Website: A collection of various types of files that together create the content, visual appearance, and functionality of a site

Browser: retrieve and display web pages containing content, (text, images, videos, and interactive elements) both locally and via the web, navigate to the internet – URL/ hyperlinks -> fetch and render web pages

Internet: global network of interconnected computers that communicate and share info, **1st Internet (1960s)**

ARPANET Advanced Research Projects Agency Network

To allow researchers from UCLA, Standard Research Institute(SRI), UCSB, and the University of Utah to work together and share resources

Networking: Establishing common standards to facilitate communication through different systems

The Web (1989 by Tim Berners-Lee) originally World Wide Web (www): a subset of the internet; just one of the ways information/documents are shared using a protocol called **HTTP (HyperText Transfer Protocol)** Other protocols: POP3/IMAP/SMTP, FTP, SSH

Servers: A computer connected directly to the internet. Special computers that "serve up" documents upon request. Web servers are called **HTTP servers**

Internet Service Provider (ISP): company provides Internet access to users, or clients and physical infrastructure for users to connect to Internet. Verizon, Spectrum, AT&T, COX, Frontier, Century Link, Mediacom, Xfinity

Router: Networking device that relays data packets between computer networks, direct flow of Internet traffic so that packets arrive at app. Destination, data sent as numeric IP address

IP(Internet Protocol) Address: every computer connected to the internet is assigned a unique 123.45.678.90

Domain Name System (DNS): developers can refer to servers by domain names (i.e. emilydidthis.com)

Wireless Tech: WiFi, radio television broadcasting, cellular comm., **global position systems (GPS),** bluetooth

Computer: A machine that processes information based on a program, Laptops, Smartphones , Smart watches, Cars, Gaming devices, Toasters, Calculators

Program: Instructions written to accomplish certain tasks**.**

Binary Language: 0/1 electrical impulses +5v/-5v. **1 Byte: 256 unique states**

Problems with Early computers: Ran on punch cards, One program at a time, Not user friendly, Limited resources, No standardization, Minimal security + protection

Hardware: tangible, physical parts of a computer responsible for executing and performing the actual physical operations. **central processing unit (CPU), memory (RAM),** hard drive, monitor, keyboard, mouse, peripheral devices (printers + scanners)

Software: programs, data, and instructions that tell the hardware what to do. operating systems, applications (like word processors, web browsers, and games), system utilities

Operating Systems: Abstract the hardware, Better resource management, Multi-programming, Security + protection

User Interface: Portion of system software that allows to interact with data. 2 types, **Graphical (GUI), Command Line (CLI)** GUI is more user-friendly, but command line is faster

Secure Shell (SSH): Allows users to securely log into remote systems and execute commands on those systems. login, change password, change file permissions

SFTP (SSH File Transfer Protocol): Used solely for transferring files between client and server, File management capabilities such as uploading, downloading, renaming, and deleting files

puTTY: open-source **terminal emulator** and **SSH client** for Windows **Cyberduck:** free and **open-source SFTP client for Mac and Windows**. Alternatives: **Transmit** and **Fetch** for **Mac**, **WinSCP** for **Windows**, and **FileZilla** for **Mac, Windows, and Linux**.

Display: flex
Flex-direction: row column row-reverse
Gap: row-gap column gap;
Justify-content: center;
Align-items: center;
Align-content: center;

HTML: Add and orders elements on a webpage. Like the skeleton of a webpage.
CSS: Handles the styling of your website. For example, fonts, colors, sizing, etc.
JavaScript: Adds action and allows user interactions. For example, buttons and text input fields.

Unix: Open-source , by AT&T Bell Labs 1969, Command line interface, Portable, multi-tasking, multi-user, Free distribution, open system, Servers (including i6), workstations, mobile devices -- MacOS, Android, iOS, Linux **Non-Unix:** MicrosoftOS **(i6 is Linux)**

% ls	list directory files
% pwd	show current directory
% cd	change directory
% cd ~	go to home directory
% cd ..	go to parent directory
% touch	create, change, modify timestamp of file
% mkdir	create directory

Chmod: sets permissions of files and directories, every file and directory has 9 permissions

Files and directories have three types of permissions (or none):
- r (read)
- w (write)
- x (execute)
- - (no permission)

The above permissions occur for each of the following classes
- or users:
- u (user/owner)
- g (group)
- o (other/world)

Permissions			Unix Commands
U	G	W	
rwx	rwx	rwx	% chmod 777 filename
rwx	rwx	r–x	% chmod 775 filename
rwx	r–x	r–x	% chmod 755 filename (Standard Dir.)
rw–	rw–	r––	% chmod 664 filename
rw–	r––	r––	% chmod 644 filename (Standard File)

Images		Documents	
PNG	2 – 4 kB	DOCX	4 – 8 kB
GIF	6 – 8 kB	PDF	18 – 20 kB
JPG	9 – 12 kB		

1 Bit	=	Binary Digit
1 Byte	=	8 Bits
1 Kilobyte (KB)	=	1024 Bytes
1 Megabyte (MB)	=	1024 KB
1 Gigabyte (GB)	=	1024 MB
1 Terabyte (TB)	=	1024 BG

Media Files	
eBook	1 – 5 MB
MP3 song	3 – 4 MB
DVD Movie	4 GB
HD Movie	5 – 8 GB
Blu-Ray	20 – 25 GB

Hypertext Markup Language (HTML)

Hypertext:	links that connect web pages to another
Markup:	uses tags to define text structure and formatting
Language:	what the computer system understands and uses to interpret commands

Whole thing -> Absolute URL

https:// (Protocol) www. (Subdomain) hostinger.com (Domain name) /tutorials/what-is-... (Path/Page) **/Relative URL**

Block Element: Starts on a new line, Takes up the full width available, can contain other block-level elements and inline elements, **<h1> <p> <blockquote>**
Inline Element: Doesn't create line breaks, Occupies only necessary width, can contain only inline elements **
 <img> **
**Self-closing elements (i.e.
, <img>)** can't contain other elements, **<ul>** should only contain **<li>** elements

`<!DOCTYPE>`	Declares the document type
`<html> ... </html>`	Container for all other HTML elements (except <!DOCTYPE>)
`<head> ... </head>`	Contains metadata about the document, such as page title and links to stylesheets or scripts
`<title> ... </title>`	Sets the title of web page, which is displayed in browser's title bar or tab
`<meta> ... </meta>`	Contains metadata information about the document, including character encoding and viewport settings

`<h1> to <h6>`	Headings that define the hierarchy of section titles
`<p> ... </p>`	Defines paragraphs of text
` `	A line break element to create new line
`<hr>`	"Horizontal rule", used to insert a visual break in content
`<ul> ... </ul>`	Defines an unordered (bulleted list)
`<li> ... </li>`	List element
`<blockquote>`	Defines a block of text that is a quotation

Ids: An attribute used to uniquely identify a specific HTML element within a webpage, **Only ONE** instance can be used on a HTML page, Used to target and manipulate an element (with CSS and Javascript), id = "x", **#**x {}
Classes: An attribute used to uniquely identify a specific HTML element within a webpage, **Several** instances of a style can be used on a HTML page, class = "x", **.**x{}

** Division Tag:** Block level container element, groups other HTML elements together, Used for layout, styling, and Javascript functionality, No specific meaning not "semantic", Often used with id

History of CSS: 1990s: Web pages were commonly styled with HTML tags and structured with tables **1996:** To separate content from presentation, the concept of CSS was proposed. In 1996, the **World Wide Web Consortium (W3C)** released the first CSS specification, **CSS1** [fonts, colors, margins] **1998: CSS2** [absolute positioning, improved support for tables, more sophisticated selectors for targeting elements] **Ongoing**: **CSS3 [gradients, animations, transitions, media queries, and more]**

Cascading Style Sheets (CSS): Separates the content (HTML) from its presentation (styling), style elements such as text, fonts, colors, backgrounds, borders, and spacing , define the layout of pages, including the positioning of elements, responsive design for various screen sizes, and multi-column layouts

1. Inline with HTML code 2. In <head> 3. In an external .css file

There are two types of length units in CSS: **relative** and **absolute**.

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)

Cascade: Applied when style rules are in conflict. **3** factors: **Inheritance, Specificity, Location**

2 File formats: Text -- Code
Binary -- Contain a sequence of bytes, or ordered groupings of eight bits. E.g. Image

CSS Rule Set: Selector Indicates which HTML element will be formatted, **Property/Value pair(s)** Specifies formatting to apply Style rules are separated by a **semicolon**

link	Describes the styles that illustrate to the user that an element is a link; default is blue underlined text
visited	What a link looks like after it has been clicked on; default is purple underlined text
hover	What a link looks like when the mouse cursor is hovering on it
active	What a link looks like when you are actively clicking on the link

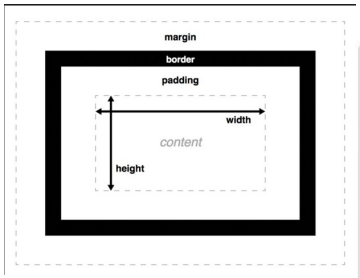
Named color	`background-color: red;`
Hexadecimal code	`background-color: #FF0000;`
Red, green, blue (RGB)	`background-color: rgb(255, 0, 0);`
Hue, saturation, lightness (HSL)	`background-color: hsl(0, 100%, 50%);`

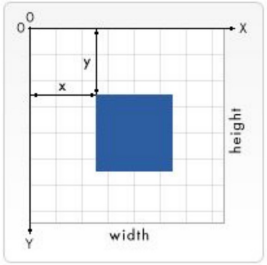
`.`	Represents current directory
`./images/pic.jpg`	Relative to the current directory
`images/pic.jpg`	Relative to the current directory
`../page.html`	One level up in the directory structure
`/css/styles.css`	Root-relative, starts from the website's root directory

On the web, an absolute URL includes the protocol (http/https), the optional subdomain (www), domain (example.com), and path.
https://www.example.com/about/team/

On your local computer:
`file:///C:/Users/YourUsername/Documents/example.txt`

Value	Taken out of the Normal Flow	Positioned Relative To...
Static	No	N/A
Relative	No	itself
Absolute	Yes	closest positioned ancestor
Fixed	Yes	the initial containing block established by the viewport
Sticky	No	closest scrolling ancestor and containing block





`i6.cims.nyu.edu/~ez560` **Absolute** **External**

`/Users/emilyzhao/Downloads/module-04.pdf` **Absolute** **Local**

`http://127.0.0.1:5502/index.html` **Absolute** **Local**

Non-Destructive Editing (NDE) is a method of editing in Photoshop that allows you to make changes to an image without losing the original image information.

For example: using separate layers for text and brush strokes, layer masks, blending modes, layer styles, smart objects!

Destructive Editing is a method of editing in Photoshop where you are permanently changing the original image.

For example: merging layers, adding filters, deleting pixels with an eraser.

You can still 'undo' while you are actively editing the image and have the project open. However, once you save it, there is no going back!

JPEG	"Joint Photographic Experts Group"
PNG	"Portable Network Graphics"
GIF	"Graphic Interchange Format"
WebP Web Picture Format	Newer web image format that is gaining solid browser support
AVIF	New, open-source image format for still and animated image

Raster Graphic Editors: Photoshop, Gimp (opensource), Affinity Photo
Standard images for the web should be as small as possible **(~5-500KB)**

Vector Graphic Editors: Adobe Illustrator, Text Editor, Inkscape (opensource)

<img alt="A baby smiling with a yellow headband." src="baby-lowres.jpg" **srcset=" baby-high-1.jpg 1.5x, baby-high-2.jpg 2x, baby-high-3.jpg 3x, baby-high-4.jpg 4x, baby-high-5.jpg 100x">**

Raster Graphics: "**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. Raster graphics are, by nature, already rasterized on the **server**.

Vector Graphics: Contain **geometric** objects, such as **lines** and **curves,** can be scaled up or down without a loss of quality because the software can recalculate the shapes based on the new size, all **modern displays** are **raster-oriented**, Vector graphics are "rasterized" **client** side.

Photoshop: Created in **1987** by **Thomas Knoll**, then a PhD student at the University of Michigan. Originally called "**Display**." manipulate and create digital images, Tools and metaphors from darkroom photography (Dodge, Burn, Crop), Uses **raster** or **bitmap graphics**, **Images** are made up of **pixels**

File Compression: Compression is when algorithms minimize your file size
Lossy: some data is discarded and approximated resulting in smaller file size **Lossless:** original data is reconstructed resulting in a more accurate file with a larger file size

Scalability: 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

Where to get images?
Unsplash.com, pexels.com, Wikipedia commons, the noun project

File Size

File size is determined by...

- Bit depth (the number of bits used to indicate the color of a single pixel)
- Image dimensions
- Image resolution
- File type

Screen Resolution vs Image Resolution

PPI is frequently associated with screen displays, where it suggests the pixel density of a screen. However, the same image can be displayed on screens with different PPI values without changing the image file itself. The image's resolution, in terms of pixel dimensions, remains the same.

Scaling vs. Image Quality

Increasing the PPI of an image (without adding more pixels) through software settings or image metadata does not enhance the image's quality or resolution. It only changes how the image is displayed or printed at a given physical size, which can be misleading for those expecting improved image detail.

The same image with a fixed number of pixels may have a higher PPI on a smaller screen and a lower PPI on a larger screen. However, the image's resolution remains unchanged. PPI is a relative measure that depends on both the image and the display size.

CMYK CYAN, MAGENTA, YELLOW, BLACK (KEY)

The standard color mode for anything that will be printed

- Subtractive color: light is removed as ink is added
- 100% of each makes black
- 0% of each makes white
- Values for CMYK are specified as a percentage between 0 and 100

EPS, AI, PDF

RGB RED, GREEN, BLUE

The standard color mode for anything that will be viewed on a screen

- Additive color: light is added together
- 100% of each makes white
- 0% of each makes black
- Values for RGB are specified as a number between 0 and 255

JPG, PNG, PSD

TABLE 23-1. Choosing the best bitmapped (raster) file format

If your image...	use...	because...
Is graphical, with flat colors	8-bit PNG or GIF	PNG and GIF excel at compressing flat color.
Is a photograph or contains graduated color	JPEG	JPEG compression works best on images with blended colors. Because it is lossy, it generally results in smaller file sizes than 24-bit PNG.
Is a combination of flat and photographic imagery	8-bit PNG or GIF	Indexed color formats are best at preserving and compressing flat color areas. The pixelation (dithering) that appears in the photographic areas as a result of reducing to a palette is usually not problematic.
Requires transparency	GIF or PNG-8	Both GIF and PNG allow on/off transparency in images.
Requires multiple levels of transparency	PNG-24 or PNG-8	Only PNG supports multiple levels of transparency. PNG-24s with alpha transparency have a much larger file size, but it is easier to find tools to create them. WebP also supports alpha transparency, and may be a better option once it is better supported.
Requires animation	GIF	GIF is the only supported format that can contain animation frames. APNG and WebP may be better options in the future.

We want the highest resolution images, but **quality is usually proportional to file size** The bigger the file sizes, the longer it takes for images to load, the slower our website runs!

HEX

A pair of digits and letters that represent R, G, B values

- Each digit represents a number between 0 and 255
- These hex color codes are composed of 16 possible values for each digit (0-9 and A-F), allowing for a total of 256 possible colors (16^6) in the web-safe palette

- First image format supported by web browsers
- Supports animation
- Preserves transparency
- Lossless compression
- 8-bit: only 256 colors

GIF Graphic Information Format

👍 Smaller file size, great for simple graphics

👎 PNG can do everything the GIF can and better (except animation)

- Great for photographs
- Capable of displaying millions of colors in RGB space
  - 24 bit color (RGB all defined with 8 bits of information)
- Lossy compression

JPEG Joint Photographic Experts Group

👍 Works well with gradient and blended colors

👎 Struggles with flat colors and hard edges

- Developed by Google intended as a replacement for JPEG, PNG, and GIF file formats.
- Preserves animations and transparency
- Can produce smaller file sizes than JPEG
- Supports both lossless and lossy compression

👍 “the Swiss Army knife of image formats”

👎 Not well supported (yet)

WebP Web Picture Format

- Newest image format (meant to replace GIF)
- Preserves transparency
- Lossless compression
- 24-bit (PNG-24): millions of colors
- 8 bit (PNG-8): 256

PNG Portable Network Graphics

👍 Good for flat color (logos, line art, icons)

👎 Works for photos but won’t be saved as efficiently, resulting in larger files sizes

PPI is not Inherent to the Image

The PPI value is not an inherent property of a digital image file. It's a metadata tag that suggests how the image should be displayed or printed at a particular size. Changing the PPI metadata does not alter the actual image data.

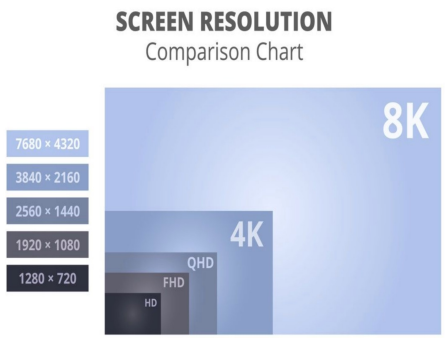
Standard Resolutions

- Standard resolution for **screen-based** images is 72 ppi (pixels per inch)
- Standard resolution for **print** is 300 dpi (dots per inch) or higher
- Document or Canvas size specifies the height and width of an image but does not refer to the image’s resolution

PPI vs DPI

PPI (Pixels per Inch) and DPI (Dots per Inch) are often used interchangeably, but they refer to different things.

- PPI relates to the number of pixels in a digital image
- DPI is a measure of the printer's resolution



Scalable Vector Graphics

- 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 (eXtensible Markup Language)

- XML is a general-purpose markup language used to structure data in a way that's both human-readable and machine-readable.
- It doesn't define how data should be presented; instead, it defines the data's structure and hierarchy.
- In XML, you define your own tags and document structure; they are “extensible.” XML doesn't have predefined tags like HTML.
- SVG provides a rich, structured description of vector and mixed vector/raster graphics with pure XML.

SVG Viewbox

`viewBox` defines the logical coordinate system and aspect ratio for the SVG content, allowing for flexible and responsive scaling. Rectangular are is specified in user coordinates (`x y width height`)

```
<svg viewBox="0 0 100 100">
    <!-- SVG content goes here -->
</svg>
```

`width` and `height` set the physical dimensions of the SVG element on the screen or within the document but may not preserve the content's aspect ratio. It's typically used for fixed-size SVGs.

```
<svg width="200" height="100">
    <!-- SVG content goes here -->
</svg>
```

title
Provides an accessible, short-text description of any SVG; not rendered as part of graphic but displayed rather as a tooltip

```
<title> This is a description </title>
```

group
Used as a container to group SVG elements

```
<g>
    <circle cx="40" cy="40" r="25" />
    <circle cx="60" cy="60" r="25" />
</g>
```

```
<svg width="550" height="300" xmlns="http://www.w3.org/2000/svg">
    <defs>
        <linearGradient id="gradient1">
          <stop offset="0%" stop-color="yellow" />
          <stop offset="100%" stop-color="magenta" />
        </linearGradient>
    </defs>

    <rect
     x="30"
     y="30"
     width="240"
     height="240"
     fill="url(#gradient1)"
     opacity="0.8"
    />
</svg>
```

Advantages of SVG

- 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!

All the ways we can embed SVGs

- Inline with HTML
- External link using the HTML <a> element
- Embedding by reference using the HTML <img> element
- Referenced from a CSS property (i.e. background image)
- A stand-alone SVG web page

Common SVG Styling Properties

`fill`
sets the color inside the shape/object

`stroke`
sets the color of the line drawn around the shape/object

`stroke-width`
defines the width of the stroke
supply a value that is a number; don’t use px units!

`opacity`
specifies the opacity/transparency of a shape/object
supply a value that is a floating point number from 0 to 1 (i.e. 0.5)

SVG Drawing Elements

Rectangle
Specify attributes for top, left point of rect (x, y) and size (width and height)

```
<rect x="100" y="100" width="100" height="100" />
```

Circle
Specify attributes for center point (cx, cy) and radius (r)

```
<circle cx="100" cy="100" r="50"/>
```

Line
Specify attributes for 2 points (x1, y1) and (x2,y2) as well as the line color (stroke)

```
<line x1="0" y1="80" x2="100" y2="20" stroke="black" />
```

```
<svg width="200" height="200">
    <!-- Uppercase "M" command to move to (20, 20) -->
    <path d="M 20 20 L 80 20 L 50 80 Z" stroke="yellow"/>
</svg>

<svg width="200" height="200">
   <!-- Lowercase "m" command to move relative to the current position -->
   <path d="m 20 20 l 60 0 l -30 60 z" stroke="red"/>
</svg>
```

`<defs>`

- You define reusable elements, patterns, gradients, filters, and masks that can be referenced and applied within an SVG document
- `<defs>` is used to separate and store these definitions, making the SVG document more organized and efficient
- After you define elements within the `<defs>` section, you can reference and apply them in the main body of your SVG document using elements like `<use>`, `<linearGradient>`, `<radialGradient>`, `<pattern>`, `<filter>`, or `<mask>`

`xmlns`

- An XML Namespace is a way to avoid naming conflicts when elements and attributes in an XML document
- Namespaces are identified by a unique URI (Uniform Resource Identifier)

```
<library xmlns:books="http://example.com/books">
```

`<path>`

Most powerful element in the SVG library of basic shapes

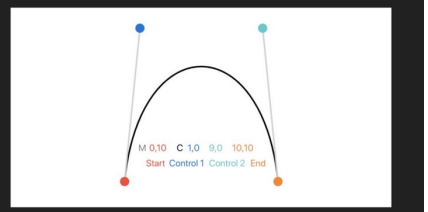
- Can be used to create rectangles, circles, ellipses, polylines, and polygons.
- Can create lines, curves, arcs, and basically any other shape, too.

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.

M = moveto
L = lineto
H = horizontal lineto
V = vertical lineto
C = curveto
S = smooth curveto
Q = quadratic Bézier curve
T = smooth quadratic Bézier curveto
A = elliptical Arc
Z = closepath

`Bezier curve`




```
<path d="M 0,10 C 1,0 9,0 10,10" />
```

Digital Accessibility: a11y
World Wide Web Consortium (W3C): international community for setting WWW standards and guidelines
Web Accessibility Initiative (WAI)
produces the **Web Content Accessibility Guidelines (WCAG)**
15% ~ 1.3 billion people! **self-identify as having a disability**
Content Structure: When you build your website using structural elements instead of styles, you give critical context **assistive technologies (AT)**, such as **screen readers**.
Contrast: Provide sufficient contrast between foreground and background colors. This includes: — Text on images — Background colors and gradients — Buttons — Other interactive elements
Color: Don't use color alone to convey information — Use asterisks to indicate required fields — Use labels to identify areas on graphics, graphs, and charts — Keep your color palette small (1-5) colors
60-30-10 Rule: Primary-Secondary-Accent
Fonts: 1-2 fonts, If you are using multiple fonts, they should complement each other
Hierarchy: visual hierarchy refers to how elements are arranged in a design. This helps the user better understand the flow so they know where to look first.
Interactivity: Ensure that interactive elements are easy to identify — All buttons or links should have a distinct style so they are visually different from other content — This includes hover, focus, and touch

User Feedback: All website interactions should provide the user with some sort of feedback — Alerting a user when something goes wrong — Proving submission confirmation — Notifying users that content has changed on a webpage

Content Structure + Semantic HTML

When you use semantic HTML elements, the inherent meaning of each element is passed on to the accessibility tree and used by the AT, giving more meaning to the content than non-semantic elements.

- **Landmarks**: <header>, <footer>, <nav>, <main>, <section>, <form>
- **Headings**
- **Lists**
- **Tables**
- **Alt Text**

Voiceover Shortcuts + Accessibility Tree Demo

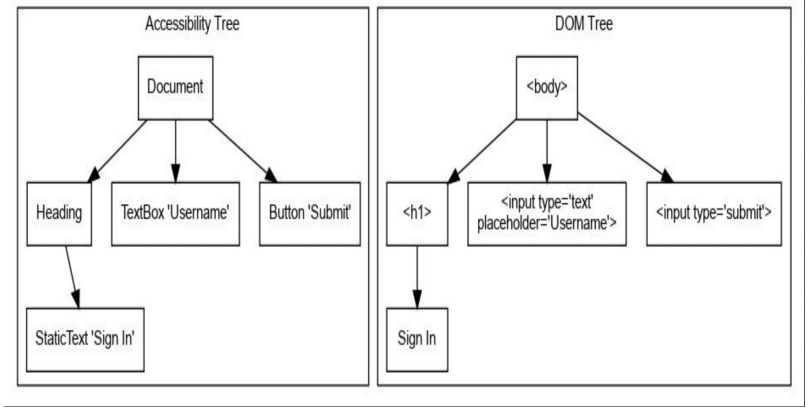
Start voiceover:	COMMAND+F5
VoiceOver Rotor:	VO (CTRL+OPTION) + U
Headings:	VO (CTRL + OPTION) + COMMAND + H

Additional beneficiaries of accessibility

- Temporarily disabled
- Situationally disabled
- Mildly disabled
- Non-native speakers
- Older people with age-related diminishing senses
- Search engine optimization (SEO)

Everyday products that evolved via accessibility

- Telephones
- Typewriters/keyboards
- Email
- Kitchen utensils
- Easy-open pull-out drawers
- Automatic door openers
- Voice controls
- Eye gaze technology



Types of Impairment

Visual: blindness, low vision, color blindness

Mobility: paralysis, limited mobility

Hearing: deafness, varying degrees of hearing loss

Cognitive: dyslexia, ADHD, and autism

Seizure and vestibular disorders: photosensitivity, seizures triggered by visual stimuli (flashing lights, certain patterns, etc...)

Speech impairments: communication disorders, difficulty speaking due to conditions such as cerebral palsy, ALS, or congenital disabilities

Core Principles of Web Accessibility (POUR)

Perceivable: content should be presented in a way that all users can perceive, including those with visual or hearing impairments

Operable: Users should be able to interact with and navigate the website through various input methods, such as keyboard or voice commands

Understandable: Information and user interface elements should be clear and easy to understand

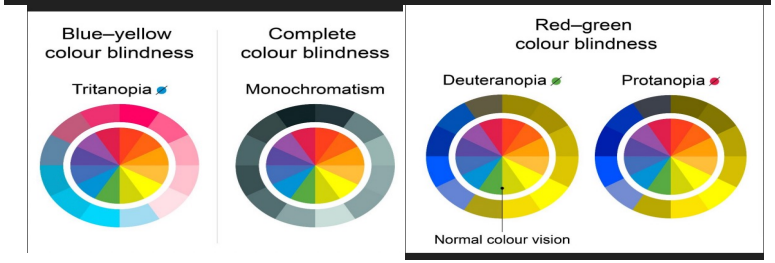
Robust: Websites should be compatible with current and future technologies, ensuring long-term accessibility

Accessibility Tree

The accessibility tree includes four properties for each element:

- The name of the element.
- A description of the element, which provides more context than just the name.
- The role of the element, which describes what kind of object it is (such as a button or a checkbox).
- The state of the element, where applicable (such as whether a checkbox is checked or unchecked).

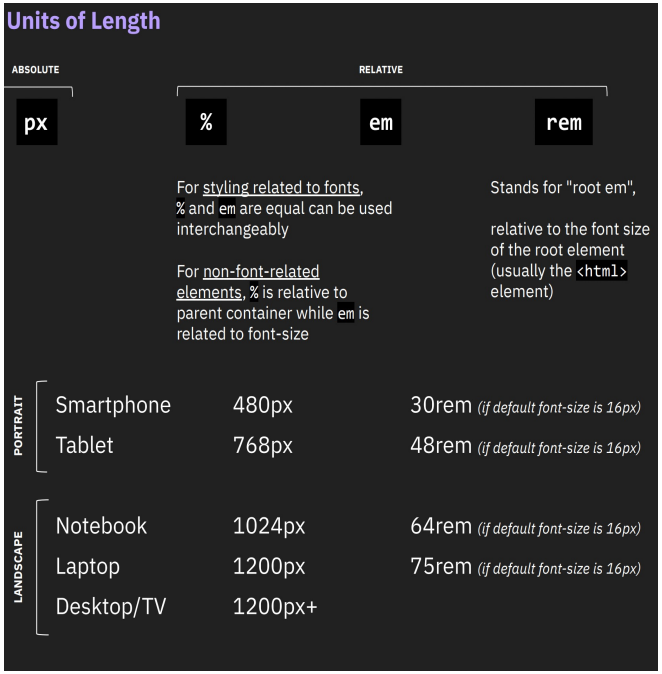
*** The best way to provide these properties is to use native HTML!**



Wireframing: visual representation/blueprint
Key principles of Wireframing:
— Keep wireframes **simple and uncluttered. Use lines and basic shapes to represent elements**. Avoid excessive details and overdesigning!
— Define **the hierarchy of content** – prioritize and differentiate elements like headings, body text, images, and calls to action
— Always **design with the user in mind**. Consider the user's needs, goals, preferences, and abilities
Wireframing Tools: Non-Digital: — Pen and paper Free Websites: — **Figma, Adobe XD, Wireframe.cc, Canva**
An **iterative process** to wire-framing that can be adapted to a variety of design projects: — Think — Design — Implement — Revise This sequence can be looped through as necessary.
Importing Your Own Fonts: 1. Choose a custom font a. **Google Fonts b. Adobe Fonts c. Font Squirrel d. Dafont** 2. Host the font a. Upload the font to your website's server b. Use a third-party service that hosts for you (i.e. Adobe Fonts) 3. Add CSS rule that references the font 4. Apply the custom font to your site
CSS Float Property: The CSS float property allows you to position block elements **inline**. This means that any element, block or inline, can be positioned alongside another element. The CSS float property is an **outmoded** technique of web page layout
CSS Positioning: The CSS position property specifies the type of positioning used for an element on a page.
static Default document flow
absolute Element is positioned relative to its first positioned (not static) parent element
fixed Element is positioned relative to the browser window
relative Element is positioned relative to its normal position
sticky Positioned based on the user's scroll position
CSS FlexBox: Use the CSS Flexible Box Layout Module (Flexbox) for arranging items along **one axis**. Flexbox consists of flexible containers and flexible items within. A flex container expands items **to fill available free space or shrinks them to prevent overflow**. In practice, flexbox can **accommodate different screen sizes** and different display devices **more easily** than the CSS float property.
Flexbox = 1 dimension, Grid = 2 dimensions

CSS Grid: Web pages are **often** laid out using grid systems. Intended to make this process **more intuitive** by defining a grid and then specifying where content should be placed within it. The CSS Grid Layout Module can be used for the **overall structure of a page**. **grid-template-columns** Specifies the number of columns and their widths
grid-gap Set the gap between rows and columns
gap: 10%; /* same spacing for both */
gap: 10px 50px; /* row space col space */
grid-template-areas: Specifies a unique grid using a series of nicknames for each cell of the layout
grid-template-areas: "a a a"
"b c c"
"b c a";
Foundations of Responsive Design:
Mobile-First Approach, Relative Units, Media Queries, Flexible Layouts, Flexible Images
Mobile-First: Content-first — Essential functionality — Lifestyle/news-focused
Desktop: Traditional — Feature-rich — Office-based
auto (browser calculates length)
inherit (from the parent element)
Media Q Conditions:
width and height: You can set conditions based on the width and height of the device's screen.
min-width and max-width: Specify a range of screen widths.
orientation: Adjust styles based on the device's orientation (landscape or portrait).
device-pixel-ratio: Target high-resolution screens (e.g., Retina displays).
aspect-ratio: Set styles based on the aspect ratio of the screen.
color: Detect whether the device supports color or is grayscale.
Media Q Type: Media types describe the general category of a device. **Except** when using the **not** or **only** logical operators, the media type is **optional** and the **all** type will be **implied**.
all: Suitable for all devices
print: Intended for paged material and documents viewed on a screen in print preview mode
screen: Intended primarily for screens
speech: Intended for speech synthesizers

```
<link rel="stylesheet" media="only screen and
(min-width: 640px)" href="tablet.css">
```



Logical Operators: The logical operators **not**, **and**, **only** can be used to compose a **complex media query**. You can also combine **multiple media queries** into a **single rule** by **separating them with commas**.
and: used for combining multiple media features into a single media query, requiring each chained feature to return true for the entire query to be true
not: used to negate a media query, returning true if the query would otherwise return false
only: used to apply a style only if an entire query matches and is useful for preventing older browsers from applying selected styles

Viewport Meta Tag

To ensure that media queries work correctly on mobile devices, it's important to include the viewport meta tag in the HTML <head> section of your web pages. This tag helps control the initial scale and width of the viewport.

```
<meta name="viewport" content="width=device-width,
initial-scale=1">
```

```
@media only screen and (min-width: 480px) {
    body {
        background-color: orange;
    }
}
```

Responsive Images: Responsive images refer to images that **adapt and change based on the viewport size** and device they are being displayed on. — They are an important aspect **of responsive web design**, allowing images to look good and load fast across different screen sizes. — Responsive images **serve better images to clients** and also **improves website loading time**.

Srcset: Srcset is an HTML **image attribute** that specifies the list of images to use in different browser situations. — The browser will pick the **most optimal image** version, based on the screen size and resolution.

srcset + image density — The more common way to to set include size information in the srcset attribute is to label each file by image density. — You do this by putting **1x, 2x**, 3x and so forth after the URL.

srcset + image width — The other way to inform the browser about the different sizes is to actually specify the image **width in pixels**. — This gives the browser more information about the images, so it can make a better decision about which one to select. — This is also good if your image versions **aren't in exact proportion** to each other.

sizes — Allows you to specify the **layout width** of the image for each of a list of media conditions — Each condition is specified using the same conditional format used by media queries.

alt text — Helps **screen-reading tools** describe images to visually impaired readers — Try to be as descriptive as possible in your alt text — Used by search engines to understand the content of image, improving your **SEO** (**search engine optimization**)

Browsers that do not support srcset and sizes will fallback to src

Srcset: specify a **list of images** in different sizes. Behind the file name of each image you specify the width of the image in pixels (**with w not px**). For example, small.jpg 240w means that this image is 240px wide.

Now you may notice that in our example of srcset there's no image with a width of 540px. That's not a problem. The browser will select the best image available upwards in size. In this case, large.jpg will be used with a width of 720px.

URI Fragment: The URI fragment is the **optional part at the end** of a URL which starts with a **hash (#) character**. It lets you refer to a specific part of the document you've accessed. You can **create IDs** in your page and then append them to the URL to navigate to different parts of pages.

Universal Selector (*): used to **select and apply styles to all elements** on a web page.

Child combinator (>): used to **select and style elements that are direct children of a parent element**. The > selector is used to target elements that are **one level deep within** the **parent element and no further**.

CSS Variables (custom properties**):** allow you to define **reusable values** and store them as variables

Favicon: "**favorite** icon" is a small, **typically square** icon that represents a website, web application, or webpage. Favicons are displayed in various places within web browsers to provide a **visual identity** for a site, enhance user experience, and help users easily **identify** and **distinguish** between **multiple open tabs**.

SEO (Search Engine Optimization): the practice of optimizing websites and online content to **improve** their **visibility** in **search engine results pages (SERPs)**. The goal of SEO is to increase **organic** (**non-paid**) **traffic** to a website by improving its **search engine rankings** and making it more **relevant and user-friendly.**

DOM Events: As you navigate the web, your browser registers different types of events.

Common events include: — Clicking or tapping on a link — Hovering or swiping over an element — Resizing the browser window — A web page loading JavaScript can be used to respond to the multitude of events that occur within the DOM.

This: In Event handlers, such as those used in HTML or with the DOM, this typically refers to the **DOM element that triggered** the **event**.

Functions are **blocks** of **reusable code** that can be **defined** and **invoked** (called) to perform a specific task.

Functions play a crucial role in: — Organizing and structuring code — Promoting code reuse — Enabling modular and maintainable programs

```
<link rel="icon" href="favicon.ico" type="image/x-icon">
```

```
<!-- descriptive text for SEO -->
    <meta name="description" content="History and distinctives of rangefinder cameras">
<!-- keywords for SEO -->
    <meta name="keywords" content="rangefinder, SLR, vintage, film, camera">
```

```
li > div {
  /* CSS rules here apply only to div elements that are direct children of an li element. */
}
```

```
li div {
  /* CSS rules here apply to all div elements inside an li, whether they are direct children or nested descendants. */
}
```

```
<a href="http://example.com/page.html#foo">Jump to #foo on page.html</a>
```

```
<a href="#foo">Jump to #foo on the same page</a>
```

To define:

```
:root {
  --primary-color: #3498db;
  --font-family: 'Arial', sans-serif;
  --spacing-unit: 1rem;
}
```

To use:

```
.element {
  color: var(--primary-color);
  font-family: var(--font-family);
  margin: var(--spacing-unit);
}
```

abstract	arguments	await*	boolean
break	byte	case	catch
char	class*	const	continue
debugger	default	delete	do
double	else	enum*	eval
export*	extends*	false	final
finally	float	for	function
goto	if	implements	import*
in	instanceof	int	interface
let*	long	native	new
null	package	private	protected
public	return	short	static
super*	switch	synchronized	this
throw	throws	transient	true
try	typeof	var	void
volatile	while	with	yield

JavaScript: invented by **Brendan Eich** and introduced by **Netscape** in **1995**. — At that time, **the** Java language was **ascendant** and the name "JavaScript" was an attempt to **ride** this **popularity**. — Eventually, browsers **other than Netscape** began to support JavaScript functionality, calling it "**ECMAScript**." — Today, JavaScript is not only a **lingua franca** of the web but a basis for many other **computational media** projects

3 ways you can apply JavaScript to HTML:
1. Inline 2. Embedded 3. External

External and embedded JavaScript are preferable for their separation of content and behavior.

Like HTML and CSS, **JavaScript is usually rendered in the web browser**. — Because it's rendered in the browser rather than on a server, JavaScript is considered a "**front-end**" language. — A browser's "**JavaScript engine**" interprets and executes JavaScript code in the browser. — There are different JavaScript engines for different browsers.

Compatibility: Computationally speaking, **there isn't much JavaScript can't do; it's a robust programming language**. — Core functionality includes **modifying HTML and CSS, communicating with the server, and storing data**.

A markup language, like **HTML**, is designed to — Structure and present content — Provide annotations to define elements such as paragraphs, headings, and links — Its primary role is to describe how content should be displayed.

A programming language, like **Javascript**, is intended to — Create algorithms, implement logic, and perform computations. — It enables the development of dynamic, interactive applications by allowing developers to write instructions that a computer can execute.

Variables: "**Buckets**" that store information in your computer's memory

Naming Variables:
— Can't contain spaces (can use "_" in place) or special characters (!@#%^&*)
— Can only start with a letter or underscore ; can be followed by any alphanumeric character after that
— Can't use Javascript's "reserved" words
— Javascript is case-sensitive

3 different ways to bind an event to an element:
— HTML event handler
— DOM event handler
— DOM Event listener

Data Types: — **Strings** (character-based data) — **Numbers** — **Boolean** Values (true / false) — **Arrays** (a list of values) — **null** / undefined

console.log() allows you to write a message to the Javascript console in the developer tools.

alert() displays an **alert box** with **message** and an **OK** button. Only use this for special cases!

A **boolean expression is a logical statement that** evaluates to either true or false.

Conditionals (**if**): are constructs that allow you to control the **flow of your code** based on specified conditions. **If () {} else if {} else {}**

Document Object Model Tree: When a browser loads a web page, it creates a model of that page. It is stored in the **browser's memory**. Every element, attribute, and piece of text in the HTML is represented by its own "DOM node."

4 main types of DOM nodes:
— The Document node, which represents the entire page
— Element nodes, which represent individual HTML tags
— Attribute nodes, which represent attributes of HTML tags, such as a class
— Text nodes, which represents the text within an element, such as the content of a tag

We talk about the relationship between element nodes as "**parents**," "**children**," and "**siblings**."

DOM queries: JavaScript methods that find elements in the DOM tree
- **Which DOM query you use** depends -> **what you want to do** and **the scope of browser support required**.

Math Operators

`+`	addition
`-`	subtraction
`*`	multiplication
`/`	division
`%`	modulus (remainder)
`++`	incremeter (add 1 to current value)
`--`	decremeter (subtract 1 to current value)

Logical Operators

`&&`	and
`||`	or
`!`	not

Relational Operators

`>`	greater than
`<`	less than
`==`	equality (compares values only)
`===`	strict equality (compares values and data type)
`<=`	greater than or equal to
`>=`	less than or equal to
`!=`	not equal to
`!==`	strictly not equal to

`Math.random()`

`Math.random()`
returns a random decimal number between 0 (inclusive), and 1 (exclusive):

```
num = Math.random()
console.log(num) // 0.5566941489945507
```

`Math.floor()`
Used to round down to the nearest integer

```
num = Math.floor(5.95)
console.log(num) // 5
```

Combined, **`Math.floor()`** and **`Math.random()`** can return random integers

```
num = Math.floor(Math.random() * 10);
console.log(num) // Returns a random integer from 0 to 9:
```

What can we change?

`.textContent`
sets or returns the text content of the specified node

```
let element = document.getElementById("changeMe")
element.textContent = "I have changed!";
```

`.innerHTML`
sets or returns the HTML content (inner HTML) of an element

```
let element = document.getElementById("changeMe")
element.innerHTML = "<a href='#'>I have changed!</a>";
```

`.style.CSSPropertyName`
sets or returns the value of a given CSS property

```
let element = document.getElementById("changeMe")
element.style.color = "red";
```

```
Date.now()       // returns the current number of milliseconds
.getHours()      // returns the hour (military time) of the current local time
.getMinutes()    // returns the minutes of the current local time
.getSeconds()    // returns the seconds of the current local time
.getDay()        // returns the day of the week as a number (Sunday is 0)
```

Methods that return a **single element node**:
.getElementById() .querySelector()
DOM Queries Methods that **return one more more elements** as a **node list:**
.getElementsByClassName() .getElementsByTagName()
.querySelectorAll()
binding: Specifying which event will **trigger** the response.

HTML Event Handler

```
<button onclick="myFunction()">Click me</button>
```

***Downsides**: Mixing HTML markup with JavaScript can make the code less maintainable and harder to debug. It's generally considered a best practice to separate HTML and JavaScript code.*

DOM Event Handler

```
let btn = document.querySelector("button");
btn.onclick = myFunctionName;
```

***Downsides:** Assigning multiple event handlers to the same event on the same element will overwrite the previous assignment.*

DOM Event Listener

```
let btn = document.querySelector('button');
btn.addEventListener('click', myFunctionName);
```

***Most recommended!** They're most flexible and powerful. They allow you to attach multiple event handlers to a single event on a DOM element without overwriting existing ones.*

Keyboard Events

- keydown
- keyup
- keypress

Mouse Events

- click
- dblclick
- mousedown
- mouseup
- mouseover
- mouseout

Focus Events

- focus
- blur

Touch Events

- touchstart
- touchmove
- touchend
- touchcancel

Pointer Events

- pointerover
- pointerenter
- pointerdown
- pointermove
- pointerup
- pointercancel
- pointerout
- pointerleave
- gotpointercapture
- lostpointercapture

Form Events

- input
- change
- submit
- reset
- cut
- copy
- paste
- select

User Interface (UI) Events

- load
- unload
- error
- resize
- scroll

Mutation Events

- DOMSubtreeModified
- DOMNodeInserted
- DOMNodeRemoved
- DOMNodeInsertedIntoDocument
- DOMNodeRemovedFromDocument

Event Handling

1. Select an element for the script to respond to
2. Specify which event will trigger the response
3. Run code specific to that event

```
// Step 1: Select an element for the script to respond to
const button = document.getElementById('myButton');

// Step 2: Specify which event will trigger the response
button.addEventListener('click', myFunction);

// Step 3: Run code specific to that event
function myFunction() {
  alert('Button clicked! This is the response to the click event.');
})
```

HTML Form

```
<form action="my-script.php">
    First name:
    <input type="text" name="firstname">
    Last name:
    <input type="text" name="lastname">
    <input type="submit" value="Submit">
</form>
```

- Forms always begin with the <form> element.
- The <form> element's action attribute specifies how the form will be processed.
- The <input> element is used for various kinds of user input.
- The <input> element's type attribute determines what kind of input is received from users.
- Each <input> element must also have a name attribute and value in order for the data to be sent.

Input Types

Text Input: Enter text

Number Input: Enter number

Date Input: mm/dd/yyyy

Checkbox: Check me

Radio Buttons: Option 1 Option 2

Select: Option 1

Textarea: Enter text

Button: Click me

Color Picker:

Datetime Local: mm/dd/yyyy, --:-- --

File Input: Choose File No file chosen

Range Input:

Reset Button: Reset

Search Input: Search

Submit Button: Submit

```
<input type="button">
<input type="checkbox">
<input type="color">
<input type="date">
<input type="datetime-local">
<input type="email">
<input type="file">
<input type="hidden">
<input type="image">
<input type="month">
<input type="number">
<input type="password">
<input type="radio">
<input type="range">
<input type="reset">
<input type="search">
<input type="submit">
<input type="tel">
<input type="text">
<input type="time">
<input type="url">
<input type="week">
```

FORM VALIDATION: Before form data gets sent, it's important to validate the input. — You may want to make certain form fields **required**. — You probably want to make sure that **certain fields are completed properly**. — You should also **verify that malicious code is not sent along** with form input. Form validation can be done **client-side, server-side, or both**

Use **Formspree**, free

<input type="text" id="lname" name="lname" value="Doe">
Name is for **processing in submission and JS**, **value** is the **default text displayed** to the user.

Date represents the number of milliseconds since **January 1st, 1970 (UTC).**

The **<audio> and <video>** tags use **src** attribute or the **<source>** tag to specify **one or more media resources**.

Version Control: A system that **records changes to a file or set** of files over time so that you can recall specific versions later. Commonly used for software source code but any type of file can be placed under version control
A Version Control System (VCS) allows you to:
— Revert files back to a previous state — Review changes made over time — Collaborate more efficiently — Maintain project backups
Centralized Version Control Systems: developed to allow collaboration with developers on other systems. — With a CVCS, a **single server contains all the versioned files** and **clients** "**check out**" files from that **central** place. — For many years, this has been the standard for version control. — The **downside** of centralized version control is the **vulnerability** of having the **entire history of a project in one place.**
Distributed Version Control Systems: clients don't just check out the latest snapshot of files, they **fully mirror the entire history of the project**. — If a server dies, anyone with a copy of all the versioned files can restore it to the server. — Every checkout is really a **full backup of all the data**. — You can also **collaborate** with different groups of people in different ways simultaneously within the same project
Git: created in **2005** by **Linus Torvalds** and the **Linux development community** for **Linux kernel maintenance.** Linux is an open source operating system project of fairly large scope. Its goal was to be a fully **distributed VCS** with a simple design, support for **non-linear development**, and the **ability to handle large projects efficiently**
Git: thinks of its data like a set of **snapshots** of a mini file system. Every time you save the state of your project, it basically takes a picture of what all your files look like then and stores a reference to that snapshot. **To be efficient, if files have not changed, Git doesn't store the file again—just a link to the previous identical file it has already stored.** This makes Git more like a mini file system with some powerful tools built on top of it.
Git has 3 main states that your files can reside in: **modified, staged, and committed.**
— **Modified** means that you have changed the file but have not committed it to your database yet.
— **Staged** means that you have marked a modified file in its current version to go into your next commit snapshot.
— **Committed** means that the data is safely stored in your local database.
Git Workflow:
1. Modify files in your **working directory** 2. Stage the files, adding snapshots of them to your **staging area**. 3. Commit changes, which takes the files as they are in the staging area and stores that snapshot permanently to your **Git directory**.
GitHub is a **web-based hosting service** that uses the **Git VCS**. — The site also provides **social networking functionality** such as **feeds, followers, wikis, and statistics**. — The company was founded in **2008** and is located in **San Francisco**. — In addition to computer programmers, architects, musicians, municipal governments, and academics are among its users.

HTML Video

```
<video width="320" height="240" autoplay>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

1. The `controls` attribute adds video controls, like play, pause, and volume.
2. The `<source>` element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.
3. It is a good idea to always include `width` and `height` attributes. If height and width are not set, the page might flicker while the video loads.
4. The text between the `<video>` and `</video>` tags will only be displayed in browsers that do not support the `<video>` element.

HTML Audio

```
<audio controls autoplay muted>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

1. The `controls` attribute adds audio controls, like play, pause, and volume.
2. The `<source>` element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.
3. The text between the `<audio>` and `</audio>` tags will only be displayed in browsers that do not support the `<audio>` element.

HTML Inline Frame (`iframe`)

— Another way to embed media on a web page is with the HTML inline frame element: `<iframe>`
— An inline frame represents a nested browsing context, embedding another HTML page into the current one.
— Embedding all or part of one web page into another is way to present content on a website.

```
<iframe src="https://www.w3schools.com"
     title="W3Schools Free Online Web Tutorials">
</iframe>
```

Domain Names: Domain names serve as a more memorable reference to Internet resources. — Domain names are used to identify **Internet Protocol (IP) addresses**. — An IP address is an **identifier for a node**—a computer or device on a network. When you register a domain name, **you are not its owner,** rather **you have the exclusive right to use it**.
Factors to consider when selecting a domain name:
— Relevance to site — Communicability — Availability
List of all domain name registrars:
www.internic.net/alpha.html
Domain Name Registrars:
— Network Solutions — Google Domains — GoDaddy
Whois: a protocol for querying databases to obtain information about the registration of domain names and IP addresses, revealing details like ownership and registration dates.
Internet Corporation for Assigned Names and Numbers (ICANN): nonprofit organization for coordinating the global Internet's systems of unique identifiers, including managing the **domain name system** and **IP address allocation**.
Web hosting service: allows individuals and organizations to make their website accessible to others. — The host usually **provides storage space** on a server as well as **Internet connectivity**.
— Theoretically, **any computer can** serve as a **web host**, but it needs to **always be on and implement measures for security and stability**.
Analytics tools: allow you to observe data about the traffic
This can include the following information:
— Number of visits — Geographic location of visitors — Time spent on pages — Referring web page — Browser information — Real-time activity.
Free: — GitHub Pages — Glitch
Paid: — Pair Networks — Media Temple (now GoDaddy) — Reclaim Hosting
Semantic URLs ("Clean URLs", "SEO-friendly URLs"**)**: web addresses that are designed to be **easily readabl**e and understandable by **both humans and search engines**. For example, a non-semantic URL might look like www.example.com/page?id=123, whereas a semantic version of the same URL could be www.example.com/products/. The latter is more **descriptive and user-friendly**.

Generic TLDs Generic top-level domains initially consisted of:

- GOV: Government agencies
- EDU: Educational institutions
- ORG: Nonprofit organizations
- MIL: Military
- COM: Commercial business
- NET: Network organizations

Some of these, such as .com and .net, are no longer restricted to their original intended usage.

More generic TLDs have since been added and are being added today.

Top Level Domain

Every domain name has a suffix that indicates which top level domain (TLD) it belongs to.

Top-level domains today are grouped as follows:

- Generic top-level domains (`.com .org .net`)
- Country-code top-level domains (`.us .uk .jp`)
- Infrastructure top-level domain (`.arpa`)
- Sponsored top-level domain (`.museum .cat .post`)
- Special-use top-level domain (`.localhost .example`)

On-Page Techniques of SEO

On-page techniques are the methods you can use to improve search results for your site.

This involves identifying and implementing keywords in seven particular places in your page.

1. Page title
2. URL
3. Headings
4. Text
5. Link text
6. Image alt text
7. Page descriptions

Selecting a Web Host

- Dedicated vs. shared server space
- Disk space
- Bandwidth (data transfer)
- Up time (reliability)
- Overage
- Extras: databases, mailboxes, and types of customer support