Computer <-(html, css, java, images)-> website <-> app server – database, apps, cloud

Lots of developing apps is writing html code

HTML – hypertext markup language

Tags – define blocks of content

Tags open <#> and close </#> and have attributes

<h1> largest heading

<h2> 2nd largest heading (down to h6)

<p> defines a paragraph

<br> single breakline (no close needed)

<b> bold text

<i> alternative voice/mode (italics)

<a> defines a hyperlink

<a href = <http://...>> Schools </a>

<img> defines an image

<img src = “smile.gif” alt = “Smile” height =”2” width = “2”>

<table> defines a table

<tr> defines a row in a table

<td> defines a cell in the table

<ol> ordered list (numbers the list. Can start the number. Else, default = 1)

<ul> unordered list (bullets)

<li> defines a list of items

New to HTML5: header, footer, main, section, article (all have open and close tags)

<header> defines the header of a section

<footer> defines the footer of a section

<main> specifies main content of a document

<section> defines a section of the document. A thematic grouping of content (with a heading usually)

<article> defines an article of content. An independent, self-contained content

Close tags are needed if it is not an empty element (like br)

HTML attributes: lang (<html lang=en-US>…</html>), title, href, src, width, height, alt

<table>, <tr>, <td> work together to define a table. Inside the <table> tag, you have <tr> with many <td> in it for each column. Each <td> has content in it before </td> and all <tr> have same amount of <td> in them to make a nice table

Header  
nav  
section article  
article  
footer

Can create links to other pages with <a> tag

<nav> defines navigation links usually with links <a> inside of it

<output> defines the result of a calculation

<figure> <img…> </figure>

<figcaption> around <figure> </figure> </figcaption> if you want a word caption around your figure

<main> for the main contents of an article

<mark> highlights

“semantics” – elements where the word describes what is happening / clearly defines the content

Browsers + screen readers + terminal/command line  
 can consume websites outside of a browser -> if structured correctly then screen readers and command line read it correctly

80s – HTML  
89 – on internet  
90 – more internet

Engines under the browsers:  
 internet explorer / Edge (IE)  
 Mozilla (Firefox)  
 Webkit (Safari/Chrome…and Google is making their own too)

HTML through tags and attributes

Semantic – describes document / content

Do NOT do presentational in HTML (how it should look) because can do that in CSS

Content – separation of concerns  
 “Account data, not what it should look like”

Design – CSS / Content – HTML

DOM (Document Object Model):  
 Cross-platform and language independent convention for representing interaction  
 -\*tree structure of our html -> describes the layers of html in a linear fashion because so much is nested -> what css looks at -> css rules cascade through the DOM and finds matches and does stuff to it  
 -can be through CSS or Java  
 manipulated through scripting

<html>  
 <head> describes things but not actually with our content  
 <title> what shows on the tab of the browser (title) </title>  
 </head>  
 <body>  
 content here!  
 </body>  
</html>

Right-click -> inspect : can see DOM of a website

Target = “\_blank” opens link in a new browser tab  
 attribute of anchor <a>

Inside of <ul> you need your <li>content</li> that you want to be listed within the </ul>

Block -> div

Inline -> span

Block and inline tags are similar (do we use inline (span) ever? In notes it says something about styling text but shouldn’t that be CSS?)  
 can use italics and bold in it -> it is semantic because that part is important <i> or <em>. We are not worried about what it looks like still because just saying it is something that should be emphasized  
 similar with labels -> but should just style all labels to be bold because its not emphasizing

All HTML tags have ID (unique) and class

Invalid HTML does not show error. It may run right or it may not show up (depends on the browser)

HTML file is a text file (when open in a browser, text keeps it formatting (spaces etc), but html looses it (does what it is commanded to do))

Comments: <!-- and -->

Src = “/images/…”will look in root of hard drive or webserver  
 “images/…” looks in the current directory (of that document)  
 “image.jpg” if the image is in that same directory  
 ../image… goes up a directory  
 ../../image… goes up two directories  
 http:// links directly to image on another (or your) website

Default -> text starts at bottom of image and keeps wrapping

Alt = “words” are the words that show up if the device can’t render the image. Make it descriptive. Important for screen readers

Title=”words” what is say when you hover over the picture/link/etc

Boarder -> do this is css

Width can be in pixels or % like width=”100” or width=”100%”

CSS:  
body {…}  
h1{…}  
p{…}

CSS: put ; after each argument

3 ways to insert a style sheet:  
external style sheet: <head> <link rel=”stylesheet” type=”txt/css” href=”mystyle.css”></head>  
internal style sheet: just put the styles directly into the <head><style>body{…}</style></head>  
inline style: for adding a style attribute to a single element: <h1 style=”color:blue; margin: 30px;”>

head</h1>

if there is a discrepancy, use whichever style was defined last (further down on the list/closer to the text)

cascading order: inline use 1st, then external/internal, then browser default

Syntax: CSS: selector {declaration block with property:value  
 }  
 ex) h1 (selector) {color(property):blue(value); font-size:12px}

Declarations always end with a semi-colon

Property name colon value

Declaration block surrounded by {}

CSS selectors: used to find/select HTML elements based on element name, ID, class, attribute, and more

Element selector: selects based on element name ex) p{…} for all <p> elements on a page  
ID selector: uses ID attribute of a HTML element which should unique within a page (so it uniquely selects 1 element)  
 #elementID  
 ex) #para1{text-align:center;}  
 ID name can NOT start with a number  
Class selector: selects elements with a specific class attribute  
 .class  
 ex) all elements with class=”center” will be right and centered -> .center{text-align:center;  
 color:red;}

And HTML says <h1 class=”center”> Heading </h1> and can only do specific HTML elements affected by a certain class with p.center{}

<p class= “center large”> text <p> refers to 2 classes! Center AND Large

Grouping selectors: if all have the same style definitions can do:  
 h1,h2,p{text-align:center}

CSS comments (single or multiline) /\* comment \*/

Box model -> for design/layout; box around every HTML element

Inside to outside: content (text/images)->padding (transparent area around content) -> boarder (boarder that goes around content and padding) -> margin (transparent area around boarder)

Ex) div{  
background-color:lightgrey;  
width:300px;  
boarder:25px solid green;  
padding:25px;  
margin:25px;  
}

Width/height sets sizes for CONTENT (not the full size of the box)

<!DOCTYPE html> to begin of document to fix padding/boarder issue in IE </html>

Can make list of items inline with CSS (rather than in a block), but you should still write them as a list

CSS applies formats and styles to DOMs (Document Object Models)

Separate content and design:  
more flexibility in changing design  
allow for table-free layouts  
more accessible

Cascade: read top down for styles

P#intro (paragraph with ID of intro)  
p#intro.class (paragraph with ID of intro and a class of class)

The more specific, the more weight it has (overrides less specific formatting)

Style attribute inline is bad -> in html and overrides everything  
<style> tag at top of page: for all in that document, but just page specific  
link to a style sheet\*\*: should do this all the time; all style in its own CSS page

ID overrides class

Class: when you have multiples of one element use this (a set of things)  
class-> “type of”; like for a specific type of paragraph, etc

<div id=”primary-block”>…</div>  
<div class=”news-article”>…</div>

<style> says CSS elements will follow -> correct way to do it: in the <head> part connect to a file with all the CSS in it

[selector]{[declaration]:[value];}

White space does not matter

Common declarations: float, width, text-align, boarder, background-color, font-family, font-style, font-size, margins, padding, display (block/inline(/none (hides the element)))

List items default to block

Jquery -> can frade out capacity 100-0 then display=none

Colors: by name, hexadecimal code #CC0000; #000066

Style by class: .my-style{…}  
Style by ID: #myIDS{…}  
Style by tag: h1{…}

Often lots of standard styles in body tag

ID->class->tag (what overrides what (ID is most powerful))

Can do nested styles: div.header p{} does all paragraphs within a div that has a header class

.header>h1 does for a h1 DIRECTLY within header class (children)

Ex) div.section #primary ul >li

Width with px, %, em, rem  
try to avoid px because of different font sizes, em is standard font size, rem is relative (based on the class size already has and adjust it accordingly)

--------------------------------------------------

~~SASS-> syntactically awesome style sheet  
preprocessing, variables, nesting, partials, mixins~~

Img#profile{float:left;margin-right:10px;boarder-radius:12px (rounds corners);}  
NO space between img#profile or else it will look for ID of profile inside of image, NOT an image tag with an ID of profile

Order does not matter with .callout and p.callout because you are doing all .callout and then specifically doing p.callout for the ps  
goes most specific overrides least specific

Float specifies whether or not an element should float

Clear: used to control the behavior of floating elements

Float can wrap text around images

~~Clear: to avoid elements after a floating element that by default float around it~~

Overflow:auto; fixes if element is taller than element containing it will overflow outside its container problem

Block starts on a new line

Grouping tags: div and span

Span example: <h1> My <span style=”color:red> red</span>text</h1>

Div default: line break before and after, but this can be changed in CSS

Global attributes: **id**, style, title**, class**? Every single element has ID and Class -> why its important for css

Span groups inline elements in a document; does no usual change on its own; provides a way to add a hook to part of the text/document; can style with CSS or javascript; supports global and event attributes; no default css

<meta> inside the head  
 <meta charset=”UTF-8”>   
 uniform transformation format 8 bit-blocks represent a character  
 a character encoding declaration  
 this UTF-8 is default for HTML5

~~New HTML APIs  
 HTML geolocation  
 HTML drag and drop, local storage, application cache, web workers, SSE~~

~~91-94- HTML versions 1-4  
2000 – HTML made syntax short  
04-06 – WHATWG (web hypertext application technology working group) used  
 helps to standardize it (can add functionality but cannot remove functionality)  
W3C (worldwide web consortium) developed definitive HTML5, HTMLX~~

Semantic elements = elements with meaning (form, table, article)  
non-semantic = div, span

Section – thematic grouping, typically with a heading

Article – independently self-contained content (makes sense on its own)

Can nest sections and articles as you wish

Can have several header/footer elements in 1 document

Nav – set of navigation links (block of links)

Figure groups image and figcaption

figcapture add visual explanation to the image

new semantic elements help html5 produce greater consistency

div is a block until apply it to float elements, and then it is inline  
 float added to any block element automatically turns the behavior into an inline element -> why sometimes have to add width to divs ~~(css display: block for making inline into a block)~~

responsive with twitter bootstrap, foundation (by zurb) and this changes size with changing screens

there is default margin/padding which becomes a problem when trying to float things so do:  
margin: 0px; (for all)  
margin: 0px 0px 10px 10px; (top, left, bottom, right)  
margin-left: etc..

padding:4px; mozilla and edge work with width 70% and 30% but IE does not because 100% and 8px

padding: 10px 5px; (top/bottom, left/right)

{float: left; width: 100%; margin: 0px;}

%s are good on smaller size monitors, but not good on phones -> use responsive for that!

Files: bio UP pages OVER styles IN style:  
from bio do: ../styles/style.css

<form>   
words  
<input type= “text” name=”first name”><br>  
words  
<input type=”text” name=”last name”><br>  
</form>

Form itself is not visible  
default text field = 20 characters

<input type=”radio” name=”gender” value=”male” checked> Male <br>  
<input type=”radio” name=”gender” value=”female”> Female <br>

Submit button -> submitting data to a form handler  
<form action=”action\_page.pnp”>  
 <input type=”submit” value=”Submit”>  
</form>

Action attribute -> defines action to be performed when it is submitted  
in example above, form is sent on the served called “action\_page.pnp”  
page contains a server side script that handle the form data

<form action…> can also have attribute method = “get” or “post”  
 get puts stuff in a url and puts it in another page  
 post puts stuff in the header and puts it in another page  
 can still see it in the header when you do right click, inspect

---------------------------------------  
accessibility issues: screen readers, etc -> if bad then you can make sense of it even if pictures don’t show up, etc ---🡪 need this for labels for forms

div.row.indented {} -> <div class=”row indented”>

feildset and legend just for organizing forms -> lengend and feildset wraps around the next things  
 good because descriptive and semantic and then can style it to look however you want it to look  
 fieldsets do not have data, they are just an organizational tool

2 ways of using labels

Head is not what user is actually consuming

<DOCTYPE! html>  
<head></head>  
<body></body>  
</html>

Input tag’s name never shows up but it is just where the data you submit is attached to

IDs override classes

Label for connects to the id for input -> have to do it if they are not next to each other and you can’t just wrap the label around it   
NOT the name, the ID (that’s what the for points to)

Value is default  
what you enter is put in the value

Textarea has rows and columns  
has no value  
how to set the text area with no value?

Input type text -> small box

Textarea -> big box (can have paragraphs/hit enter)

Textarea is an open/close tag and the value is what is inside of there

Type submit creates a button and has specific action -> submits the form

In HW, name form elements

Get -> when you want to get something from the server [puts stuff in url; can bookmark this; easier to see (security problem) and gets written to log]  
post -> when you want to put something to the server (like updating profile) [hides stuff from url; cannot bookmark this]  
--------------------------------------------

Get is the default, but the submitted form data is visible in the page address field   
 do NOT use sensitive info with get

Post has no size limitations and can handle sensitive data

Name attribute -> necessary for an input field to be submitted

Grouping form data with <fieldset>  
<fieldset> used to group related data in a from  
<legend> defines a caption for fieldset

<form action=”action\_page.php”>  
 <fieldset>  
 <legend> Personal info:<legend>  
 FN: <br>  
 <input type=”text” name=”firstname” value=”Mickey”> <br>  
 LN: <br>  
 <input type=”text” name=”lastname” value=”Mouse”><br>  
 <input type=”submit” value=”Submit”>  
 </fieldset>  
</form>

Forms let you get info into a website  
forms are html -> same rules apply  
forms: semantic markup, tags and attributes, style with CSS  
forms = interactivity  
forms can only handle part of it  
html =/= process forms   
 need back-end technology to receive the info  
with forms, you have a lot more security risk  
with interactivity, protect against malicious users; validate

Form tags:  
input  
select/option  
textarea  
button  
label  
feildset/legend

Input -> need type attribute:  
textfield words  
password letters but little dots hiding them  
hidden send more info than you offer on the form (this is hidden from the user)  
checkbox can select one or more (square boxes)  
radiobuttons can only select 1 if html is written correctly

Back-end just sees all of the input types as input

When you submit, send the name field and whatever you enter

Value -> default value. Blank by default

If you don’t check the check box “remember me” then remember\_me I s just not included. If the processing page shows remember\_me then it exists and was checked

Select can have multiple argument  
<select name=”country\_id” multiple=”true”>  
 <option value=””>Please Select</option>  
 <option value=”IN”>Indiana</option>  
 <option value=”NC”>North Carolina</option>  
</select>  
in this example, value is an attribute of option, but the words between the open and close tags are what you see in the form

For a default have <option selected=”true”value=”MI”>Michigan</option>

When multiple=true when default multiple is selected, both are just selected  
when multiple =/= true it should be the first one it finds it selects and ignores the rest

Textarea: an open-close tag  
a bigger box of text

<textarea name=”bio” cols=”50” rows=”5”>  
This is the default bio  
</text area>

White space -> one element of it is used so would have space before and after stuff and ignores the other spaces

HTML5 with forms:  
with input tag, added many new types  
ex) number for # keyboard to show up  
if HTML5 tag is not supported, then it just reverts back to text field so there is no harm putting it in there  
same idea with:  
input type button   
input type submit  
 button type button does not work/do anything if HTML5 is not supported

\*use labels to connect the label with the input tag  
wrap in label to connect it all  
<label>UN:  
 <input type=”text” name=”Username”>  
</label>

If label and input, etc are not next to each other then pass in ID of input element  
<label for=”users”>Username:</label>  
<input type=”text” id=”user” name=”username”>

With label in it, if you click it, it checks the box or puts the cursor in the text box -> good for phones!

Fieldset and legend  
<fieldset>  
 <legend>Login form:</legend>  
 <input type=”text” name=”Username”>  
 <input type=”password” name=”password”>  
 <input type=”submit” value=”Login”>  
</fieldset>

Fieldsets and legends all have ID and class -> can refer to them that way!

<form action=”submit.php” method=”post”>  
…  
</form>

Action -> where you want the from to go. Like href in the <a> tag -> it’s a filepath

Method -> “post” or “get”  
get is the default  
have “?” in the address now

Post->URL goes wherever it should. Don’t see any input.  
it puts the info in the file headers  
it is in the request, but not in the webaddress or log  
better for preventing getting info from hacking

If have multiple of dropdown selected, its okay, it separates the values by a coma and sends the info over

\*with radio box: need name all the same!

\*label tag =/= around all inputs  
each label tag must be for 1 input tag  
name attribute must be the same

If you do not check the check box, then the element is not sent in the form

You can have many forms on one page

Each form has its own method of submit and submits just that form when you push the button

Form without action posts to itself  
Form without method defaults to get

When submit, page refreshes and the default returns

Password gets posted with get -> value is just posted in the URL (with get)

https <-give layer of security on the way to the server

with post -> just sends it in the header part

response -> device request a page. Page responds to the device/screen

media queries – new in CSS but lets apps adjust to web page

CSS makes list items from a footer in line

Break point -> where you change the layout (text wrap around picture vs just go under it)  
 standard to have 3 - large monitor, standard/tablet, phone

Mobile first -> start with mobile size and then work out when you make it responsive. Phone should be primary site. Helps with loading pictures, etc and not using all your data

With responsive website, start with <meta name=”viewpoint”, initial-scale=”1”> in the header (forces 100% and then applies styles accordingly)

At the end of the style sheet, media query style -> @media or @1000px, then add that specific style to it

Just put media query at the bottom of CSS

Media queries dictate breakpoints

Can do max-width or min-width

Always do normal CSS before media queries

Can use twitter bootstrap, foundation (by zurb) and materialized to make these responsive sites. Use a grid with 12 boxes and at each breakpoint can say how many boxes you want each element to take up

Foundation-kitchen sink shows you all the things you can do

Jquery – allows for freedom/flexibility. Uses CSS selectors if something happens to an element on the page, do something

Add jquery link in the html head (can connect to the google link because using CDN – content delivery network, so don’t have to go all the way to the server to get stuff people around you have already loaded locally)

Script tag says another language is coming in the html head

Better to link to java script than to write it all inline -> similar to CSS in this way  
Can do functions that tell certain elements to fade out etc in javascript

Parent is what is directly above it (or can do closest in javascript)

<script type= “text/javascript” src=main.js>

With javascript can refresh just one part of an app (like the stock ticker)

<form> - defines a form used to collect user input; contains form elements

<input> - most important element -> displayed depending on type (text, radio, submit)

<textarea> - <textarea name=”comment” form”form\_id” rows=”10” cols=”10”>text here!</textarea>; an open-close tag

<select> - <select name=”carlist” form=”carform”>  
 <option>…</option>  
 <option>…</option>  
 </select>  
 the open/close tag needed for a drop down that comes out from the form  
can also have the multiple=”true” inside of there if you want that to be true

<option> - where you put your dropdown list items; open/close tag;   
 <option value=”volvo> Volvo </option>  
 “volvo" is what is sent if selected;  
 Volvo is what is displayed  
default is the 1st item selected and you can say selected inside the <>

<label> - provides usability improvement because if click label goes to input box  
 <label>title  
 <input….>  
 </label>  
OR, if they are not next to each other, then use the input’s ID to reference it  
 <label for=”input\_id”>Label</label>  
 <input type=”text” name=”firstname” id=”input\_id”>

<fieldset> - groups together related data in a form

<legend> - defines a caption for the fieldset element

<input type=”text” name=”firstname”> - width = 20 characters – a 1-line input field

<input type=”radio” name=”gender” value=”male” checked> Male  
<input type=”radio” name=”gender” value=”female”> Female  
can select 1 of the choices only  
the name must be the same!

<input type=”submit” value=”Submit”> (value is what shows up on the button)   
 submits data to a form-handler (server page with a script for processing input data – specified in the form’s action attribute (<form action=”action\_page.pnp” id=”userform”>) -> this webage on the server contains the server-side script that handles form data)

Form has a method attribute -> method=”get” and method=”post”

Get is visible from webpage address field and has a size limitation

Name attribute – needed for data in input field to be sent at all

<button type=”button onlick=”alert(‘Hello World!’)”> Click me!</button>

Input with text, radio, submit  
input tag has name, type(, id, value)  
type determines how input tag is displayed  
select list has <select name=”X” form=”formitbelongsto”>  
 <option value=”Y”>Y</option>  
 </select>  
labels are good because connect so if click on label, can enter data and also connects with text readers  
label directly around input OR use for=”inputid” to reference an input that’s not right there  
new HTML5 input types=>color, date, month, **number**, time, url, week, range, datetime-local, email, search, tel  
 these are good because if HTML5 not supported, then it doesn’t change anything and reverts back to text, but if it is, it gives you a keypad, etc.