.html">* Redirect to another page
  + There are also other <meta>'s. Most importants are: ***description***, ***keywords***, ***Cache-Control***, ***expires*** and ***copyright***.
  + <title>
  + <link>
  + <style>
  + <script>
* HTML hierarchy: Similar to CSS: Useful for CSS and JS to select an element.
  + Parent
  + First child
  + First grand childs
* HTMLEntities: **&lt;** **&gt;**
* **New elements are for semantic solution**
* <header>
* <footer>
* Spellchesk & counteditable: Useful for spell cheking
* <p spellcheck = "true">Spel check in this sentence</p>
* <p spellcheck = "true" counteditable = "true">Spel check in this sentence</p>
* Video:

        <video autoplay="autoplay" controls="controls" loop="loop">

            <source src="qncd.mp4" type="video/mp4">

            <source src="https://qncd.ogg" type="video/ogg">

            <source src="qncd.mp4" type="video/mp4">

        </video>

        <button onclick="**PlayorPause**()">Play or Pause</button>

        <button onclick="**bigScreen**()">Big screen baby!</button>

        <button onclick="**normalScreen**()">just regular size for me</button>

        <br>

        <video id="lecture" width="300" controls="controls">

            <source src="./video/video.mp4" type="video/mp4">

            Your browser is either old if doesn't support MP4 or OSG formats

        </video>

        <p>This word is <em>emphesized</em></p>

        <script>

            var lecture = document.**getElementById**("lecture");

            function **PlayorPause**(){

                if (lecture.paused){

                    lecture.**play**();

                } else {

                    lecture.**pause**();

                }

            }

            function **normalScreen**(){

                lecture.width = 300;

            }

            function **bigScreen**(){

                lecture.width = 500;

            }

        </script>

* Audio
* It is similar to Video
* Drag and drop: Use the attributes: *draggable*, *ondragstart*, *ondragover*, *ondrop*. *Onclick*, *onfocus*, *onblur*.

    <head>

        <meta charset="utf-8">

        <title> Video</title>

        <link>

        <script type="text/javascript">

            function **allowDrop**(e){

                e.**preventDefault**() ;

                // prevents the image from moving back to its natural area

            }

            function **drag**(e) {

                e.dataTransfer.**setData**("Text", e.target.id);

                // allows the item to be dragged

            }

            function **drop**(e) {

                e.**preventDefault**();

                d = e.dataTransfer.**getData**("Text"); // d = data

                e.target.**appendChild**(document.**getElementById**(d));

                // this code allows you to drop the item

            }

        </script>

    </head>

    <!--this is a comment-->

    <body>

        <img draggable="true" id="DragMe" src="./img/po\_1\_Capture.JPG" ondragstart="**drag**(event)" height="300" width="300">

        <!-- drop Zone area -->

        <div ondrop="**drop**(event)" ondragover="**allowDrop**(event)" id="dropzone" style="border: 1px; height: 350px; width: 350px;"></div>

    </body>

* Geolocation: We can access the location of a user at any time. But not without their permission first, since this very clearly invades their privacy.
* We use the ***getCurrentPosition()*** function in JavaScript to find where someone is

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="utf-8">

        <title>Geolocation</title>

        <link>

        <script type="text/javascript">

        </script>

    </head>

    <body>

        <p>Hello</p>

        <p id="coords"></p>

        <button onclick="**getMyPosition**()">Find me!</button>

        <script>

        var MyLoc = document.**getElementById**("coords");

        function **getMyPosition**() {

            if (navigator.geolocation) {

                navigator.geolocation.**getCurrentPosition**(showPosition);

            } else {

                MyLoc.innerHTML = "Geolocation is not supported by this browser.";

            }

        }

        function **showPosition**(position){

            MyLoc.innerHTML = "Latitude : "+position.coords.latitude + "<br>" + "Longitude : "+position.coords.longitude;

        }

        </script>

    </body>

</html>

* If you try\* this in your browser, you may run into a security problem The browser will, by default, protect your location.
* So you might need to go into your settings and change your location position settings.
* Once you upload this script to a website, the browser should ask you if you want to allow *mywebsite.com* to access your position and it will save that for you. (Browsers vary Settings depend on the browser and version)
* Adding envent to a button outside a form: You may have also noticed that we used the ***onclick*** attribute in our ***<button>*** element. When you click the button, by default nothing will happen because it's not in a ***<form>.*** But when you add the ***onclick*** attribute and set the value to a JavaScript ***function***, it will perform that function for you.
* localStorage & sessionStorage:
* With older HTML we cannot store user data. If they put information into a form, neither they or us can do anything with that information.
* With the help of ***localStorage***, we can store the ***users data*** - and any amount of it! With websites such as Google or Facebook, they hold our information in a database. With HTML5, they can make the *users browser* store *their own information*. This is beneficial because they can *store any amount of information* without *hurting the performance of our server*. However, the user needs an HTML5 browser and needs to have Ja%aScript turned on To test this, we can use die following script

        <script>

            if (typeof(Storage)!=="undefined"){

                // we can accept localStorage

**alert**("Accepted");

            } else {

                // we cannot accept localStorage

**alert**('No localStorage accepted');

            }

            localStorage.myName = "Plinki Bao";

**alert**(localStorage.myName);

        </script>

* If put that code in your ***<header>*** and run it in your browser, you will be prompted with either "Accepted" or "No localStorage accepted".
* If you are not accepted, you should turn on your JavaScript in your settings or update your browser. If the previous HTML? classes didn't work for you, this is probably why
* To store information in the browser using *localStorage*, we can simply use:

localStorage.myName = "Plinki Bao";

And to access this, we use locaStorage.KyName as our variable, and call it like so:

**alert**(localStorage.myName);

* The great thing about **localStorage** is that it will stay in the browser until you either delete it using another script, or the user clears their cache from the browser.
* In the past websites used *cookies* to *store information*. But a *cookie* can only hold *x amount of data*, is not as secure as **localStorage**, and can be accessed from other sources. With **localStorage**, this information can only be accessed from the *website* that *stored* it.
* If you left this page and came back to it at a later time, you can **still** access **locaStorage.myName** and it will have the same value as it does now. There is a way to create a more **temporary** storage and well cover this next
* sessionStorage:

Along the lines of ***localStorage***, we also have ***sessionStorage***. The difference is *how long the data stays* with the user for.

* localStorage = Stays alive until a *script* from that *same website destroys it*, or the user clears their *cache*
* sessionStorage = Stays ali\e until the *window is closed*

To tell if the user can use sessionStorage, use the same code as localStorage.

            if (typeof(Storage)!=="undefined"){

                // we can accept localStorage

**alert**("Accepted");

            } else {

                // we cannot accept localStorage

**alert**('No localStorage accepted');

            }

* Instead of "localStorage" we now use "sessionStorage" to store data

            sessionStorage.seName = "Pow Baw";

**alert**(sessionStorage.seName);

* Add this code into a new html page, and dick the button that appears. Once you've clicked the button a few times, refresh the page (f5 usually). Then dose the window, and re-open it Notice what happens

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="utf-8">

        <title>BoilerPlate</title>

        <link>

        <script type="text/javascript">

            function **counter**(){

                if(typeof(Storage)!=="undefined") {

                    if(sessionStorage.counter) {

                        sessionStorage.counter = **Number**(sessionStorage.counter)+1;

                    } else {

                        sessionStorage.counter = 1;

                        }

                    document.**getElementById**("number").innerHTML= sessionStorage.counter;

                    if (sessionStorage.counter>1) {

                        document.**getElementById**("number").innerHTML+="<p>Refreah the page</p>";

                    } else {

                        document.**getElementById**("number").innerHTML+="<p>Sorry, your browser does not support web atorage...</p>";

                    }

                }

            }

        </script>

    </head>

    <body>

        <p>Hello</p>

        <button onclick="**counter**()" type="button">Add another one!</button>

        <div id="number"></div>

        <script>

        </script>

    </body>

</html>