SSDP 1100 – Day 1

Course: Front-End Web Development Essentials

Instructor: Gabbie Bade

Agenda

- Course Overview
- VSCode
- Introduction to the Web
- Introduction to HTML
- Images on the Web
- Assignment #1

About Me

Gabbie Bade
gabbie_bade@bcit.ca

Marketing Strategy
& Project Management (2012-2023)
Front-End Web Developer (2020 to now)
Completed FWDP (2023)
Instructor (2023 to now)

Course Overview

All course materials will be provided via the Learning Hub: https://learn.bcit.ca/

This includes the course outline, grades, feedback, assignments, projects, etc.

At the start of each class, download the files for the day from the Learning Hub.

Assignments & Projects

There are **4 assignments** for the course. They are all due on the day they are given, and you will have the afternoon to complete them.

There is **1 project** for the course. I would recommend starting the Final Project *after* Day 2.

Don’t be shy, ask for help!

Class is better when you talk too. So, interrupt me…

- if you do not understand something,
- if I go too fast,
- if you have questions or want more information.

Don’t fall behind… speak up in class.

BCIT Student Supports

If you need help outside of a specific course, please use the resources BCIT has available:

Learning Commons (Tutoring) - https://www.bcit.ca/learning-commons/

Accessibility Services - https://www.bcit.ca/accessibility/

Counselling - https://www.bcit.ca/counselling/

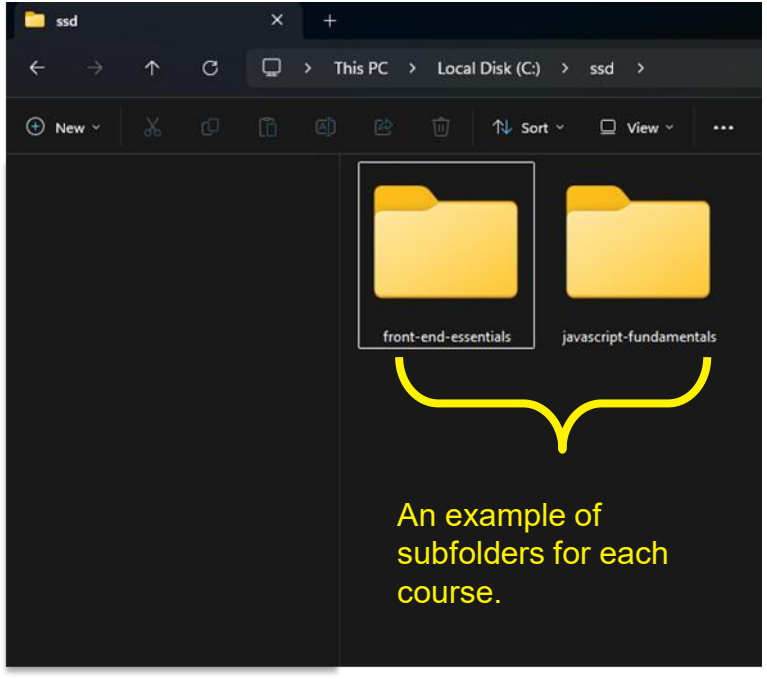
Student Health - https://www.bcit.ca/health-services/

Folder & File Structure

Create a folder on your computer with subfolders for each course.

Do not use cloud storage.

Get in the habit of using lowercases and no spaces when creating folders and files.



Show File Extensions

Seeing file extensions is extremely helpful so you should enable this on your computer.

Mac:
https://support.apple.com/en-ca/guide/mac-help/mchlp2304/mac

Windows:
https://www.thewindowsclub.com/show-file-extensions-in-windows

Folder & File Naming Rules

When naming your files and folders, follow these main rules:

- No spaces. Use **-** or **_** instead of spaces.
- Only use **lowercase** letters, no capital letters.
- Use descriptive names like **tree-with-building.jpg** instead of **IMG00314.jpg**.

Computer Software

- Browsers (in addition to the OS built-in browsers (Safari (Mac), Edge (Windows))
  - Chrome
  - Firefox
- Text Editor
  - Visual Studio Code (primarily for web dev languages or Python)
  - *You are free to use other text editors like Atom or Sublime Text, I will be using VSCode in class.*

VSCode Extensions

- There are a lot of community-created extensions to enhance VSCode:
  - Live Server or Live Server (Fiver Server) – *for quickly previewing your HTML on a browser, automatically updates preview on save*
  - ESLint – *can help you find and fix errors in your code based on patterns*
  - Prettier – *helps with code formatting*
  - Image Preview
  - Live Preview – *opens a preview within VSCode*
  - px to rem

Use of AI

- Just like Google, Stack Overflow etc. treat AI tools like ChatGPT, MDN Plus, and GitHub Copilot as an extra resource. **It is not the ultimate solution to everything.**
- For your foundational courses like this one, please try **not to use AI-generated code.**
- AI can be a great tool to check syntax, definitions, and validate for errors.
- Make sure that **you understand the code** that AI generates—master your foundations, refer to what you learned, correct the syntax, and validate the code.
- You will have the opportunity and freedom to use AI in future courses.

Use of AI and other tools

Please add a comment in your code if you…

- Use AI to generate more than a few lines.
- Got a code snippet from sites like StackOverflow or CodePen.
- Use code we learned in class.
- Use a mockup you found online for your design.

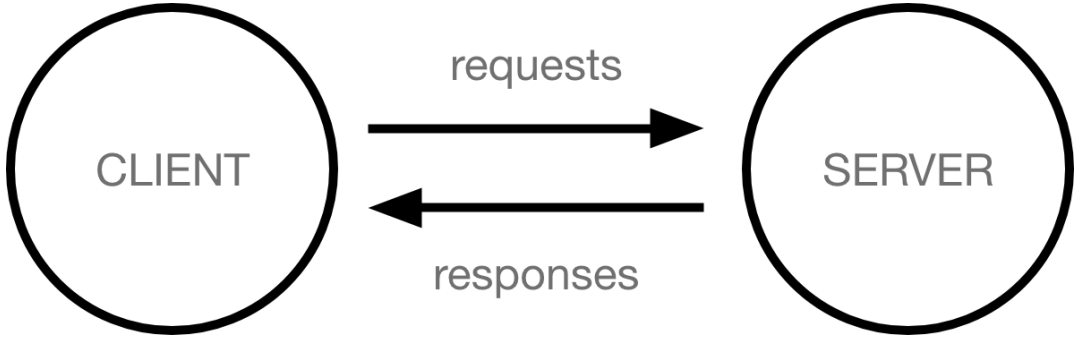
Please add a link in your comment.

Introduction to the Web

Introduction to the Web

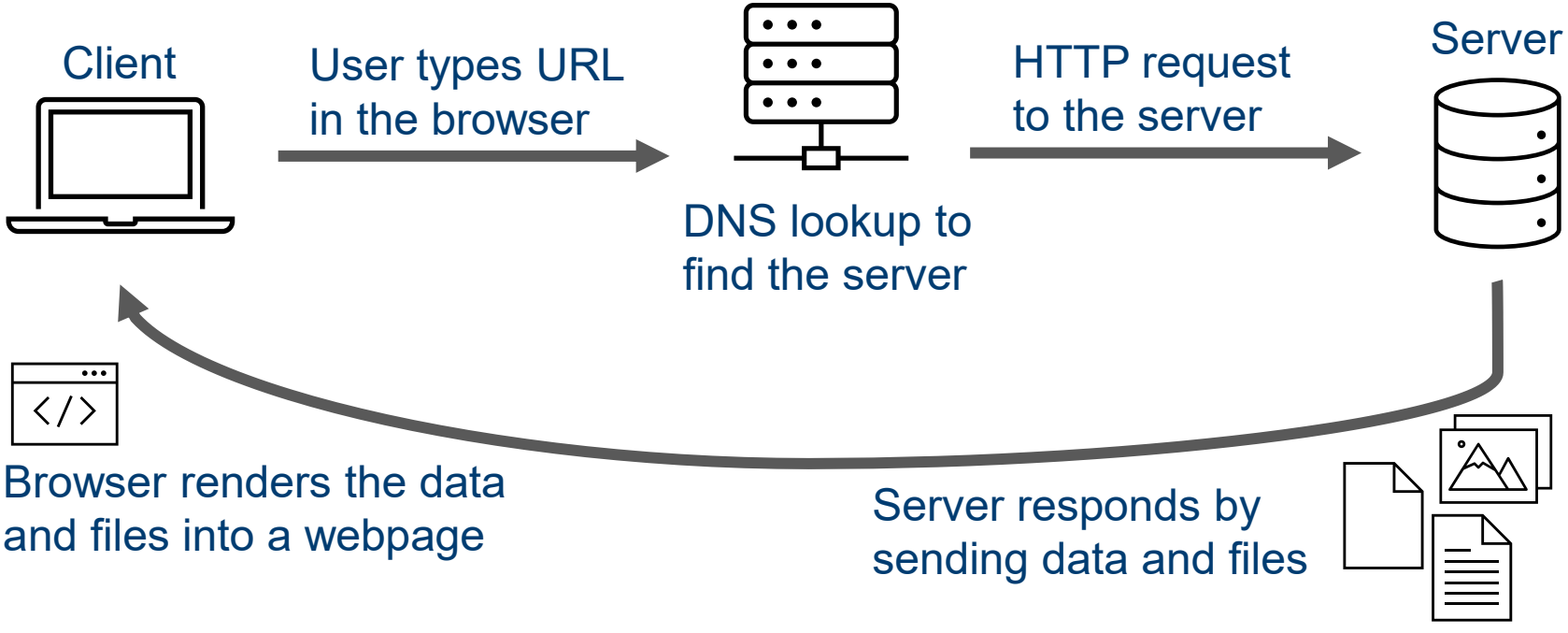
Client = internet connected device + browser

Server = computer that stores websites



https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/How_the_Web_works

How it all works



Domain Names

Domain Names

- Domain names are the human readable name you type into the browser.
  - Example: https://google.com/
- They are mapped to IP addresses.
  - Example: 74.125.129.94
- Central Domain Name Servers (DNS) contain tables of domain names and their corresponding IP addresses.

Domains


Second Level Subdomain
developer.mozilla.org
Third Level Subdomain
Top Level Domain

Top Level Domains (TLD)

`.com` `.ca` `.net` `.org` `.edu` `.info`

`.biz` `.gov` `.tv` `.co` `.tech` `.ai`

- Some TLDs are restricted. For example, you must have an address in Canada to register a **.ca** domain.
- Not every company sells all the TLDs.
- Every TLD: https://data.iana.org/TLD/tlds-alpha-by-domain.txt

Choosing a Domain Registrar

Consider the following:

- Privacy – By default your name, address, email, phone number is publicly associated with the domain.
  - Lookup this information at https://www.who.is/
  - All **.ca** domains include WHOIS privacy by default. Choose a domain registrar that includes it in the base price otherwise.
- Price – The prices will be about $10 to $20 a year. Check the renewal price, not just the initial price.

Domain Registrars

These are some popular domain registrars:

- GoDaddy – https://ca.godaddy.com/domains
- Namecheap – https://www.namecheap.com/
- Porkbun – https://porkbun.com/products/domains
- Hover – https://www.hover.com/
- NameSilo – https://www.namesilo.com/

Price & Feature Comparisons

These sites will compare prices and features across several domain registrars:

https://www.domcomp.com/

https://tld-list.com/

Hosting Provider Options

In addition to a domain, you also need a hosting account to put your files online.

These are examples of hosting providers:

https://www.siteground.com/web-hosting.htm

https://www.hostinger.com

https://www.greengeeks.ca/web-hosting

https://whc.ca/canadian-web-hosting

https://www.bluehost.com/hosting/shared

Introduction to HTML

Introduction to HTML

- HTML stands for Hypertext Markup Language.
- HTML is markup language that provides the *structure* of a webpage.
- It includes elements used to label pieces of content.
  For example: a paragraph, a list, a table, etc.

https://www.w3.org/standards/webdesign/htmlcss

HTML Files

An HTML file is a text file with the extension **.html** that includes HTML code.

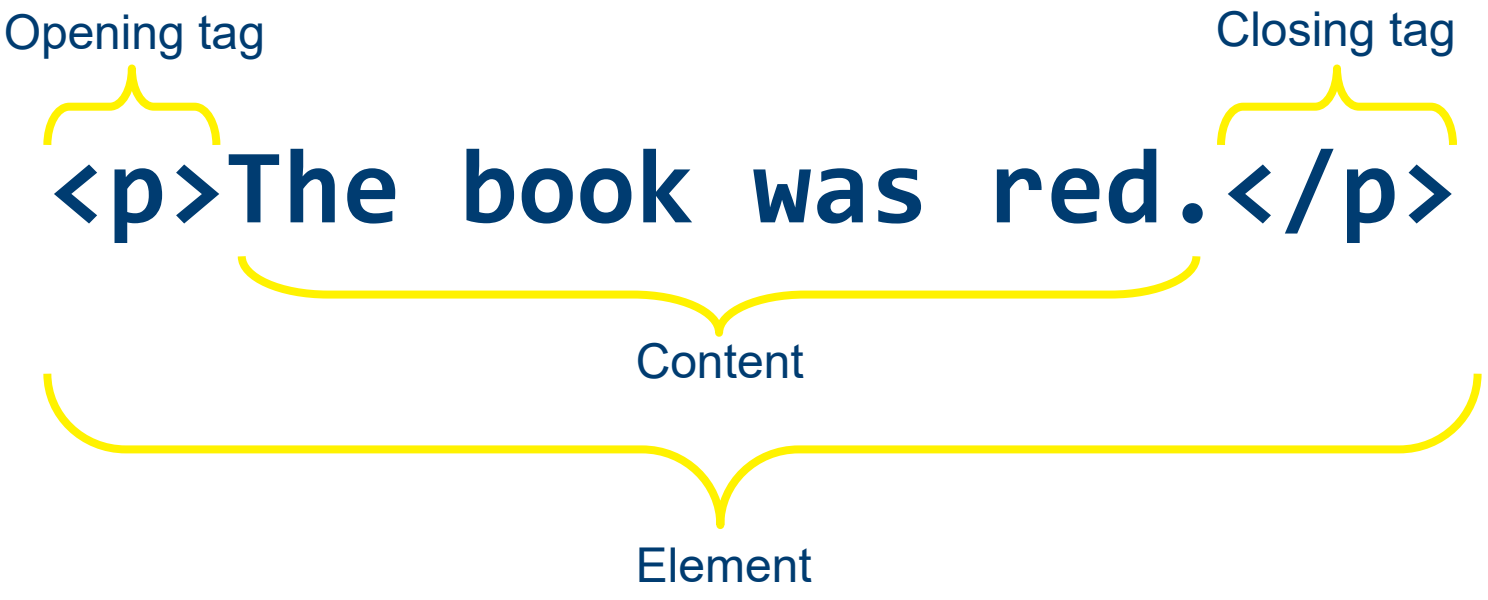
HTML + CSS

While HTML provides the basic structure of a webpage…
…CSS (Cascading Style Sheets) provides the visual design.

But before we use CSS, we need to understand how to write proper HTML.

Don't worry about how the webpage looks in the browser yet!

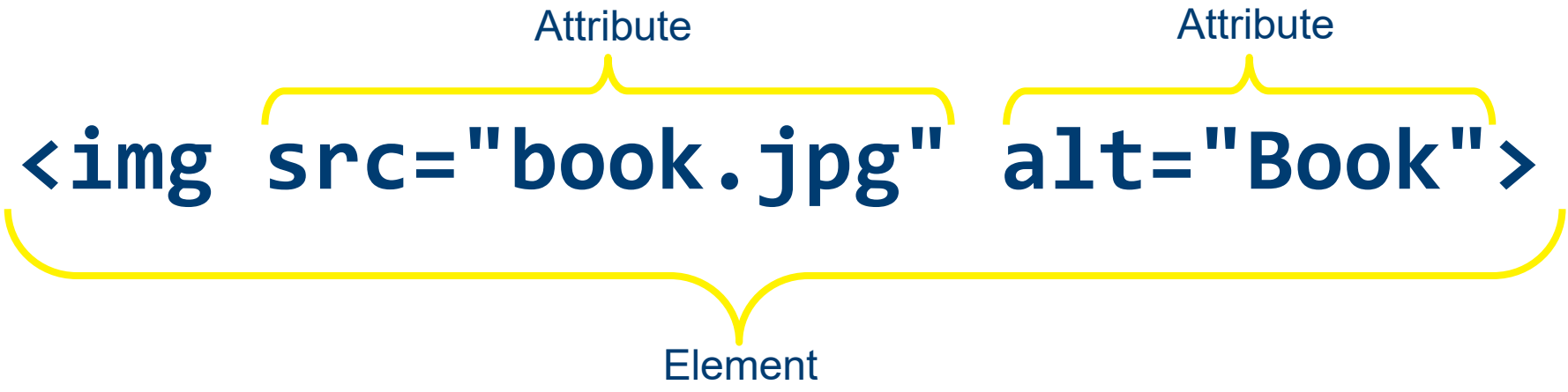
HTML Element Structure



HTML Element Structure

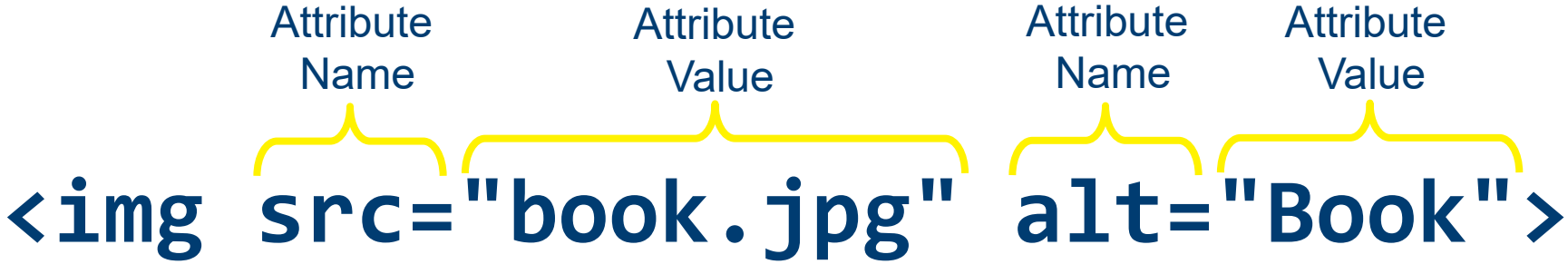
Some HTML elements, like images, do not require a closing tag.

This example also shows HTML attributes.



HTML Element Structure

HTML attributes should have a **name** followed by an equals sign and then a **value** wrapped in double *or* single quotes.



Double or Single Quotes

Which of the following is correct?

```
<img src="book.jpg" alt="Book">
```

```
<img src='book.jpg' alt='Book'>
```

Both! As long as you are consistent, you can use *either* double quotes *or* single quotes in your attribute values.

HTML Document

The code to the right is the basic structure of an HTML document.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Page Title</title>
  </head>
  <body>
    <h1>Page Title</h1>
  </body>
</html>
```

https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started

Doctype

Every HTML file is required to include the doctype at the top.

This is how you tell the browser which version of HTML the document uses. We all use the current version, HTML5, so this is essentially a relic of the past.

Always set your doctype to the following:

```
<!doctype html>
```

HTML Element

The HTML element wraps all of your code, except the doctype. It is also called the root element.

It is best to specify the primary language of the webpage for screen readers here as well.

```
<html lang="en"></html>
```

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/html

https://www.w3schools.com/tags/ref_language_codes.asp

Head Element

The Head element is used for adding meta data about the webpage but **not** the content of the webpage.

```
<head>
  <meta charset="utf-8">
  <title>Page Title</title>
</head>
```

Head Element

You will commonly use the Head element for the following:

- Define the character set,
- Set responsive design meta tags,
- Specify search engine titles and descriptions,
- Include favicons,
- Include CSS files and JavaScript files,
- and more…

Body Element

The body element contains **all** the content of the webpage that appears below the address bar in the browser.

The example below currently contains only a single heading element as the content of the webpage.

```
<body>
  <h1>Page Title</h1>
</body>
```

Semantic HTML

It is important that your HTML make sense *without* any CSS.

We do so by using **semantic HTML.**

Semantic HTML provides meaning to the content of our webpages.

It is the difference between saying “a thing” versus saying “a numbered list”.

Benefits of Semantic HTML

Some of the reasons we write semantic HTML are:

- Search Engine Optimization (SEO)
- Accessibility, especially for screen readers
- For developers and designers to understand what content will populate the webpage

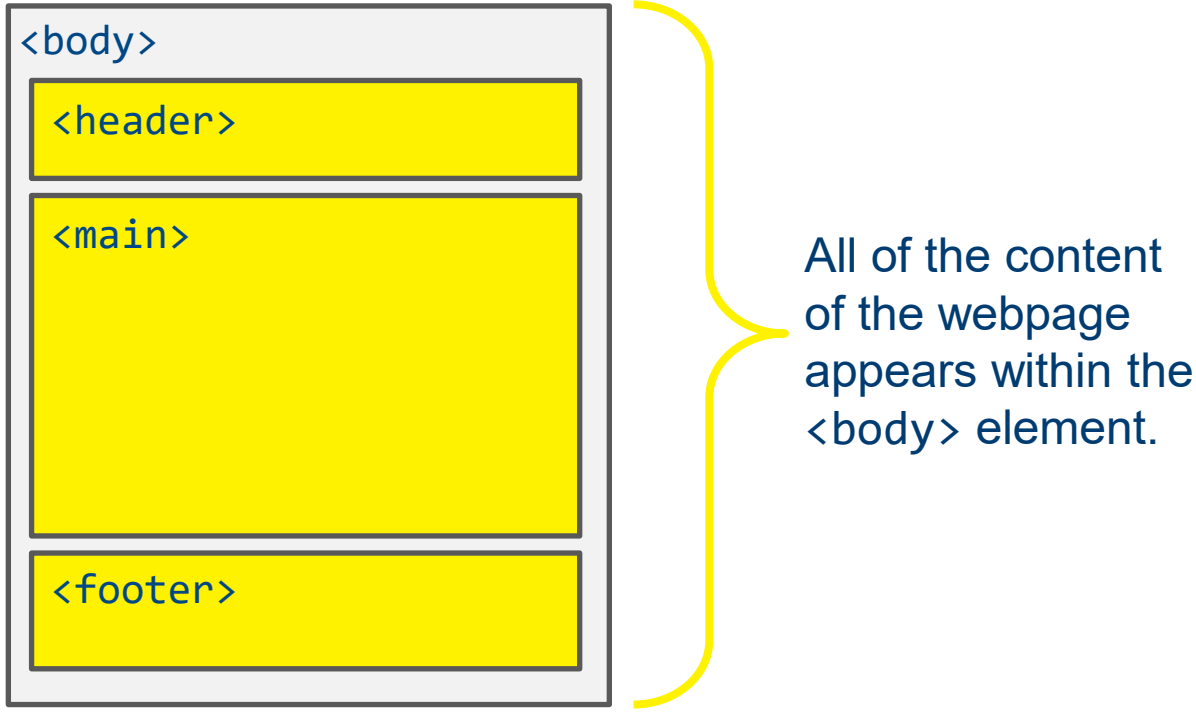
https://developer.mozilla.org/en-US/docs/Glossary/Semantics

Writing Semantic HTML

Let’s create a basic, semantic structure to our new webpage using three HTML elements.

https://developer.mozilla.org/en-US/docs/Web/HTML/Element#Content_sectioning

Basic Semantic Structure



Header Element

The `<header>` element is typically used at the top of the `<body>` element.

It will often contain elements like a heading, a logo and navigation.

It can also be used within an `<article>` element (more on those later).

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/header

Main Element

The `<main>` element defines the main content area of the webpage within the `<body>` element.

Unlike the `<header>` and `<footer>` elements, the content of `<main>` should be unique to the webpage.

The `<main>` element should only be used **once** on a webpage.

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/main

Footer Element

The `<footer>` element is typically used at the bottom of the `<body>` element.

It will often contain elements like copyright information, the author information, and navigation.

It can also be used within an `<article>` element (more on those later).

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/footer

Whitespace in HTML

By default, browsers ignore more than a single space in your HTML code.

Keep this in mind when writing your code and make it readable by using indents with the tab key to show how elements are nested underneath one another.

Indent Nested Elements

This is easier to read…

```
<section>
    <h2>Section Title</h2>
    <p>This is some text.</p>
    <ul>
        <li>List Item 1</li>
        <li>List Item 2</li>
        <li>List Item 3</li>
    </ul>
</section>
```

…than this is.

```
<section>
<h2>Section Title</h2>
<p>This is some text.</p>
<ul>
<li>List Item 1</li>
<li>List Item 2</li>
<li>List Item 3</li>
</ul>
</section>
```

Adding Content

Now that we have a basic structure, we can add content to our webpage using some common HTML elements.

When adding placeholder content, we often use Lorem Ipsum for text. You can use the Emmet shortcut or copy it from online generators:

https://lipsum.com/

https://www.shopify.ca/partners/blog/79940998-15-funny-lorem-ipsum-generators-to-shake-up-your-design-mockups

Heading Elements

There are six heading elements:

```
<h1> <h2> <h3> <h4> <h5> <h6>
```

The `<h1>` element is used only **once** per webpage and serves as the heading of the webpage. This is a best practice because of SEO and screen readers.

Do not skip heading numbers when creating subheadings.

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Heading_Elements

Paragraph Element

The <p> tag is used to wrap blocks of text to form the paragraph element.

It can contain a single sentence, multiple sentences, images, stylized text, form fields, links, etc.

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p

Section Element

The `<section>` tag is used to wrap a section of the webpage and typically includes a heading at the top.

It is used to define the structure of your webpage.

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/section

List Elements

There are three types of list elements in HTML:

- Unordered lists `<ul></ul>`
- Ordered lists `<ol></ol>`
- Description lists `<dl></dl>`

Unordered Lists

Displays an unordered list of items with bullet points. The style of the bullets can be changed (or removed) with CSS.

```
<ul>
    <li>List Item 1</li>
    <li>List Item 2</li>
    <li>List Item 3</li>
</ul>
```

- List Item 1
- List Item 2
- List Item 3

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul

Ordered Lists

Displays an ordered list of items with numbers. The style of the bullets can be changed (or removed) with CSS.

```
<ol>
    <li>List Item 1</li>
    <li>List Item 2</li>
    <li>List Item 3</li>
</ol>
```

1. List Item 1
2. List Item 2
3. List Item 3

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ol

Description Lists

Displays a list of terms and descriptions. Less used than the other two but useful if creating a glossary of terms.

```
<dl>
    <dt>Term 1</dt>
    <dd>Description 1</dd>
    <dt>Term 2</dt>
    <dd>Description 2</dd>
</dl>
```

Term 1
    Description 1
Term 2
    Description 2

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/dl

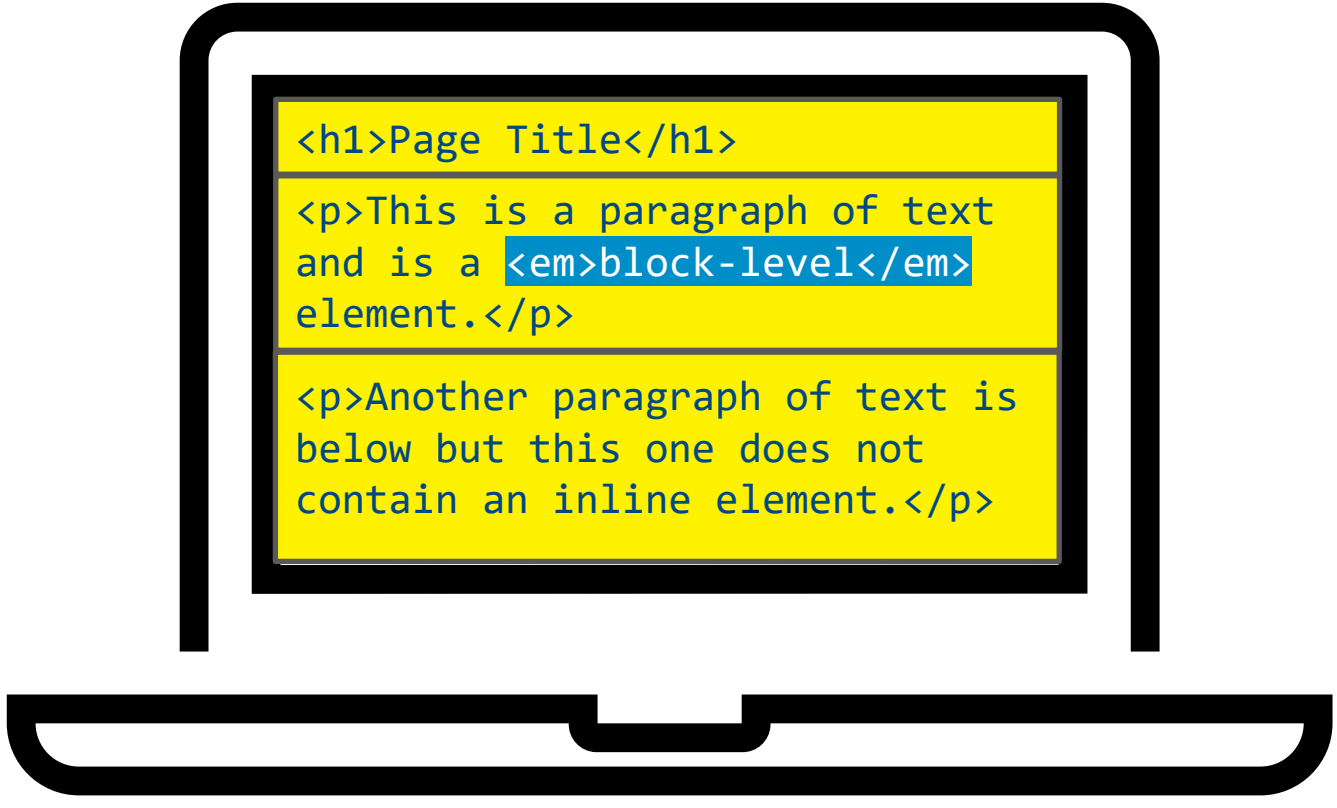
Block vs. Inline Elements

Most HTML elements fall into one of two categories: **block** or **inline**.

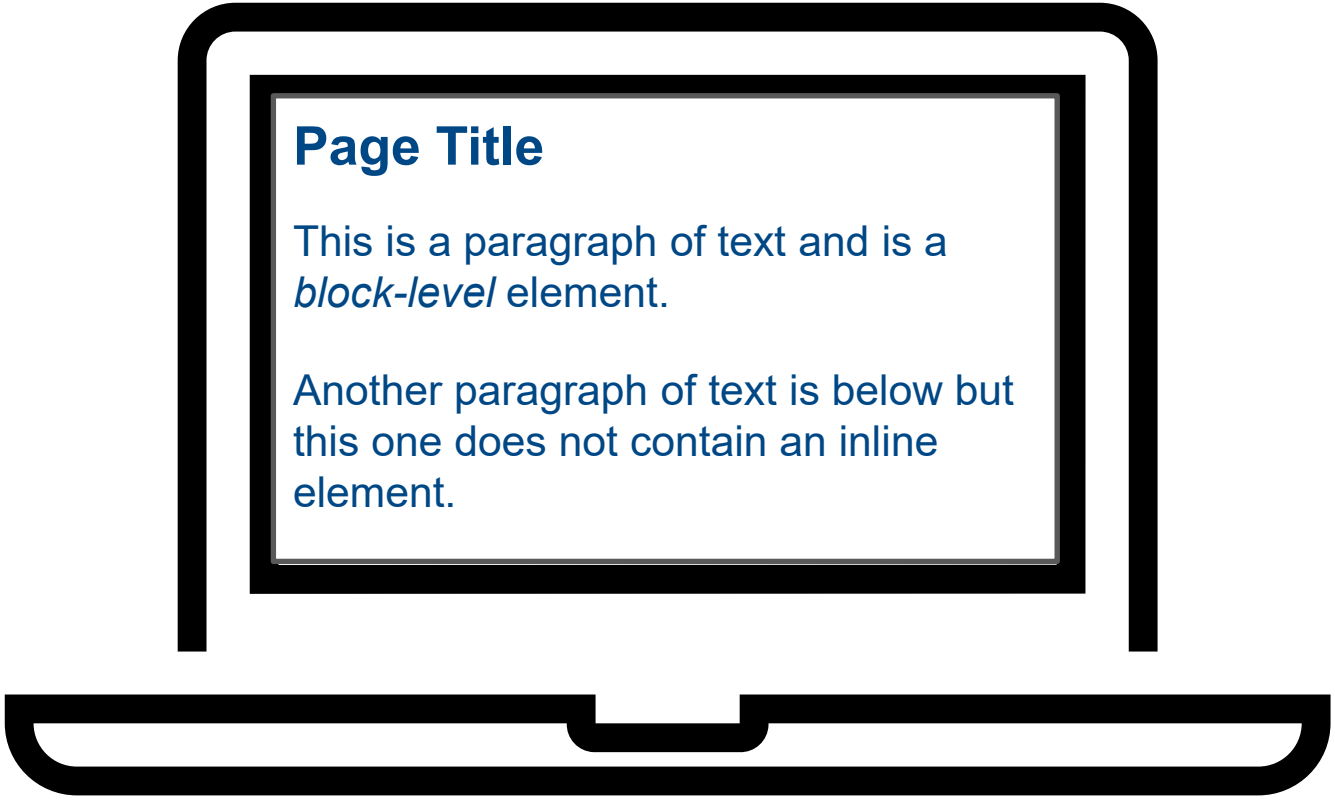
A block-level element will cause a line break and take up the entire width of the webpage. Headings, paragraphs and lists are all block-level elements.

An inline element is *generally* used inside of a block-level element. It does **not** cause a line break or take up the entire width of the webpage.

Block vs. Inline Elements



Block vs. Inline Elements



Semantic Text Elements

There are 30+ inline elements for text.

Some of them are very specific and niche. Others you will use somewhat frequently.

When in doubt, use the link below!

https://developer.mozilla.org/en-US/docs/Web/HTML/Element#Inline_text_semantics

Line Break Element

The
 element creates a line break within a block-level element like a paragraph.

Useful for poems, lyrics, and addresses.

```
<p>Ho! Tom Bombadil, Tom Bombadillo!<br>
By water, wood and hill, by the reed and willow,<br>
By fire, sun and moon, harken now and hear us!<br>
Come, Tom Bombadil, for our need is near us!</p>
```

Ho! Tom Bombadil, Tom Bombadillo!
By water, wood and hill, by the reed and willow,
By fire, sun and moon, harken now and hear us!
Come, Tom Bombadil, for our need is near us!

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/br

Anchor Element

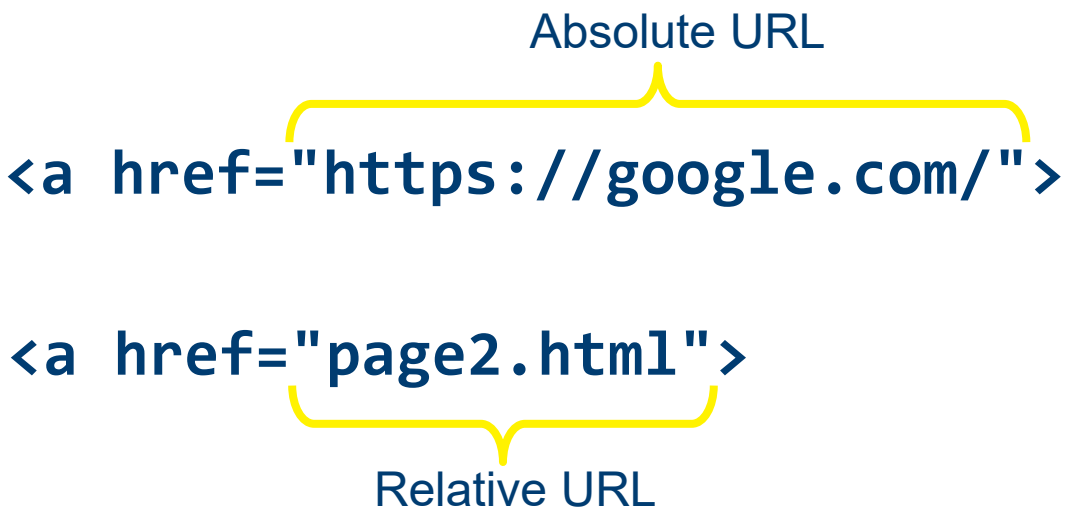
The <a> element is how we create hyperlinks. It is the HTML element that makes “the web” what it is.

You can link to other webpages, locations on your webpage, files, email addresses, phone numbers, etc.

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a

Anchor Element Structure

When linking to another webpage, there are two types of addresses: **absolute** URLs and **relative** URLs.



Absolute vs. Relative URLs

Absolute URLs are the full path to where the file exists on the internet.

Most often used for linking to external websites.

Relative URLs are the path to the file, *relative* to the current file. Moving either file to another folder will break the link.

Most often used for linking within your own website.

Opening Links in a New Tab

- **`target="_blank"`** is the attribute that allows you to have links open a new tab when clicked.
- Along with it, you need one more attribute: **`rel="noreferrer"`** prevents the new site from seeing where you were from/original tab (helps with privacy)

```
<a href="https://google.com" target="_blank"
                    rel="noreferrer">
```

Linking to an Element

By setting the id attribute on an element, we can link to that specific element of the page.

ID Value

```
<h2 id="section-about">
```

and the ID Value

```
<a href="#section-about">About Section</a>
```

ID Attribute

The ID attribute can be set on **any** HTML element but the value must only be used **once** per webpage.

For instance, if we set an id value of “top” on *two* elements, which would the browser link to?

The value of an ID also cannot contain whitespace.

https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes/id

Anchor Element

The content inside of the <a> tags is the clickable content.

If this is text, it should indicate where the link goes.

```
<p>Read more about us <a href="about.html">here</a>.</p>
```

```
<p>Read more <a href="about.html">about us</a>.</p>
```

Nav Element

The <nav> tag is used to wrap navigation links.

Anytime you have multiple links together, wrap them in the nav tag.

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/nav

HTML Comments

You can write comments in your HTML code that will not display on the webpage.

This is only visible when viewing the code and very helpful to explain to yourself and other developers what your code does.

Code Comments

Writing good comments in your code is extremely helpful for yourself and other developers.

A great article with some general rules:

https://stackoverflow.blog/2021/07/05/best-practices-for-writing-code-comments/

Images on the Web

Images on the Web

Traditionally, images on the web were in these formats:

JPG (or JPEG) for Photographs

PNG or GIF for Graphics

Images on the Web

In addition to JPG and PNG files for images, there are three others you are likely to use on the web as well:

- WebP
- AVIF
- SVG

https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Image_types

WebP

The **.webp** image format is a modern replacement for JPG and PNG files on the web. It can provide the same quality at a smaller file size.

It has wide support across browsers.

https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Image_types#webp_image

AVIF

The **.avif** image format is a modern replacement for JPG and PNG files on the web as well. It provides even smaller file sizes than .webp images at the same quality.

Browser support is still somewhat limited however, but it's getting there:
https://caniuse.com/avif

https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Image_types#avif_image

Scalable Vector Graphics

Scalable Vector Graphics (SVGs) are generally what we use for creating basic graphics and icons on the web.

They are a great replacement for PNG and GIF files.

https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Image_types#svg_scalable_vector_graphics

Optimizing Images for the Web

An image from a phone/camera is meant to be printed and must be optimized before using them on the web.

This means… determining the optimal dimensions, the best file format, and saving it at the smallest file size possible without giving up quality.

Smaller file sizes = faster websites

Image Dimensions

Images should generally not be displayed larger than their dimensions (width and height in pixels).

For example: using CSS to display an image larger than the dimensions it was saved at will make it blurry and pixelated.

It is fine to shrink an image with CSS though.

Image File Sizes

Every image file should be **below 1MB** if it is being used on the web and usually much lower than 1MB.

Even large, full width photographs can be under 500KB without degrading the image quality when viewed on a screen.

In your design course you will learn more about optimizing images for the web.

Animated GIFs

Instead of using animated GIFs like the one below, you should use video files since the quality is better and file size is lower.

Image Element

This is the basic structure of the HTML image element.

Source: The name and location of the image file.

Alternative Text: Used by screen readers or if the image does not load.

```
<img src="book.jpg" alt="Book">
```

Optimizing Images

- Download modern file formats
- Download the size you need
- Use tools

Helpful Tools

- Web Tools
  - TinyPNG
  - TinyJPG
  - Compressor

Helpful Tools

- Softwares and CLIs
  - Sharp – *using Node.js*
  - ImageMagick – *software and also CLI*
  - Imagine – *software*

File Paths

Relative vs Absolute Paths

A relative path points to a file relative to the current file.

```
<img src="images/book.jpg" alt="Book">
```

An absolute path is the full URL to a file beginning with **http**.

```
<img src="https://example.com/images/book.jpg" alt="Book">
```

When to use each

Use an absolute path to load an external resource.

Use a relative path to load a resource in your files.

Note: This rule will change slightly when working with content management systems later in the program.

File Path Syntax

These two lines mean the exact same thing:

```
<img src="images/book.jpg" alt="Book">

<img src="./images/book.jpg" alt="Book">
```

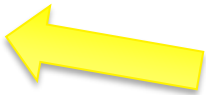
The second one is using an older syntax that is not necessary for HTML, CSS, or JavaScript.

File Path Syntax

Beginning a path with a slash means “go to the root folder on the server”. These do **NOT** point to the same place:

```
<img src="images/book.jpg" alt="Book">
```

```
<img src="/images/book.jpg" alt="Book">
```



Avoid this. You will almost never need to do this.

“Live Server” in VS Code will act like these are the same but it is wrong and your code will likely break on a real server.

File Path Syntax

This path says “go up **one** folder to find the images folder, inside of the images folder load book.jpg”:

```
<img src="../images/book.jpg" alt="Book">
```

This path says “go up **two** folders to find the images folder, inside of the images folder load book.jpg”:

```
<img src="../../images/book.jpg" alt="Book">
```

More on File Paths

If that isn't completely clear yet, have a look at the links below for further explanation.

https://www.w3schools.com/html/html_filepaths.asp

https://css-tricks.com/quick-reminder-about-file-paths/

Assignment #1

Assignment #1

- Please refer to Assignment #1 in the Learning Hub.
- To submit the assignment, you can do **one** of these:
  - **Have me check your assignment in class before 4pm.**
  - Zip today's **folder** and upload it to the Learning Hub before next class.
- If you have questions or need guidance, just ask!

Resources

MDN Web Docs – HTML
https://developer.mozilla.org/en-US/docs/Web/HTML

World Wide Web Consortium
https://www.w3.org/

HTML Specifications
https://html.spec.whatwg.org/multipage/

HTML Specifications
https://validator.w3.org/

Video Tutorials

LinkedIn Learning – HTML5: Structure, Syntax, Semantics
https://www.linkedin.com/learning/crafting-meaningful-html/craft-meaningful-html

LinkedIn Learning – HTML Essential Training
https://www.linkedin.com/learning/html-essential-training-4/

Resources (VS Code)

Emmet Cheat Sheet
https://docs.emmet.io/cheat-sheet/

Creating Snippets
https://code.visualstudio.com/docs/editor/userdefinedsnippets

Creating Custom Emmet Snippets In VS Code
https://www.smashingmagazine.com/2021/06/custom-emmet-snippets-vscode/

QUESTIONS & ANSWERS