

Links and Images

CSCI 1210 · Essentials of Web Development

Today's Topics:

- 🔗 HTML Links — the `<a>` element
- 📁 Absolute vs. Relative URLs & paths
- 🖼️ HTML Images — the `` element
- 📐 Image formats, resolution & optimization
- 👤 Web Design Lifecycle: Users & Requirements

East Tennessee State University · Department of Computing



What makes the Web the Web

What Makes the Web the Web

Hyperlinks Changed Everything

One of the things that makes the web unique is **hyperlinks**. They changed the way consumers interact with content — no longer a linear, start-to-finish activity.

Before Links

- ▶ Content consumed top-to-bottom
- ▶ Linear, start-to-finish
- ▶ Like reading a book
- ▶ No jumping between sources

With Hyperlinks

- ▶ Users follow a "breadcrumb trail"
- ▶ Jump site-to-site, or within a site
- ▶ Find exactly the content they want
- ▶ Non-linear exploration

The Anchor Element

The element that creates a hyperlink:



Whatever appears between `<a>` and `` becomes a clickable link.



Default Browser Styling

By default, links are displayed blue and underlined. CSS can override this.

Where to Go — href

`href = "hypertext reference"`

An HTML-speak way of saying "address"— tells the browser where to go when the user clicks the link.

```
<a href="https://www.csci1210.com">CSCI 1210</a>
```

// Will display like this:

[CSCI 1210](https://www.csci1210.com)

Remember: Attributes

- ▶ Go inside the *opening* tag
- ▶ Provide extra info to the browser
- ▶ Make the page behave correctly
- ▶ Format: `name="value"`

href Values

- ▶ Absolute URLs (other sites)
- ▶ Relative URLs (your own site)
- ▶ Page fragments (`#section-id`)
- ▶ Other protocols (`mailto:`, `ftp:`)

Two Types of URLs

Absolute URL

Points to another website. Provides the *entire* URL — a unique, unchanging point on the web.

```
href="http://www.example.com/default.html"
```

Easy to spot — always contains **http://** or **https://**

Relative URL

Points to a file *within your own site*. The path is relative to the current document's location.

```
href="default.html"
```

```
href="#top"
```

No

```
http://
```

— just a filename or path.

The Rule

Linking within your own site? Use *relative URLs*.

Linking to another site? Use *absolute URLs*.

Relative URLs are shorter, portable, and faster (no DNS lookup needed).

Navigating the File Tree

Think of Your Files as a Tree

Big branches = main folders · Smaller branches = sub-folders · Leaves = files. Any branch can have 0, 1, or many leaves.

```
csci1210.com/
├── assignments/
│   └── hw4/
│       ├── hw4.html
│       ├── css/
│       │   └── hw4.css
│       └── images/
│           ├── 001.jpg
│           └── 002.jpg
└── labs/
    ├── lab1.html
    └── lab9.html
└── index.html
```

🔑 Key Concepts

- ▶ **Parent** — folder containing another folder
- ▶ **Child** — folder inside a parent
- ▶ **Grandparent/grandchild** — two levels up/down
- ▶ **..** — go up one level
- ▶ **../** — "back out" of a directory

Making Internal Links

```
<!-- Same directory -->  
<a href="myinterests.html">My Interests</a>  
  
<!-- Subdirectory (going down) -->  
<a href="hw2/homework2.html">Homework 2</a>  
  
<!-- Going UP one level -->  
<a href="../index.html">Home</a>  
  
<!-- Going UP two levels -->  
<a href="../../index.html">Home</a>
```

// Scenario summary:

Same dir: filename only
Going down: folder/file.html
Going up: ../file.html
Up 2 levels: ../../file.html

// No leading "/" for relative!

⚠ Common Gotcha

When going into a sub-folder, do **not** start with a slash — `hw2/homework2.html` is correct; `/hw2/homework2.html` starts from the web root instead.

Links Within a Page (Fragments)

1 Name the Target

Give the destination element an `id`:

```
<h1 id="top">Top of Page</h1>  
  
<p id="footer">This is the bottom</p>
```

2 Link to the Fragment

Use `#` followed by the `id`:

```
<a href="#top">Back to top</a>  
  
<a href="#footer">Go to bottom</a>
```

Common Use Cases

- ▶ "Back to top" buttons on long pages
- ▶ Tables of contents (clicking a chapter heading jumps to it)
- ▶ Single-page sites with anchor navigation

Special Link Types

mailto: Links

```
<a href="mailto:someone@example.com">  
  Email Us  
</a>
```

- ▶ Opens the user's default mail client
- ▶ Addressed to the specified email
- ▶ Requires a default email app to be set
- ▶ Rarely used in modern web development

target="_blank"

```
<a href="https://etsu.edu"  
  target="_blank">  
  ETSU  
</a>
```

- ▶ Opens the link in a *new tab*
- ▶ Keeps your site open in the original tab
- ▶ Good for external links
- ▶ Default behavior overwrites current page

Links — Summary

| Concept | Details |
|------------------------------|---|
| <a> | The HTML anchor tag — creates a hyperlink |
| href | "Hypertext reference" — the destination address |
| Absolute URL | Full path including <code>http://</code> — for external sites |
| Relative URL | Partial path — for pages within your own site |
| Fragment (#id) | Jump to a specific element on the page |
| <code>target="_blank"</code> | Open link in a new browser tab |
| <code>mailto:</code> | Open the user's email client |



The Golden Rule

Use **relative URLs** for links within your site. Use **absolute URLs** for any other site. Path is important — test your links!

Images on the Web

"A Picture is Worth a Thousand Words"

Images are an important part of the web experience. Knowing how to make a raw image suitable for the web is a critical skill for any developer.



What We'll Cover

- ▶ The `` HTML element
- ▶ Required & optional attributes
- ▶ Image file formats
- ▶ Captions with `<figure>`

Optimization Skills

- ▶ Understanding resolution (PPI)
- ▶ File size vs. quality trade-offs
- ▶ Resizing & cropping with GIMP
- ▶ Compression ratios

The Element



Standalone (self-closing) · Inline element

src *

Source URL for the image — absolute or relative path

alt *

Alternative text — required for accessibility & validation

title

Optional title — appears as a tooltip on hover (in some browsers)

```

```

* = required attribute

The alt Attribute

Accessibility

- ▶ Screen readers speak the `alt` text aloud
- ▶ Critical for users with visual impairments
- ▶ Displays when the image fails to load
- ▶ Required for HTML validation

Writing Good Alt Text

- ▶ Describe what the image actually shows
- ▶ Be concise but descriptive
- ▶ Don't start with "image of..." — the browser already knows
- ▶ Decorative images: `alt=""` (empty string)

```
<!-- Good -->
![ETSU Sherrod Library with autumn leaves](campus.jpg)
 
<!-- Bad -->
![image](campus.jpg)
![campus.jpg](campus.jpg)
```

Image Formats

GIF

Graphics Interchange Format

- Small file sizes
- Transparent backgrounds
- Can be animated
- Good for logos, icons, cartoons
- Only 256 colors
- Not good for photos

JPG / JPEG

Joint Photographic Experts Group

- 16+ million colors
- Great for photos
- Lossy compression → small files
- No transparent backgrounds
- Cannot be animated
- Not ideal for logos/icons

PNG

Portable Network Graphics

- 16+ million colors
- Transparent backgrounds (PNG-24/32)
- Lossless compression
- Good for logos, icons, cartoons
- Larger file sizes than JPG
- Cannot be animated

Captions — `<figure>` & `<figcaption>`

Adding Captions to Images

The `<figure>` element wraps an image and its caption together as a semantic unit. Use `<figcaption>` inside it for the caption text.

```
<figure>
  
  <figcaption>Glass Globe</figcaption>
</figure>
```

// Browser renders:



Why Use `figure`?

It's *semantic HTML* — it tells the browser (and screen readers) that the image and caption belong together as a unit. Better than wrapping in a `<div>`.

Image Links

Using an Image as a Link

Instead of using text for a link's display, you can use an image. Simply *nest* an `` inside an `<a>` tag.

```
<a href="https://csci1210.com/index.html" target="_blank">
  
</a>
```

Key Concept: Nesting

- ▶ The `` is nested *inside* the `<a>`
- ▶ The entire image becomes clickable
- ▶ We'll do a lot of nesting in this class
- ▶ Parent → child relationship

Best Practices

- ▶ Always include `alt` text
- ▶ Use `target="_blank"` for external links
- ▶ Describe where the link goes in the alt text
- ▶ Consider adding a `title` attribute for clarity

Image Resolution

Resolution Defined

- ▶ Number of pixels per inch (PPI)
- ▶ Web standard: **72 PPI**
- ▶ Print standard: **300 PPI**
- ▶ Higher PPI = sharper but larger file

Why Not Use 300 PPI for Web?

- ▶ Massive file size impact
- ▶ Screens can't display more than ~96 PPI anyway
- ▶ Wasted bytes = slower page loads
- ▶ File size is the #1 factor in page load time

Two Words:
File Size

Not Everyone Has Fast Internet

- ▶ Satellite connections are slow
- ▶ 3G/4G mobile is still common
- ▶ ~3% of US adults still used dial-up as of 2013
- ▶ File size directly affects user satisfaction

Magic Sizes that Impact Load Speed!

< 14 KB (The "Instant" Zone)

A "magic" rule of thumb, this is the limit for critical, above-the-fold content to load in a single round-trip (TCP slow start). Fitting initial HTML and CSS into this limit makes a site feel instantaneous.

1 MB - 1.5 MB (Ideal Range)

The optimal size for a balance between rich content and speed

< 3 MB (MAX Recommended)

Going over 3 MB typically results in a slow, frustrating experience, particularly on mobile networks, leading to higher bounce rates.

Key Optimization Targets

- ▶ Images: Keep total image weight under 500 KB, with individual images ideally under 100–200 KB.
- ▶ Layout: Limit to 50–60 "blocks" of content to ensure rapid rendering.
- ▶ Code: Minify and compress CSS, JavaScript, and HTML, which can reduce file sizes by up to 60–90 %.
- ▶ Assets: Use lazy loading to defer loading of images, videos, and scripts until they are needed.

Practical Tip

- ▶ Use the same logo file on every page
- ▶ Load it from the same location every time
- ▶ Browser downloads it once, caches it
- ▶ Every subsequent page loads it instantly

The Browser Cache

What is Caching?

The browser stores a local copy of recently downloaded files. If you revisit a site, it doesn't have to download everything again — it loads from the cache instead.

Cache Benefits

- ▶ Dramatically speeds up repeat visits
- ▶ Accessing local memory is very fast
- ▶ Reduces server bandwidth usage
- ▶ Better experience on slow connections



Practical Tip

- ▶ Use the same logo file on every page
- ▶ Load it from the same location every time
- ▶ Browser downloads it once, caches it
- ▶ Every subsequent page loads it instantly

Optimizing Images for the Web

What Does "Optimize" Mean?

Edit image files so the resulting file size is as small as possible *without* unacceptably losing quality. Three main techniques:

Cropping

Remove unimportant parts of the image. Only show what's relevant.
Can dramatically reduce file size.

Resizing

Reduce dimensions and/or resolution. A 1920×1080 photo doesn't need to be displayed full-res on a 400px column.

Compression

Increase JPEG compression ratio. Adjust the quality slider. A quality of 75–85 is often indistinguishable from 100 at a fraction of the size.

552 KB → 24 KB

Real example: by adjusting resolution, dimensions & compression — 4.3% of original file size, still looks great on a web page!

Photo Editors

The Problem with Camera Photos

Most cameras — even cell phone cameras — take pictures that are *way too big* for web pages. Social media sites (Facebook, Instagram) resize automatically, but for your own site you need to do it yourself.

Adobe Photoshop

- ▶ Industry-standard photo editor
- ▶ Extremely powerful feature set
- ▶ Subscription-based (not free)
- ▶ Available in campus computer labs

GIMP

- ▶ GNU Image Manipulation Program
- ▶ Free and open source
- ▶ Comparable features to Photoshop
- ▶ Available for Windows, Mac, Linux

Covered In Lab

Resizing (dimensions & resolution), cropping, and compression — the bare basics of photo editing sufficient for web optimization.

Targeted Site Users

Who Are You Building For?

Every well-designed site has a **targeted set of users**. You cannot create a site that appeals to everyone — and trying to do so results in a site that appeals to no one.

Why Define Users?

- ▶ Profile users based on site mission & goals
- ▶ Define groups to focus the design
- ▶ Know what content and design will appeal
- ▶ Know what to leave out

Where Information Comes From

- ▶ Client's marketing department
- ▶ Sales/marketing employees
- ▶ Existing website analysis
- ▶ Surveys & customer research

User Personas

What is a Persona?

A **persona** is a description of a hypothetical site user — their interests, goals, technology comfort, and concerns. We create one per targeted user group to guide design decisions.

? Questions to Answer

- ▶ Who will visit this site?
- ▶ What do they have in common?
- ▶ What are their interests & goals?
- ▶ What do they like / dislike?
- ▶ What technology do they use?

📋 Focal User Characteristics

- ▶ Age ranges & gender
- ▶ Education level & occupation
- ▶ Computer & web experience level
- ▶ Hobbies and personal goals
- ▶ Devices used (desktop, mobile, etc.)

Example Persona — Joan

Joan is a new teacher in her second year, recently certified. She's starting to get the hang of things but realizing how much she needs to improve. She wants to get better, but also wants to be smart about it.

NEEDS

- Measurable impact
- Project that self-manages

VALUES

- Student growth
- Time savings
- Products that fit her current workflow

WANTS

- Something that doesn't take long to set up
- Products that integrate with existing tech

FEARS

- Cutting into already sparse free time
- Starting something unsustainable
- Doesn't actually improve student outcomes

User Groups — Case Study: Auto Parts Store

Sites Often Serve Multiple User Types

Group users by shared characteristics that distinguish them from other groups. Example: an auto parts store website.



Professional Mechanics

Use: Search by part # & availability
Concern: Delivery speed



Hobbyists

Use: Search by make / model / year
Concern: Getting the right part



DIY Repair

Use: Parts by pictures / tutorials
Concern: Price



Resellers

May be a group — but client may choose *not* to target them

Prioritizing Groups

Rank groups by importance to the client. The designer doesn't decide — the designer works with the client to build and prioritize user profiles. Some groups may be intentionally excluded.

Requirements

What are Requirements?

Requirements are the **elements that must be present** in our final product to achieve success. They come from multiple stakeholders — not just the designer.

Designer

Site must have correct utility and be usable.

Site Owner (Customer)

Site must allow the sale of products.

End User

Site must allow me to select products by browsing by color, style, product name, or size.

⚠ Key Point

Designers do not determine specific requirements. They *gather* requirements specified by clients and end users. The designer facilitates the process.

Requirements Elicitation

Key Terms

- ▶ **Requirements Analysis** — learning and understanding user needs
- ▶ **Requirements Elicitation** — discovering requirements by communicating with users
- ▶ **User Task Analysis** — determining how requirements should be implemented

Elicitation Techniques

- ▶ **Interviews & questionnaires** — specific, detailed questions
- ▶ **Brainstorming sessions** — group discussion of ideas
- ▶ **Storyboards** — diagram alternatives, discuss pros/cons
- ▶ **Role playing** — walk through a specific requirement
- ▶ **Prototyping** — rough drafts for examination



Important Note

Users generally will not have thought through their requirements in advance. They must be guided in the process of stating them. A client saying "I want an online store" leaves many design decisions open.

Types of Requirements

Specific Requirements

Tied to functional demands or design attributes:

- ▶ "Our company color is royal blue; we'd like that featured in the design"
- ▶ "We'd like site visitors to be able to chat with technical support"

General Requirements

Derived from knowledge of targeted users:

- ▶ Older audience → avoid flashy presentation
- ▶ High color contrast for readability
- ▶ Larger text for accessibility
- ▶ Mobile-first if users are on phones

Requirements vs. Mission Statement

Sometimes stated requirements conflict with the site's mission statement. The client/site owner must be made aware of conflicts so they can decide how they'll be handled. The designer facilitates, but doesn't decide.

Review

Lecture Quiz

10 questions — click "Reveal Answer" to check

Lecture Quiz — Questions 1–5

1 Which element do we use to create a link?

- A <href>
- B <a>
- C
- D <link>

Reveal Answer

2 How are links displayed by default?

- A Like this (black, no underline)
- B Like this (blue, underlined)
- C Like this (red, no underline)
- D Like this (green, italic)

Reveal Answer

Lecture Quiz — Questions 3–5

3 Which two attributes are required for an `` element?

- A `src` & `alt`
- B `src` & `title`
- C `title` & `figcaption`
- D `alt` & `title`

[Reveal Answer](#)

4 Images can be used in place of text for hyperlinks.

- A True
- B False

[Reveal Answer](#)

5 Which image format is most likely for a logo with a transparent background?

- A `.bmp`
- B `.jpg`
- C `.svg`
- D `.gif` or `.png` (supports transparency)

[Reveal Answer](#)

Lecture Quiz — Questions 6–10

6 Designers determine specific requirements.

A True

B False

Reveal Answer

7 A site mission statement is usually identical to an organization's mission statement.

A True

B False

Reveal Answer

8 What technique can we use to understand our site's users?

A Persona

B Surveys / research

C Client's marketing / sales department

D All of the above

Reveal Answer

9 Which of the following is a user characteristic we would identify when building personas?

I All of the above (age, gender, education, occupation, hobbies, experience, goals, devices)

A Just age ranges

Reveal Answer

10 What does `target="_blank"` do?

A Makes a page launch in a new tab

B Overwrites the current page

C Nothing — it's deprecated

Reveal Answer