* Web Design: Open Source, Accessibility and Net Neutrality, Web Standards
* HTML, CSS, JavaScript, SVG, WOFF
* The web (or World Wide Web) is a specific system for accessing information on the Internet. It relies on the Internet's infrastructure to function and is just one of many services and applications that operate on the Internet.
* Megabits (Mb or Mbit) (megabytes is MB) measures the speed or capacity of digital data transfer such as internet connection speeds, network bandwidth, and data transmission rates.
* A router is a device that directs the flow of internet traffic by relaying data packets between computer networks. Routers are responsible for determining the best path for data to travel from the source to the destination across a network. They use routing tables and logical rules to make decisions about how to forward data packets, ensuring that they reach their intended destination. Routers play a crucial role in the functioning of the Internet and local area networks (LANs).

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

* Operating System
  + Software that manages a computer’s resources
  + Allocates resources among other programs
  + Resources include the central processing unit (CPU), computer memory, file storage, input/output (I/O) devices, and network connections
  + Runs indefinitely and terminates only when the computer is turned off
  + Example: Microsoft OS, Mac OS, Linux iOS, Android, Symbian OS
* graphical user interface (GUI) - visual; command line interface (CLI) - textual
* Unix
  + An open source OS produced by AT&T Bell Labs
  + Originally developed in 1969
  + Command line interface Portable, multi-tasking, multi-user
  + Free distribution, open system
  + Servers (including i6), workstations, mobile devices
  + Basis of Linux and MacOS
  + cd - change directory, chmod - changes permission, cp - copy, ls - list of directory, man - manual, mkdir - creates directory, pwd - print directory, rm - delete file, rmdir - remove empty directory, df -h - check space left
  + chmod:
    - r(read), w(write), x(execute), -(no permission);
    - u(user/owner), g(group), o(other/world)
  + Permissions:
    - U   G   W
    - rwx rwx rwx 777
    - rwx rwx r-x 775
    - rwx r-x r-x 755
    - rw- rw- r-- 664
    - rw- r-- r-- 644
  + Unix Commands: $ chmod 777 filename
  + Standard file permission: 644
    - Owner can read and write file;
    - group can read file;
    - others can read file
  + Standard directory permission: 755
    - Owner can read, write and execute file;
    - group can read and execute file;
    - others can read and execute file

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

* HTML - Hypertext Markup Language
  + A language for describing Web pages
  + HTML is not a programming language, it is a markup language
  + A markup language is a set of markup tags
  + HTML uses markup tags to describe web pages
  + “Hypertext” is the ability to link one page to another
  + First established in 1990s
* HTML Tags: keywords surrounded by angle brackets, e.g. <html>
  + The purpose of the tags is to group and describe page content
  + HTML tags normally come in pairs, like <h1> and </h1>
  + The first tag in a pair is the “start tag,” the second tag is the “end tag”
  + Start and end tags are also called “opening” and “closing” tags
  + Some tags, such as <img>, are self-closing
* HTML element: everything from the start tag to the end tag.
  + <p>This is a paragraph.</p>
  + Start tag: <p>
  + Element content: This is a paragraph.
  + End tag: </p>
* HTML documents describe web pages. All they consist of is HTML tags in plain text. Networked HTML documents are web pages.
* Web browsers read HTML documents and display them as web pages Web browsers do not display HTML tags, but use them to interpret the content of the page.

<!DOCTYPE html> tells browsers that they are interpreting an HTML document

<html> Text between describes the web page

    <head>

        <meta charset="utf-8">

        <title>Page Title</title> Text between is displayed as the page title (usually at the top of the browser window)

    </head>

    <body> Text between is the visible page content

    </body>

</html>

* SFTP: SSH (Secure) File Transfer Protocol
  + Web pages are usually created “locally” on a personal computer, then uploaded to a web server
  + A web page is not publicly accessible until it’s published to a web server
  + An FTP client is used to transfer files from a personal computer to a server
  + Cyberduck, Fetch, WinSCP, Transmit, and FileZilla are a few FTP clients to choose from
  + “Local” files are those on a personal computer, “remote” files are those on a web server—“live”
* A Uniform Resource Locator (URL), colloquially termed a web address, is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it. A URL is a specific type of Uniform Resource Identifier (URI), although many people use the two terms interchangeably. URLs occur most commonly to reference web pages (http), but are also used for file transfer (ftp), email (mailto), database access (JDBC), and many other applications. Most web browsers display the URL of a web page above the page in an address bar. A typical URL could have the form http://www.example.com /index.html, which indicates a protocol (http), a hostname (www.example.com), and a file name (index.html). Slashes are used to separate directory and filenames. Conventions already existed where server names could be prefixed to complete file paths, preceded by a double slash (//).
* Static vs. Dynamic Pages: A static website is a group of self-contained, individual pages (or page), sent to the browser from the server one-page-at-a-time.
* HTML, CSS, JavaScript - three layers of web design: structure, style, behavior
* id in HTML should be unique within a page; class in CSS can be applied to multiple elements in the page

<!DOCTYPE html>

<html>

    <head>

        <title>Example Page</title>

    </head>

    <body>

        <h1 id="main-heading">Welcome to My Website</h1>

        <p>This is a sample paragraph.</p>

        <ul>

            <li id="list-item-1">List Item 1</li>

            <li id="list-item-2">List Item 2</li>

        </ul>

    </body>

</html>

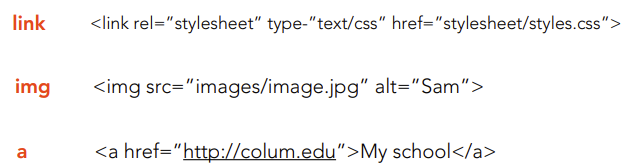
* Nesting: when tags “wrap” other tags. When you create markup, you should indicate nesting by indenting the nested tags with 2 spaces (preferred) or a tab.
* Essential element tags
  + Primary structure: html, head, body



* + Head elements: title, meta, link
  + Structural elements (block): p, br, h1 - h6, ul, ol, a, img, (div)



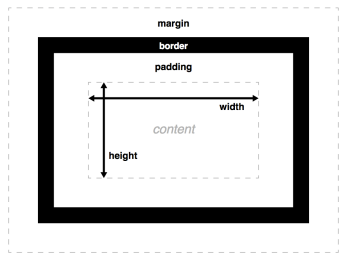
* + Formatting elements (inline): em, i, strong, b, q, blockquote, (span)
* Attributes
  + Most elements can have attributes, which provides additional information about the element.
  + Attributes always follow the same format: name=”value”. You can use either single or double quotes.
  + Essential attributes



* <img src="logo.png"> instructs the web browser to display the image file named "logo.png" on the web page. It does not create a file, set an anchor link, or display text; its primary purpose is to render images on the web page.
* The alt attribute in HTML is used to provide alternative text for an image. This alternative text serves multiple purposes, including accessibility, search enginie optimization, and fallback text.
* Style
  + The style global attribute contains CSS styling declarations to be applied to the element. \*Note that it is recommended for styles to be defined in a separate file or files. This attribute and the <style> element have mainly the purpose of allowing for quick styling, for example for testing purposes.
  + Definition and Usage
    - The style attribute specifies an inline style for an element.
    - The style attribute will override any style set globally, e.g. styles specified in the <style> tag or in an external style sheet.
    - The style attribute can be used on any HTML element (it will validate on any HTML element. However, it is not necessarily useful).
  + Syntax: <*element* style= “*style\_definitions*”>
  + Color: change the text color, change the background color
  + Alignment
  + Font family, font style, font weight, font size
  + <img> attributes
  + <img> width x height
  + Borders and <img> radius
  + <img> link
  + Positioning
  + The <style> HTML element contains style information for a document, or part of a document. It contains CSS, which is applied to the contents of the document containing the <style> element.

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

* CSS - Cascading Style Sheets - a language that web developers use to style the HTML content on a web page
  + <link href="style.css" type="text/css" rel="stylesheet">
  + <link rel="stylesheet" type="text/css" href="mystyle.css">
  + CSS defines a web page’s appearance
  + CSS separates style and content
  + Consists of a plain text file with rules for the display of HTML elements
  + Formatting includes fonts and colors as well as layout and position
  + Can be created outside of your HTML and applied to multiple web pages
  + Well-formed HTML is important for your CSS to work properly
* CSS History
  + Prior to CSS, Web pages were commonly styled with HTML tags and structured with tables, which was both tedious and inefficient
  + Nine different style sheet languages were proposed, two were chosen as the foundation
  + CSS Level 1 emerged as a W3C Recommendation in December 1996
  + Browsers began to support CSS over the next few years
* CSS Application
  + CSS can be applied in three different ways to a web page:
    - In an external .css file
    - In the section of an HTML document
    - Inline with HTML code
* Conflict
  + Three primary factors determine which style rule wins out:
    - Inheritance, Specificity, Location
* Display
  + Elements in HTML are primarily “inline” or “block” elements.
  + An inline element allows content to flow around its left and right sides.
  + A block element fills the entire line and nothing is displayed on its left or right side, e.g. headings and paragraphs.
  + The CSS display property allows you to specify the type of box used for an HTML element.
* CSS Box Model
  + The “CSS box model“ is a set of rules that define how every web page on the Internet is rendered. CSS treats each element in your HTML document as a “box” with a bunch of different properties that determine where it appears on the page.
  + The box model is our toolkit for customizing this default layout scheme.
  + In a web page, every element is rendered as a rectangular box.
  + This box includes the following changeable properties:
    - Content – The text, image, or other media content in the element.
    - Padding – The space between the box’s content and its border.
    - Border – The line between the box’s padding and margin.
    - Margin – The space between the box and surrounding boxes



* Box behaviors
  + **Block boxes** always appear **below** the previous block element. This is the “natural” or “static” flow of an HTML document when it gets rendered by a web browser.
  + The **width of block boxes** is set automatically based on the width of its parent container. In this case, our blocks are always the width of the browser window.
  + The default **height of block boxes** is based on the content it contains. When you narrow the browser window, the <h1> gets split over two lines, and its height adjusts accordingly.
  + **Inline boxes** don’t affect **vertical spacing**. They’re not for determining layout—they’re for styling stuff inside of a block.
  + The **width of inline boxes** is based on the content it contains, not the width of the parent element.
* “Floats” let you put block-level elements side-by-side instead of on top of each other. This is a big deal. It lets us build all sorts of layouts, including sidebars, multi-column pages, grids, and magazine-style articles with text flowing around an image.
* To position an HTML block element directly to the left or right of another block element, you can use the CSS float property. The float property allows you to move an element to the left or right within its containing element, and other content will flow around it. This is commonly used in creating layouts where elements are positioned next to each other, such as floating an image to the left of a block of text.
* Floating an Element
  + The CSS float property gives us control over the horizontal position of an element. By “floating” the sidebar to the left, we’re telling the browser to align it to the left side of the page.
* Float options:
  + left align - float: left;
  + center align - margin: 0 auto;
  + right align - float: right;
* CSS Units of Length
  + Relative units of length include:
    - em (relative to font size)
    - % (relative to the containing element)
  + Absolute units of length include:
    - px (pixels)
  + Alternatively specifications:
    - auto (browser calculates length)
    - inherit (from the parent element)
* CSS3: latest standard for CSS
  + CSS2 is best supported
  + CSS3 is still evolving but offers new features for designers and developers
  + Modern browsers support many aspects of CSS3
  + CSS3 is backwards compatible with CSS2
* CSS Additive (light) (sum is white) vs. Subtractive (paint) (sum is black) Color
* CSS Level 1 only included 16 basic colors, called the VGA colors as they were taken from the set of displayable colors on VGA graphics cards.
* CSS Level 3 Values
  + Although various colors not in the specification (mostly adapted from the X11 colors list) were supported by early browsers, it wasn't until SVG 1.0 and CSS Colors Level 3 that they were formally defined. They are called the extended color keywords, the X11 colors, or the SVG colors.
* RGB Color Codes
  + RGB color values are supported in all browsers, specified with: rgb(red, green, blue).
  + Each parameter (red, green, and blue) defines the intensity of the color as an integer between 0 and 255, e.g., rgb(0, 0, 255) is rendered as blue, because the blue parameter is set to its highest value (255) and the others are set to 0
* Hexadecimal Color Values
  + Hex triplet: a six-digit, three-byte hexadecimal number used in HTML, CSS, SVG, and other computing applications to represent colors.
  + The bytes represent the red, green, and blue components of the color. One byte represents a number in the range 00 to FF (in hexadecimal notation), or 0 to 255 in decimal notation. This represents the least (0) to the most (255) intensity of each of the color components. Thus web colors specify colors in the 24-bit RGB color scheme.
  + The hex triplet is formed by concatenating three bytes in hexadecimal notation, in the following order:
    - Byte 1: red value (color type red)
    - Byte 2: green value (color type green)
    - Byte 3: blue value (color type blue)
* These all produce the same red, but only one is needed
  + color name  
    background-color: red;
  + hexadecimal code  
    background-color: #ff0000;
  + red, green, and blue values  
    background-color: rgb(255, 0, 0);
  + hue, saturation, and lightness values  
    background-color: hsl(0, 100%, 50%);
* Fonts
  + @font-face / @font-family
    - The @font-face CSS at-rule specifies a custom font with which to display text; the font can be loaded from either a remote server or a locally-installed font on the user's own computer.
    - The @font-family CSS property specifies a prioritized list of one or more font family names and/or generic family names for the selected element.

@font-face {

  font-family:"Open Sans";

  src: url("/fonts/0penSans-Regular-webfont.woff2")format("woff2")，url("/fonts/0penSans-Regular-webfont.woff")

  format("woff");

}

p.a{

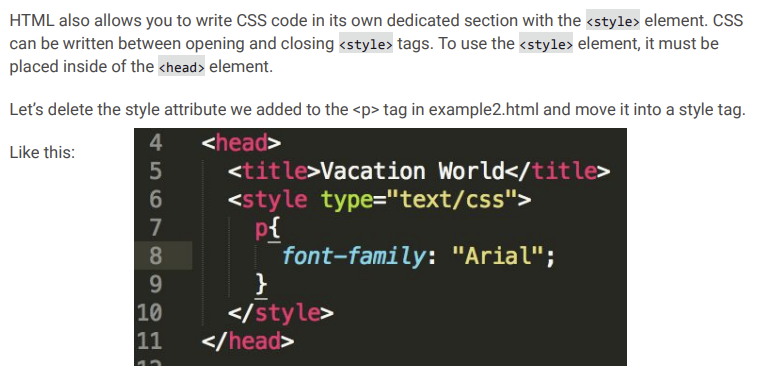
  font-family: "Times New Roman"， Times， serif;

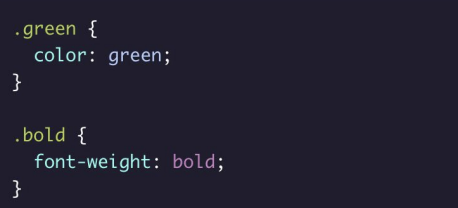
}

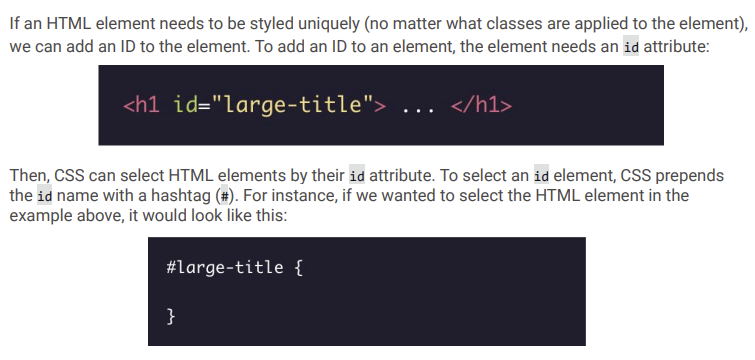
p.b{font-family: Arial，Helvetica, sans-serif;

}

* + Web-safe browser fonts
    - Arial(sans-serif)
    - Verdana(sans-serif)
    - Tahoma(sans-serif)
    - Trebuchet MS(sans-serif)
    - Times New Roman (serif)
    - Georgia (serif)
    - Garamond (serif)
    - Courier New(monospace)
    - Brush Script MT(cursive)
* Background Styling
  + The background shorthand CSS property sets all background style properties at once, such as color, image, origin and size, or repeat method.
* Inline styling
  + Although CSS is a different language than HTML, it's possible to write CSS code directly within HTML code using inline styles. To style an HTML element, you can add the style attribute directly to the opening tag. After you add the attribute, you can set it equal to the CSS style(s) you'd like applied to that element, e.g. style=“*font-family*: Arial;”



* CSS Selectors
  + Tag Names
    - The basic CSS selector targets HTML tag names. We’ve seen how we are able to target HTML elements like the <p> and <img> tags in the code you wrote previously. For example, let’s target the <div> elements in our source code and change the color to maroon. Any element with the tag of <div> will be affected by the CSS styling we specify.
  + Class Names
    - CSS is not limited to selecting elements by tag name. HTML elements can have more than just a tag name; they can also have *attributes*. One common attribute is the class attribute. lt's also possible to select an element by its class attribute.
    - For example, consider the following HTML:
    - <p class="brand">Sole Shoe Company</p>
  + Multiple Classes
    - It's possible to add more than one class name to an HTML element's class attribute. For instance, perhaps there's a heading element that needs to be green and bold. You could write two CSS rules like so:
    - 
  + CSS classes are meant to be reused over many elements. The convention is to use classes to target multiple HTML elements at once, i.e. if multiple elements all need the same styling.
* ID Name



* While classes are meant to be used many times, an ID is meant to style only one element. IDs override the styles of tags and classes. Since IDs override class and tag styles, they should be used sparingly and only on elements that need to always appear the same.
* CSS pseudo classes
  + A CSS pseudo-class is a keyword added to a selector that specifies a special state of the selected element(s). For example, :hover can be used to change a link’s color when the user's pointer hovers over it.
  + /\* Any link over which the user's pointer is hovering \*/
  + a:hover { color: blue; }
  + CSS Buttons Pseudo-classes aren’t just for styling text links—they can be applied to any kind of selector (not just type selectors).

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

* Raster Graphics
* Binary Files
  + All files can be categorized into one of two file formats:
    - Text
    - Binary
  + When you write code, that is a text file.
  + An image file, on the other hand, is a binary file.
  + Binary files typically contain a sequence of bytes, or ordered groupings of eight bits.
* Raster Graphics, also referred to as “bitmap” graphics, are binary files.
  + Raster graphics consist of a grid of picture elements, pixels, each of which contain color and brightness information.
  + Pixels can be changed individually or as a group with program algorithms.
  + This is contrast to vector graphics, which describe points and lines.
* Formats:
  + JPEG(Joint Photographic Experts Group), does not support transparency
  + PNG(Portable Network Graphics), supports transparency
  + GIF(Graphic Interchange Format)
  + WebP(Up-and-coming web image format)
* Photoshop
  + Photoshop was created in 1987 by Thomas Knoll, then a PhD student at the University of Michigan. It was originally called “Display.” Its purpose was to display and manipulate grayscale images scanned into a computer. Display was acquired by Adobe in 1988. It was released as Photoshop 1.0 for Macintosh in 1990. Layer support was introduced in version 3 (c. 1993).

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

* Vector Graphics
  + Unlike raster graphics, which are defined by a specific resolution (measured in pixels per inch or PPI), vector graphics are resolution-independent
  + Vector graphics contain geometric objects, such as lines and curves.
  + This has advantages compared to raster-only formats.
  + Since all modern displays are raster-oriented, the difference between raster-only and vector graphics comes down to where they are rasterized.
  + Vector graphics are “rasterized” client side; raster graphics are, by nature, already rasterized on the server.
* Scalable Vector Graphics (SVG) is a markup language for describing two-dimensional graphics.
  + SVG allows for three types of graphic objects: vector graphic shapes, images, and text.
  + SVG drawings can be interactive and even styled with CSS.
  + SVG defines vector graphics in XML format.
    - XML stands for “Extensible Markup Language”
    - It is a markup language designed to transport and store data.
    - Whereas HTML is about describing and displaying information, XML is about carrying information.
    - XML tags are not predefined; they are “extensible.”
    - Most XML grammars represent either textual information or raw data; they only provide rudimentary graphical capabilities.
    - SVG provides a rich, structured description of vector and mixed vector/raster graphics with pure XML
* Scalability- To be scalable means to increase or decrease uniformly.
  + In terms of graphics, it means not being limited to a single, fixed, pixel size.
  + On the web, scalability means that a particular technology can grow over time.
  + SVG is scalable in both senses of the word
* Advantages of SVG
  + SVG supports transparency
  + SVG images can be created and edited with any text editor.
  + SVG images can be searched, indexed, scripted, and compressed.
  + SVG images are scalable, can be printed at any resolution, and are zoomable without degradation.
  + SVG is an open standard!
  + The advantages of style sheets are generally accepted, certainly for use with text and layout. SVG extends this control to the realm of graphics. It allows for script-based manipulation of the document tree and the style sheet
* SVG Path Element
  + The <path> element is foundational to drawing with SVG; it allows you to create all kinds of shapes.
  + The shape of a <path> element is defined by one attribute: d
  + The d attribute contains a series of commands and parameters used by those commands.
  + All of the commands also come in two variants: an uppercase letter specifies absolute coordinates; a lowercase letter specifies relative coordinates.
* SVG Path Commands
  + M - moveto
  + L - lineto
  + H - horizontal lineto
  + V - vertical lineto
  + C - curveto
  + S - smooth curveto
  + Q - quadratic Bezier curve
  + T - smooth quadratic Bezier curveto
  + A - elliptical arc
  + Z - closepath
* SVG on the Web
  + There are several ways in which SVG content can be included within a web page:
    - A stand-alone SVG web page
    - Embedding by reference, using the HTML <img> element
    - Embedding SVG code inline with HTML
    - From an external link, using the HTML <a> element
    - Referenced from a CSS property
* The fill property specifies the fill color of the SVG element
* SVG includes <line>, <rect>, <circle>, <polygon> but not <triangle> element

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