* Browser- It is important to note that many users across the web do not use the latest browsers and may not support all of the latest features
* Devices- Many people access your webpage in a range of devices including desktop computers, laptops, mobile phones. It is important to remeber your website must suppost several screen sizes.
* Sreen Readers - Mayn laws have passed that require websites be accessbile to those with disabilites.
* e-commece - commercial transactions conducted electroncally on the internet
* tag and element are often used interchangeably
* opening tags can contain an attribute that provide additional information about the element
  + the attribute is written in a name:value fashion > lang=”en-us”
* above the body you will often see a head element with a title, the title shows up at the top of your browser (above the url)
* HTML stands for HyperText Markup Language
* Html has 6 levels of headings
* The <p> tag is used for paragraphs. Paragraphs by default will start on anew long and have some spacing between each other.
* The <b> tag allows for a piece of text to be bolded.
* The <i> tag allows for piece of text to be italicized
* The <sup> tag allows for a piece of text to be superscripted
* The <sub> tab allows for piece of text to subscripted
* HTML will collapse extra white space from the html file. Extra white space can be used for readability
* <br /> tag is used for a line break
* <hr /> tab will create a horizontal separator between paragraphs

**<p>**Venus is the only planet that rotates

clockwise.**</p>**

**<hr />**

**<p>**Jupiter is bigger than all the other planets

combined.**</p>**

* Semantic Markup - do not change the structure of your page but they provide extra information to the pages.
* <strong> indicates that its content is important, browser will bold this content by default
* <em> puts emphasis on its content. The browser will show the content in italics
* <blockquote> element is used for longer quotes that take entire paragraph. Browsers tend to indent the contents inside a blockquote.
* <q> tag can be used to insert quotations into content, but IE does not.
* <abbr title=”Professor”>Prof</abbr> Will show the title when the user mouses over the abbreviated text
* <cite> is used for citations, it will italicize the text
* <dfn> is used for definitions
* <address> element is used for addresses
  + <href> can be used as well to link to mailto: addresses
* <s> to put a strikethrough

Lists

* <o1> can be used to create an ordered list
  + Each item in the list is written with the <li> tag
  + An ordered list will list items with numbers
* <ul> is for unordered lists
  + An unordered list will list items with bullet points
* <dl> is used for definition lists
  + <dt> for the term
  + <dd> for the definition
  + The definition list with show the term with nothing before it and then the definition indented below it

Links

* Links are created using the <a> element
  + You can specify the link using the href attribute
  + The text in between the <a> tag will be the hyperlink text
* You can an absolute URL’s to link any other page on the internet
* Or you can use a relative URL that will link to another page on your website
  + These files can be referenced by name if they are in the same folder
  + Or you will need to specify the folder name if they are not
* You can link to element id’s that you have assigned to other elements.
  + The href will start with a #
  + <a href=”#top”>
  + You can also link to a part of another page by specifying the href followed by the id
* Each sub-directory of a website must contain an index.html
  + Same folder: index.html
  + Child Folder: music/index.html
  + Grandchild Folder: music/artist/index.html
  + Parent Folder: ../index.html
  + Grandparent Folder: ../../index.html
* mailto wil open the user’s email program
* use the target attribute to open a new link in a new window
  + target=”\_blank”

Images

* It is good practice to create a folder for all of the images that site uses
  + If you’re site is large, you may want to create further sub directories
* <img> is an empty element used to add images
  + The src attribute specifies the url to the image
  + The alt attribute will show a text description of the image that describes the image if you cannot see it. This attribute will also be used by screen reader software.
    - This is required but can be left with empty quotes
  + The title attribute will show as a tooltip
  + Height and width can also be defined, but is now used in CSS
* Where you place the image in relation to a paragraph matters
  + Before a paragraph: the paragraph starts on a new line
  + Inside the start of the paragraph: the paragraph text will align with the bottom of the image
* The align attribute effects how other elements flow around the image
  + Left: allows text to neatly flow around the right side
  + Right: allows text to neatly flow around the left side
  + Alignment is now handled by css
  + Top,middle and bottom: are used to specify where ONE line of text will be in relation
* Three rules for images
  + Websites use jpeg, gif or png
    - Images with a lot of colours should use jpeg
    - Images with few colours should use gif or png
  + Size matter, it could be stretched or shrunk if its size does not match the CSS style
  + Use the correct resolution
* Bitmap images are drawn by pixel whereas vector image are drawn by lines
* Transparent images should is a png format
* Images can have a parents element called <figure>
  + Figured can have a caption using the <figcaption> tag

Tables

* The <table> tag is used to create tables
  + <tr> will create a table row
    - <td> will create a table cell for the row
  + Table headings can also be added using the <th> tag
    - A table header uses a scope attribute to specify row or column
      * Scope=”col” is used for columns, column headers use their own table row
      * Scope=”row” is used for rows, the will appear to the left of the table
  + You must specify and empty table cell to render the data properly
  + The colspan attribute can be used to stretch a cell across multiple columns
  + The rowspan attribute can be used to stretch a cell across multiple rows
* A table can use three elements to help distinguish the content
  + <thead> for table headers
    - This allows for a table header to still be visible when the content is too long to be displayed on one screen
  + <tbody> for table content
  + <tfoot> for the footer element
* Tables also have width, cellpadding and cell spacing attributes.
  + Replace by css
* Border attribute is used to specify the border width
  + Replaced by css

Forms

* <form>
  + Action attribute of the form should point to a URL that will receive the form data
  + Method attribute can be either “get” or “post”
    - get is used for retrieving data from the web server and are added to end of the URL
    - post sends the values via HTTP headers which allows for a user to upload data
* <input> elements can be used to add input tools on your website
  + The name of the input field is what the server will expect along with its value. Each form control must have a name.
  + Can have a required=”required” attribute
  + Type=”text” will display one-line text field
    - The name attribute will specify the input field’s name
    - Can use maxlength
    - Size limits the field size in characters, done with CSS now
  + Type=”password” creates a text field with blocked out characters
    - Name, maxlength, size
    - This does not secure the data, you must use Secure Sockets Layer (SSL) to transmit secure data
  + Type=”textbox”
    - This is not an empty element. The text that you write for this element is used as a placeholder
  + Type=”radio” creates a multiple choice input field
    - Name attribute is sent to the server with the value
    - Unlike the other types, this type will have several instances of the same name, but with different values
    - The checked attribute will allow you to select an option when the page loads
      * **<form action="http://www.example.com/profile.php">**
      * **<p>**Please select your favorite genre:
      * **<br />**
      * **<input type="radio" name="genre" value="rock"**
      * **checked="checked" />** Rock
      * **<input type="radio" name="genre" value="pop" />**
      * Pop
      * **<input type="radio" name="genre" value="jazz" />**
      * Jazz
      * **</p>**
      * **</form**
  + Type=”checkbox” allows users to select or deselect one or more options
    - Similar to the radio type this type needs several instances with the same name but different values
* <select> element will create dropdown list box
  + The name attribute indicates the name of the form control sent with the value
  + The <option> element is used to create a list item
    - The text option will have a value attribute that will be sent to the server
    - The text option will also have some content in between the opening and closing tags for the user to see on the website
  + The selected attribute can be used to select an item when the page loads
  + You can set multiple=”multiple” to allow a user to select multiple options
* <input type=”file”>
  + This will create a box that looks like a text input followed by a browse button
  + This input type requires that the form uses the “post” method.
* <input type=”submit” value=”Submit”>
  + Creates a button to send a form to a server
  + The value specifies the text displayed in the button
* The <button> element will create a button
  + You can add an image element inside of a <button> element
* Input type=”hidden”
  + will hide a form control from the user but adds value to the author
  + For example, the author can use this control to indicate which page the user was on when the form was submitted
* <label for=”element id”> elements are used to display text for an input rather than naming it in the input. This makes the form accessible to vision-impaired users.
  + The for attribute should be named after the input element’s id
  + The position of your label in comparison to the input will affect the structure of the site
* You can group form controls together using a <fieldset> element
  + The <legend> element can follow the fieldset element to provide a form caption
* You can have a date input type
* You can have an email input type
  + Some browsers will perform validation on this input
  + Some phones also display a special keyboard
* You can have an url type
  + Some browsers will perform validation on this input
  + Some phones also display a special keyboard
* You can have a search type
  + Some browsers will add a clear search button to the input field
* Any text input fields can have a placeholder attribute

Extra Markup

* Each web page should begin with a DOCTYPE declaration to tell a browser which verion of html the page is using
* Web pages that use the XTHML strict DOCTYPE will have a XML declaration at the beginning of the file.
* Comments can be added by: <!-- -->
* Element id’s must start with a letter or underscore
* Every HTML element can be assigned a CSS class using the class attribute
* Block elements are those that will always start on a new line
* Inline elements will always continue from the previous element
* The <div> element allows you to group a set of elements together. This is useful to apply CSS styling to a group of elements. Divs are block elements.
* The <span> element is an inline equivalent of a div. It is used to:
  + Contain a section of text where no other element is suitable
  + Contains a number of inline elements
* <iframe> is like a page that has been put inside of your page.
  + Common use onf an iframe is google maps
  + Src specifies the url of the page to show
  + Height and width should also be defined
* <meta> element lives inside the pages <head> element and is used to provide information about that web page but is not visible to users.
  + It can tell search engines about your page, who created it or even specify an expiry time.
  + The meta element commonly contains a name and value:
  + Name: description, Value: the description is used by search engines to understand the content of the page. Maxlength is 155 characters.
  + Name: keywords, value: comma-separated words that a user might search
  + Name: robots, value: indicates if a search engine should add this page to search-results
  + The <meta> element also has an http-equiv and content attribute pair.
    - http-equiv=”author”
    - http-equiv=”pragma”: prevents a browser from caching page
    - http-equiv=”expires”: indicates when a cache should expire
* Some characters have been reserved in HTML and need to be specified by escape characters
* All elements are treated as boxes

Flash, Video & Audio

Introducing CSS

* The <link> element can be used in an HTML document to tell the browser where to find the CSS file
  + Href specifies the path to the file
  + Type specifies the type of the document, the value should be “text/css”
  + Rel=”stylesheet”j specifies the relationship between the HTML page and the file link
* There are several different types of selectors:
  + Universal selector: \*
  + Type Selector: h1, h2 match element names
  + Class selector: .class match class attributes
    - P.class match elements with the class
  + ID selector: #ID match element ID’s
  + Child selector: ID>.class matches direct children of ID with the class
  + Descendant selector: p a {} matches an element that is a descendent of another
* There are precedence rules for CSS
  + The last definition will take precedence over another preceding definitions
  + A more specific selector will take precedence over another.
* Some CSS properties are inherited by default. Or you can specify inheritance by setting the value of a property to “inherit”

Color

* The color property changes the color of the text and can be defined in three ways, RGB, Hex, or predefined color names.
* Background-color property sets the color of the background for that box
* Opacity can be used to give transparency to an element. Its value is between 0 and 1.
* Colors can also be defined using HSL (Hue, saturation, lightness)
  + Hue is like a color wheel, value between 0 and 360
  + Saturation specifies the amount of gray in a colour
  + Lightness specifies the amount of black in a colour
  + Example: color: hsl(0, 100%, 100%)
* Alpha values can be added to RGBA and HSLA to specify opacity

Text

* Serif font have extra detail on the strokes of letters, they are considered easier to read
* Sans-serif fonts have straight ends to letters, making them more compact. Screens that have lower resolution look cleaner with sans-serif.
* Monospace font makes each letter have the same width, making alignment nice.
* Font-weight not only adds emphasis to a text, but also affects the whitespace
* Note that the font must be installed on the user’s computer to be displayed
  + You can specify a list of fonts separated by commas
* Font-size property specifies the size of the font (can be describe in px or %)
  + Default size is 16px
  + And em is equivalent to the width of letter ‘m’
  + Use pt for printer-friendly pages, why?
* @font-face can be used to specify a font that will be downloaded if it is not installed
  + Font-family specifies the name of the font
  + Src specifies the location of the font
    - Each browser has its own font format, so you may need to specify multiple src’s
  + Format specifies the format that the font is supplied in.
  + [www.fontsquirrel.com/fontface/generator](http://www.fontsquirrel.com/fontface/generator)
    - This will generate various formats for you and provide the CSS code for the @font-face
* Font-weight allows you to create bold text. This value can be normal or bold.
* Font-style defines italic text. This value can be normal, italic or oblique
* Text-transform used to change the text casing. Uppercase, lowercase, capitalize
* Text-decoration: none, underline, overline, line-through, blink
* Line-height sets the height of the entire line of text
* Kerning is the term used for space between letters. CSS: Letter-spacing and word-spacing
* Text-align: left, right, center, justify.
  + Justify: every line will take up the fill width except the last line
* Vertical-align is used to align inline elements such as <span> and <img>. Can take many values.
* Text-indent used to indent the first line of text.
* There are pseudo-elements and classes in CSS that can also have their own style.
  + i.e. first-letter, first-line, link, visited, hover, active, focus, etc.
* There are selectors for a CSS rule:
  + Existence: p[class] targets any paragraph with a class attribute
  + Equality: p[class=”dog”] targets any paragraph with a class with the value of dog
  + Space: p[class~=”dog”] targets any paragraph with a list of classes that contain dog
  + Prefix: p[attr^”d”] targets any paragraph with an attribute whose values begins with ‘d’
  + Substring: p[attr\*”do”] targets any paragraph with an attribute whose values contain ‘do’
  + Suffix: p[attr&”g”] targets any paragraph with an attribute whose value ends with ‘g’

Boxes

* By default a box is just big enough to hold its contents
  + Width and height can be defined the pixels, ems, or percentage.
  + Recently, developers use ems or percentages to create designs flexible across different screen sizes
* Min-width, max-width, min-height and max-height can be used to specify the size of a box when a browser is too large or too small.
  + These are helpful properties to maintain the legibility of content
* The overflow property specifies what happens when the text is too large to fit in a box.
  + Can have the value of hidden or scroll to show a scroll bar
* Every box has a border that is set to a width of 0 by default
* Margins sit outside the edge of the border, you can set the width of a margin to create space between boxes
  + Margin will added to a specifies width of a box
  + If two boxes have margining between them, the larger margin will be used
  + If you want content to be centred within a box, set the left and right margin to “auto”
    - A box needs a defined width for this to work, otherwise it will span across the page.
* Padding is the space between the border of a box and the content within it.
  + Padding will be added to a specified width of a box
* Border-width: used to control the width of a border. Can be pixels or ‘thin’, ‘medium’ or ‘thick’.
  + You can define border-top, border-right…
  + Or border-width: 2px 1px 1px 2px;
* Border-style is used to add some styling to the border. There are several values this can have.
* Border-color is self-explanatory. You can define the color of each side separately.
* You can change an element to be inline/block elements by using the “display” property
  + Values:
    - Inline
    - Block
    - Inline-block: changes an element to inline but maintain block features
    - None: hides an element from the page
* Visibility: will hide the element but leave space where it would have been
  + Values: hidden or visible
* Border-image applies an image to the border of a box
  + The specified image will be sliced into 9 pieces. Each corner of the slicing will be placed in the border’s corners but we have options with the edges.
    - An edge can be stretched, repeated or rounded.
      * Stretch: stretches the image to fill the border
      * Repeat: repeats the image to fill the border
      * Round: like repeat but will scale a larger image to fit the border
* Border-radius is used to create rounded corners on a box
  + You can create an elliptical shape for radii as well

Lists, Tables and Forms

* The list-style-type property allows you to control the shape or style of a bullet-point
  + Unordered list values:
    - None, disc, circle, square
  + Ordered list values:
    - Decimal, decimal-leading-zero, lower-alpha, upper-alpha, lower-roman, upper-roman.
* List-style-image can be used to assign images as bullet points
  + Syntax: list-style-image: url(“images/star.png”);
  + This property can be used in <ul> and <li> elements
* List-style-position can be used to move the list marker inside/outside of the box
* The above three properties can be defined in one list-style property
* Use various text properties for a table to make it easier to read such as:
  + Bold the headers
  + Give cells padding
  + Align text to the left and numbers to the right
  + Change background-colour of alternating rows
* Empty-cells property can be used to show, hide or inherit the cell borders
* Border-spacing can be used on a table to control the gap between cells
* You can also add styling rules to input fields!
  + Font-size
  + Color
  + Border
  + Border-radius
  + Background-image
* Fieldset’s and legends can have styling too. This can be useful when you want to change the style of multiple form controls.
  + Width
  + Color
  + Background-color
  + Border
  + Border-radius
  + Padding
* Aligning form controls can be tricky, use a combination of block elements and inline elements to achieve the desired look.
  + One trick to assign a global width for all labels/spans and align them.
* The cursor property can be used to change the style of cursor.

Layout

* Four positioning schemes:
  + Normal flow: each block level element appears on a new line
    - Position: static;
  + Relative positioning: The normal flow is maintained but an element can be moved in relation to its original position but does not affect its surrounding elements.
    - Position: relative;
      * Offset properties: top, bottom, left, right (px, ems, %)
  + Absolute position: The element is taken out of the normal flow and positioned in relation to its container.
    - Position: absolute;
      * Offset properties: top, bottom, left, right (px, ems, %)
  + Fixed position: positions an element in relation to the browser window. This element will stay in place even if the user scrolls up or down.
    - Position: fixed;
      * Offset properties: top, bottom, left, right (px, ems, %)
  + When using relative, absolute or fixed positioning, it is possible to run into overlapping elements. You can assign an element an z-index property which will give a priorities for which element should be in front.
    - The z-index can range from 0-10, 10 being high priority.
  + Floating Element: takes the element out of normal flow and positions it in the container. Other content can flow around it.
    - When using this property, you should use the width and height properties as well
    - This is commonly used for quotes or images that are inline with text
    - The clear property allows you to limit an element to not touch any other elements within the same container on any particular sides
      * Clear: left; means that an element cannot touch any other element on its left side. It will only touch the container on the left.
      * Clear: right;
      * Clear: both; neither left or right sides will touch other elements
      * Clear: none;
    - If a container contains only floats, some browsers may render the border oddly. To solve this problem, wrap the contents in a div and set the overflow: auto and width: 100%.
* Creating multi-column layouts with floats
  + First wrap the columns in separate <div>
  + Assign a width to each div
  + Set a float position for each div, which will position them beside each other
  + Assign a margin for each to div to create a gap
* The battle of resolutions:
  + Fixed width layout: the entire page is controlled with a set width, uses pixel
    - Set the body width then set all other element’s width, margin and padding according to the body width
  + Liquid layout: the layout changes according to the user’s browser window, uses %’s
    - To ensure proper displays, min and max values are used
* Style sheets can import rule from other sheets using @import url(“tables.css”);
* An HTML file can link to multiple style sheets. The last link takes precedence.

Images

* Control the size of an image using the width and height properties
* Most sites use a consistent image size, so you can set a few image rules and then apply the class accordingly
* Rather than using an images align property, it is common to use the float property.
  + You can create separate rules with float properties and margins and assign them to images along with the size rules
* To centre an image, make it a block-level element and use one of two techniques:
  + Text-align: center;
  + Margin: auto;
* The background-image property allows you to place an image behind any element
  + By default the image will repeat to fill the box
  + You can find a tile piece online that will repeat nicely, referred to as wallpaper
  + Background-repeat can have four values
    - Repeat
    - Repeat-x: repeat horizontally
    - Repeat-y: repeat vertically
    - No-repeat
  + Background-attachment can be used to specify if an image should be fixed or move as the user scrolls up and down the page
    - Fixed: image will stay in positions
    - Scroll: the image will move up and down as the user scrolls
  + Background-position will default to center
    - Left top, left center, left bottom
    - Center top, center center, center bottom,
    - Right top, right center, right bottom
  + You can summarize the background properties into one using “background”. Specify the properties in this order, you can skip some if you want:
    - Background-color
    - Background-image
    - Background-repeat
    - Background-attachment
    - Background-position
* Image rollover is when a background image moves positions on different mouse actions. This allows for a developer to have one image for a button that itself contains 3 images. You are only showing part of the image for inactive, hover and clicked.

HTML5 Layout

* Traditionally, everything was grouped into div elements. HTML5 introduces new elements such as header, nav, article, footer and aside.
  + These new elements help author’s describe their page better within the code
  + These features also help screen readers skip unnecessary content
* <header> and <footer> are at the top and bottom of a page. They can also appear at the top and bottom of an article.
* The <nav> element contains the page’s navigation tools.
* An <article> element acts as a container for a standalone section of a page
* <aside> can be used for related by unessential information
* <section> elements help to group content together, they typically have their own header and articles
* <hgroup> can be used to group headings
* Older browsers do now know how to treat these new elements so add a CSS rule for them:

**header, section, footer, aside, nav, article, figure**

**{**

**display: block;}**

Process & Design

* Who is the site for?
  + Invent fictional characters for your website that your target audience would be like. When you run into a design question, ask yourself what your fictional characters would want.
* Why people visit your website?
  + What are their motivations: entertainment?
  + Do they have a goal: buy, contact?
* What are your visitors trying to achieve?
  + Make key tasks easy to achieve
* What information your visitors need?
  + Needs to be quick and efficient to gather info
* How often will people visit?
  + Some sites require constant updates while others don’t