IST 263

What is the difference between the internet and the web?

* The web is part of the internet
* World wide web, FTP, E-mail, Telnet all part of the internet

What decade was the internet conceived? 1962. Licklider of MIT proposed the idea of a network of computer that could talk to eah other

First message sent on the ARPAnet?

* 1969 first message sent
* First message was sent from UCLA to Stanford: “LOGIN” – it crashed the fledging ARPA network anyway

What decade was email invented? 1971 Ray Tomlinson.

* Came up with the @ symbol to tell the difference between person and location in an email address

Modern technology that united all computers on early networks: TCP/IP (we still use it)

* 1971 Vincent Cerf. Unifies lots of mini networks like ARPAnet

Throughout the 1980s, researchers and scientists used it to send files and data (no images)

When was the web invented? 1991 Tim Berners-Lee (working at CERN)

* Helped solve the difficulties scientists were having sharing info
* Created HTML
* Created URLs
* Developed HTTP the protocol
* Director of the world wide web consortium, which oversees the continued development of the web

The first popular web browser was called: Mosaic (first to show pictures) 1993

* In 1993 congress decided that the web could be used for commercial uses
* The first thing ever purchased on the internet was pizza

How does the web work? When u type a URL your IP address is also sent out so you receive the correct website.

DNS: domain name server

URL: uniform resource locator

Protocol: https: the s means your site is secure, your credit card info will be encrypted

top-level domain: .com,.net, etc.

sub-domain:

port:

**Unit 02: Lecture Video – HTML Tags**

Tags and elements are synonyms.

Tags:

<strong></strong> … bold

<h1></h1> …heading

<p></p> … paragraph

<br>…line break

Block vs. Inline elements:

Block tags create a space in between tags (heading, subheading, paragraph, div tag..etc.)

Inline tags don’t create a space (bold)

Void elements – tags that have no closing tag

<br>… line tag has no closing tag

<hr> … horizontal line

Image tag is also a void tag

Character Escapes:

Freeformatter.com 🡨 where you can find code for symbols such as the copyright symbol

Lists:

<ul>

<li> first item </li>

<li> second item </li> 🡨 bulleted list

<li> third item </li>

</ul>

<ol>

<li> first item </li>

<li> second item </li> 🡨 numbered list

<li> third item </li>

</ol>

Browser Caching – not seeing the most current version of a logo or picture.

**Unit 03: Lecture Video – links, images, and video**

Copyright and intellectual property

HTML Attributes and links:

* All HTML elements can have attributes
* Attributes provide additional information about elements
* Attributes are always specified in the start tag
* Attributes usually come in name/value pairs like: name=“value”

Link Paths:

* External link – full url including http
* Relative link – refer to page based on the current page’s relative location
* Site root link - / specifies root site directory – specify path from root

Note: paths to files and paths to images work the same way

Images:

* Web image formats: PNG, JPEG, GIF
* Element – img is a void element
* src attribute specifies image location
* alt attribute describes image as text

\*A void tag doesn’t have a closing tag\*

Embedding:

Embedding with the <iframe> tag:

* an inline frame is used to embed another document within the current HTML document
* title attribute describes the content of the iframe and should always be used

Embedding a youtube video:

* find your video
* click share and choose embed
* copy and paste the HTML code that pops up between <body> tags

**Unit 04: Lecture Video – Tables and Forms**

Tables are used to display a lot of data in an organized manner.

Element: Purpose:

* <table> </table> marks start and end of the table
* <tr> </tr> one row in a table
* <th> </th> header in a table
* <td> </td> one cell in a table

Form Element:

<form action=””method””>

</form>

* Tells you where a form begins and ends
* Will contain elements with form control elements (checkboxes, dropdowns text boxes, etc.)
* Defines how form is processed and what method is used to send data
* You can’t overlap form tags so once you start and element you must close it before another form is started
* The action part of the tag tells you where the form is going to go once it is submitted
* The method part of the tag tells you how to information is going to be sent to the action page

Form attribute method:

* What is “get”
  + Sends the form data across the URL
  + Used for searches
* What is “post”
  + Sends the form data on a special header with the http request
  + Most commonly used

Text box and label elements:

<label for=”firstname”>Enter first name: </label>

<input type=”text” id=”firstname” name=”firstname”>

The input element is a void tag

The id attribute in the input element determines the label that the tag is associated with.

Radio Button: The for, type and id attributes change depending on the value. The name attribute stays the same for all values.

Checkboxes: have different name values, the different questions for check boxes don’t affect each other.

Textarea:

* Attributes
  + Id
  + Name
  + Rows
  + Columns
* Textarea is **NOT** a void tag
* Creates a textbox on a form

Select tag:

* Creates a dropdown list with options

Form selects can allow the user to select more than one value in the drop down.

**Unit 05: Lecture Video – photoshop**

The formats:

* PNGs allow transparency
* PNG-24 millions of colors
* PNG-8 only 256 colors ..used for cartoons, drawings, anything that’s not a photo
* JPEGs do not allow transparency

Transparency – putting a logo on a shirt without the square background

Pixel – one dot of an image

* There are 72 pixels per inch on a screen
* The number of pixels per inch is abbreviated PPI

Pixelated – an image does not have enough pixels to be scaled larger

Dimensions – width and height

**Unit 06: Lecture Video – wireframes, CSS**

Wireframe - allows us to design basic layout of a website without thinking about design. They help s focus on user experience

The primary purpose of wireframe is layout.

CSS – standard for defining presentation of HTML documents

Selector {property: value;}

|

Declaration

The style attribute means you have an inline style

Embedded – style tag inside the head tag

External – you need link void tag, rel=”stylesheet”, relative link to location of stylesheet

Most common styles to use are either embedded or external styles

It is frowned upon to use inline style sheets

Units of measure:

Absolute vs. Relative

Absolute – have predefined meanings or real-world equivalents. They are NOT appropriate for webpages

Relative – based on the size of something else, such as the default text size

Units of measure that are relative:

* em
* cx
* ch
* rem
* vw
* vh
* vmin
* vmax
* %

Inheritance and the DOM (document object model)

Body tag is a parent of the p tag.

When styling body tag everything inside of it will follow those same styles, except links which are styled differently.

Fonts load off user’s computer, if the font isn’t on their machine it won’t load

Line height controls space between lines

Selectors:

Group Selectors

Element selector p { color: navy; }

Group selectors p, ul, td, th {color: navy; }

Descendent selectors – style specific tags not all of them. For example, a specific a tag, not all of the a tags in the file.

p a {color: orange}

id selectors

#myheading {color: pink;}

Class selectors

.myclass {color: grey;}

**Unit 07: Lecture Video – Site maps and CSS colors, background and box model**

Site map – a listing of all of the pages that are going to be on your website. They are hierarchical

Why build a site map?

* Organize your content
* Having clear organization helps users find content
* Need for SEO (search engine optimization)

CSS Colors

Hex colors – hexadecimal RGB values must be preceded by # symbol. Hex colors are 6 digits 0-9 or a-f for each digit

Pseudo Selectors – applies a style to a state of an element

* Changes the color of a link in different states, before you visited, after, hovering
* You need the colon after the element and then state name
* Examples:
  + a:link
  + a:visited
  + a:hover

CSS Background colors

* styling the body tag means the entire browser window content area will be affected

CSS Box Model

* margin – area outside the box
* padding – space immediately around the content
* boarder
* content area
* width of the box + the margin x 2 + border x 2 + padding x 2
* inline elements box will not create space height wise

**Unit 08: Lecture Video – CSS Layout**

Semantic containers – help get your website to appear first when searched on google. Helps disabled people if they’re using a screen reader.

* <article>
* <aside>
* <details>
* <figcaptain>
* <figure>
* <footer>
* <header>
* <main>
* <mark>
* <nav>
* <section>
* <summary>
* <time>

Non-semantic container elements – used to style things

* <div> ….block element
* <span> …..inline element

Floating inline and block elements:

* The floating starts where the element occurs in the document.

Positioning

* Relative – maintains the flow of a normal document, position according to top and left from there
* Absolute – takes element out of document flow and positions according to container
* Fixed – takes the element out of the document flow and positions according to viewport

Flexbox:

* Layout things in a grid like format
* They don’t have the normal document flow
* You can use percentages or ratios to size the elements

**Unit 09: Lecture Video – Responsive design**

Responsive web design (RWD): making your site look good on desktops, tablets and mobile devices

Flexible grids

What will we do to size our sites?

* Media queries – allow us to give CSS rules strictly by width of screen
* Media query
  + Different styles based on:
    - Width of browser
    - Orientation of browser
    - Screen resolution
    - Whether you are printing or view on screen
* Common Breakpoints:
  + Extra small devices (phones, 600px and down)
    - @media only screen and (max-width: 600px) {…}
  + Small devices (portrait tablets and large phones, 600px and up)
    - @media only screen and (min-width: 600px) {…}
  + Medium devices (landscape tablets, 768px and up)
    - @media only screen and (min-width: 768px) {…}
  + Large devices (laptops/desktops, 992px and up)
    - @media only screen and (min-width: 992px) {…}
  + Extra large devices (large laptops and desktops, 1200px and up)
    - @media only screen and (min-width: 1200px) {…}

Viewport:

* The browser’s viewport is the area of the window in which web content can be seen. This is often not the same size as the rendered page, in which case the browser provides scrollbars
* If you don’t control the viewport, your site will shrink to fit the device size. That’s when you see a teeny tiny webpage shrunken down to fit your phone screen.

Viewport

This belongs in the head tag

<meta name = “viewport” content= “width=device-width, initial-scale=1”>

Flexible Image sizes:

Img { max-width: 100% }

Flexbox wrapping

.flex-container { flex-wrap: wrap; }

**Unit 10: Lecture Video – Accessibility**

Making your site look good on:

* Tab order
* Alt tags video subtitles
* Semantic tags
* Color contrast
* Resizable text

How to make site accessible:

* Semantic tags
  + Make sure you are using the semantic tags properly. People with screen readers are going to hear the names of your semantic elements
  + For example – don’t use an article tag for an element that is not an article
  + Make sure you use header, footer, main, etc. semantic elements
* Headings
  + Respect the H1-H6 tags
  + You should only have 1 H1 tag on the page as the main heading
  + Use the other heading tags in order as subheadings (h2), sub-subheadings (h3), etc.
  + Headings are hierarchical
    - H1 should only appear once on the page, h2 is a subheading
* Alt attribute for img element
  + Describe your image
  + Don’t say image of sunset, just say sunset
  + If the image is purely decorative, not content, use alt tag but leave it blank – screen readers skip over this
* Color contrast
  + This is for people who are color blind and can also help older people and people with visual impairments
* Aria roles
  + For dynamic content for screen readers
* Tab index
  + Manually order the elements on your page
  + The tab index attribute allows us to change the order the user gets when tabbing through links
* Skip to content button
  + This will link you down to the content a tag with the id of skip-to-content
  + In css left: -999px – positions skip to content off screen
* Video captions
  + Help with SEO because of searchable text

**Unit 11: Lecture Video – Intro to JavaScript**

What is JavaScript – scripting or programming language that allows you to implement complex features on webpages

* Just like CSS and HTML, JavaScript is interpreted by your browser
* Most popular programming language in the world

What is JavaScript used for?

* Auto typing on google. When you’re typing and google is giving suggestions of what you might be searching for.
* Validation on forms
* Dropdown navigation
* Accordion navigation on phone aka hamburger navigation
* Show or hide info with click of more button
* Slideshow

How do you use it?

* Types of JavaScript
  + Embedded JavaScript (can be anywhere in HTML) src attribute tells you where the file is
    - <script>

Java code goes here

</script

* External Java Script
  + <script src=”my\_script.js”></script>

How do you make a JavaScript file?

* You only need to make a JavaScript file for external JavaScript
* It is a text file just like .html and .css
* Use .js as the file extension

1st JavaScript Program

* Pointers
  + JavaScript is case sensitive
  + Comments
    - // single line
    - /\* multiple line\*/
  + Example of a command in JavaScript
    - Alert(‘Hello World’);
      * First we wrote our command
      * The info the command takes is sent in parenthesis
      * Since we’re sending a string it must be in quotes, single or double quotes are fine
      * End of command gets a semicolon

Event Handling

* What are events?
  + Actions that happen in the browser like:
    - The page loading
    - A user clicking on something

Event handlers and their descriptions:

* onlbur
  + An element loses focus
* onchange
  + The element of a form field changes
* onclick
  + the mouse clicks an object
* onerror
  + an error occurs when the document or an image loads
* onfocus
  + an element gets in focus
* onkeydown
  + a key on the keyboard is pressed
* onkeypress
  + a key on the keyboard is pressed or held
* onkeyup
  + a key on the keyboard is released
* onload
  + a page or an image is finished loading
* onmousedown
  + a button on the mouse is pressed
* onmousemove
  + the mouse is moved
* onmouseout
  + the mouse is moved off of the element
* onmouseover
  + the mouse is moved over an element
* onmouseup
  + the mouse button is released
* onsubmit
  + the submit button is clicked on a form

Functions:

* A JavaScript function is a block of JavaScript code, that can be executed when “called for”
* Named code needs curly brackets around it
* Our named code must start with the word function

**Unit 12: Lecture Video –JavaScript DOM and Nodes**

DOM – document object model

What is a node?

* Each element on this page is referred to as a node
* The attributes inside the tags are nodes
* Content inside the elements are nodes
* We can use these nodes to change things and make our pages dynamic
* HTML elements are JavaScript nodes

Accessing DOM nodes:

* Document
  + In JavaScript, document defines the webpage. When you are identifying an object on a page you start with document. After that you can chain together different methods in JavaScript to select nodes

DOM node locations:

document.getElementById(“messages”)

* Document refers to whole web page
* Getelementbyid says to look at the id attributes in the tag
* (“messages”) is the value in the id attribute

Manipulating Nodes

innerHTML

Put your DOM node together with innerHTML:

* document.getElementById(“messages”).innerhtml=”<p>Hello World</p>”;
  + innerHTML refers to what we are changing
  + =”<p>Hello World</p>”; refers to value after change

Setattribute()

* document.getElementById(“myid”).setAttribute(“src”, “image.PNG”);

style

* document.getElementById(“myid”).style.color=”#0000”;
* input type is a color picker

Hamburger Nav

HTML

1. we need to add the image for the hamburger

CSS media query

1. we only want the image to show when the screen is small
2. we need to hide the stacked navigation when the screen is small

JavaScript

1. we want to turn on the stacked navigation when the user clicks on the hamburger
2. we want to turn the navigation back off when the user clicks the hamburger again