

# J220

## Coding for Journalists

LECTURER

Yoli Martinez

### PROMPTS

- Go to this Poll Everywhere to participate in today's class:  
<https://pollev.com/yolim>

start Zoom recording + captions

# Agenda

Announcements

Introduction to the Text Editor (making your code readable)

Semantic HTML and Hierarchy

Catching Bugs

**BREAK**

File Structure

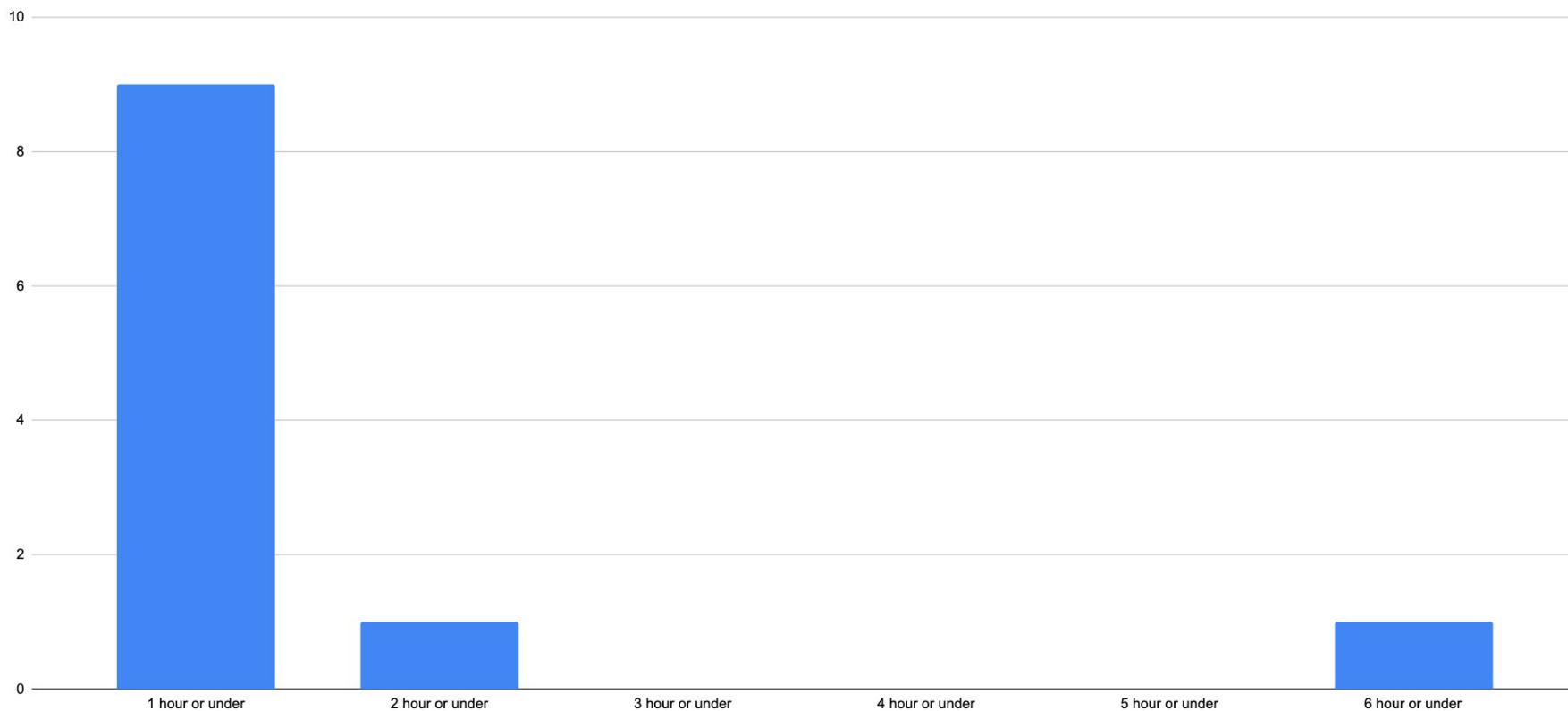
In-Class exercise

Homework

# Announcements

We'll wait 10 minutes for office hours. TK.

## Week 01-23: Number of students grouped by hours spent outside of lecture and office hours



# Text Editor, and making your code readable



# Which text editor will you be using?



Powered by  **Poll Everywhere**

Start the presentation to see live content. For screen share software, share the entire screen. Get help at [pollev.com/app](https://pollev.com/app)

# Text Editor

You can write code anywhere, but a text editor will:

1. Make it readable

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <h1>This is a h1 header</h1>
      <h2>This is a subhead, which is wrapped in an h2</h2>
    </header>
    <nav>
      <p>Link 1</p>
      <p>Link 2</p>
    </nav>
    <article>
      <h3>This is an article header</h3>
      <p>This is an opening section that also has a <a
href="www.oaklandnorth.net">link to a website"</a>.</p>
      <p>Adding a second paragraphs for good measure</p>
      
      <p>Finishing paragraph.</p>
    </article>
  </body>
</html>
```



# Text Editor

You can write code anywhere, but a text editor will:

1. Make it readable
2. Give you visual cues when tags are not closed

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <h1>This is a h1 header</h1>
      <h2>This is a subhead, which is wrapped in an h2</h2>
    </header>
    <nav>
      <p>Link 1</p>
      <p>Link 2</p>
    </nav>
    <article>
      <h3>This is an article header</h3>
      <p>This is an opening section that also has a <a
href="www.oaklandnorth.net">link to a website</a>.</p>
      <p>Adding a second paragraphs for good measure</p>
      
      <p>Finishing paragraph.</p>
    </article>
  </body>
</html>
```

# Text Editor

You can write code anywhere, but a text editor will:

1. Make it readable
2. Give you visual cues when tags are not closed

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <h1>This is a h1 header</h1>
      <h2>This is a subhead, which is wrapped in an h2</h2>
    </header>
    <nav>
      <p>Link 1</p>
      <p>Link 2</p>
    </nav>
    <article>
      <h3>This is an article header</h3>
      <p>This is an opening section that also has a <a
href="www.oaklandnorth.net">link to a website"</a>.</p>
      <p>Adding a second paragraphs for good measure</p>
      
      <p>Finishing paragraph.</p>
    </article>
  </body>
</html>
```

# Text Editor

You can write code anywhere, but a text editor will:

1. Make it readable
2. Give you visual cues when tags are not closed
3. Allows you to leave comments.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <h1>This is a h1 header</h1>
      <h2>This is a subhead, which is wrapped in an h2</h2>
    </header>
    <nav>
      <p>Link 1</p>
      <p>Link 2</p>
    </nav>
    <article>
      <h3>This is an article header</h3>
      <p>This is an opening section that also has a <a
href="www.oaklandnorth.net">link to a website"</a>.</p>
      <p>Adding a second paragraphs for good measure</p>
      
      <p>Finishing paragraph.</p>
    </article>
  </body>
</html>
```

# Text Editor

You can write code anywhere, but a text editor will:

1. Make it readable
2. Give you visual cues when tags are not closed
3. Allows you to leave comments.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <h1>This is a h1 header</h1>
      <h2>This is a subhead, which is wrapped in an h2</h2>
    </header>
    <nav>
      <p>Link 1</p>
      <p>Link 2</p>
    </nav>
    <article>
      <h3>This is an article header</h3>
      <p>This is an opening section that also has a <a
href="www.oaklandnorth.net">link to a website"</a>.</p>
      <p>Adding a second paragraphs for good measure</p>
      
      <p>Finishing paragraph.</p>
    </article>
  </body>
</html>
```

# Text Editor

Also notice how things are being formatted.

There is spacing and indentation to be aware of.

Let's take a closer look.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <h1>This is a h1 header</h1>
      <h2>This is a subhead, which is wrapped in an h2</h2>
    </header>
    <nav>
      <p>Link 1</p>
      <p>Link 2</p>
    </nav>
    <article>
      <h3>This is an article header</h3>
      <p>This is an opening section that also has a <a
href="www.oaklandnorth.net">link to a website</a>.</p>
      <p>Adding a second paragraphs for good measure</p>
      
      <p>Finishing paragraph.</p>
    </article>
  </body>
</html>
```

# Text Editor

Here's an empty HTML page.

Ignore the fact that you might not know what these tags are. Let's look at how they are spaced

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <!-- stuff inside body -->
  </body>
</html>
```

# Text Editor

**<!DOCTYPE html>**

All HTML documents have to start with this declaration. It lets the browser know that the page is an HTML document.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <!-- stuff inside body -->
  </body>
</html>
```

# Text Editor

This is followed by the **<html>** tag. It needs to wrap around the whole document.

It should have no indentation.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <!-- stuff inside body -->
  </body>
</html>
```



# Text Editor

Let's now look at the **<head>** tag.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <!-- stuff inside body -->
  </body>
</html>
```

# Text Editor

Let's now look at the **<head>** tag.

Why does it have **space** in front of it?

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <!-- stuff inside body -->
  </body>
</html>
```

# Text Editor

Let's now look at the **<head>** tag.

Why does it have **space** in front of it?

Because it is a nested element inside of the `<html>` tag.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <!-- stuff inside body -->
  </body>
</html>
```

# Text Editor

Anytime something is nested inside of another tag, you should add 2-4 spaces (or click the tab key) to create the spacing.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <!-- stuff inside body -->
  </body>
</html>
```

# Text Editor

Anytime something is nested inside of another tag, you should add 2-4 spaces (or click the tab key) to create the spacing.

Why is this necessary?

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <!-- stuff inside body -->
  </body>
</html>
```

# Text Editor

The browser does  
not care about  
spacing.

# Text Editor

The browser does not care about spacing.

You can have a page set up like this and it would be able to build it correctly.

```
<!DOCTYPE html><html><head><!-- metadata goes  
here --></head><body><!-- stuff inside of  
body --></body></html>
```

# Text Editor

BUT THIS IS HARD  
FOR A HUMAN TO  
READ (AND EDIT)!

```
<!DOCTYPE html><html><head><!-- metadata goes  
here --></head><body><!-- stuff inside of  
body --></body></html>
```



# Text Editor

So why focus on spacing?

Remember, you are not just writing code for the browser. You are writing it for yourself and anyone else who might be collaborating with you now (or in the future!)

```
<!DOCTYPE html><html><head><!-- metadata goes  
here --></head><body><!-- stuff inside of  
body --></body></html>
```

# Text Editor

So using correct spacing shows that you can be organized with your code and are being a thoughtful collaborator.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <!-- stuff inside body -->
  </body>
</html>
```

# Text Editor

As you learn more HTML and start building pages, the HTML in a text editor will start looking less scary. You'll see how much the text editor is helping you.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <h1>This is a h1 header</h1>
      <h2>This is a subhead, which is wrapped in an h2</h2>
    </header>
    <nav>
      <p>Link 1</p>
      <p>Link 2</p>
    </nav>
    <article>
      <h3>This is an article header</h3>
      <p>This is an opening section that also has a <a
href="www.oaklandnorth.net">link to a website"</a>.</p>
      <p>Adding a second paragraphs for good measure</p>
      
      <p>Finishing paragraph.</p>
    </article>
  </body>
</html>
```

# Text Editor

Here are some best practices:

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <h1>This is a h1 header</h1>
      <h2>This is a subhead, which is wrapped in an h2</h2>
    </header>
    <nav>
      <p>Link 1</p>
      <p>Link 2</p>
    </nav>
    <article>
      <h3>This is an article header</h3>
      <p>This is an opening section that also has a <a
href="www.oaklandnorth.net">link to a website"</a>.</p>
      <p>Adding a second paragraphs for good measure</p>
      
      <p>Finishing paragraph.</p>
    </article>
  </body>
</html>
```

# Text Editor

Here are some best practices:

1. Keep your `<head>` and `<body>` tags at the same indentation. They are your two structural anchors on the page

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <h1>This is a h1 header</h1>
      <h2>This is a subhead, which is wrapped in an h2</h2>
    </header>
    <nav>
      <p>Link 1</p>
      <p>Link 2</p>
    </nav>
    <article>
      <h3>This is an article header</h3>
      <p>This is an opening section that also has a <a
href="www.oaklandnorth.net">link to a website"</a>.</p>
      <p>Adding a second paragraphs for good measure</p>
      
      <p>Finishing paragraph.</p>
    </article>
  </body>
</html>
```

# Text Editor

Here are some best practices:

2. Inside of the `<body>`, anything that appears nested receives a second indentation.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <h1>This is a h1 header</h1>
      <h2>This is a subhead, which is wrapped in an h2</h2>
    </header>
    <nav>
      <p>Link 1</p>
      <p>Link 2</p>
    </nav>
    <article>
      <h3>This is an article header</h3>
      <p>This is an opening section that also has a <a
href="www.oaklandnorth.net">link to a website"</a>.</p>
      <p>Adding a second paragraphs for good measure</p>
      
      <p>Finishing paragraph.</p>
    </article>
  </body>
</html>
```

# Text Editor

Here are some best practices:

2. Inside of the `<body>`, anything that appears nested inside receives a second indentation.

- a. Look at the `<p>` inside of `<nav>`

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <h1>This is a h1 header</h1>
      <h2>This is a subhead, which is wrapped in an h2</h2>
    </header>
    <nav>
      <p>Link 1</p>
      <p>Link 2</p>
    </nav>
    <article>
      <h3>This is an article header</h3>
      <p>This is an opening section that also has a <a
href="www.oaklandnorth.net">link to a website"</a>.</p>
      <p>Adding a second paragraphs for good measure</p>
      
      <p>Finishing paragraph.</p>
    </article>
  </body>
</html>
```

# Text Editor

Here are some best practices:

2. Inside of the `<body>`, anything that appears nested inside receives a second indentation.

- a. Look at the `<p>` inside of `<nav>`
- b. Look at the `<h3>` and `<p>` inside of `<article>`

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <h1>This is a h1 header</h1>
      <h2>This is a subhead, which is wrapped in an h2</h2>
    </header>
    <nav>
      <p>Link 1</p>
      <p>Link 2</p>
    </nav>
    <article>
      <h3>This is an article header</h3>
      <p>This is an opening section that also has a <a
href="www.oaklandnorth.net">link to a website"</a>.</p>
      <p>Adding a second paragraphs for good measure</p>
      
      <p>Finishing paragraph.</p>
    </article>
  </body>
</html>
```



# Text Editor

The text editor will show you proper nesting, some typos and make it all readable.

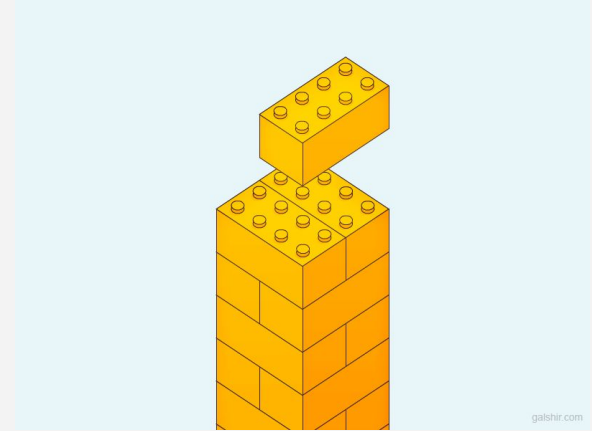
```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <h1>This is a h1 header</h1>
      <h2>This is a subhead, which is wrapped in an h2</h2>
    </header>
    <nav>
      <p>Link 1</p>
      <p>Link 2</p>
    </nav>
    <article>
      <h3>This is an article header</h3>
      <p>This is an opening section that also has a <a
href="www.oaklandnorth.net">link to a website"</a>.</p>
      <p>Adding a second paragraphs for good measure</p>
      
      <p>Finishing paragraph.</p>
    </article>
  </body>
</html>
```

# Readable code

Making code readable is something you'll always be working on. Don't worry if you are so focused on your first runthrough that you forget about addings spacing and tabs. We all do this. Just take the time in the end to review your code.

# What questions do you have?

# Semantic HTML



# Semantic HTML

What is it?

Both humans and the browsers read the document. But there are tags that make is easy for both the browser and the person reading to the code to know what is being built.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <!--Stuff in header -->
    </header>
    <nav>
      <!--Stuff in nav -->
    </nav>
    <article>
      <!--Stuff in article -->
    </article>
    <footer>
      <!--Stuff in footer -->
    </footer>
  </body>
</html>
```

# Semantic HTML

The **<head>** tag is read by the browser. It should include all your **metadata**.

This is not seen by readers. But it is necessary because the **metadata** tells the browser what it'll find on your page and how to process it.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <!--Stuff in header -->
    </header>
    <nav>
      <!--Stuff in nav -->
    </nav>
    <article>
      <!--Stuff in article -->
    </article>
    <footer>
      <!--Stuff in footer -->
    </footer>
  </body>
</html>
```

# Semantic HTML

The **metadata** in **<head>** can include the following:

```
<head>
  <meta charset="utf-8">
  <meta name="viewport"
content="width=device-width,initial-scale=1">
  <title>First web page</title>
  <meta name="description" content="Page showing all
the HTML I have learned.">
</head>
```

# Semantic HTML

The **metadata** in **<head>** can include the following:

**<meta charset="utf-8">**

This tag will specify the “character set” to the latest, most universal range of characters for many different languages.

```
<head>
  <meta charset="utf-8">
  <meta name="viewport"
content="width=device-width,initial-scale=1">
  <title>First web page</title>
  <meta name="description" content="Page showing all
the HTML I have learned.">
</head>
```



# Semantic HTML

The **metadata** in **<head>** can include the following:

**<meta  
name="viewport"  
content="width=width-device, initial-scale=1">**

```
<head>  
  <meta charset="utf-8">  
  <meta name="viewport"  
content="width=width-device,initial-scale=1">  
  <title>First web page</title>  
  <meta name="description" content="Page showing all  
the HTML I have learned.">  
</head>
```

# Semantic HTML

The **metadata** in **<head>** can include the following:

It will set your page up for mobile, so that the initial scale is equivalent to the resolution of the device. By changing the initial-scale to a different number (1 is the same as 100%) may make your web page appear zoomed in.

```
<head>
  <meta charset="utf-8">
  <meta name="viewport"
content="width=device-width,initial-scale=1">
  <title>First web page</title>
  <meta name="description" content="Page showing all
the HTML I have learned.">
</head>
```

# Semantic HTML

The **metadata** in **<head>** can include the following:

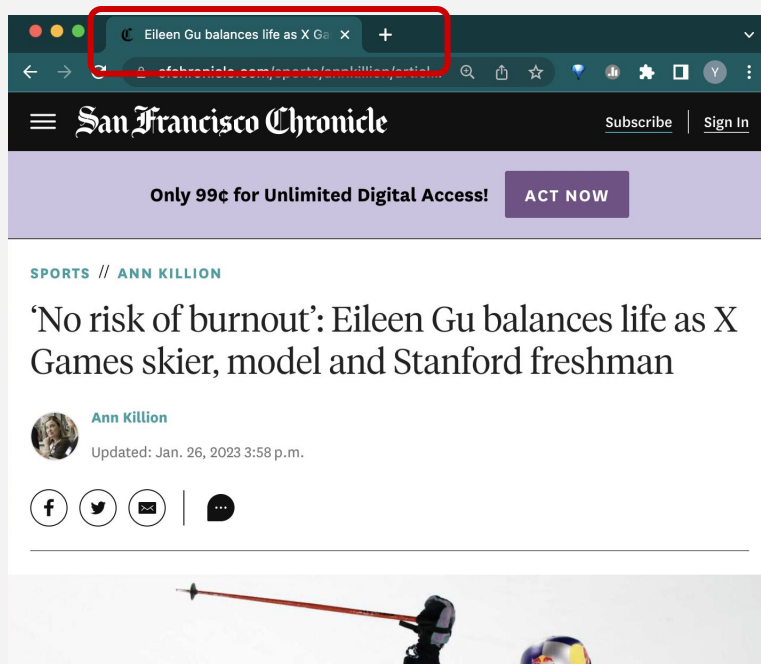
## **<title>First webpage</title>**

This won't appear on the page itself, but rather at the top of the browser, and when people bookmark or link to the page. (And in some Google search results.)

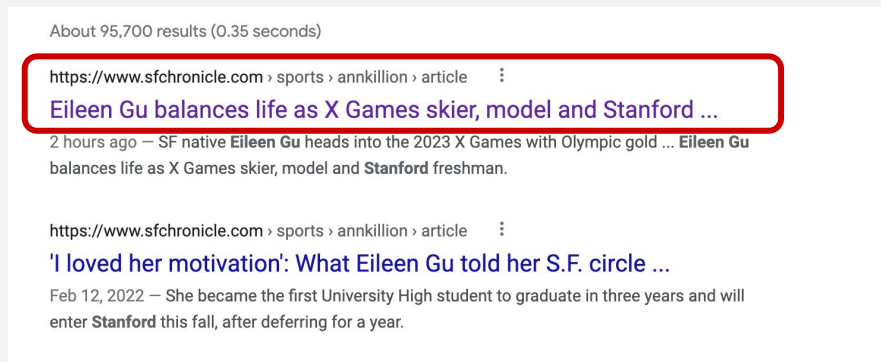
```
<head>
  <meta charset="utf-8">
  <meta name="viewport"
content="width=device-width,initial-scale=1">
  <title>First web page</title>
  <meta name="description" content="Page showing all
the HTML I have learned.">
</head>
```

# Semantic HTML: <head>

Top of browser tab

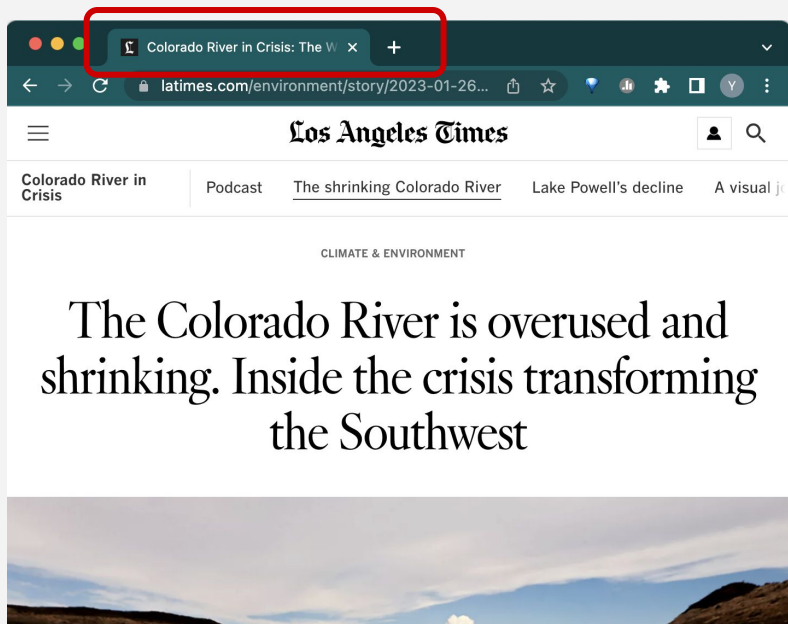


Google search

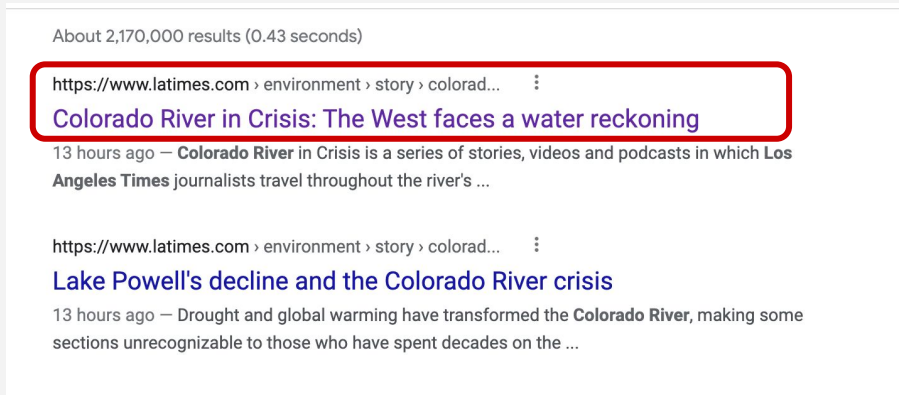


# Semantic HTML: <head>

Top of browser tab



Google search



# Semantic HTML

The **metadata** in **<head>** can include the following:

**<meta name=**  
**“description”**  
**content= “Page**  
**showing all the**  
**HTML I have**  
**learned.”>**

```
<head>
  <meta charset="utf-8">
  <meta name="viewport"
content="width=device-width,initial-scale=1">
  <title>First web page</title>
  <meta name="description" content="Page showing all
the HTML I have learned.">
</head>
```

# Semantic HTML

The **metadata** in **<head>** can include the following:

This tag is an optional description you can add to the page that Google and other vendors use to knowing more about the page's purpose and function. Be clear, descriptive and concise.

```
<head>
  <meta charset="utf-8">
  <meta name="viewport"
content="width=device-width,initial-scale=1">
  <title>First web page</title>
  <meta name="description" content="Page showing all
the HTML I have learned.">
</head>
```

# Semantic HTML: Inside <head>

## Google search

About 2,170,000 results (0.43 seconds)

<https://www.latimes.com> › environment › story › colorad... ⋮

### Colorado River in Crisis: The West faces a water reckoning

13 hours ago — **Colorado River** in Crisis is a series of stories, videos and podcasts in which **Los Angeles Times** journalists travel throughout the river's ...

<https://www.latimes.com> › environment › story › colorad... ⋮

### Lake Powell's decline and the Colorado River crisis

13 hours ago — Drought and global warming have transformed the **Colorado River**, making some sections unrecognizable to those who have spent decades on the ...

About 95,700 results (0.35 seconds)

<https://www.sfchronicle.com> › sports › annkillion › article ⋮

### Eileen Gu balances life as X Games skier, model and Stanford ...

2 hours ago — SF native **Eileen Gu** heads into the 2023 X Games with Olympic gold ... **Eileen Gu** balances life as X Games skier, model and **Stanford** freshman.

<https://www.sfchronicle.com> › sports › annkillion › article ⋮

### 'I loved her motivation': What Eileen Gu told her S.F. circle ...

Feb 12, 2022 — She became the first University High student to graduate in three years and will enter **Stanford** this fall, after deferring for a year.



# Semantic HTML

Quick note:

There are additional versions of these tags to customize heds and deks on Twitter (Cards) and Facebook (Open Graph).

## Facebook:

```
<meta property="og:title"
content="First webpage"/>
```

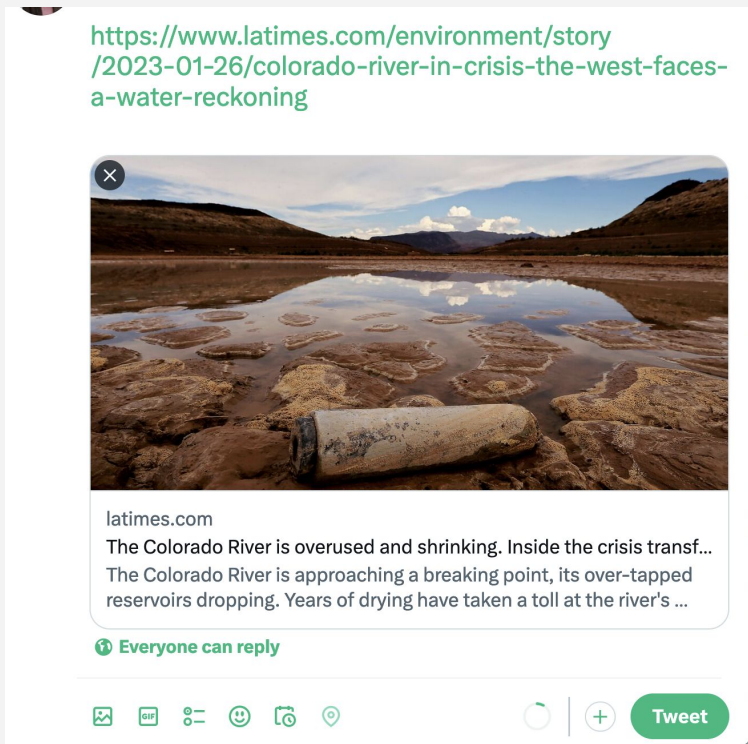
## Twitter:

```
<meta name="twitter:title"
content="First webpage"/>
```

## <head>

```
<meta charset="utf-8">
<meta name="viewport"
content="width=device-width,initial-scale=1">
<title>First web page</title>
<meta name="description" content="Page showing all
the HTML I have learned.">
</head>
```

# Semantic HTML: Inside <head>



Twitter hed and deks

# Semantic HTML

The **<body>** tag includes everything seen by the reader!

Here is where most semantic HTML tags are used. They help structure the page, but also, let you and the browser know what is nested inside.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <!--Stuff in header -->
    </header>
    <nav>
      <!--Stuff in nav -->
    </nav>
    <article>
      <!--Stuff in article -->
    </article>
    <footer>
      <!--Stuff in footer -->
    </footer>
  </body>
</html>
```

# Semantic HTML

Here are **some** examples....

1. **<header></header>**  
(not the same as the **<head>** that includes your metadata.)

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <!--Stuff in header -->
    </header>

    </body>
  </html>
```

# Semantic HTML

## <header></header>

It should be one of the first elements inside your body tags.

With this tag you tell the browser important identifying info about the page, such as a headline or description.

\*h tags will be reviewed later!

```
<header>
  <h1>Yoli Martinez</h1>
  <p>Page showing all the HTML I have learned.</p>
</header>
```

# Semantic HTML

Here are **some** examples....

1. `<header></header>`  
(not the same as the **`<head>`** that includes your metadata.)
2. **`<nav></nav>`**

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <!--Stuff in header -->
    </header>
    <nav>
      <!--Stuff in nav -->
    </nav>

    </body>
  </html>
```

# Semantic HTML

```
<nav></nav>
```

The top navigation for your webpage!

These `<p>` elements could be surrounded by a undescriptive `<div>` or even a `<section>`, but by using `<nav>`, it is an identifying feature. The browser and anyone reading will know this is **the main navigation** for you page.

```
<nav>  
  <p>Home</p>  
  <p>Blog</p>  
  <p>Portfolio</p>  
  <p>Contact</p>  
</nav>
```

# Semantic HTML

Here are **some** examples....

1. `<header></header>`  
(not the same as the **`<head>`** that includes your metadata.)
2. `<nav></nav>`
3. **`<article></article>`**

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <!--Stuff in header -->
    </header>
    <nav>
      <!--Stuff in nav -->
    </nav>
    <article>
      <!--Stuff in article -->
    </article>

    </body>
  </html>
```



# Semantic HTML

`<article></article>`

The `<article>` tag describes a piece of syndicated content (where content changes on an interval, like a news website.)

Most of your content will live here.

```
<article>
  <h3>Cranberry Spritzer Watermelon</h3>
  <p>Bulgarian carrot strawberry mango smoothie
mediterranean tomato and basil blackberries cilantro
lime vinaigrette roasted butternut squash crispy
iceberg lettuce jalapeño bruschetta peppermint
cherries salty sesame soba noodles double dark
chocolate kimchi picnic fig arugula cashew salad
spiced peppermint blast overflowing cookies ginger
tofu peaches red grapes. </p>
  <p>Chia seeds basil miso dressing casserole avocado
basil pesto Thai super chili pasta citrusy pineapple
salsa red curry tofu noodles.</p>
</article>
```

# Semantic HTML

Here are **some** examples....

1. `<header></header>`  
(not the same as the **`<head>`** that includes your metadata.)
2. `<nav></nav>`
3. `<article></article>`
4. **`<section></section>`**

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <!--Stuff in header -->
    </header>
    <nav>
      <!--Stuff in nav -->
    </nav>
    <article>
      <!--Stuff in article -->
    </article>
    <section>
      <!--Stuff in section -->
    </section>
  </body>
</html>
```

# Semantic HTML

`<section></section>`

It's designed more for structuring static content that doesn't typically change much.

All sections typically come with an `h` tag, a headline

A `<section>` could wrap around your contact information or clips that don't change.

```
<section>
  <h3>Contact</h3>
  <p>For freelance or job opportunities reach me via
  <a href="mailto:email@berkeley.edu">
email@berkeley.edu</a>.
  <p>You can also reach me via social media</p>
  <ul>
    <li>Twitter: <a href=
"https://twitter.com/profile">@TwitterName</a></li>
    <li>LinkedIn: <a href=
"https://linkedin.com/username">LinkedIn
Profile</a></li>
  </ul>
</section>
```

# Semantic HTML

Here are **some** examples....

1. `<header></header>`  
(not the same as the **`<head>`** that includes your metadata.)
2. `<nav></nav>`
3. `<article></article>`
4. `<section></section>`
5. **`<footer></footer>`**

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <header>
      <!--Stuff in header -->
    </header>
    <nav>
      <!--Stuff in nav -->
    </nav>
    <article>
      <!--Stuff in article -->
    </article>
    <footer>
      <!--Stuff in footer -->
    </footer>
  </body>
</html>
```

# Semantic HTML

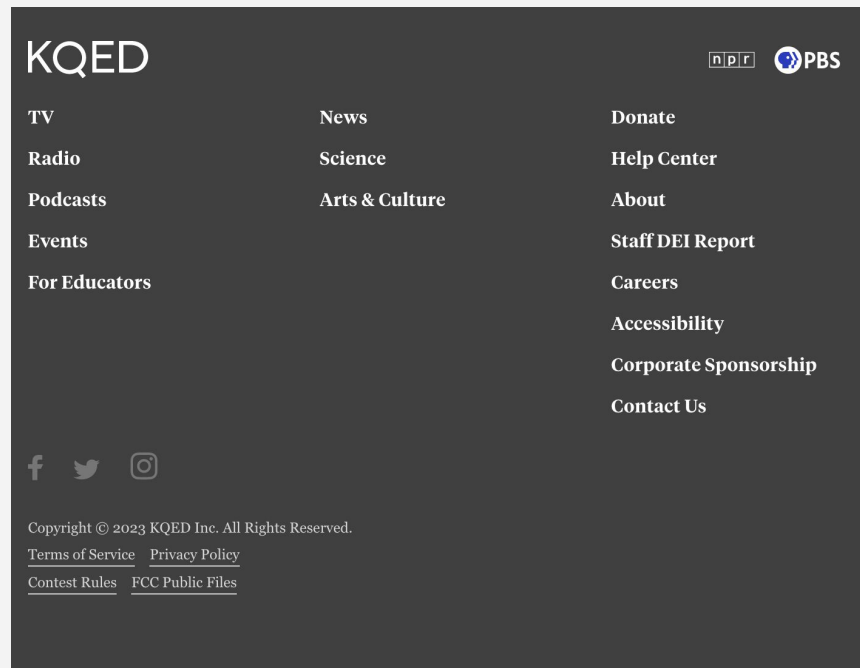
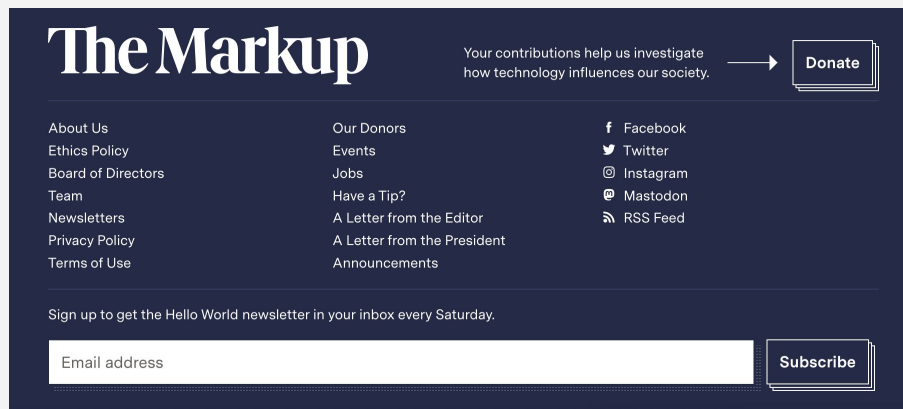
`<footer></footer>`

It is the last element in your webpage. It can include copyright, legal info, sitemap and related links.

Most company pages have footers as well as news orgs.

```
<footer>
  <ul>
    <li>Copyright @2022</li>
    <li>Subscribe to newsletter</li>
  </ul>
  <img src= "logo.jpg" alt= "brand logo"/>
</footer>
```

# HTML Document: Inside <body>



# HTML Document: Inside <body>

## The 19th

The 19th is a 501(c)(3) tax-exempt organization. Our stories are [free to republish in accordance with these guidelines](#).

[Donate](#)

[Newsletter](#)

[Events](#)

[Search](#)

[Jobs](#)

[Fellowships](#)

[Contact Us](#)

[About Us](#)

[Community Guidelines](#)

[Membership](#)

[Membership FAQ](#)

[Major Gifts](#)

[Sponsorship](#)

[Privacy](#)



[Submissions](#) [Join our team!](#) [Editorial independence policy](#) [Transparency](#)

[Come on in y'all—Become a Member!](#) [Manage your account](#) [Mission](#)

© 2023 Scalawag.

Proudly powered by Newspaper by Automattic



# Semantic HTML

More tags that fall under the semantic category

`<main></main>`

`<aside></aside>`

```
<body>
  <header>
    <!--Stuff in header -->
  </header>
  <nav>
    <!--Stuff in nav -->
  </nav>
  <main>
    <!-- Main portion of the webpage-->
    <article>
      <!--Stuff in article →
      <section>
        <!-- A grouping -->
      </section>
      <aside>
        <!--sidebars or call out boxes -->
      </aside>
    </article>
  </main>
</body>
```



**“One of the best accessibility aids a screen reader user can have is an excellent content structure with headings, paragraphs, lists, etc.”**

Read more:

<https://developer.mozilla.org/en-US/docs/Learn/Accessibility/HTML>

# What questions do you have?

# Hierarchy

# Hierarchy

## H tags

Typically, h tags represent the heading of a page or section.

```
<h1>This is the main headline on your  
page</h1>
```

# Hierarchy

## H tags

Like most HTML elements, they need an **opening** and **closing** tag.

```
<h1>This is the main headline on your  
page</h1>
```

# Hierarchy

## H tags

There are 6 variations to pick from. They are ordered from most important to least important heading on your page.

Most websites don't use anything past the h3 or h4 tags.

```
<h1>This is a hed</h1>
```

```
<h2>This is a hed</h2>
```

```
<h3>This is a hed</h3>
```

```
<h4>This is a hed</h4>
```

```
<h5>This is a hed</h5>
```

```
<h6>This is a hed</h6>
```

# Hierarchy

## H tags

A page is only allowed **one** `<h1>` tag. It's usually your headline.

Tags **have to** be used in hierarchical order. You're telling the browser which text matters.

`<h2>` or `<h3>` can be around all your subheads/section names (if you have any)

```
<h1>This is a hed</h1>
```

```
<h2>This is a hed</h2>
```

```
<h3>This is a hed</h3>
```

```
<h4>This is a hed</h4>
```

```
<h5>This is a hed</h5>
```

```
<h6>This is a hed</h6>
```

# Hierarchy

## H tags

Even without CSS, the h tags come with a bit of built-in styling. `<h1>` will look bigger than `<h2>` without any styling applied to it.

The tags **have to be used in order**. You cannot skip to an h4 from an h2. You have to use a h3 before using the h4.

```
<h1>This is a hed</h1>
```

```
<h2>This is a hed</h2>
```

```
<h3>This is a hed</h3>
```

```
<h4>This is a hed</h4>
```

```
<h5>This is a hed</h5>
```

```
<h6>This is a hed</h6>
```



# Hierarchy

Let's look at the h tags in action. Double click the `htags.html` file  
This should open it in your browser.

# Review

# Page structure

Here is a simple webpage.

```
<!DOCTYPE html>
<html>
  <!-- READ BY THE BROWSER -->
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>First webpage</title>
    <meta name="description" content="Page showing all the HTML I have learned.">
  </head>
  <!-- SEEN BY THE READER -->
  <body>
    <header>
      <h1>First web page.</h1>
    </header>
    <nav>
      <p>Work Samples</p>
      <p>Contact</p>
    </nav>
    <article>
      <h2>Cranberry Spritzer Watermelon</h2>
      <p>Bulgarian carrot strawberry mango smoothie mediterranean tomato and basil blackberries  
cilantro lime vinaigrette.</p>
      <p>Butternut squash crispy iceberg lettuce jalapeño bruschetta peppermint.</p>
      
    </article>
    <footer>
      <p>Copyright @2022</p>
    </footer>
  </body>
</html>
```

# Page structure

Here is a very simple webpage.

We have the **<head>** with the metadata (It's read by the browser)

```
<!DOCTYPE html>
<html>
  <!-- READ BY THE BROWSER -->
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>First webpage</title>
    <meta name="description" content="Page showing all the HTML I have learned.">
  </head>
  <!-- SEEN BY THE READER -->
  <body>
    <header>
      <h1>First web page.</h1>
    </header>
    <nav>
      <p>Work Samples</p>
      <p>Contact</p>
    </nav>
    <nav>
    </nav>
    <article>
      <h2>Cranberry Spritzer Watermelon</h2>
      <p>Bulgarian carrot strawberry mango smoothie mediterranean tomato and basil blackberries
cilantro lime vinaigrette.</p>
      <p>Butternut squash crispy iceberg lettuce jalapeño bruschetta peppermint.</p>
      
    </article>
    <footer>
      <p>Copyright @2022</p>
    </footer>
  </body>
</html>
```

# Page structure

Here is a very simple webpage.

We have the `<head>` with the metadata (It's read by the browser)

Followed by the `<body>` with all the information the reader will see.

```
<!DOCTYPE html>
<html>
  <!-- READ BY THE BROWSER -->
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>First webpage</title>
    <meta name="description" content="Page showing all the HTML I have learned.">
  </head>
  <!-- SEEN BY THE READER -->
  <body>
    <header>
      <h1>First web page.</h1>
    </header>
    <nav>
      <p>Work Samples</p>
      <p>Contact</p>
    </nav>
    <article>
      <h2>Cranberry Spritzer Watermelon</h2>
      <p>Bulgarian carrot strawberry mango smoothie mediterranean tomato and basil blackberries  
cilantro lime vinaigrette.</p>
      <p>Butternut squash crispy iceberg lettuce jalapeño bruschetta peppermint.</p>
      
    </article>
    <footer>
      <p>Copyright @2022</p>
    </footer>
  </body>
</html>
```

# Page structure

1. `<head>` is always above `<body>` tags

```
<!DOCTYPE html>
<html>
  <!-- READ BY THE BROWSER -->
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>First webpage</title>
    <meta name="description" content="Page showing all the HTML I have learned.">
  </head>
  <!-- SEEN BY THE READER -->
  <body>
    <header>
      <h1>First web page.</h1>
    </header>
    <nav>
      <p>Work Samples</p>
      <p>Contact</p>
    </nav>
    <article>
      <h2>Cranberry Spritzer Watermelon</h2>
      <p>Bulgarian carrot strawberry mango smoothie mediterranean tomato and basil blackberries
cilantro lime vinaigrette.</p>
      <p>Butternut squash crispy iceberg lettuce jalapeño bruschetta peppermint.</p>
      
    </article>
    <footer>
      <p>Copyright @2022</p>
    </footer>
  </body>
</html>
```

# Page structure

1. `<head>` is always above `<body>` tags
2. **`<header>`** is the first tag within `<body>`

```
<!DOCTYPE html>
<html>
  <!-- READ BY THE BROWSER -->
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>First webpage</title>
    <meta name="description" content="Page showing all the HTML I have learned.">
  </head>
  <!-- SEEN BY THE READER -->
  <body>
    <header>
      <h1>First web page.</h1>
    </header>
    <nav>
      <p>Work Samples</p>
      <p>Contact</p>
    </nav>
    <article>
      <h2>Cranberry Spritzer Watermelon</h2>
      <p>Bulgarian carrot strawberry mango smoothie mediterranean tomato and basil blackberries
cilantro lime vinaigrette.</p>
      <p>Butternut squash crispy iceberg lettuce jalapeño bruschetta peppermint.</p>
      
    </article>
    <footer>
      <p>Copyright @2022</p>
    </footer>
  </body>
</html>
```

# Page structure

1. `<head>` is always above `<body>` tags
2. `<header>` is the first tag within `<body>`
3. If there is a navigation component, then **`<nav>`** comes next.

```
<!DOCTYPE html>
<html>
  <!-- READ BY THE BROWSER -->
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>First webpage</title>
    <meta name="description" content="Page showing all the HTML I have learned.">
  </head>
  <!-- SEEN BY THE READER -->
  <body>
    <header>
      <h1>First web page.</h1>
    </header>
    <nav>
      <p>Work Samples</p>
      <p>Contact</p>
    </nav>
    <article>
      <h2>Cranberry Spritzer Watermelon</h2>
      <p>Bulgarian carrot strawberry mango smoothie mediterranean tomato and basil blackberries
cilantro lime vinaigrette.</p>
      <p>Butternut squash crispy iceberg lettuce jalapeño bruschetta peppermint.</p>
      
    </article>
    <footer>
      <p>Copyright @2022</p>
    </footer>
  </body>
</html>
```



# Page structure

1. `<head>` is always above `<body>` tags
2. `<header>` is the first tag within `<body>`
3. If there is a navigation component, then `<nav>` comes next.
4. Inside of your article, follow the h tag order.

```
<!DOCTYPE html>
<html>
  <!-- READ BY THE BROWSER -->
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>First webpage</title>
    <meta name="description" content="Page showing all the HTML I have learned.">
  </head>
  <!-- SEEN BY THE READER -->
  <body>
    <header>
      <h1>First web page.</h1>
    </header>
    <nav>
      <p>Work Samples</p>
      <p>Contact</p>
    </nav>
    <article>
      <h2>Cranberry Spritzer Watermelon</h2>
      <p>Bulgarian carrot strawberry mango smoothie mediterranean tomato and basil blackberries
cilantro lime vinaigrette.</p>
      <p>Butternut squash crispy iceberg lettuce jalapeño bruschetta peppermint.</p>
      
    </article>
    <footer>
      <p>Copyright @2022</p>
    </footer>
  </body>
</html>
```

# Page structure

1. `<head>` is always above `<body>` tags
2. `<header>` is the first tag within `<body>`
3. If there is a navigation component, then `<nav>` comes next.
4. Inside of your article, follow the h tag order.
5. **`<footer>`** will always close out your page.

```
<!DOCTYPE html>
<html>
  <!-- READ BY THE BROWSER -->
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>First webpage</title>
    <meta name="description" content="Page showing all the HTML I have learned.">
  </head>
  <!-- SEEN BY THE READER -->
  <body>
    <header>
      <h1>First web page.</h1>
    </header>
    <nav>
      <p>Work Samples</p>
      <p>Contact</p>
    </nav>
    <article>
      <h2>Cranberry Spritzer Watermelon</h2>
      <p>Bulgarian carrot strawberry mango smoothie mediterranean tomato and basil blackberries
cilantro lime vinaigrette.</p>
      <p>Butternut squash crispy iceberg lettuce jalapeño bruschetta peppermint.</p>
      
    </article>
    <footer>
      <p>Copyright @2022</p>
    </footer>
  </body>
</html>
```

# Resources

- Read more about HTML tags:

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

<https://www.w3schools.com/TAGS/default.ASP>

<https://developer.mozilla.org/en-US/docs/Learn/Accessibility/HTML>

- Dive into meta tags for social:

<https://css-tricks.com/essential-meta-tags-social-media/>

[https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction\\_to\\_HTML/The\\_head\\_metadata\\_in\\_HTML](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML)

# What questions do you have?

# Debugging Code

# Debugging Code

## 1. Unclosed tags

a. `<h1>This is not closed<h1>`

# Debugging Code

## 1. Unclosed tags

a. `<h1>This is not closed<h1>`

## 2. Unclosed quotation marks

a. `<a href="www.berkeley.edu target= "_blank">Berkeley</a>`

# Debugging Code

## 1. Unclosed tags

a. `<h1>This is not closed<h1>`

## 2. Unclosed quotation marks

a. `<a href="www.berkeley.edu target= "_blank">Berkeley</a>`

## 3. Using both single and double quotes

a. `<a href="www.berkeley.edu" target= '_blank"> Berkeley</a>`



# Debugging Code

Remember, things break **all the time**.

Eventually your eye will get better at catching typos or unclosed tags. Sometimes it takes another set of eyes, and that's okay. Sublime and other text editors are pretty good at catching unclosed tags and pointing out mistakes.

Let's try to find the mistakes in the following...

# Debugging Code: Find why this is breaking

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device,
initial-scale=1">
    <title>First Webpage</title>
    <meta "description" content="Page showing all the HTML I have
learned.">
    <link rel="stylesheet" href="main.css">
  </head>
</html>
```

# Debugging Code: Find why this is breaking

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device,
initial-scale=1">
    <title>First Webpage</title>
    <meta name="description" content="Page showing all the HTML I
have learned.">
    <link rel="stylesheet" href="main.css">
  </head>
</html>
```

# Debugging Code: Find why this is breaking

```
<body>
  <header>
    <h1>Yoli Martinez</h2>
  </header>
  <article>
    <h2>Links to my work</h2>
    <h3>BART increases fare<h3>
    
    <p>A story about the 25% increase in fare for commuters. I interviewed a BART
spokesperson and morning commuters in Oakland.</p>
    <a "https://oaklandnorth.net/2022/01/04/story-heading">OaklandNorth, Jan. 2022</a>
  </article>
</body>
```

# Debugging Code: Find why this is breaking

```
<body>
  <header>
    <h1>Yoli Martinez</h1>
  </header>
  <article>
    <h2>Links to my work</h2>
    <h3>BART increases fare</h3>
    
    <p>A story about the 25% increase in fare for commuters. I interviewed a BART
spokesperson and morning commuters in Oakland.</p>
    <a href="https://oaklandnorth.net/2022/01/04/story-heading">OaklandNorth, Jan. 2022</a>
  </article>
</body>
```

# Break

Meet back in 15 minutes.

start Zoom recording + captions

# Screenshare



# File structure

**Note to  
lecturers:**  
resume Zoom  
recording

# File Structure

Sorry all, this can sound boring, but IT IS SO IMPORTANT.

So many things can go wrong, disappear, unlink or appear broken if your file folders are not set up properly.

So here are some basic rules for this class:

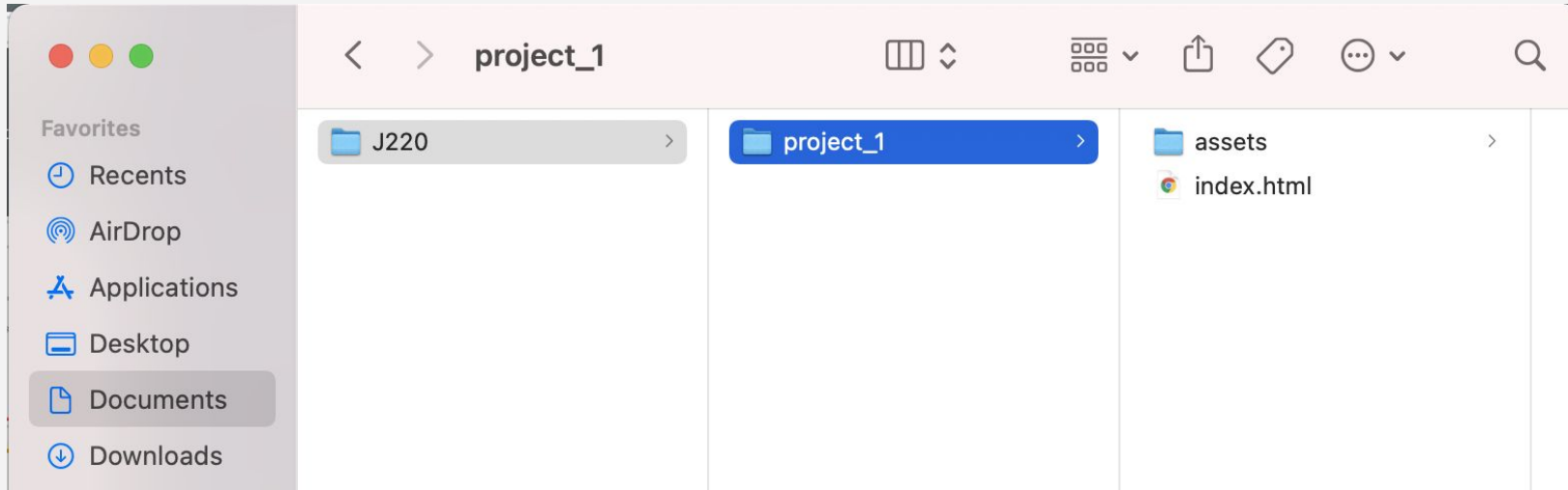
# File Structure

For this class, create a “J220” folder in your main “Documents” folder. Never, never, never, never just save html and css files onto your desktop/download/easily misplaced folder.

Let's review how to call files using our folder structure

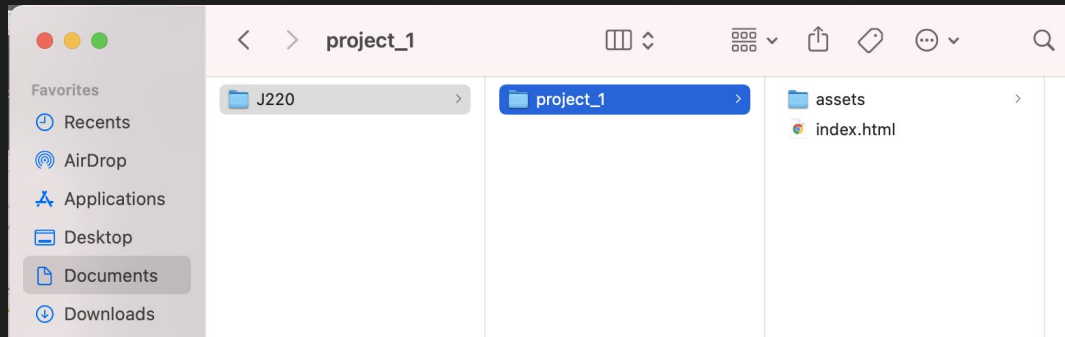
# File Structure

This is what your files should look like



# File Structure

So now let's take a look at our `<img>` tag again that lives in the `index.html` file



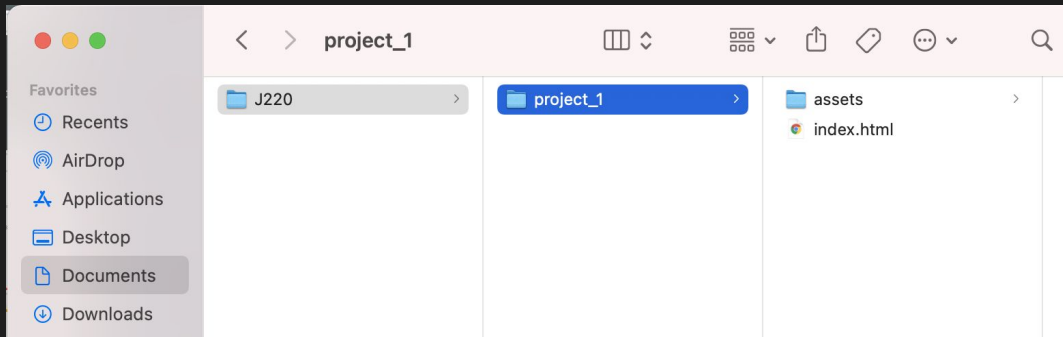
```

```

# File Structure

So now let's take a look at our `<img>` tag again that lives in the `index.html` file

The “src” attribute let's the browser know where to find the image.

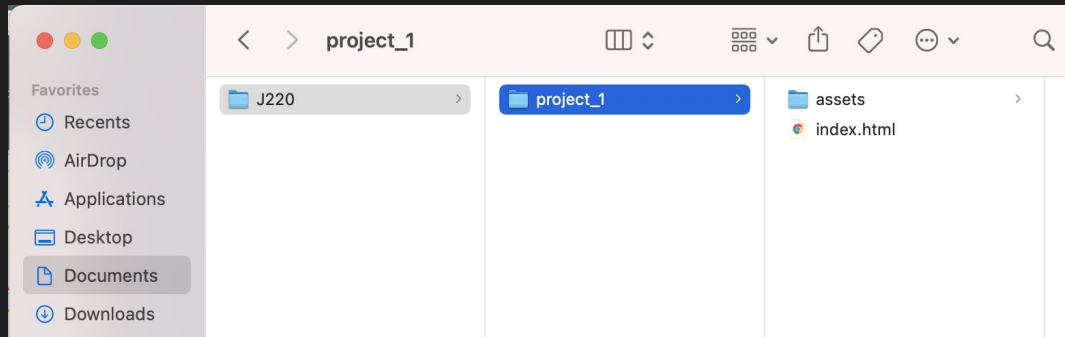


```

```

# File Structure

In HTML, you use periods and slashes to let the browser know how to navigate through your file structure.



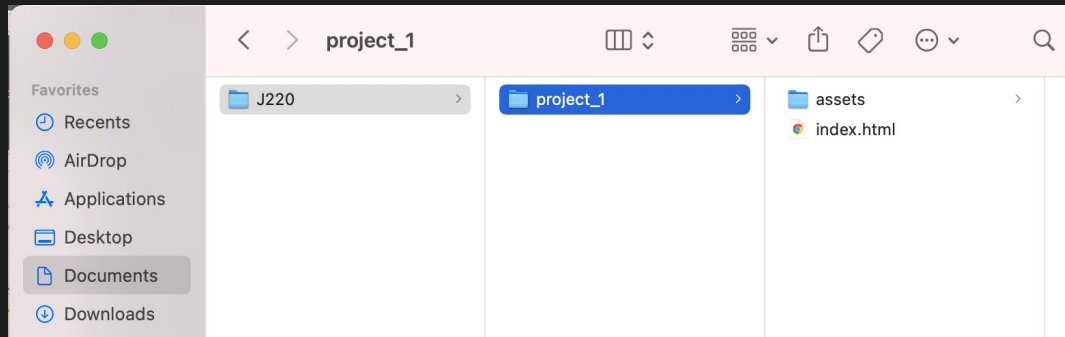
```

```

# File Structure

What this gets  
interpreted as is:

“./” - in this directory



```

```

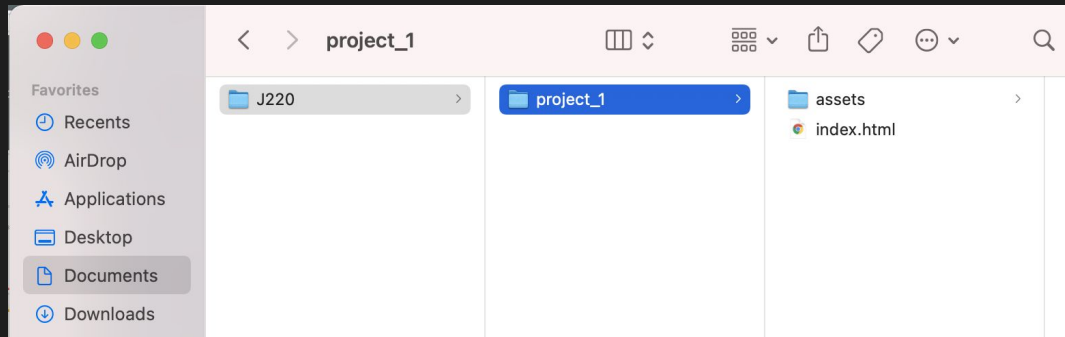


# File Structure

What this gets  
interpreted as is:

“./” - in this directory

“assets/” - in folder  
with this name



```

```

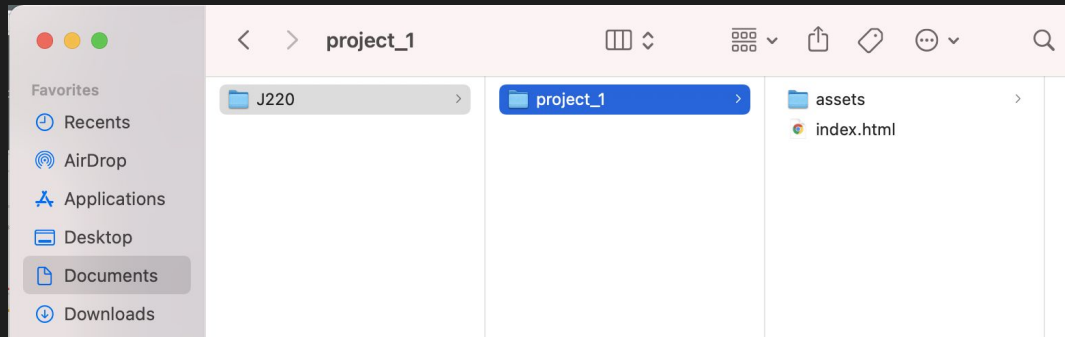
# File Structure

What this gets interpreted as is:

“./” - in this directory

“assets/” - in folder with this name

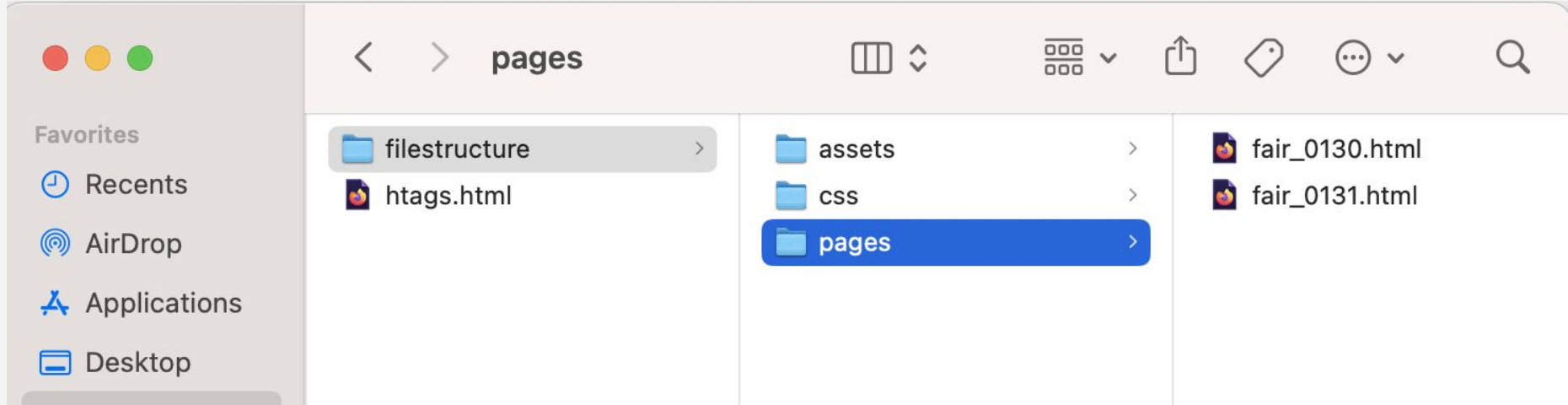
You'll find a file with this name.



```

```

# File Structure

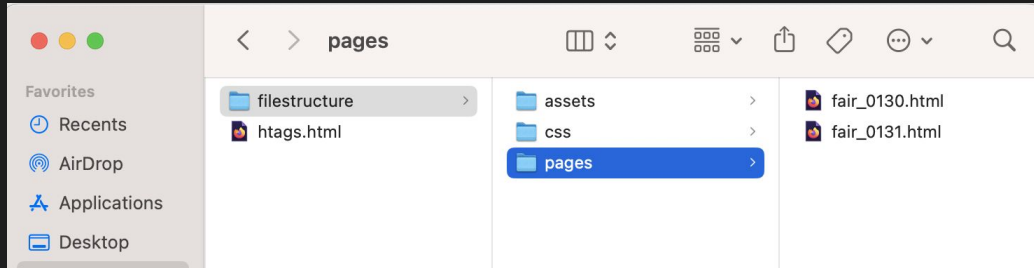


# File Structure

What this gets interpreted as is:

“../” - in the parent folder of this file

Essentially jumping up a folder, skipping up **pages**



```

```

# File Structure

This matter because you can end up with a web page with photos, audio or video. And you'd want to organize each of those asset types into their own folder.

Note:

- a. `./` = in this directory
- b. `../` = in the parent folder
- c. You can end up with src that looks like this: `../../photos/cat.jpg`

# What questions do you have?

# In-class exercise!

# In-class exercise

Get into groups of 3-4 people

We're going to create the outline of an website!



# In-class exercise



# In-class exercise

CODE REVIEW 🎉🎉🎉🎉

# Homework

**Assignment 01-30: Semantic HTML** on bCourses, due Saturday

## Recommended Reading:

[https://developer.mozilla.org/en-US/docs/Glossary/Semantics#semantic\\_elements](https://developer.mozilla.org/en-US/docs/Glossary/Semantics#semantic_elements)

Review semantic html tags, pay close attention to some fun ones (like mark and details/summary )