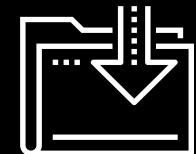


# Git'n Pro with HTML and CSS

Skills Bootcamp in Front-End Web Development

Lesson 1.2





**WELCOME**

# Where to Get Help

# Where to Get Help

---

- 01 Review in-class material (activities and slides).
- 02 Practice, practice, practice: work individually or in groups.
- 03 Watch the class videos again.
- 04 Attend office hours, which are held 45 minutes before and 30 minutes after class.
- 05 Attend one-on-one sessions with your Student Success Manager (SSM) (to be announced by your SSM).
- 06 Contact your Student Success Manager anytime!



A background composed of a dense pattern of white and light gray triangles, creating a low-poly or crystalline effect across the entire slide.

# Today's Class!

# Today's Objectives

---

By the end of the lesson today you will be able to:



Explain the importance of Git version control.



Utilize git for "adding, committing, and pushing" code to GitHub.



Link HTML/CSS using element selectors.

# Identify Your Learning Path

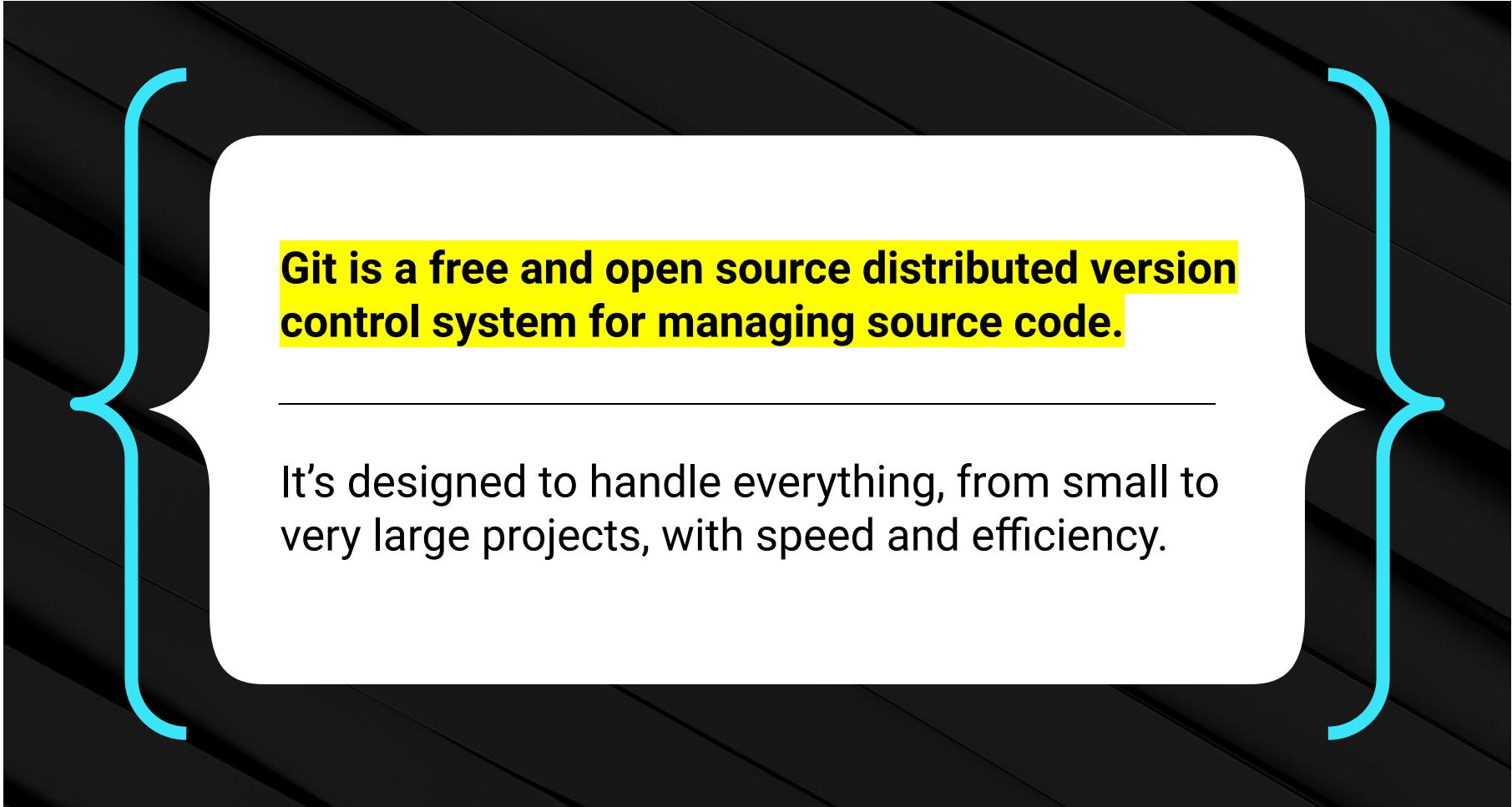
---



	New to HTML/CSS	Familiar with HTML/CSS
Level	Unfamiliar with HTML	Comfortable with HTML
Aim	Be able to write a complete, basic HTML document	Build up your skills
Goal	Understand the function of CSS and how it works with HTML	<ul style="list-style-type: none"><li>● Clear up any questions or confusion you have about HTML.</li><li>● Become knowledgeable about a wider range of HTML and CSS tags.</li><li>● Be able to selectively apply CSS to specific HTML elements.</li></ul>
Target	<b>Use Git and GitHub to upload code.</b>	



# What is Git?



**Git is a free and open source distributed version control system for managing source code.**

---

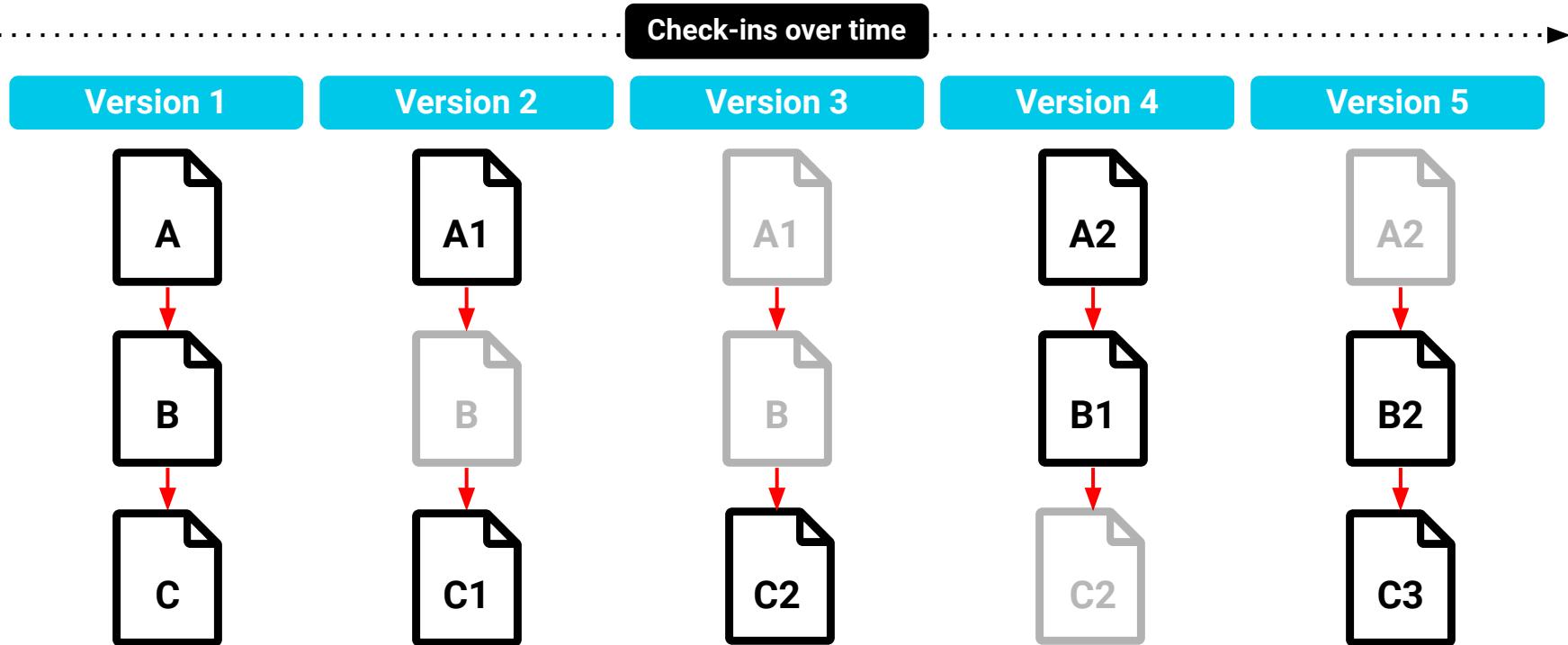
It's designed to handle everything, from small to very large projects, with speed and efficiency.



# What is version control?

# Introduction to Git

Version control is a system that allows you to manage changes over time.





# Why is version control important?

# Introduction to Git

Version control is important because:

01

Modern web development is highly collaborative.

02

Teams are often extremely large and spread out across the country or world.

03

Apps are sometimes made up of hundreds or even thousands of files.



# Introduction to Git

---

Version control allows programmers to code without being afraid of messing everything up.

It also allows developers to view histories of code changes and track issues.

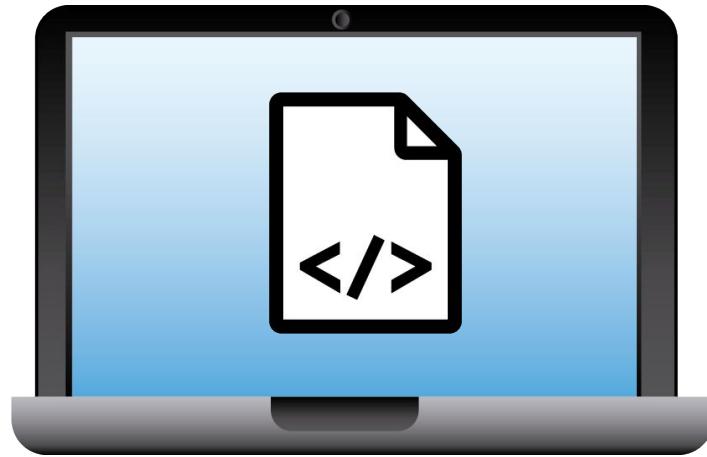




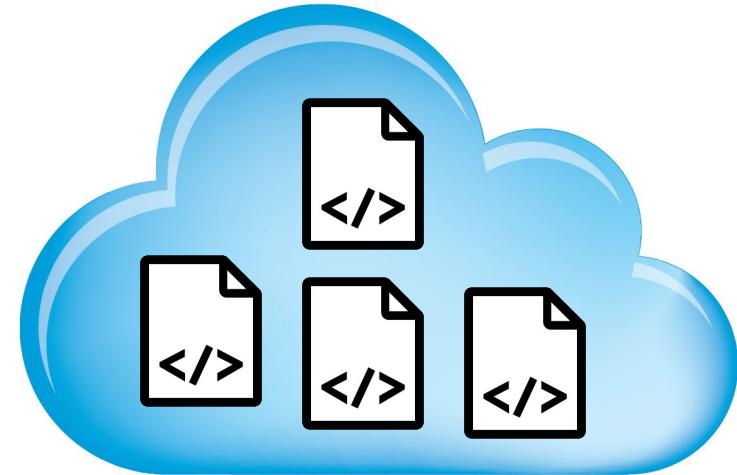
What is the difference between  
Git and GitHub?

# Git vs. GitHub

---



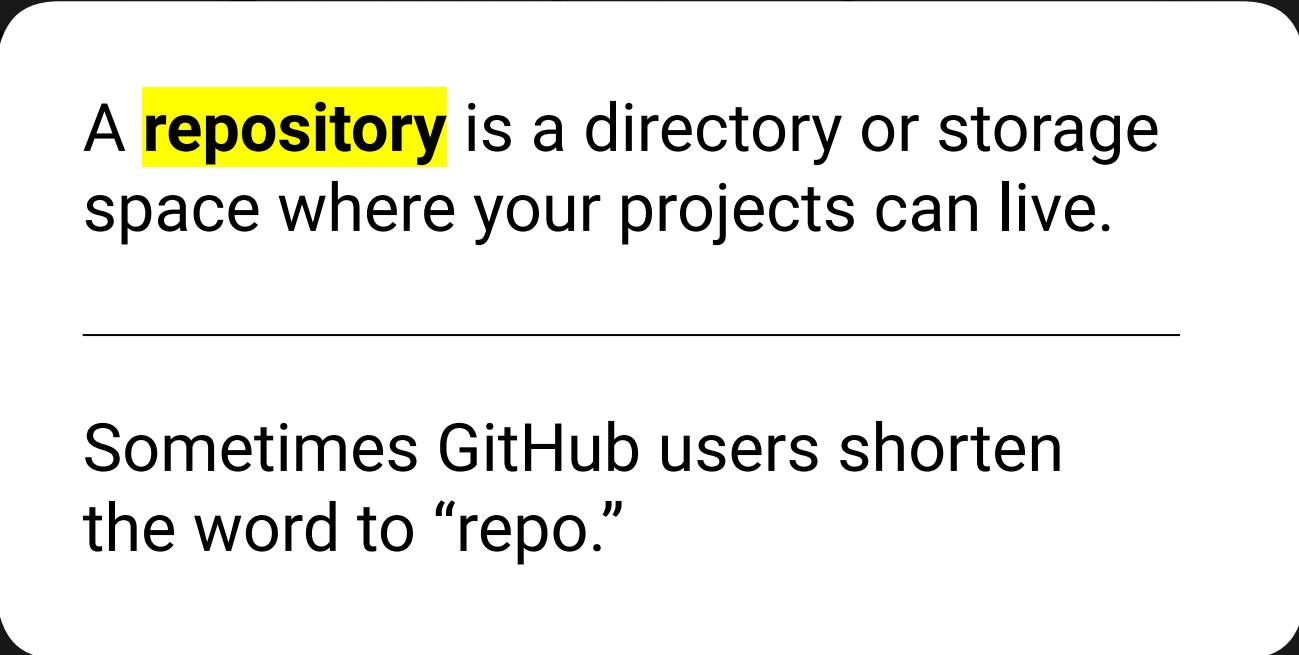
is a **version control system**  
that lets you manage and keep track  
of your source code history.



is a **cloud-based hosting service**  
that lets you manage Git repositories.



# What is a repository?



A **repository** is a directory or storage space where your projects can live.

---

Sometimes GitHub users shorten the word to “repo.”



# Why use Git?

# Why Use Git?

---

A version control system like Git makes it easy to:

01

Keep track of code history.

02

Collaborate on code as a team.

03

See who made which changes.

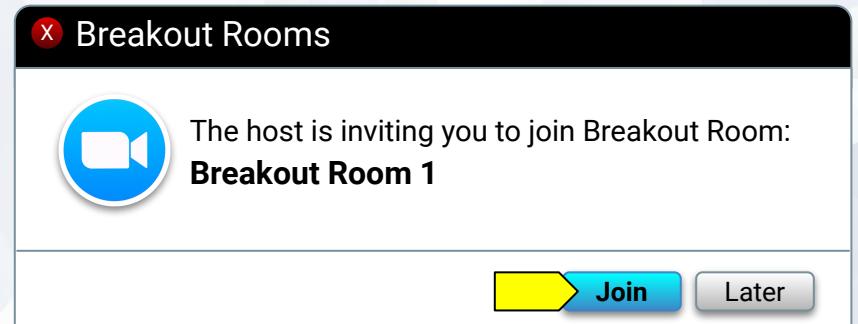
04

Deploy to staging or production.



# Quick Activity: Explain Git

Take a few minutes to explain to one another what Git version control is for.



Suggested Time:

---

3 minutes



Time's Up! Let's Review.



GitHub



# Instructor Demonstration

---

## GitHub Examples

# Questions?



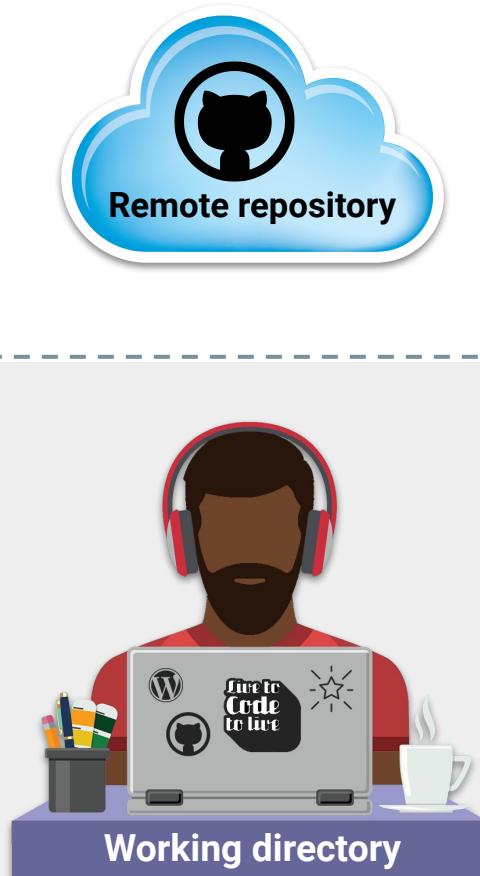


# Instructor Demonstration

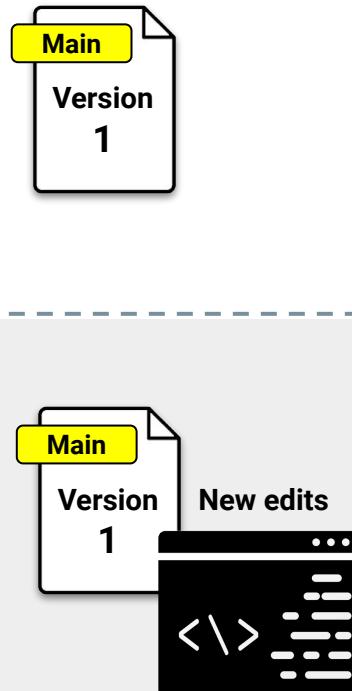
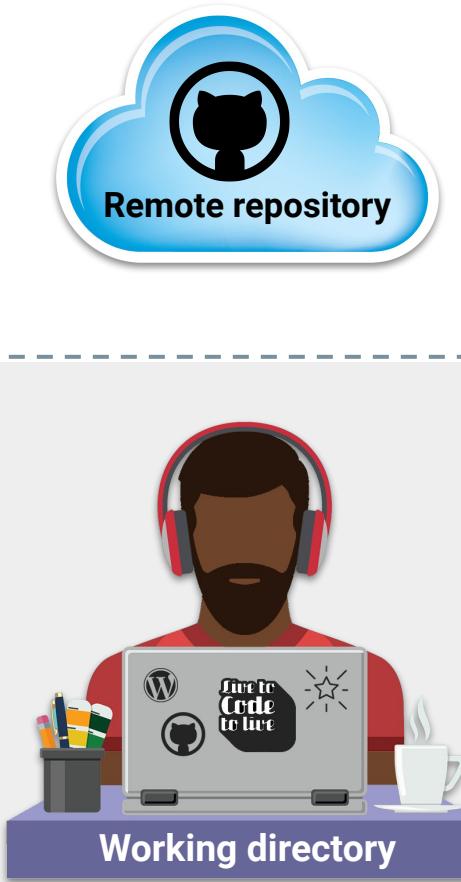
---

## Git Add, Commit, Push

# Git Clone



# Git Add



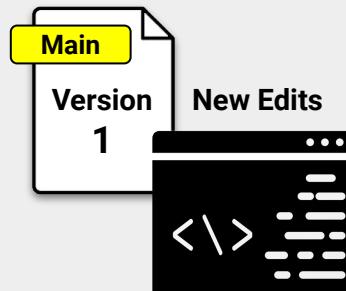
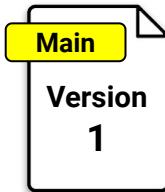
**Staging area** is where you edit the files that will be part of the next commit.

**Staging area**

`git add`

Adds your edits to be committed later.

# Git Commit



**Staging area** is the where you edit the files that will be part of the next commit.

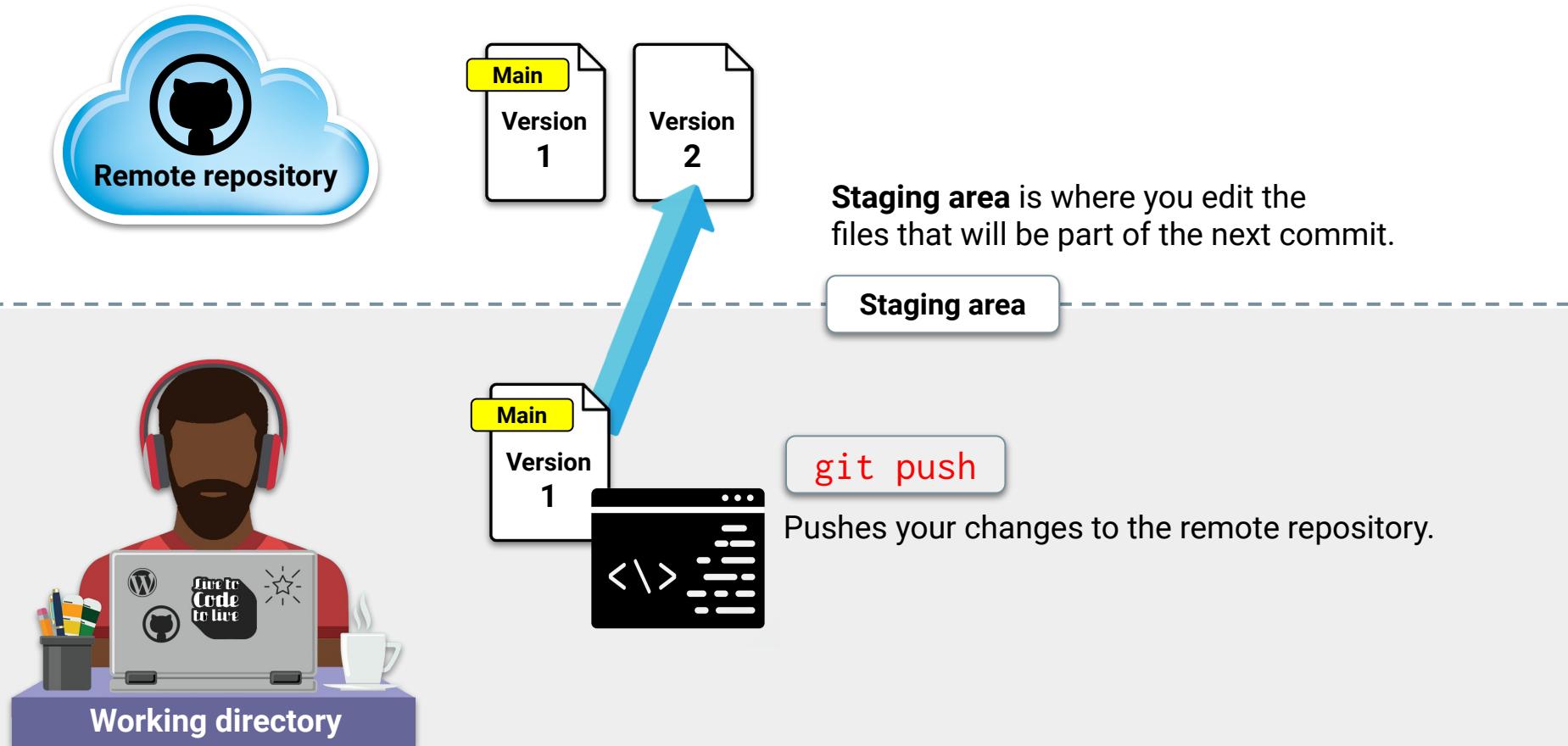
Staging area

git commit



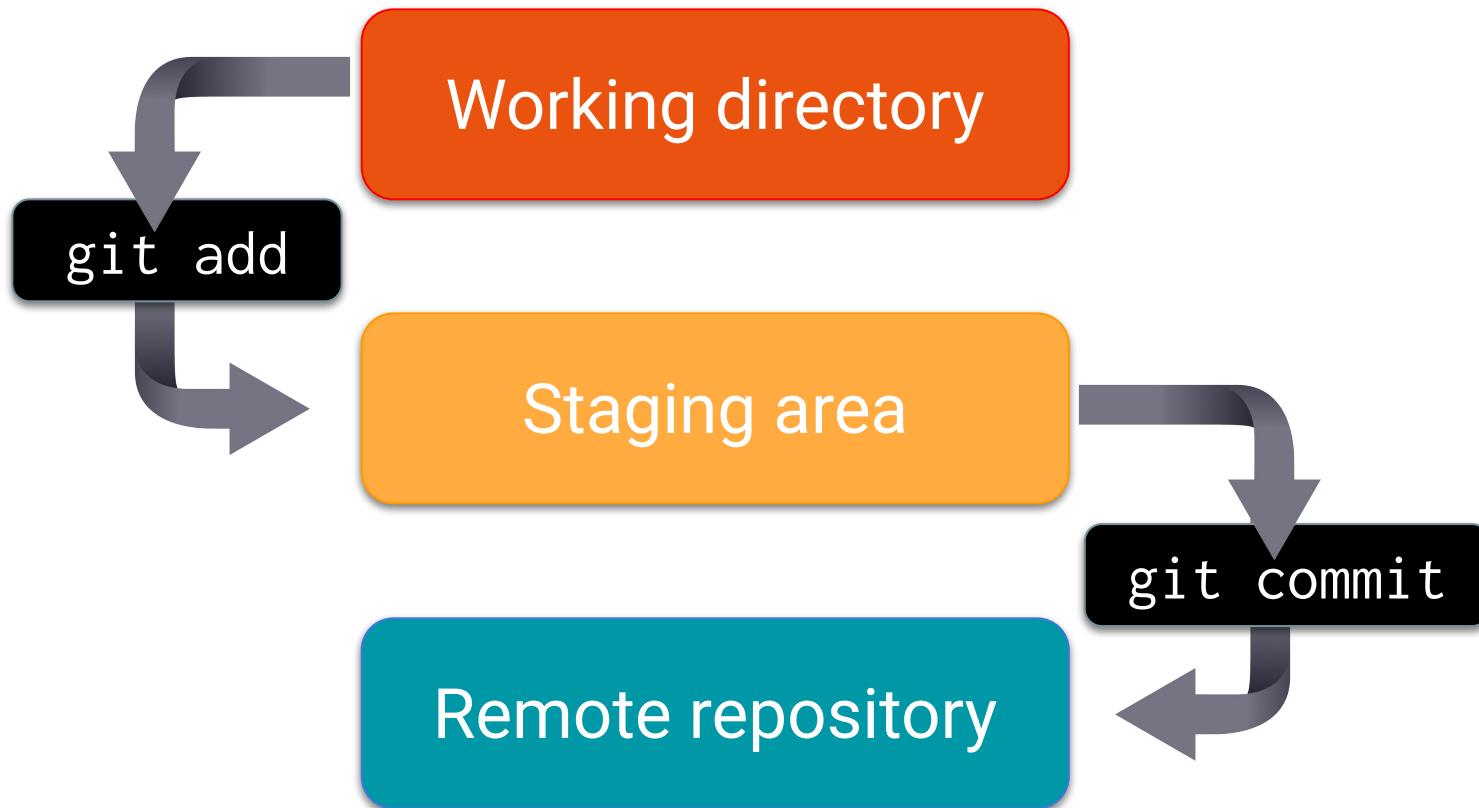
Your staged changes are saved once you commit.

# Git Push

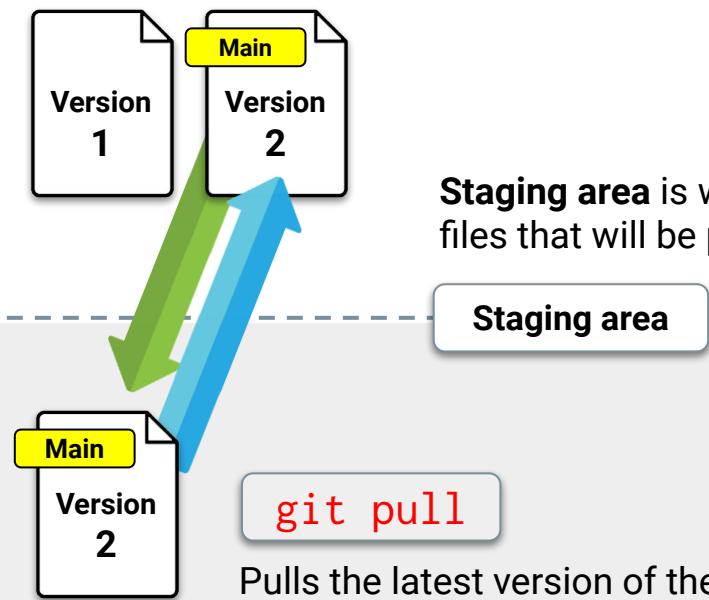
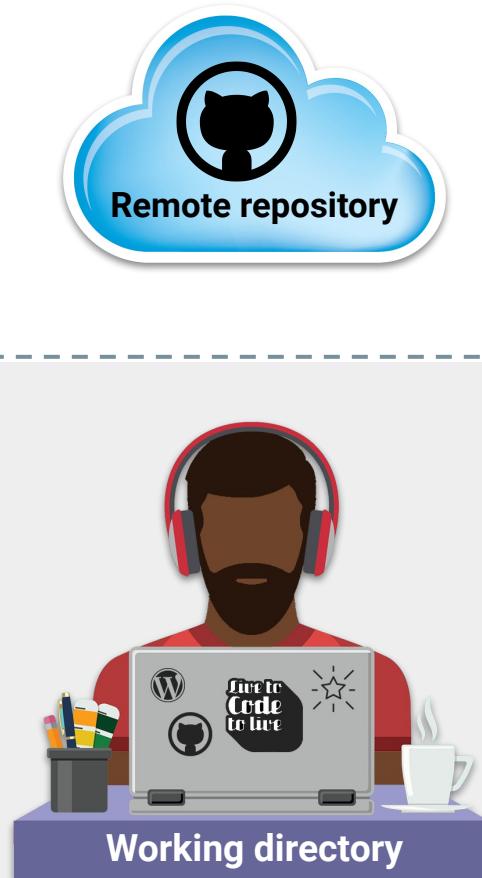


# Basic Git Commands

---



# Git Pull

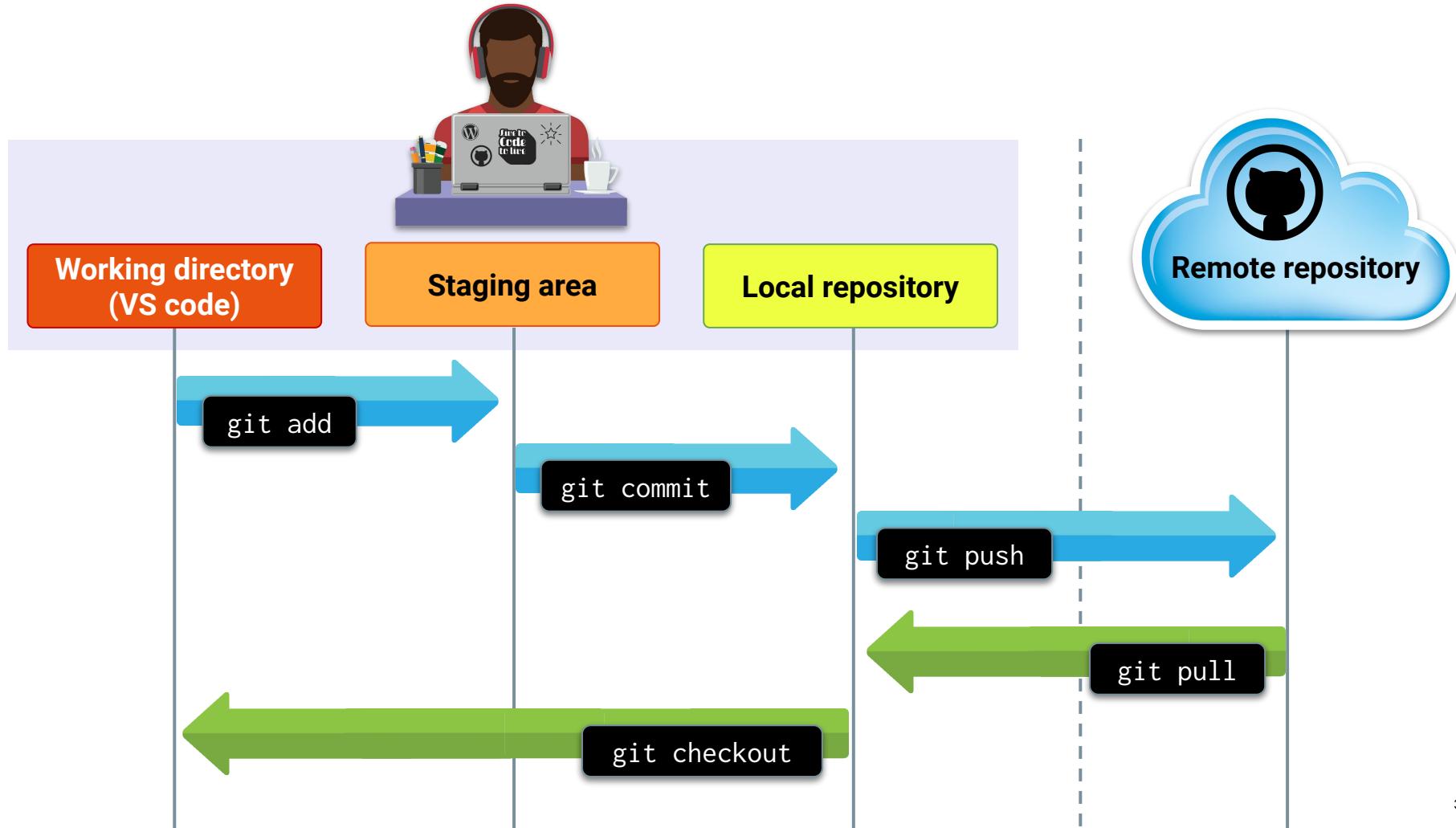


**Staging area** is where you edit the files that will be part of the next commit.

Staging area

git pull

Pulls the latest version of the remote repository and updates the local repository to match.



# Basic Git Commands

---

git clone

Copies an entire repo (to begin)

git add

Adds a file for inclusion in Git

git commit

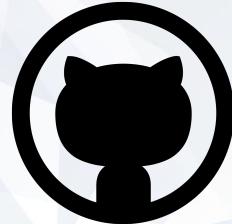
Notes a change to the local repo

git push

Sends changes to hosting service

git pull

Downloads freshest version of repo



# GitHub Guide

## Get Started with GitHub



# Activity: Git Add, Commit, Push

Using GitHub and the command line, do the following:

- Create a new public GitHub repository and name it whatever you like. Be sure to check the box for “Initialize this repository with a README.”
- Next, clone the repo to your local directory.
- Then create an HTML file inside the local directory.
- Add, commit, and push the code to GitHub.

Suggested Time:

---

15 minutes

# Activity: Git Add, Commit, Push

---

## Bonus:

- Create a new public GitHub repository and name it `zen-garden`. Be sure to check the box for “Initialize this repository with a README.”
- Clone the repo to your local directory.
- Go to CSS Zen Garden. Navigate to a few of the examples and choose a page that you like.
- Download the HTML and CSS. Each page has a link to download the code, normally near the top of the page.
- Move the HTML and CSS into your newly cloned repo and open the HTML in Chrome.
- Use inspect element to identify a page element you would like to change in some way.
- Change the CSS in any way you'd like.
- Add, commit, and push the code to GitHub.



**RECAP**



What kind of code are we learning  
in this course?

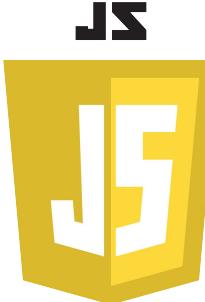
# Front-End Development



What three “languages” do we  
use to create websites?

# HTML, CSS, JavaScript

---

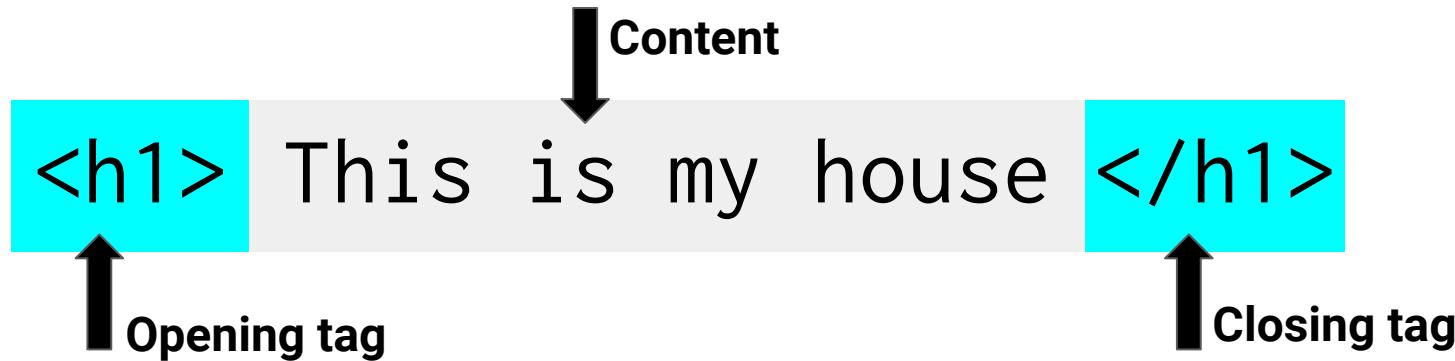
HTML	CSS	JavaScript
Used to write content  	Used to format content  	Used to create dynamic web applications that take in user inputs, change what's displayed to users, animate elements, and much more  



# What HTML Have We Learned So Far?

# HTML Syntax (Basic)

---



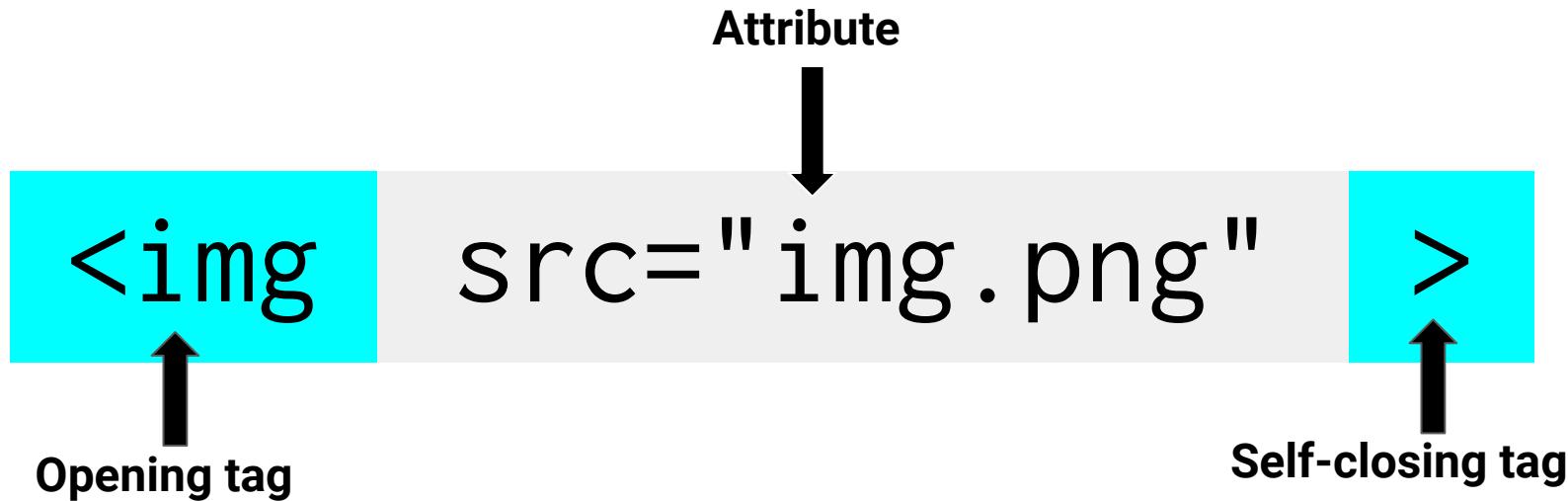
# HTML Syntax (with Attribute)

---



# Tricky Tags (Self-Closing)

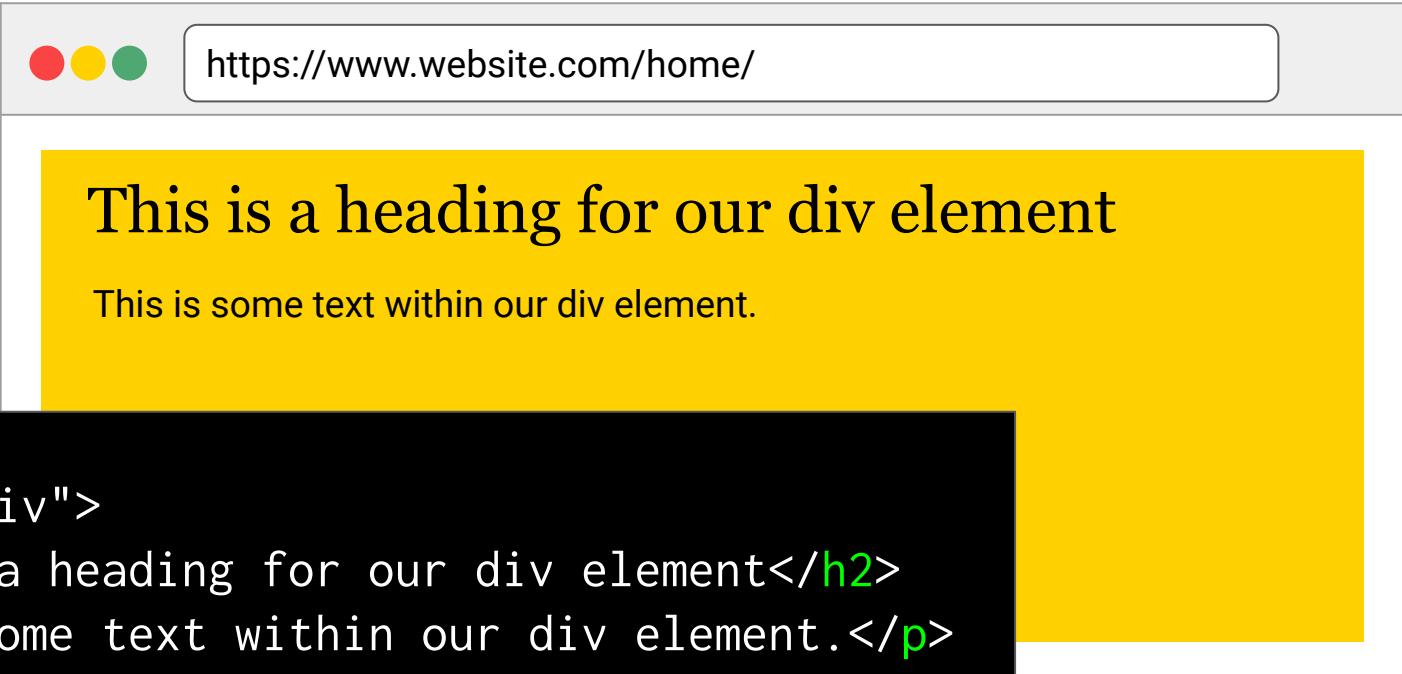
---



# Divs

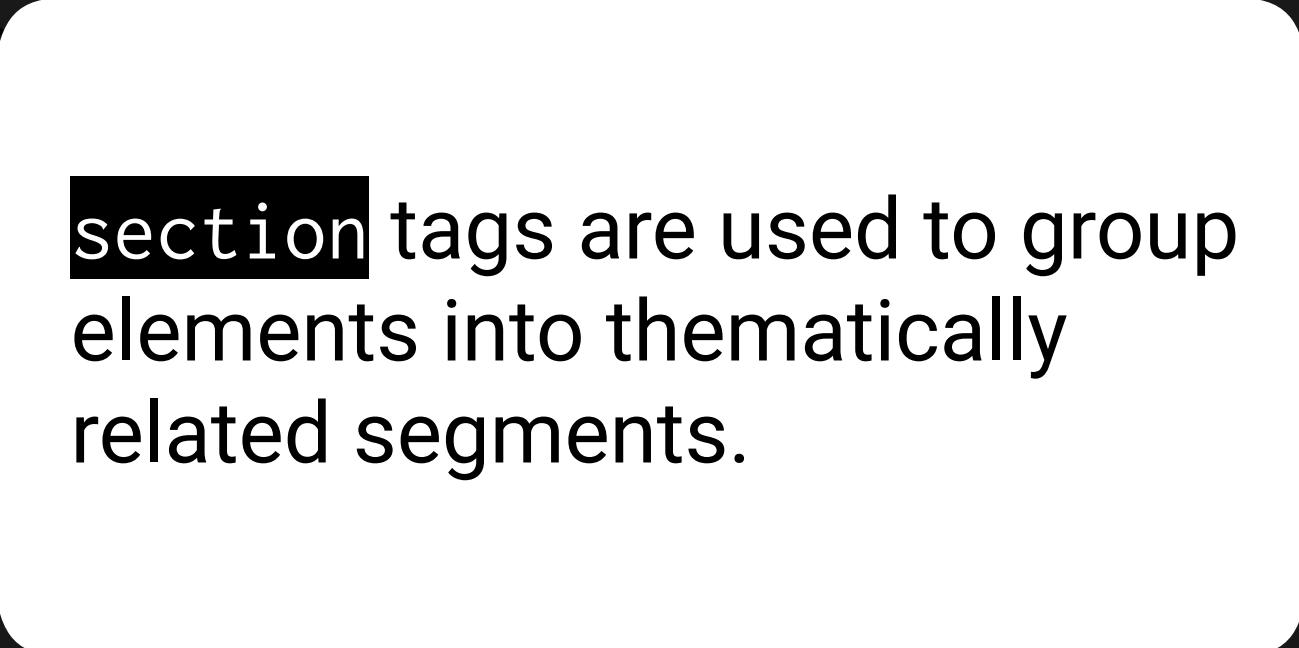
The `<div>` tag defines a division or a section in an HTML document.

# The <div> Tag



A screenshot of a web browser window. The address bar shows the URL <https://www.website.com/home/>. The page content is displayed in a yellow box. Inside the yellow box, there is an 

## element and a element. The code for this content is shown below. ``` <div class="Div"> <h2>This is a heading for our div element</h2> <p>This is some text within our div element.</p> </div> ```



**section** tags are used to group elements into thematically related segments.

---

<div>

The <div> tag creates neutral segments that require CSS to style.

<section>

<section> tags create a specific type of content that HTML5 web browsers can interpret and style.



# Activity: Basic Student Bio Activity

In this activity, you will create a basic HTML page with your own information that looks similar to the design shown on screen.

Suggested Time:

---

20 minutes

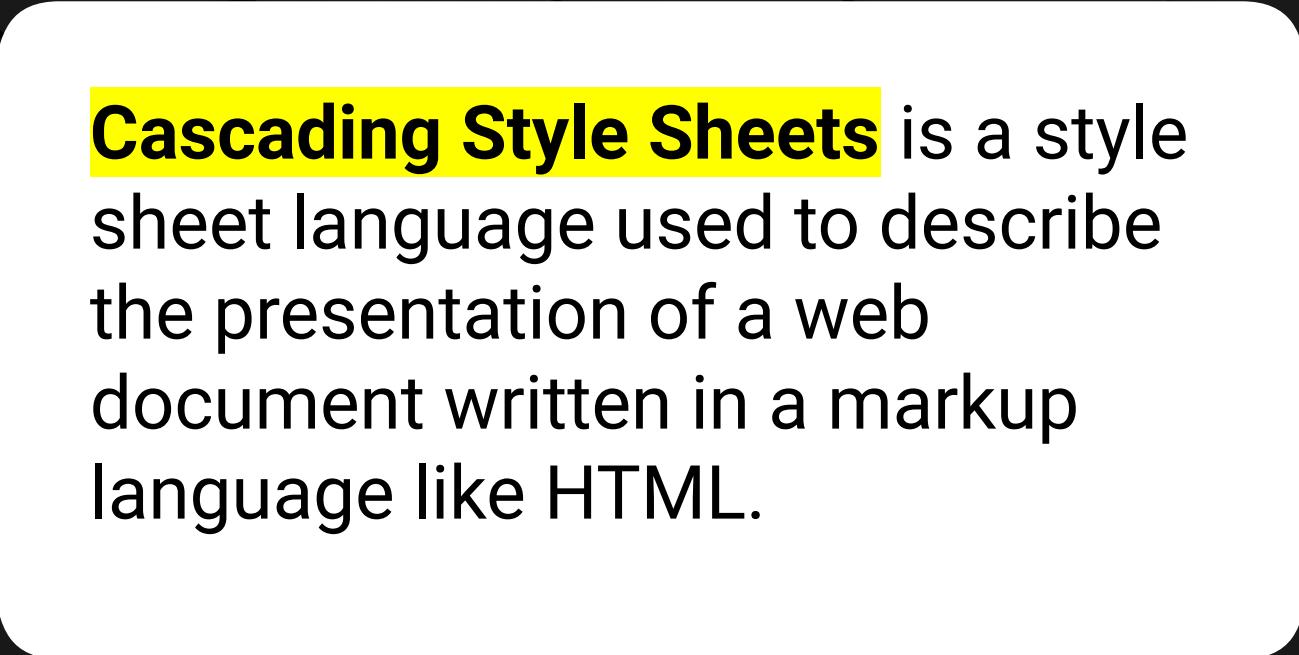


Time's Up! Let's Review.

*Break*



# CSS Stylin'—Basics of CSS



**Cascading Style Sheets** is a style sheet language used to describe the presentation of a web document written in a markup language like HTML.

# HTML/CSS Analogy

---

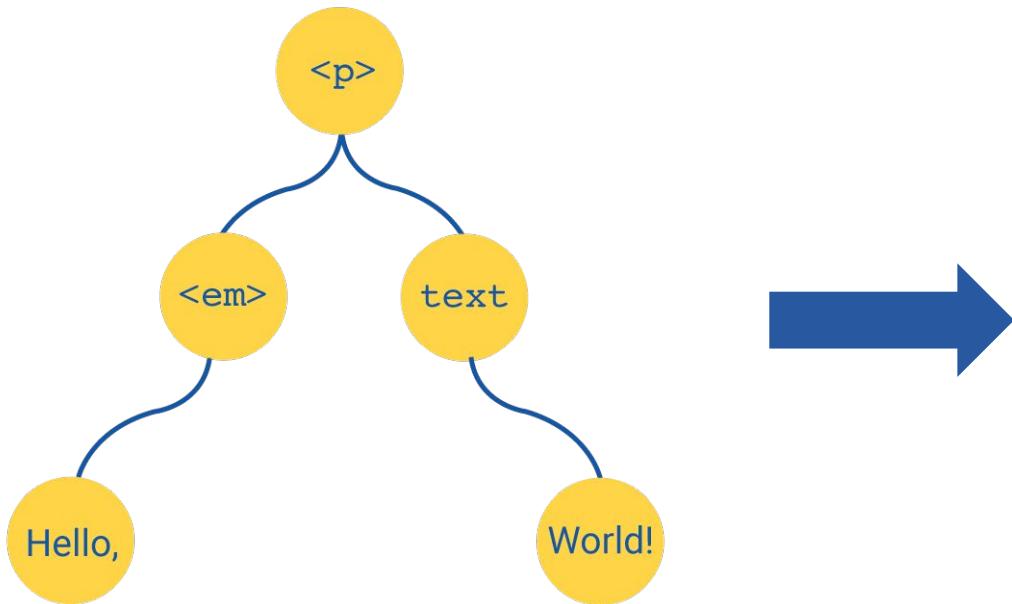
Plain HTML	HTML and CSS!
Like writing papers in Notepad	Like writing papers in Microsoft Word
Can only write <b><i>unformatted</i></b> text	Can <b><i>format</i></b> text, page layout, alignment, and more!
 A purple Notepad icon with a white 'N' on it.	 A blue Microsoft Word icon with a white 'W' on it.



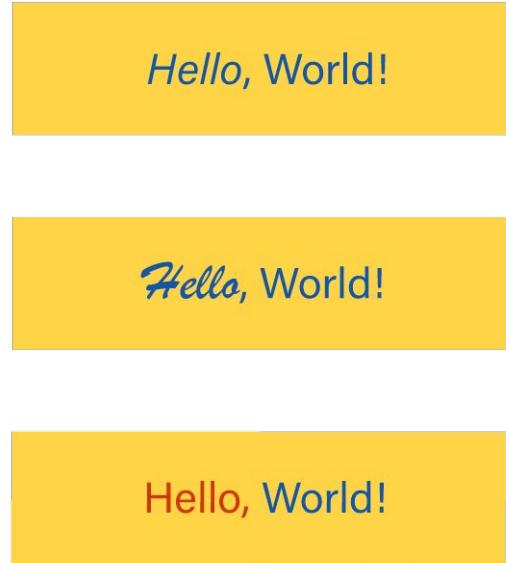
The role of CSS is to make your HTML look good! Making it look good makes it easier for people to interact with your site.

# CSS

---



**HTML**  
(structure)

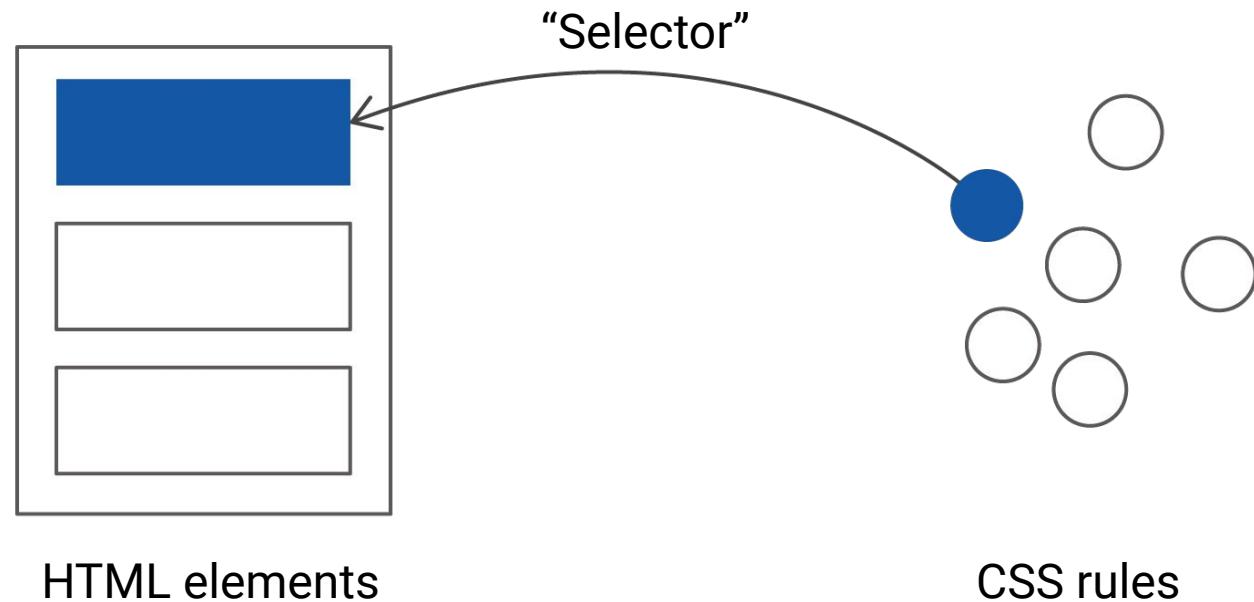


**CSS**  
(presentation)

# CSS

---

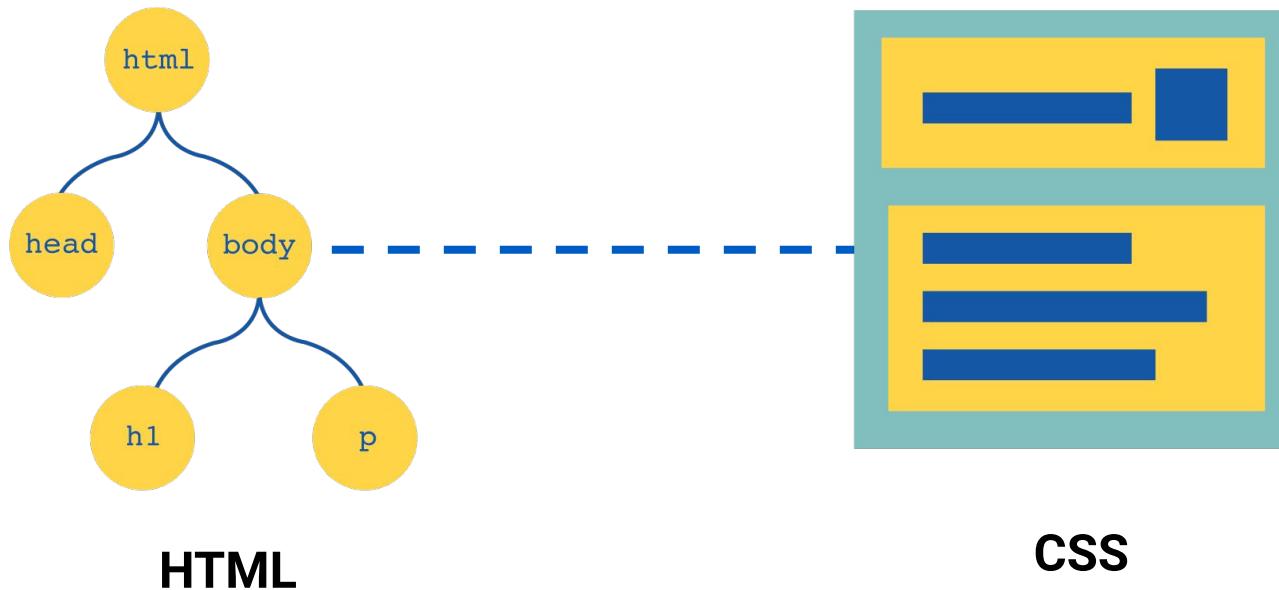
CSS is technically a selector language, which means it is used to *select* HTML elements in order to *style* them.



# How to Use CSS

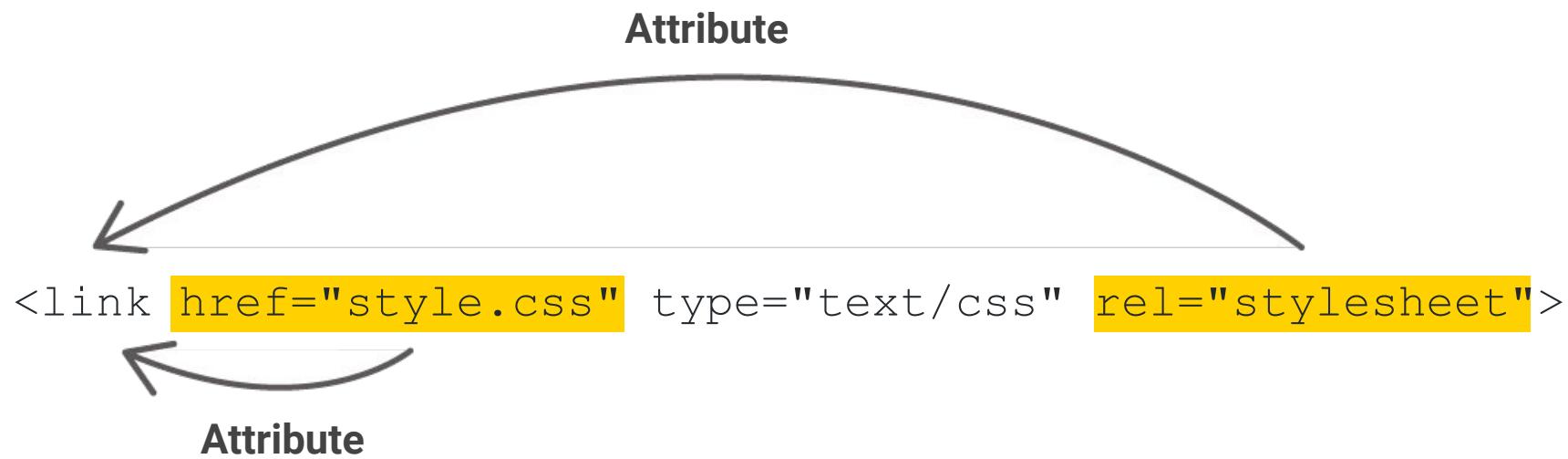
---

There are two ways to link CSS and HTML: **inline** through the use of the `style` attribute or **externally** through the use of a `link` element that connects a stylesheet to the HTML.



# How to Use CSS

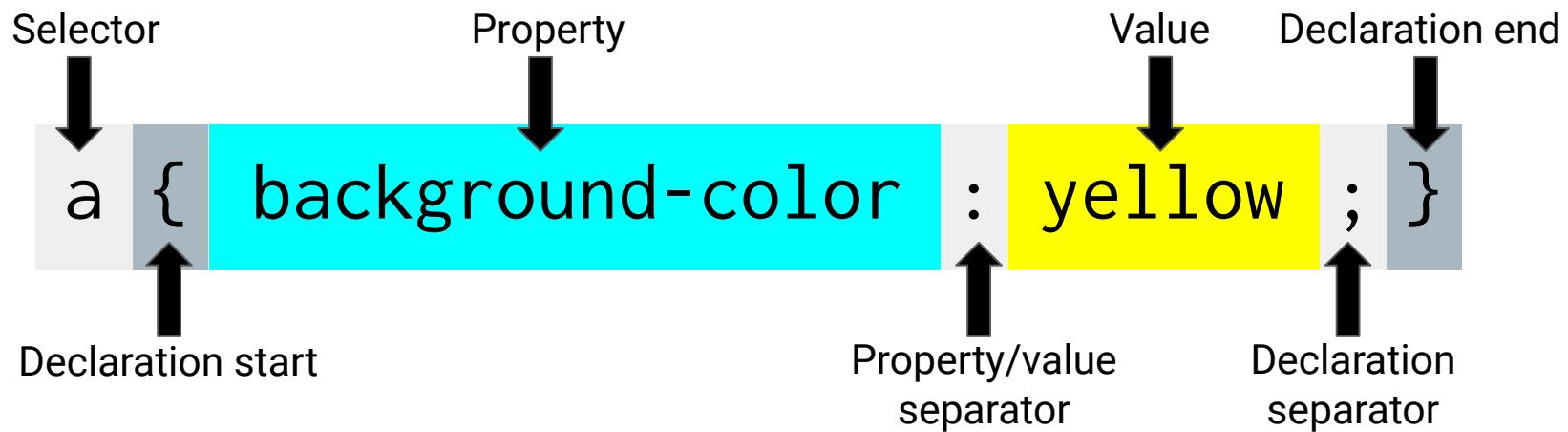
---



# CSS Syntax

CSS works by hooking onto **selectors** added into HTML using **classes** and **identifiers**.

Once hooked, we apply **styles** to those HTML elements using CSS.





With an external stylesheet file, you can change the look of an entire website by changing just one file!

# CSS Example

---

In the following example, the header would become blue and much larger because of the CSS.

We can incorporate an element's class or ID to apply a CSS style to a particular part of the document. Just remember to include the necessary symbol before the CSS: “.” for class, “#” for ID.

Example (HTML)	Example (CSS)
<p class="bigBlue">Header</p>	<pre>.bigBlue {     font-size: 100px;     color: blue; }</pre>

# Key CSS Attributes

---

Font and Color	
color:	Sets color of text
font-size:	Sets size of the font
font-style:	Sets italics
font-weight:	Sets bold
Alignment and Spacing	
padding (top/right/bottom/left):	Adds space between element and its own border
margin (top/right/bottom/left):	Adds space between element and surrounding elements
float:	Forces elements to the sides, centers, or tops
Background	
background-color:	Sets background color
background-image:	Sets background image



# Instructor Demonstration

---

## CSS Show! Demo



# Activity: CSS Styled Bio Page

Activity instructions sent via Slack.

Suggested Time:

---

20 minutes



**RECAP**

# Questions?



*The  
End*