

FWDP 1000 – Day 1

Course: Web Development 1

Instructor: Gabbie Bade

Agenda

- Course Overview
- Introduction to the Web
- Domain Names
- Introduction to HTML
- Set up a Git Repository
- Assignment #1

About Me

Gabbie Bade

gabbie_bade@bcit.ca

Marketing Strategy
& Project Management (2012-2023)

Front-End Web Developer (2020 to now)

FWD 32 (2023)

Teaching Assistant / Instructor (2023 to now)



About You

So I can personify the Teams thumbnail...

- Your name
- Any experience with design, development, coding, or any previous careers?
- Any hobbies or interests you want to share?

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 **.zip** folder for the day from the Learning Hub.

Assignments & Projects

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

There are two **projects** for the course. They are accessible now on the Learning Hub. I would recommend starting Project 1 *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/>

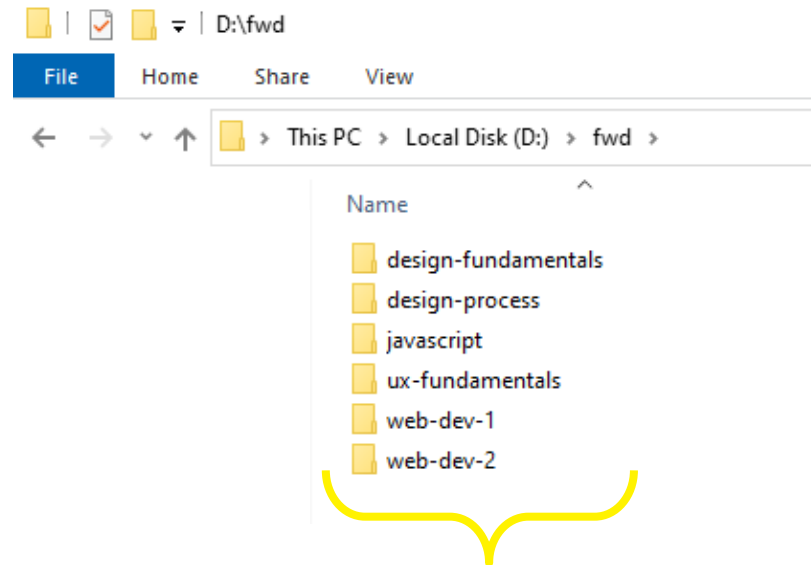
Student Health - <https://www.bcit.ca/health-services/>

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.



An example of subfolders for each course.

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>

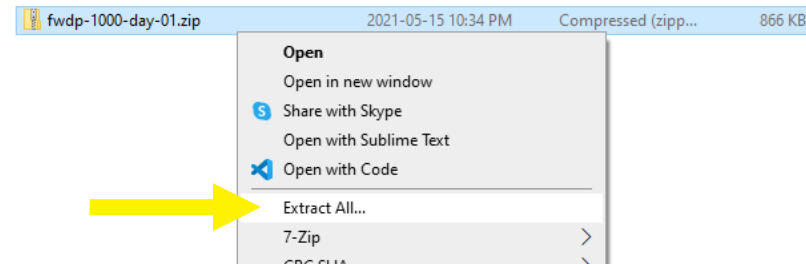
Folder & File Naming Rules

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**.

Windows - .zip Folders

On Windows, right click the **.zip** folder and choose “Extract All”.



You **must** use the Extract method.

Do **not** click into a .zip folder to copy/cut files out of it. This can cause corrupted files.

Mac - .zip Folders

On a Mac, double click the **.zip** folder to extract the folder.

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.**
- 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.
- AI can be a great tool to check syntax, definitions, and validate for errors.
- Please add a comment in your code if you use AI to generate more than a few lines.

Introduction to the Web

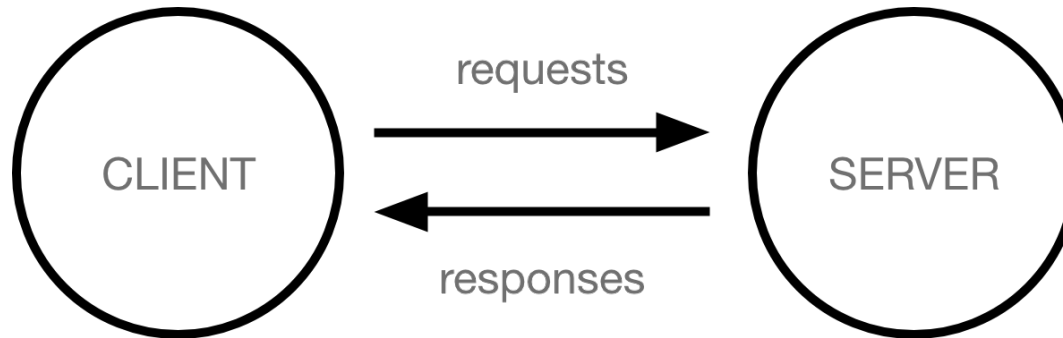
What is this website?

<http://info.cern.ch/hypertext/WWW/TheProject.html>

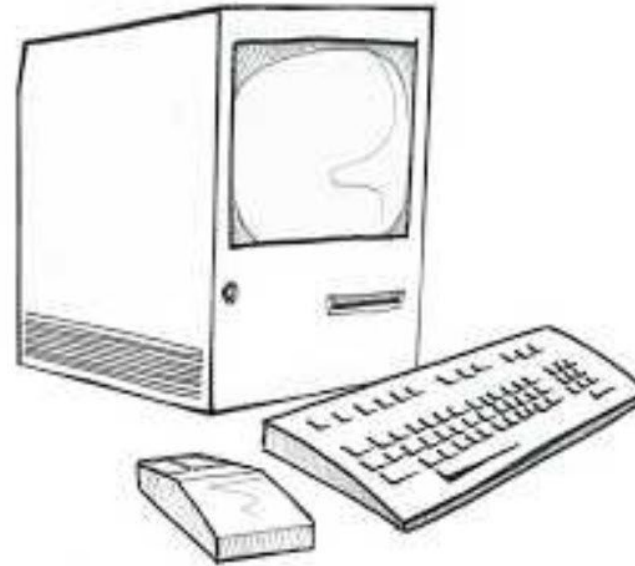
Introduction to the Web

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



https://www.youtube.com/watch?v=rZa_C5bCtUE

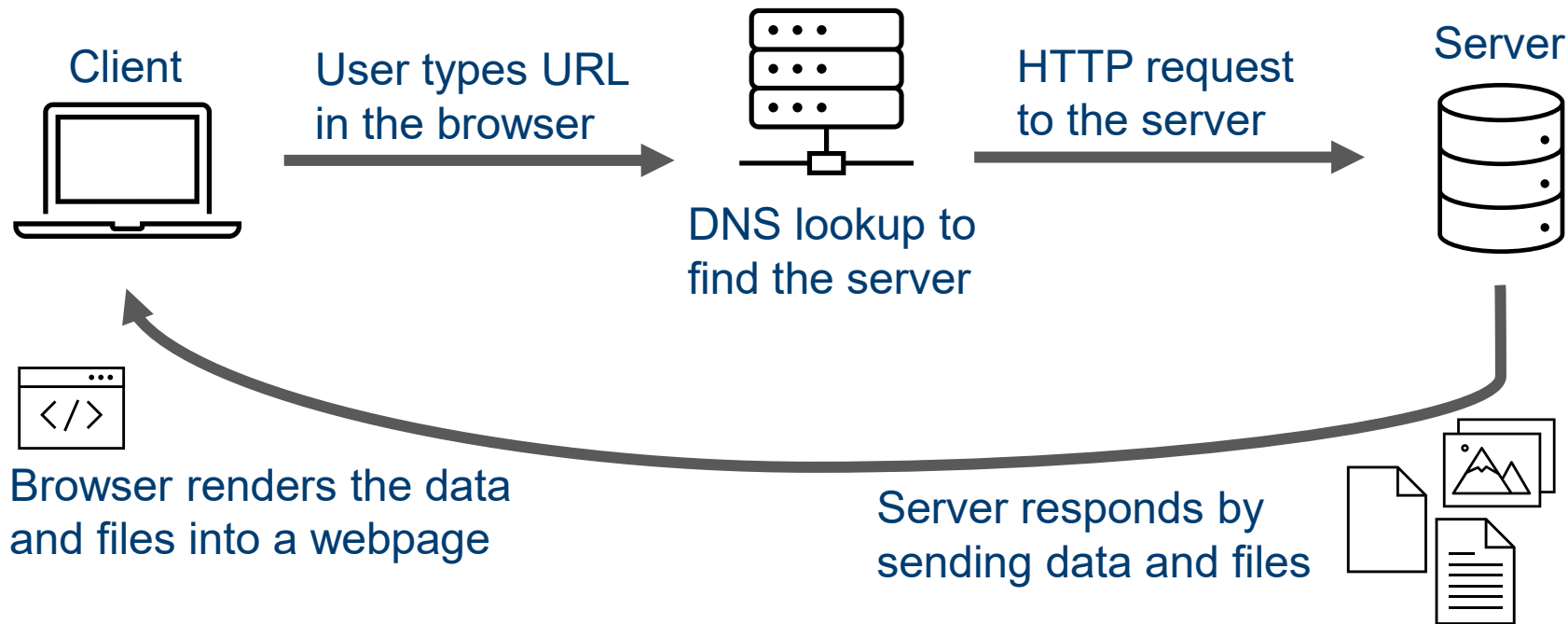
Terminology

- **TCP / IP:** Transmission Control Protocol and Internet Protocol
- **DNS:** Domain Name Servers
- **HTTP:** Hypertext Transfer Protocol
- **URL:** Uniform Resource Locator

How it all works

- “The web” is a network of connected servers.
- We access those servers by typing URLs into our browser.
- The DNS is used to find what server the URL points to.
- The browser sends an HTTP request to that server asking for the website and its files.
- The server responds and sends that data using TCP/IP.
- The browser then renders the website based on the data and files it receives.

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



mozilla.org

The diagram shows the domain 'mozilla.org' in a large, bold, dark blue font. A yellow bracket is positioned above the 'mozilla' part, and another yellow bracket is positioned below the '.org' part.

Top Level Domain

Domains



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>

Choose a Portfolio Domain Name

- If possible, try to get the **.com** ending
- For the Canadian market, **.ca** works well.
- There are profession specific ones as well:
 .design .web .tech .me .codes
- Easy to spell and remember, shorter the better, avoid dashes, try to create a consistent brand...

<https://hover.blog/choosing-domain-name-for-portfolio-website/>

Domain Registrars

There are **many** domain registrars.

You can compare prices and features: <https://tld-list.com/>

Many hosting companies offer free domains for the first year but it is usually cheaper to purchase from a domain registrar with a lower renewal price.

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/>

Pointing a Domain to a Server

We will cover this on Day 6 of the course.

For now, do some research and think about what you want your portfolio domain to be.

Have your domain purchased by Day 6 of Web Dev 1.

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 that can host all of your projects during the FWD program:

<https://www.siteground.com/web-hosting.htm> ← **My recommendation and what I will demo in class.**

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

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

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

Student Hosting Account

To get three months of free hosting with SiteGround, use this link and sign up using your **@my.bcit.ca** email address:

<https://www.siteground.com/studentsprogram/>

Note: It may be cheaper and more convenient to lock in a discounted rate with a hosting provider and not use this student account at all. It is entirely up to you. See the next slide.

When Student Hosting Ends

If you sign up for the free student hosting with SiteGround, it will expire after three months. At that point you can either...

1. Renew the SiteGround account at their regular rate.
2. Buy a new SiteGround account at a discounted rate.
3. Buy a hosting account with a different provider.

If you choose option 2 or 3, you will have to move your files and point your domain to that new account.

Choosing a Hosting Provider

Do some research to decide on a hosting provider and ask me if you have any questions about the differences between hosting providers.

If you want to use a hosting provider not listed, please check with me so I can verify it will work for all projects.

Purchase your hosting account by Day 6 of Web Dev 1.

Recap

True or False: Having a domain name is all you need to host a website on the internet.

After entering a domain name into a browser, what is the first thing the browser does?

- a) Make a request to that domain's server.
- b) Start downloading the files for that website.
- c) Find the server that domain points to online.
- d) Render the files downloaded from that website.

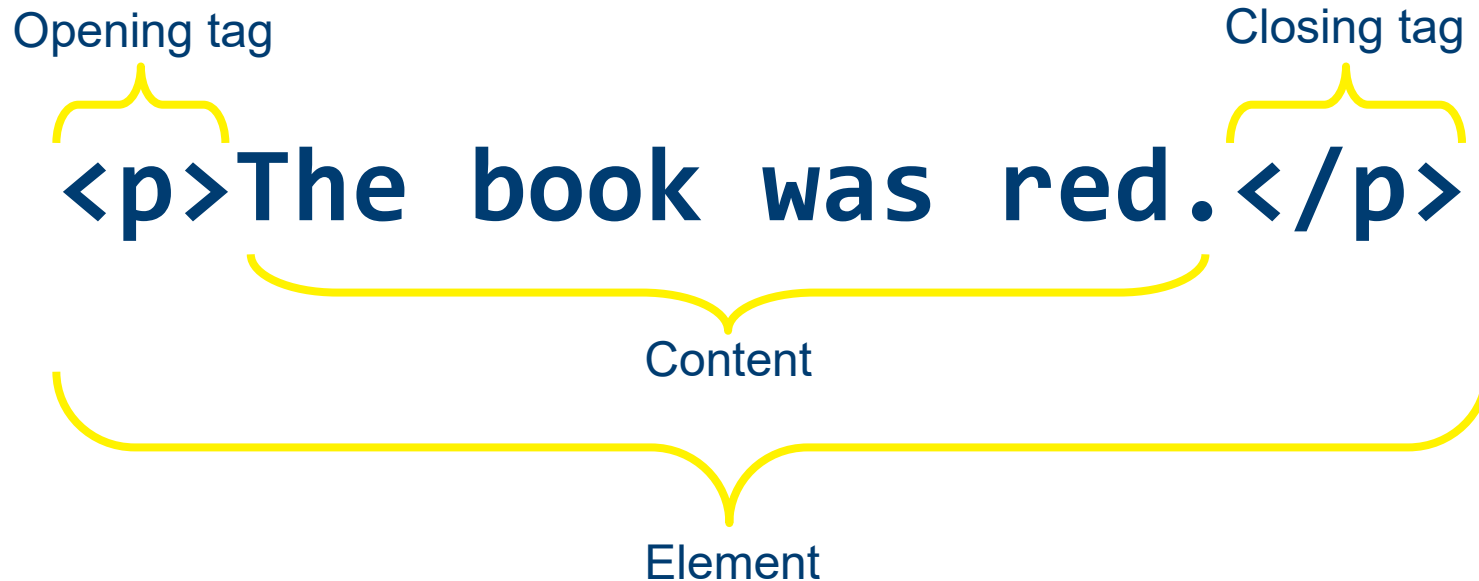
Introduction to HTML

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>

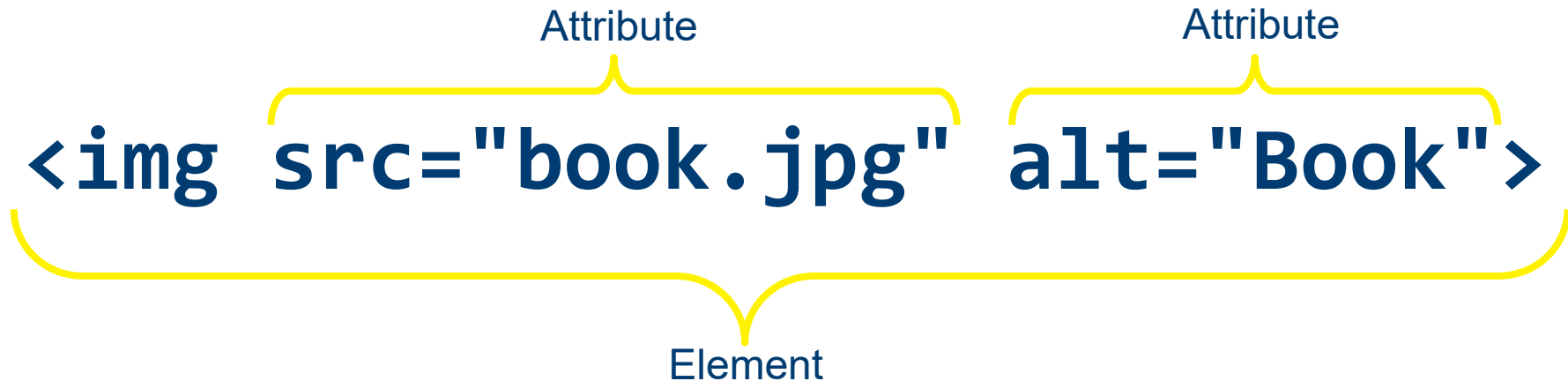
HTML Element Structure



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.

Diagram illustrating the structure of an HTML element with two attributes:

``

The diagram shows the following structure:

- Attribute Name:** `src`
- Attribute Value:** `"book.jpg"`
- Attribute Name:** `alt`
- Attribute Value:** `"Book"`

Double or Single Quotes

Which of the following is correct?

```

```

```
<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>
```

HTML Files

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

So let's make one!

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!

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

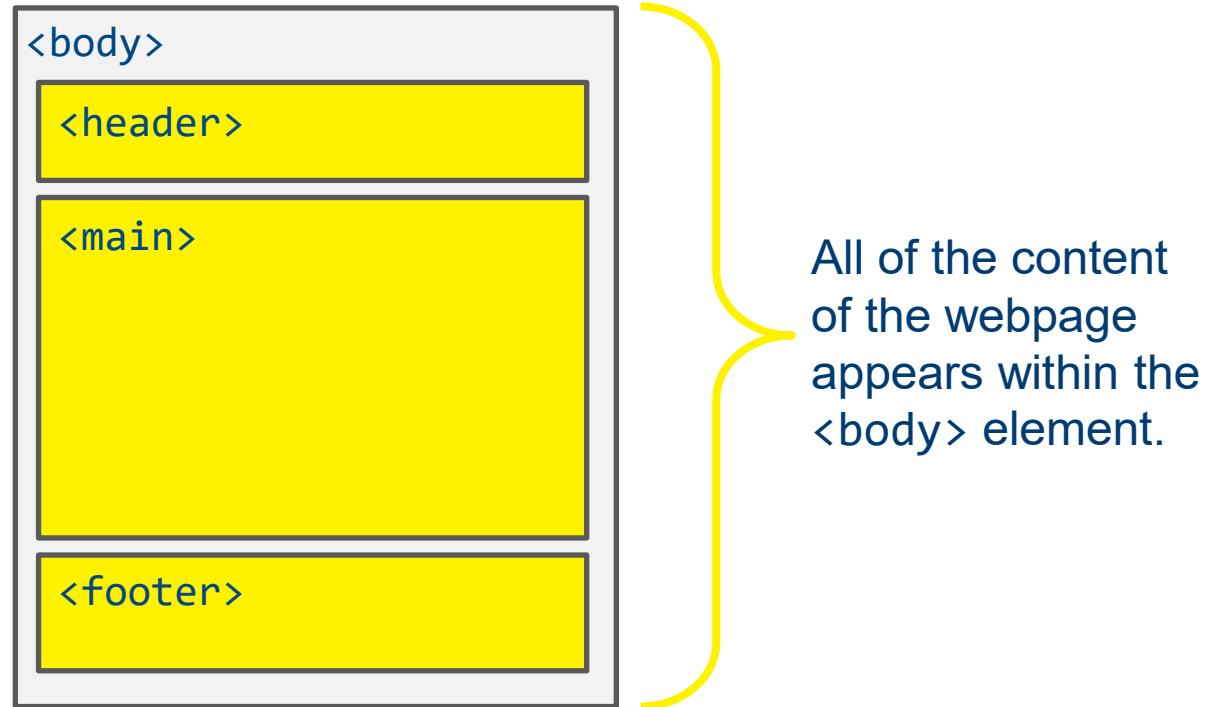
<https://developer.mozilla.org/en-US/docs/Glossary/Semantics>

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 ``
- Ordered lists ``
- 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>

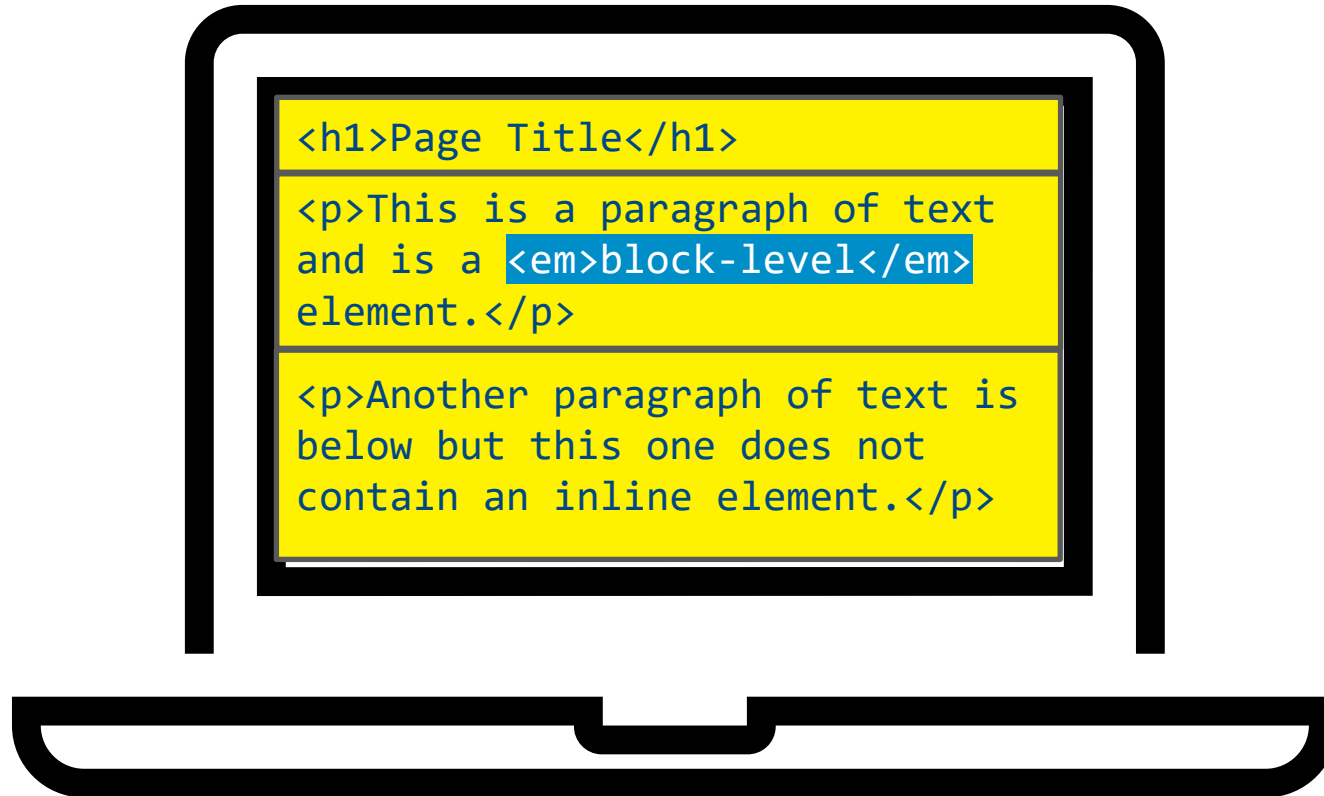
Block vs. Inline Elements

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

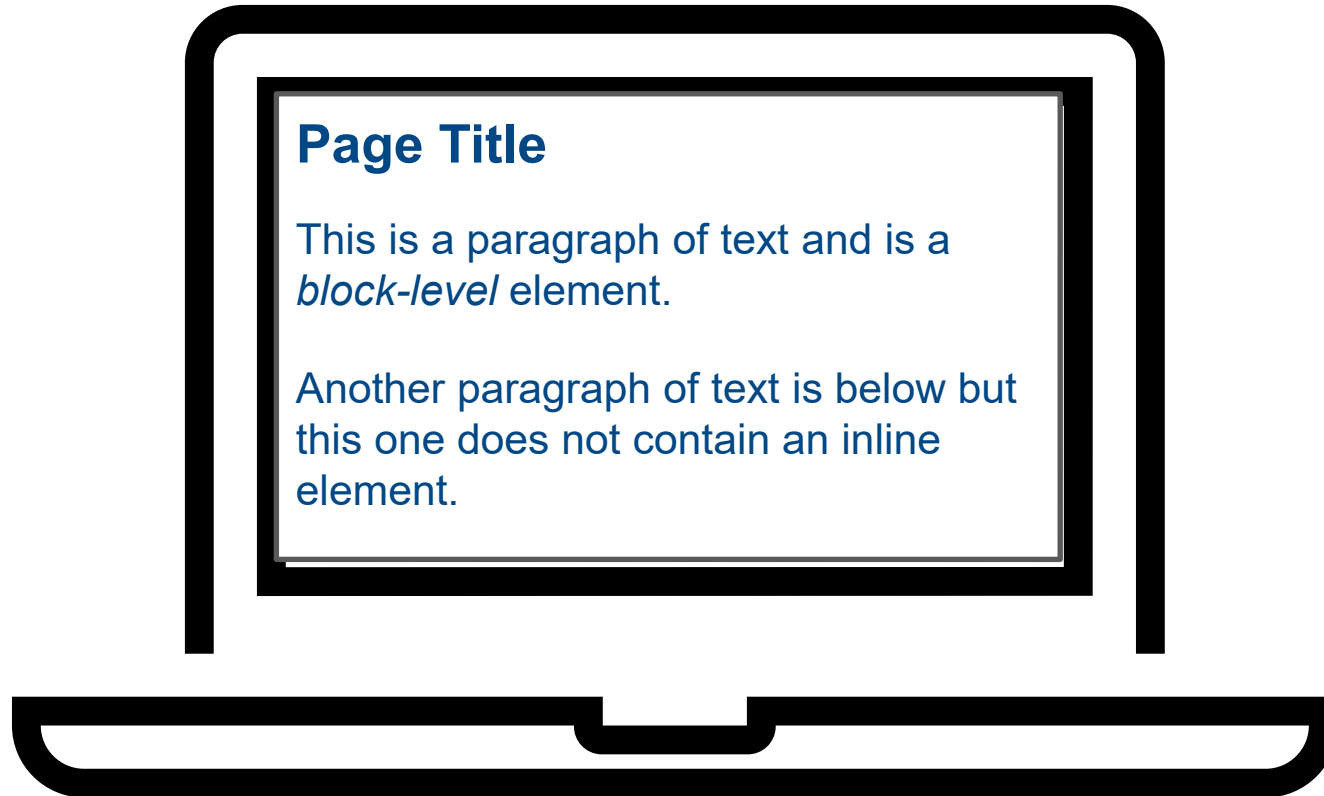
A block-level element will cause a line break and take up the entire width of the webpage. Headings, paragraphs and lists are all block-level elements.

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 URL

``

``

Relative URL

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.

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">
```

The diagram shows a yellow bracket above the text `id="section-about"` in the HTML tag `<h2 id="section-about">`. The bracket is labeled "ID Value" above it.

and the ID Value



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

The diagram shows a yellow bracket above the text `href="#section-about"` in the HTML tag `About Section`. The bracket is labeled "# and the ID Value" above it.

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 here.</p>`

 `<p>Read more about us.</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>

Start End

`<!-- I am an HTML comment -->`

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/>

Let's set up a Git repo!

Let's set up a Git repo!

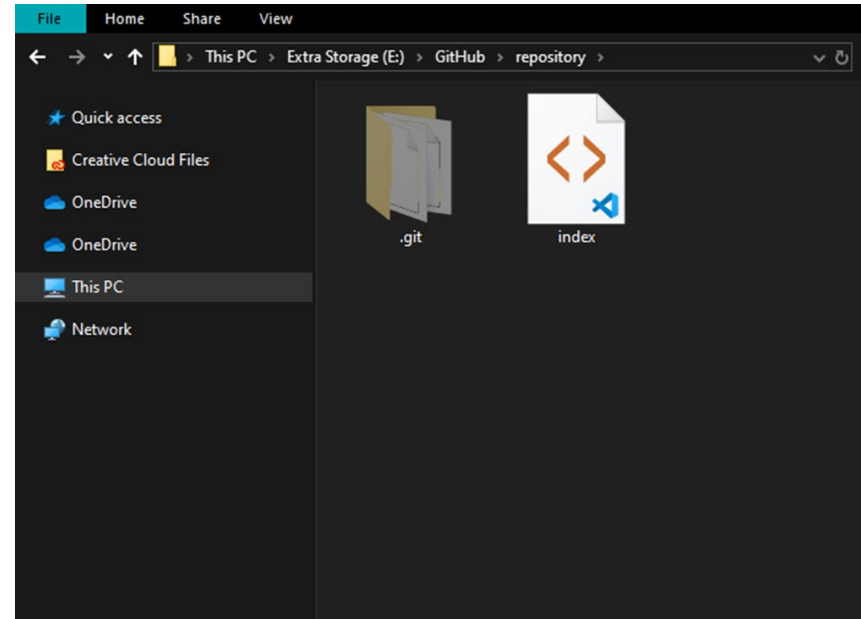
Version control is an integral aspect in development, simplistically, think of it as save points in a game.

A Repository or repo tracks all the changes you and your teammates do in your project.

Git VS GitHub

What we are doing now is a **Git repo** this is local to your device.

In the next few days, we'll use this same repo and add it on GitHub.



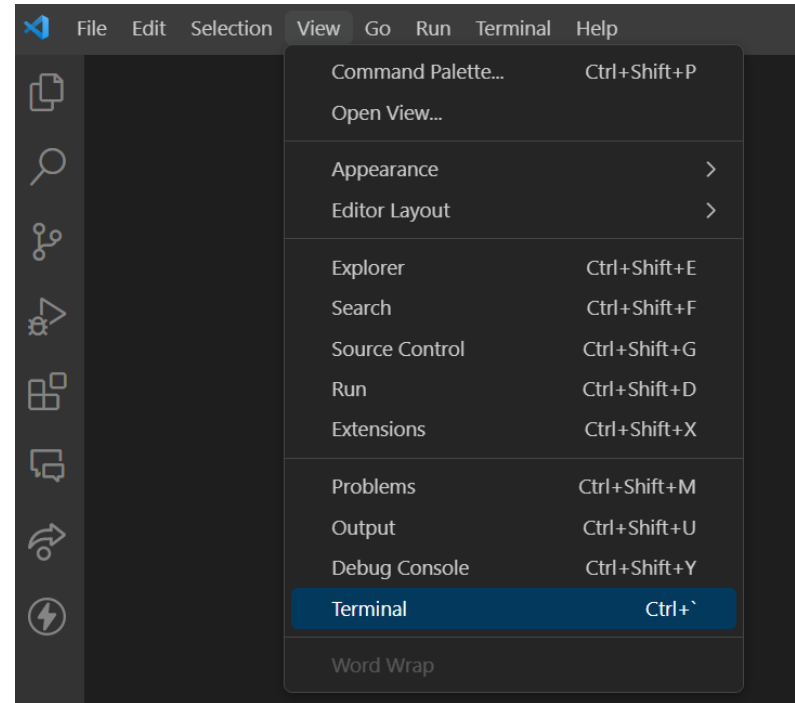
Let's set up a Git repo!

- Open the Terminal (Ctrl + `)
- Make sure you are in the right directory, if not type:

```
cd yourproject/path
```

- Once you are in the right directory, initialise your repository.

```
git init
```



Saving your changes

Whenever you make changes, on top of just saving your files, it is best to **add** and **commit** them to your repo as frequently as you could.

This ensures that you are tracking all the changes and saves your future self from having potential headaches.

```
git add .  
git commit -m "a detailed description of the change"
```

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!

Working Together

You can work on and submit this assignment alone, or in pairs.

This is a great way to get to know your classmates.

If you worked with someone and you are submitting on the Learning Hub (LH), make sure to add a comment regarding who you worked with since only one of you is required to submit.

Practice Git

This is not required but highly recommended.

Practice adding and committing until it becomes a habit.
Git/GitHub is incredibly helpful especially for more complex projects.

Practicing it even in simple projects make it less daunting!

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