Communications Lab
WEB

today

Project 1 Brief ~ already done!

VS Code Overview

File Organization

HTML Part1
- structure
- h1...
- p
- a
- img
- br
- pre

Preview Recitation Project

Brief: Project 1 *Shanzhai Web*

The Brief

Project 1: Shanzhai Web

Introduction

Shanzhai 山寨, literally meaning "mountain fortress," originally described bandits who opposed and evaded corrupt authorities during the Song dynasty. Today, *shanzhai* refers to counterfeit goods that rebel against the established commercial market, carrying a spirit of opposition and individuality. Some *shanzhai* products aim to deceive buyers, others add missing features, and many are made in jest or parody.

For our purposes, *shanzhai* is not about fakery but about **copying as creative strategy**. It mixes technical mastery (closely imitating the recognizable surface of a brand) with adaptation, deviation and play (bending it for new purposes). This tension creates room for art: by reusing familiar corporate designs — the very medium of authority — artists can surprise, critique, and play.

What we have done so far

In the first weeks of class, we explored how perception (Gestalt) and medium (McLuhan) shape meaning: the same content feels different depending on how it's arranged, framed, or delivered. In your paper websites and journeys through google sheets, you saw how structure alone (color blocks, scrollable distances, links) can create meaning.

Shanzhai Web builds on this: when you copy a corporate site, you inherit its authority and familiarity. When you twist it, you reveal how fragile that authority really is; copying then isn't passive imitation; it's a way of **speaking through an existing medium to critique, parody, or reimagine**.

The Assignment

Create a small website that copies the skeleton of an existing corporate site (brand, platform, institution, news outlet, etc.) and then bends it with your own conceptual twist.

Your project should both show layout proficiency in HTML/CSS (week 3-6) and make a clear conceptual statement (parody, critique, speculative future, personal remix, cultural translation, etc.).

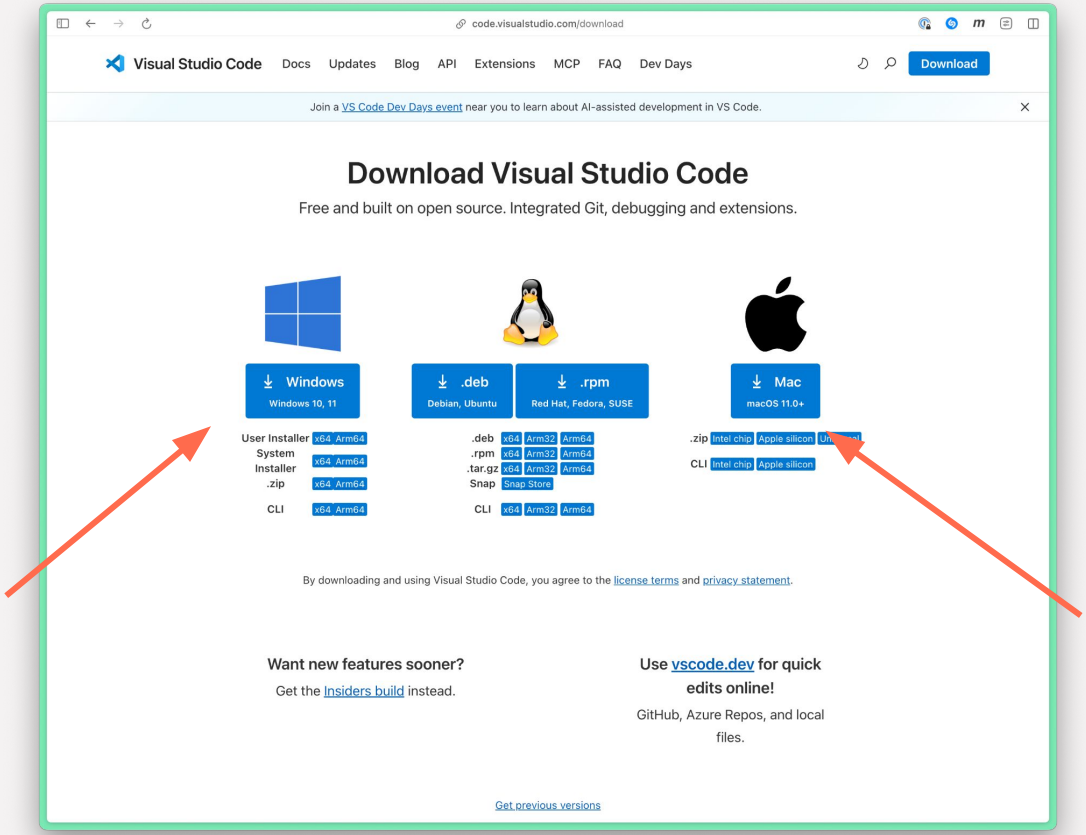
Requirements:

- Use HTML + CSS only (the techniques we cover in class).

Download VS Code

Visual Studio Code (VS Code) is an application in which you will write code. We call such an application a "**text editor**".

Download VS Code from https://code.visualstudio.com/Download.



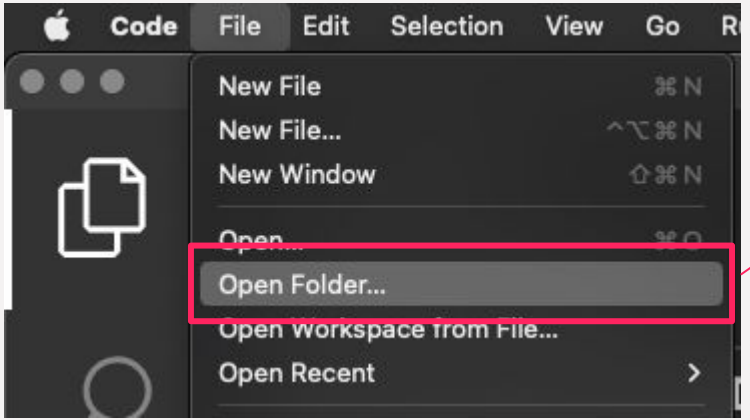
Create a folder for all of your CommLab projects

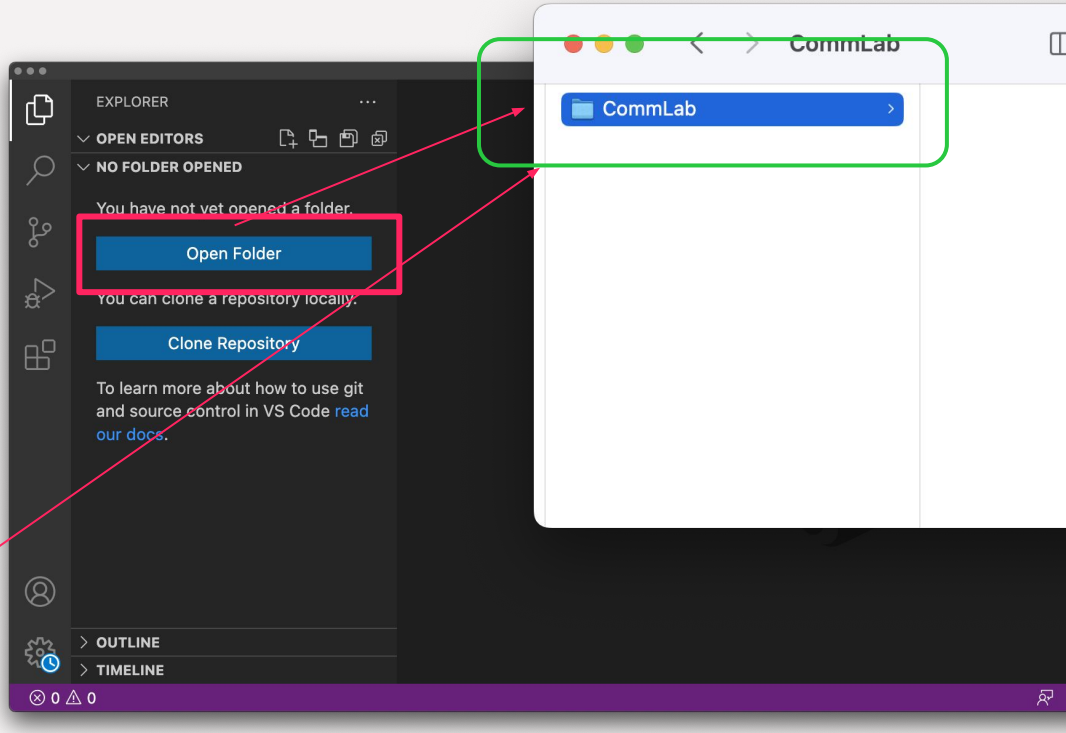
Create a folder. Choose a good location for it and give it a name that makes sense and has **no spaces.**

Work on Websites

First, open your entire CommLab folder in VS Code.

1. Open VS Code.
2. File > **Open Folder…**
3. Find your CommLab folder.



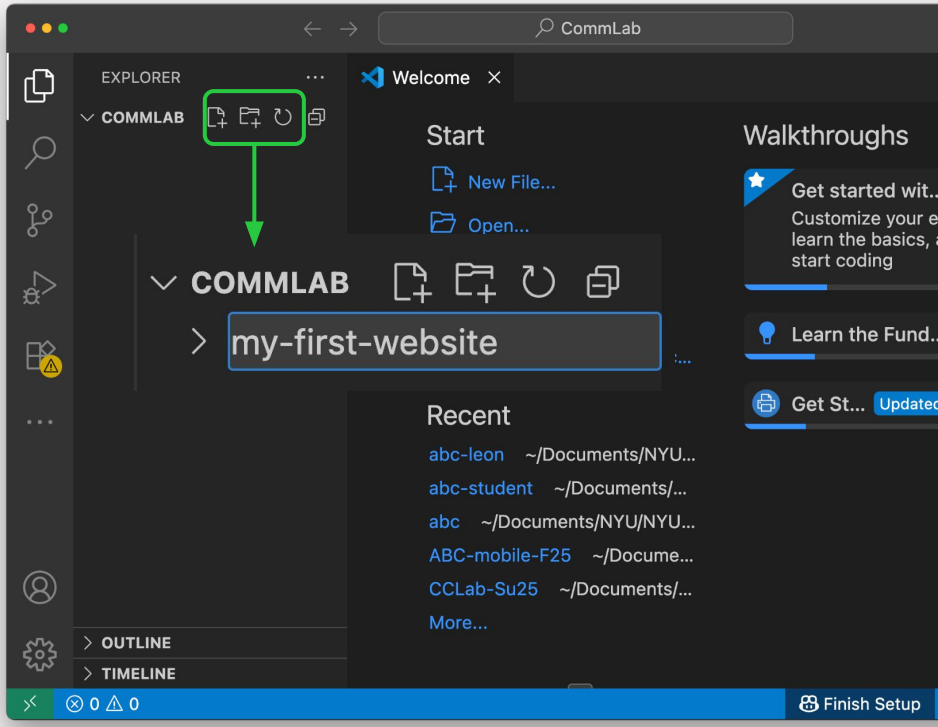


Starting a new project with VS Code

In VS Code, with your entire CommLab folder open, create a folder for your next website.

Icons will appear when hovering the mouse on the folder you just opened.

Click the folder icon and provide a folder name when prompted.



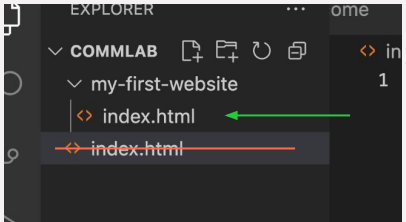
Starting a new project with VS Code

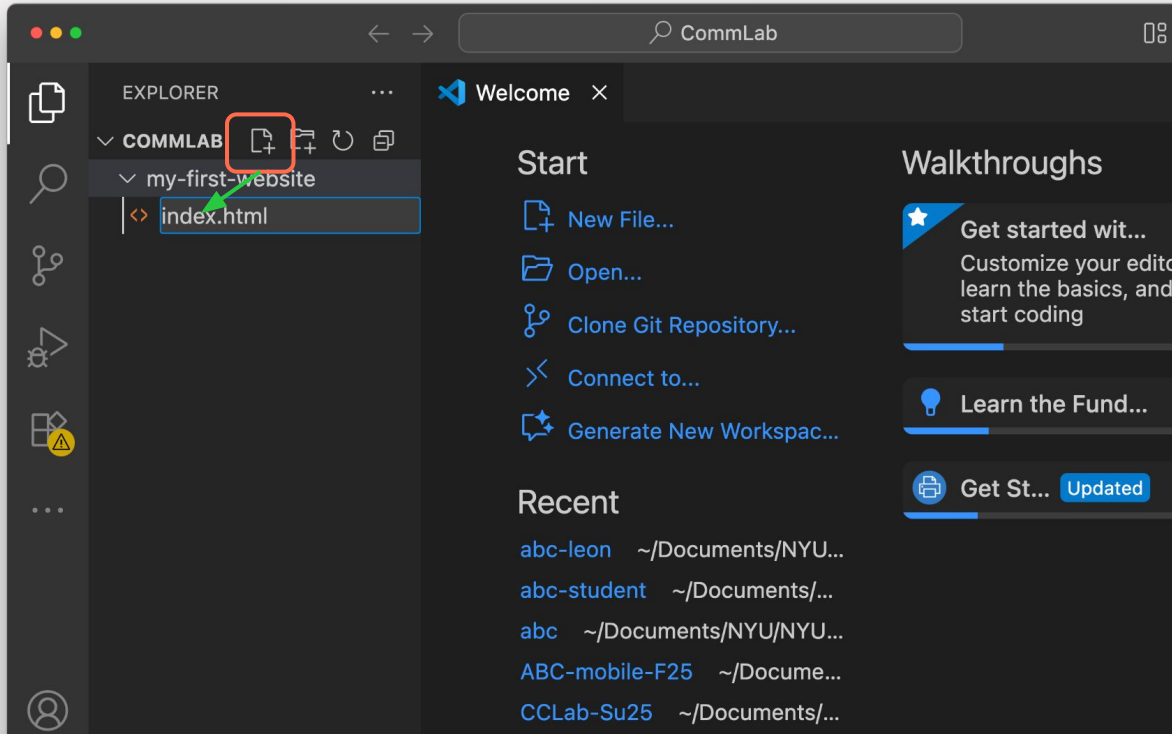
Then create the first file.

The first file will ~~often~~ always be **"index.html"**, the central page of your website.

Click the new file icon and type "index.html".

Double-check that the file was created inside your website folder (“my-first-website”, for example) and not inside the larger CommLab folder.



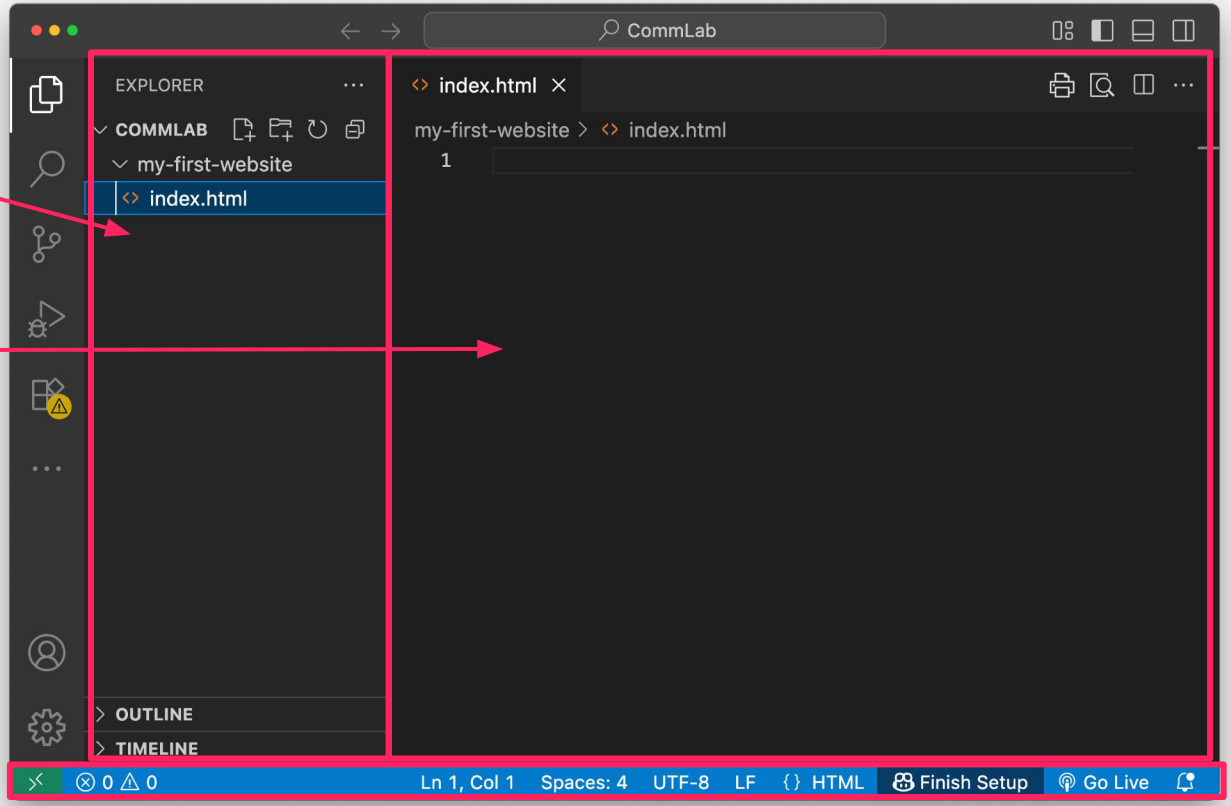


VS Code's Interface (1/3)

Tree View

Open Files/Documents

Status Bar



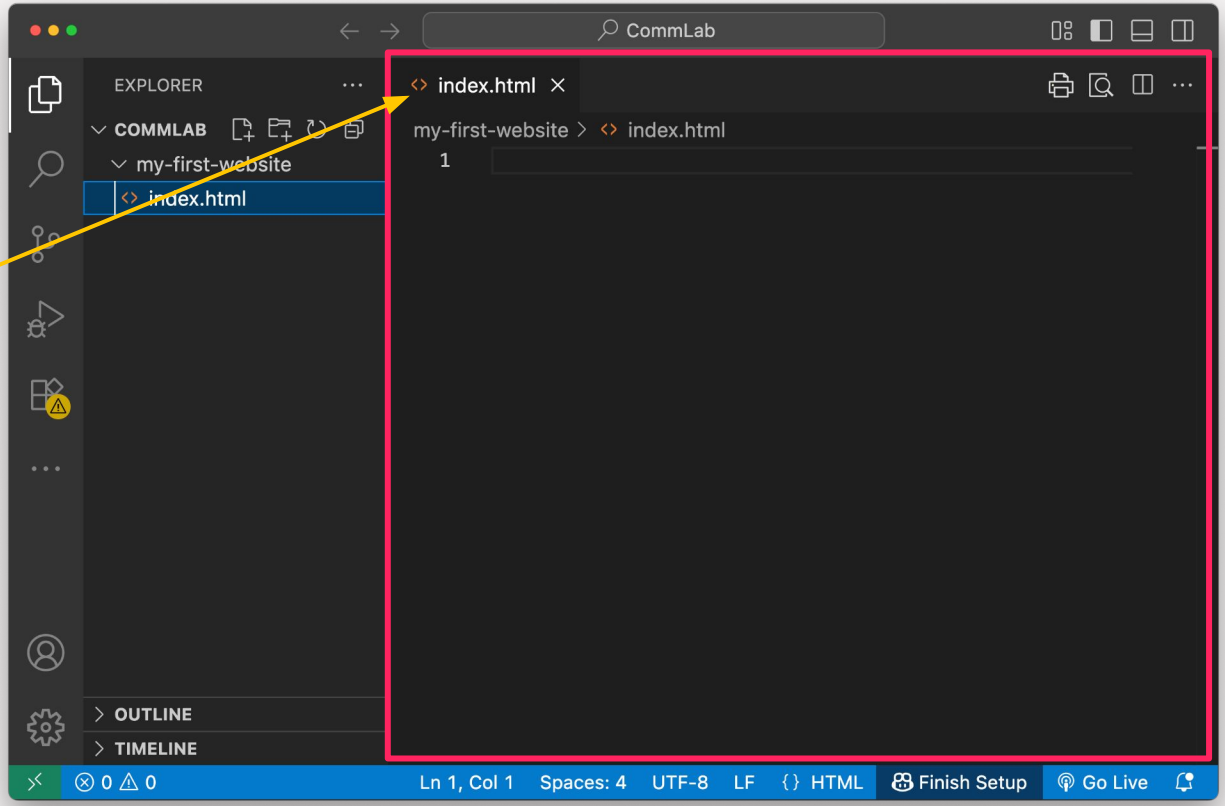
VS Code's Interface (2/3)

When you click a file in the tree view, it opens up on the right side.

Here is where you **write your code.**

You can have **multiple files** open in different **tabs**.

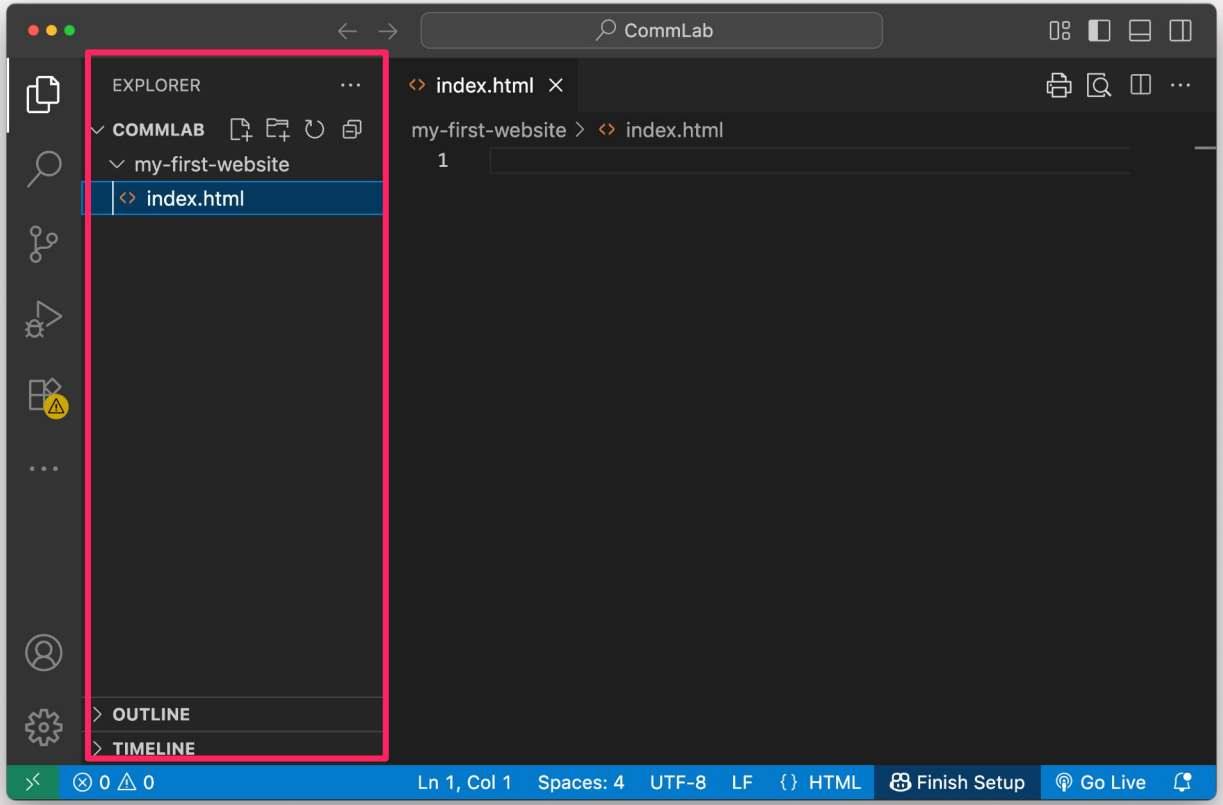
If you close a file, you can easily see it again by clicking it in the Tree View.



VS Code's Interface (3/3)

In the Tree View,
you can **see every file** in
your project folder.

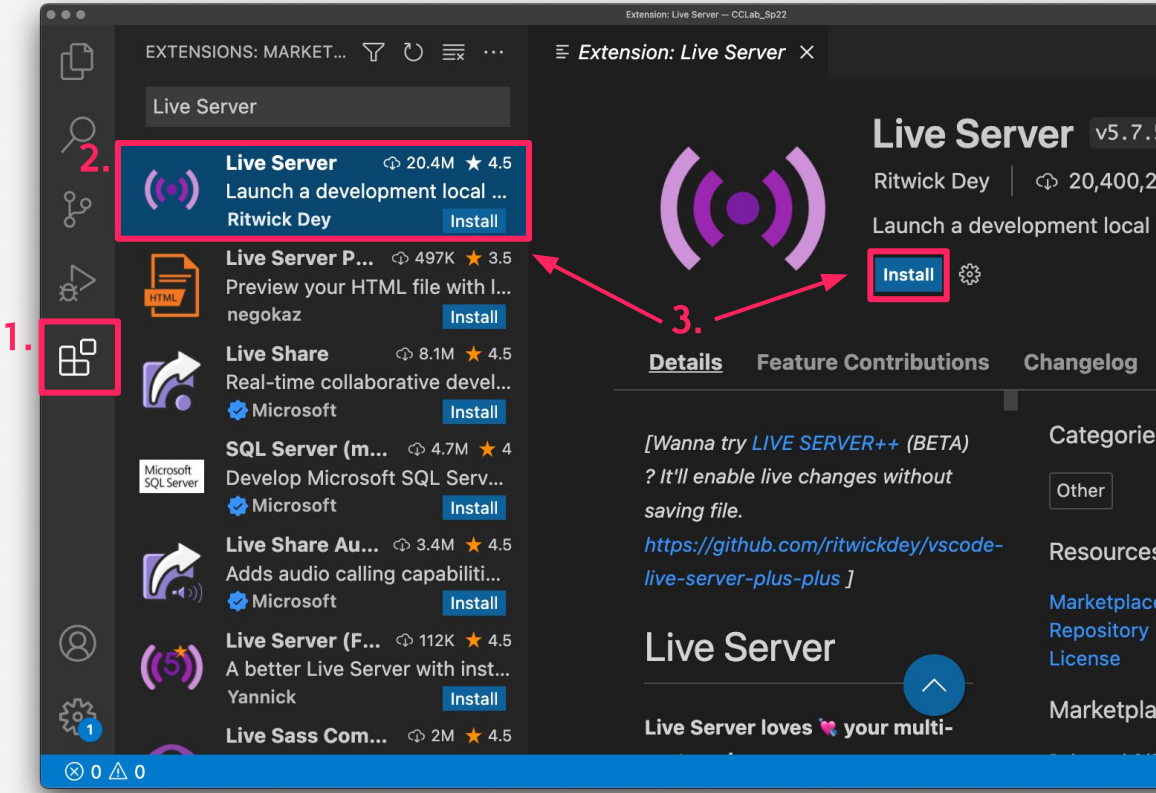
When you click a file, it will
open in the right working
space / open files area.



Install Extensions

In the Preferences, find “**Extensions**” and search and find “**Live Server**” and click **install**.

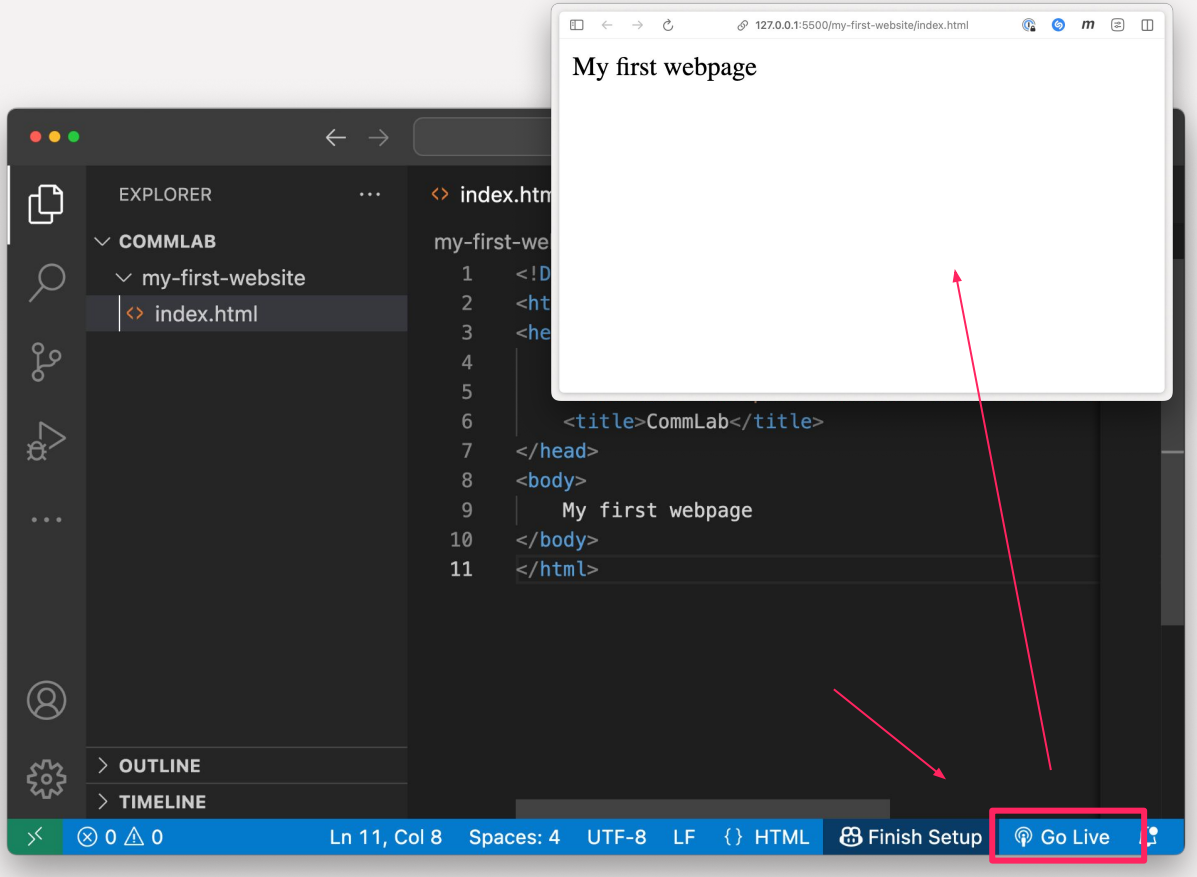
This extension will be part of you working process when building websites.



Use the Live Server Extension

Click "**Go Live**" on the Status Bar to initialize a local server and **see your website** on the browser!

Ensure that you open **the project folder**, not a single file.

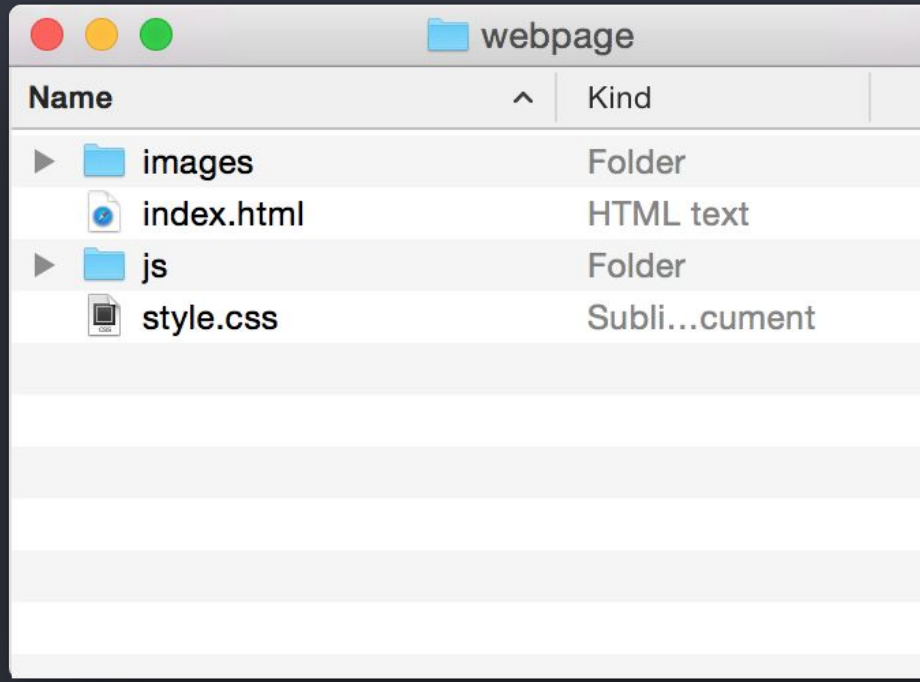


File Organization

File Organization

Web pages are usually not just one text file.

There are multiple text files (with HTML, CSS, and JavaScript), media assets (like static images, sprites, GIFs, video, and audio), and folders that organize their structure.



File Organization

It is EXTREMELY important that you keep your files (and folders) organized in this class. I will be demonstrating some best practices for this over the course of the semester.

Web projects break because of sloppy file management and then it gets very, very, hard to fix them again.

Please keep on top of this!

File Organization Tips

Lower case and upper case letters are not the same.

- `Index.html`
- `index.html`
- `INDEX.HTML`

File Organization Tips

Avoid white space

- `"My Website.html"` - WRONG
- No spaces in file or folder names
- Use dashes or underscore instead. `"my-website"` or `"my_website"`

File Organization Tips

Be consistent

- If you use upper-case letters in a folder, use it with ALL folders. Same with lower case.
- For example, if your main folder is `"my-website"`, your subfolders could be `"my-videos"`, `"my-photos"`, etc...

File Organization Tips

Short names are better

- As long as you don't sacrifice specificity.
- Example: `"website"` is probably better than `"my-first-ever-website-fall-2021"`.

File Organization Tips

1. Lower and upper case are not the same
2. Avoid white space
3. Be consistent
4. Short names are better

File Organization Final Lessons

- Don't move files into different folders unless you know exactly what you are doing.
- Don't rename folders or files for the same reason as above.
- Changes in file or folder names will not automatically update other files that reference it, so be careful!

HTML

...is the name of the language we will learn.

What is HTML?

HTML = **HyperText Markup Language**

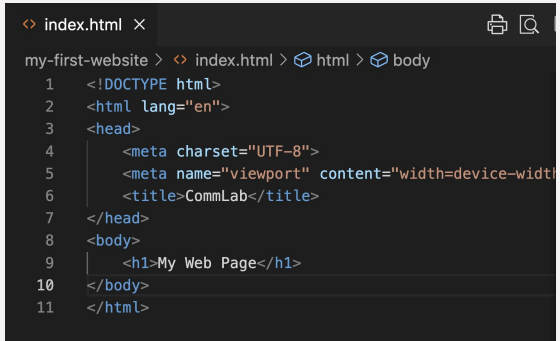
The basic building blocks of the web.

HTML is about structure/hierarchy and content, not style (CSS) or interaction (JavaScript).

It tells the browser:

> *“This is a heading.”*
> *“This is a paragraph.”*
> *“This is an image.”*
> *“This is a link.”*

A web page = a text file written in HTML, read by the browser.



What is HTML?

The Browser's main job is to
read HTML (plain text)
interpret it
then render the webpage (graphically).

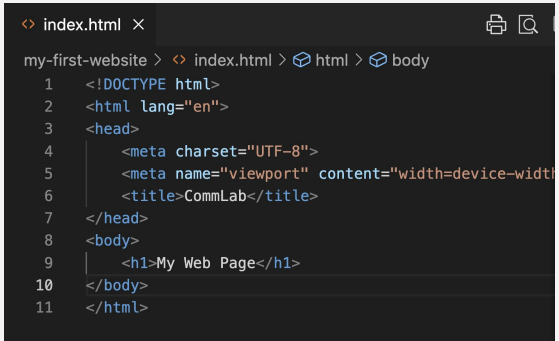
HTML communicates

"This is a heading."
"This is a paragraph."
"This is an image."
"This is a link."

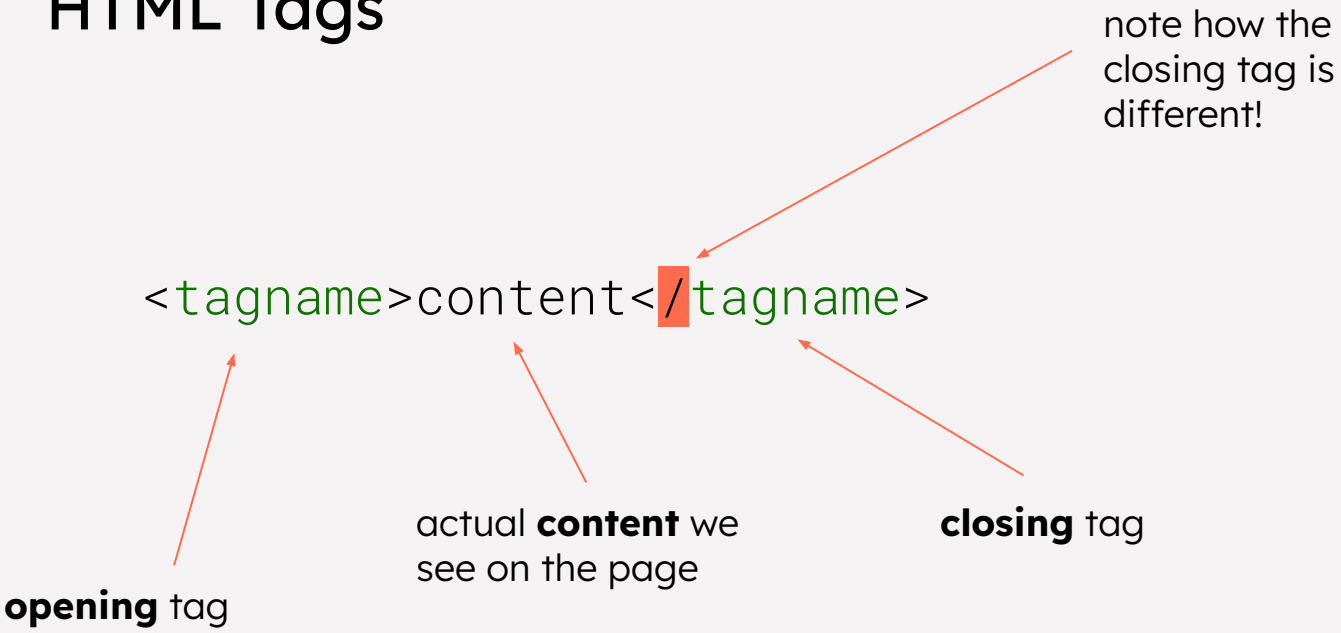
by using "HTML Tags":

```
<h1>My Web Page</h1>
```

Tags produce "HTML Elements"



HTML Tags


note how the closing tag is different!
<tagname>content</tagname>
opening tag
actual content we see on the page
closing tag

HTML Tags

```
<tagname>content</tagname>
```

element

HTML Tags

sometimes additional **attributes** give the browser additional information on how the element should be rendered

```
<tagname attribute="attributevalue">content</tagname>
```

note that the attribute(s) are located **inside** the element's opening tag

HTML Tags

```
<tagname>content</tagname>
```

There is a clearly **defined list of possible tag names** that every browser knows and can render.

In the following slides we will learn some of them.

While learning, it is very useful to pay attention to each tag's *default styling* on the page. For Example, the browser, by default, renders a headline larger than a paragraph, or colors a link in blue.

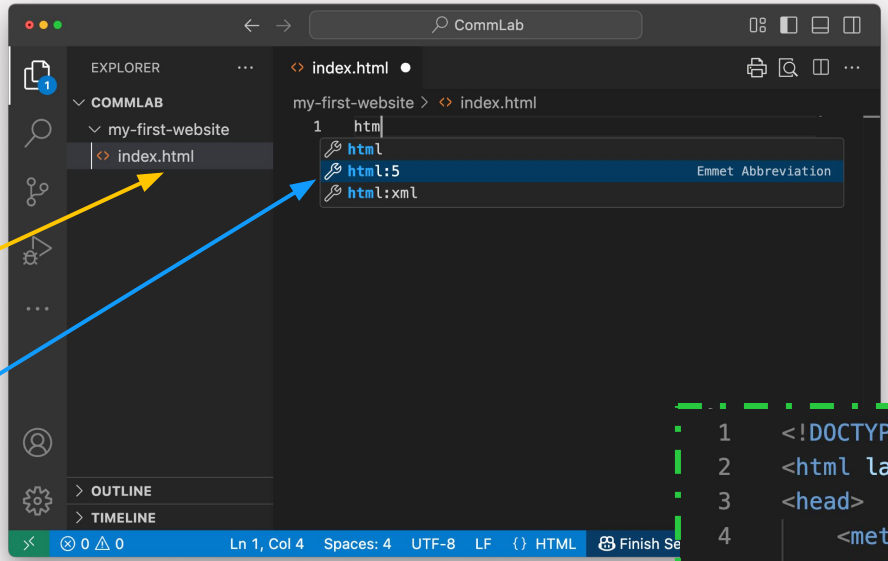
Let’s write some HTML!

Every HTML file starts with the same basic structure.

🤫super trick:

if your file has the **.html file extension**.

...then you can **type “html....”** and pick the **html:5** option from the autocomplete dropdown!




```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="wid
6      <title>Document</title>
7  </head>
8  <body>
9
10 </body>
11 </html>
```

Basic HTML Structure

tell the browser to be ready to interpret some html

```
<!DOCTYPE html>
<html lang="en">
<head>

</head>
<body>

</body>
</html>
```

an opening and closing tag!

all our html code will be inside on large HTML element!

in the head are *invisible* things.
information about our website, links to additional resources the browser needs to render the page, encoding information, etc.

you spent most of your time in the body.
***everything we actually see* on the webpage (text, images, etc.) is in the body.**

note: there is only ONE head, and ONE body on a page, never more!

Basic HTML Structure

```
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-wi
6       <title>Document</title>
7   </head>
8   <body>
9
10  </body>
11  </html>
```

in the <u>head</u> are *invisible* things.
information about our website, links to additional resources the browser needs to render the page, encoding information, etc.

the autocomplete has already put some elements into our page's head.
Can you find out what they are doing?

💡 *Reminder:* to see the page rendered, use the "Go Live" button, created by the extension we installed earlier.

the paragraph <p>

Use for **blocks of text**.

Browsers automatically
add space before/after.

```
<p>This is a paragraph of text on my page.</p>
```

headings <h1>.....<h6>

Use for **titles and subtitles**.

<h1> is the largest, <h6> the smallest.

```
<h1>Main Title</h1>
<h2>Subtitle</h2>
<h3>Smaller Heading</h3>
```

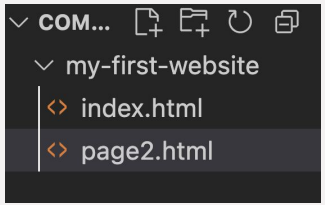
the link <a>

Creates **links to other pages or sites**.

Look, our first attribute!

Can you think about why a link has an attribute?

```
<a href="https://google.com">Visit Example</a>
<a href="page2.html">Go to Page 2</a>
```



try adding a second page to your website.

images <img>

Displays an image file on the page.

This tag has no closing tag because it has no *content*.

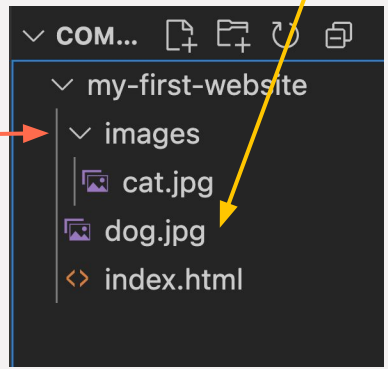
the **src** attribute describes the **path** to the image file.

```
<img src="dog.jpg" alt="A dog" width="500">

<img src="images/cat.jpg" alt="A cat" width="300">
```

the dog image is NOT in the images folder, that's why its path looks different.

the cat file is inside the images folder. This MUST be reflected in the path we specify. That way the browser can find the image.

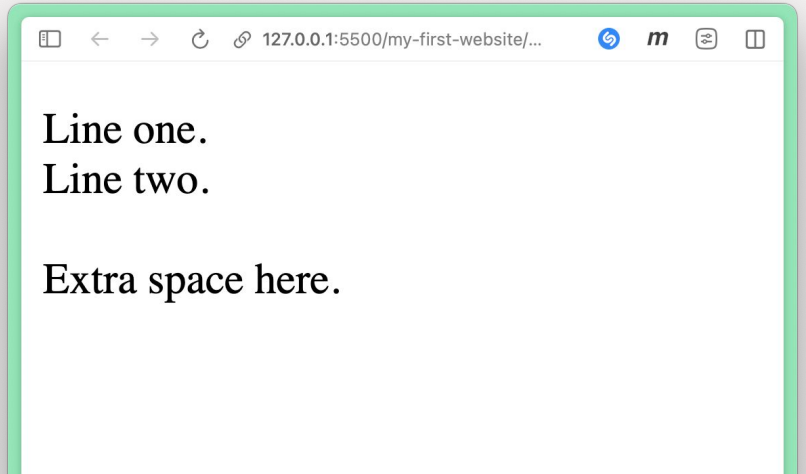


line break

Moves the next text onto a new line.

Use for **spacing** or poetic rhythm.

```
<p>Line one.<br>Line two.<br><br>Extra space here.</p>
```



This tag also has no closing tag (because it has no content).

Also, try out: **<hr>**

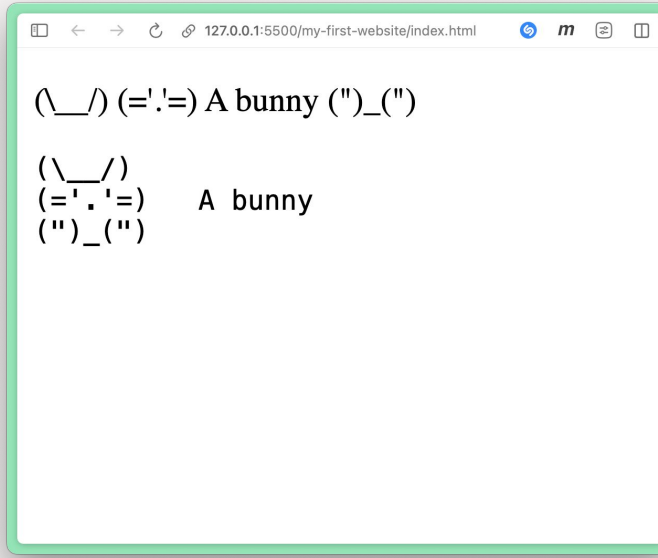
preformatted text <pre>

Preserves spaces and line breaks exactly as typed.

Useful for cute ASCII art.

Can you image what purpose (besides ASCII art) this tag has?

```
    <p>
(\__/)
('.')    A bunny
(")_(")
    </p>
    <pre>
(\__/)
('.')    A bunny
(")_(")
    </pre>
```



Tomorrow’s Recitation: Life Story as Scroll

Make a small **multi-page website** that tells a **true story from your life** (whole biography, a season, one day, one friendship—your choice). Like you did in the Google Sheet last week, **express time, space** (
) and **transitions through space, distance, scrolling, and clicking** (<a> links to other pages). Also use images, and, if you like, ASCII art (<pre>).

Use this exercise to apply and practice the elements we have covered in class. **Don’t use any techniques beyond those we covered**, but see the small selection of techniques as a **creative challenge**.

Requirements
* Pages: at least 3 HTML pages (e.g., index.html + 2 others) connected with links.
* Media: at least 2 images (<img>).
* Non-fiction: write from your real life (you can be selective, poetic, minimal).
* File hygiene: a clean file organization is important.

Submission
Compress (zip) your website folder and send it via DM to me on Slack.