|  |  |
| --- | --- |
| * How the web/ internet work - **What is the Internet**: can refer to the technical infrastructure (fiber optic cables), “last mile” networks, data centers, and TCP/IP. Can also refer to social life like content and services that run on top of internet infrastructure. Interconnected computers and devices that communicate worldwide   **- What is the web:** One application that runs on top of the internet like zoom, email, telnet, networked tv, ssh, gaming engines. System of interlinked documents that are accessed via browsers  **- How are they different:** the infrastructure, the web is the content we access through the internet  **- What is the cloud:** having access to someone else’s computer(s). Software, info, and computing services that run on someone else’s computer. A system of remote servers and services that store and manage data, run applications, and provide resources over the internet.  **- What is TCP/IP:** Transmission Control Protocol, enables reliable transmission of data across a network. Breaks down data into smaller packets before they’re sent, ensures packets are delivered, checks for errors. Provides flow control and congestion control. IP: Internet Protocol, handles addressing and routing of packets to ensure they reach the correct destination. Each device gets a unique address on the network, determines best route for data to travel, data does not arrive in order  **- What is a domain name:** human-readable address used to access a website on the internet. It’s the URL that you type into a browser to visit a specific website like google.com. MMakes websites easy to find and access  **- What is DNS:** Domain Name System, a way to assigning human-readable addresses to IP addresses so people don’t have to remember long sequences of numbers. Browser asks for a website @ a domain name, the browser looks in its cache for the IP, then to a DNS resolver  **- Who invented the internet:** The internet began as ARPANET, an academic research network funded by the military DARPA in 1969.1973 Vint Cerf and Bob Kahn worked on TCP/IP. 1981 funding for internet shifted to NSF for long-distance networks that served as the internet’s backbone. 1994 clinton admin pushes for internet  **- Who controls the internet:** No one runs the internet; it’s a decentralized network of networks. Companies, universities, and governments operate their own network traffic. Govts can filter or block traffic **- Where is the Internet:** all around us. Not a single physical location, it’s a global network of interconnected computers, servers, data centers, cables, and satellites across the world. **- What is Net Neutrality:** Principal that all internet traffic should be treated equally without discrimination or interference by internet service providers. ISPs cannot block, slow down or prioritize certain types of internet content or services **- What kind of data do you generate as you use the internet:** Account details: name, email, phone, location, payment info. Profile data: interests, preferences, photos, interactions.  Cookies: small data files to track browsing behavior, stores info for quick re-access  Browsing History, internet history of visited pages  Search Queries: search results are stored to help improve results and generate info about your interests  **- Who can see your data and how can they see it**: ISPs like Comcast, AT&T or Verizon. They can see sites you visit, cannot see encrypted data, but can track metadata. Websites like Facebook, google, amazon: collect data from your interactions, search or pages visited. You’re tracked by cookies and given personalized ads. Hackers and cybercriminals, Government and Law agencies | * The Web   -**Who invented the world wide web?** 1989 Tim Berners-Lee: ideas: HTML, URI, HTTP  **-What is a URL?** Uniform Resource Locator: reference to a web resource that specifies its location on a computer network. http indicates protocol used, domain name, file to be retrieved.  **-What is HTTP?** Hypertext transfer protocol, rules that servers and browsers must follow to transfer web files over the internet. Allows people to create, update, or delete resources. HTTPS adds encryption.  **-What is a web browser? What are some examples?** Software app that allows you to access and interact with content on the internet. Retrieves data from web servers and displays it in a format you can read like text, images, and videos. Ex. Are chrome, firefox, and safari  **-What can a browser do? What languages does it understand?** It sends a request to the web server and it shows the requested content on your device. It understands HTML, CSS, and JavaScript to display pages  **-What is a web server? What languages does it understand?**  A computer or device on a network that other computers can access for information, communication, or computational services. It understands protocol languages like HTTP(port 80)/HTTPS(Port443) and HTML.  **-What is a search engine? What are some examples?** A tool or software system that helps you find information on the internet. It allows users to enter queries (search terms) and then retrieves and displays a list of relevant websites, articles, videos, images, or other content based on the search criteria. Ex. Google, Bing, Yahoo, DuckDuckGo  **-How can someone “own” a domain name?**  To own a domain name, you need to:   1. Choose a name and check its availability. 2. Register it through an accredited domain registrar. 3. Provide your contact information and pay the registration fee. 4. Set up the domain for use (e.g., pointing it to a website or setting up email).   **CSS rules of thumb**: 1.Follow CSS naming conventions, 2.know the anatomy of a style block: h1{color:blue; font-size:12px} selector{declaration(property,value);3.Use external style sheets 4.make code readable 5. Avoid syntax errors, No angle brackets in style sheet 6.use comments |
| * HTML  1. - Know the HTML Rules of Thumb: [1. Avoid spaces, capital letters, and special characters when naming files](https://csci344.github.io/spring2025/resources/html-rules-of-thumb/#1-avoid-spaces-capital-letters-and-special-characters-when-naming-files)   No white space: Rename page 1.html → page\_1.html or page1.html  No capitalization; all lowercase Rename Page1.html → page1.html  No special characters but dashes and underscores are good   1. [2. Most tags have an opening tag and a closing tag](https://csci344.github.io/spring2025/resources/html-rules-of-thumb/#2-most-tags-have-an-opening-tag-and-a-closing-tag)   <h1>My title</h1>  Some tags that don’t: images <img src = “dog.png” alt=”photo ofdog” /> Line break: <br /> Stylesheet links: <link rel=”stylesheet” href=”mystyle.css”/>   1. [3. The browser ignores white space](https://csci344.github.io/spring2025/resources/html-rules-of-thumb/#3-the-browser-ignores-white-space) 2. [4. Make your code readable by indenting and using line breaks](https://csci344.github.io/spring2025/resources/html-rules-of-thumb/#4-make-your-code-readable-by-indenting-and-using-line-breaks) 3. [5. Attribute syntax](https://csci344.github.io/spring2025/resources/html-rules-of-thumb/#5-attribute-syntax)   Attributes are always followed by an equal sign and values surrounded by quotation marks. See img above   1. [6. Last in, first out (LIFO)](https://csci344.github.io/spring2025/resources/html-rules-of-thumb/#6-last-in-first-out-lifo)   Close tags in opposite order you opened them.  <p>welcome,  <strong> leonard</strong> </p>   1. [7. Use comments to help you understand your code](https://csci344.github.io/spring2025/resources/html-rules-of-thumb/#7-use-comments-to-help-you-understand-your-code) 2. [8. Links to CSS files go inside the <head> tag](https://csci344.github.io/spring2025/resources/html-rules-of-thumb/#8-links-to-css-files-go-inside-the-head-tag)   Put your style sheet in the <head> tag area   1. [9. All visible content goes inside the <body> tag](https://csci344.github.io/spring2025/resources/html-rules-of-thumb/#9-all-visible-content-goes-inside-the-body-tag) 2. [10. Use the Browser Inspector](https://csci344.github.io/spring2025/resources/html-rules-of-thumb/#10-use-the-browser-inspector)   - Be familiar with the basic HTML tags we’ve been working with, including:  - Images see img tag above  - Hyperlinks (absolute, relative, and internal)  Absolute gives full path to destination with http, used to link external site hosted on different server  Relative: a partial path to a destination relative to current documents location. Use this one if you own it, can move in and out of directory depending on location  - Paragraph and header tags (**<h1>...<h6>**)  - Embedding iFrames: <iframe src="URL" width="width" height="height"></iframe>  - Semantic tags (**<header>, <nav>, <section>, <article>, <footer>, <aside>**)   * + Ordered and unordered lists (**<ul>, <ol>**)   -Know how to make a comment (<!-- your comment here -->)  -Know how to link to a style sheet: <link rel="stylesheet" href="styles.css">  -Know what goes in the **<head></head>** section and what goes in the **<body></body>**.: head gets links like style sheet and body gets images and media  -Know how to make an HTML file and a CSS file from scratch using VS Code (like you did in Tutorial 3). | * CSS   -Know how to use the following 5 selectors: id, class, element, descendent, direct child.  - id example: #myElement { … }  - class example: **.**myClass { … } in html: h1,p,div, section,span,header: <p class=”name”>text</p>  - element example: h1 { … },<p>,<a> for hyperlinks  - descendent example: nav a { … }: targets any element inside the parent element .container p{color:blue}any p goes blue  - direct child example: nav > a { … } only targets element that is direct child or one level deep of parent  -Know how to set some common properties:  Box model properties (**width, height, padding, margin, border**) padding = space between content and border, creates space inside box, border separates elements from other element on page. Margin is space outside the border, creates distance between element and other elements  Color properties (**color, background-color**)  Font properties (**font-family, font-weight**, etc.) -Know how to make a comment (**/\* your comment here \*/**) HTML boiler plate code:  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <meta http-equiv="X-UA-Compatible" content="ie=edge">  <title>Your Page Title</title>  <!-- Link to External Stylesheet -->  <link rel="stylesheet" href="styles.css">  <!-- Add other meta tags as needed -->  </head>  <body>  <!-- Header Section -->  <header>  <h1>Welcome to My Website</h1>  <nav>  <ul>  <li><a href="#home">Home</a></li>  </ul>  </nav>  </header>  <!-- Main Content Section -->  <main>  <section id="home">  <h2>Home Section</h2>  <p>This is the home section of the webpage.</p>  <img src="image.jpg" alt="A beautiful image" width="600">  </main>  </body>  </html> |
| * CSS Media Queries   Allows you to apply different styles based on specific conditions like device screen or orientation, resolution, etc. media type: type of device like screen or print. Condition: max-width, min-width:700px, orientation: portrait, landscape, resolution:min-resolution: 192dpi  @media media-type and (condition) {  /\* CSS rules go here \*/  } | * Flexbox   -Know what properties and values need to be set in the **flex --container** (parent): .container{**display:flex;}** **-align-items flex-start**, flex-end, space-around, space-between, center **-justify-content** (same options as align-items) **-flex-direction row**, column, row-reverse, column-reverse **-flex-wrap** no-wrap, wrap, wrap-reverse  -Know what properties and values are typically set in the **flex item** (child)  **padding, margin, width, height, border** |
| * CSS Grid   -Know what properties and values need to be set in the **grid container** (parent) element: **display** (should be set to **grid**): .container{display:grid;} **grid-template-rows, grid-template-columns,** you can do gird-colum-start:1 g-c-end:4 = g-c:1/4 **gap:** spacing between grid items in both rows and columns  -What needs to be set in the **grid items** (child elements)? **grid-column-start grid-column-end** (with or without span keyword) | * Accessibility Principles   **- Name some of the things you need to take into consideration when designing for accessibility?:** Using semantic HTML to provide meaning/ sections to the content: <header>,<nav>,<main>, provide alternative text: helps a reader understand the content of images or icons, Sufficient color contrast: consider people with color blindness or low vision: use high contrast between text and background, provide keyboard navigation, use ARIA: makes custom/ dynamic content like buttons, widgets, insure elements are accessible like forms for readers/ cognitive impairments, readable text size/ resizable, captions for multimedia, descriptive links, clear layout, test content  **- What are things you can do in your HTML to ensure that they’re screen-reader compatible?** |