Class 3: Introduction to CSS

- Review: Structuring Content with HTML
- Using CSS to Style Structured Content (HTML)
- Common CSS Properties
- Additional CSS Rules
- CSS Selectors
- CSS Best Practices & Gotchas

Review: Structuring Content with HTML

Review: Static Websites

Website:

> A website is a collection of web pages (HTML files) and related content.

Static Website:

> A website with fixed content; the website is the same for all users.

Review: HTML

HTML: **H**yper**T**ext **M**arkup **L**anguage

HTML describes the structure of web pages.

HTML is a **markup** language.

Review: HTML Structures Content

Structure Content

- **Structural** Markup
  - “This is a heading” `<h2>`
  - “This is an aside” `<aside>`
  - “An abbreviation” `<abbr>`
- **Semantic** Markup
  - “Emphasized this” `<em>`
  - “Strong importance” `<strong>`

Do Not Style Content

- *Styling* Markup – **Never!**
  - ~~“Bold”~~ `<b>`
  - ~~“Italics”~~ `<i>`
  - ~~“Underlined”~~ `<u>`
  - ~~“Extra space”~~ `<br>`

HTML Elements

Definition: An **element** is a piece of content that is structured with an opening and closing tag.

Identify an **appropriate** HTML **element**.

Review: Define Structure Using HTML Tags

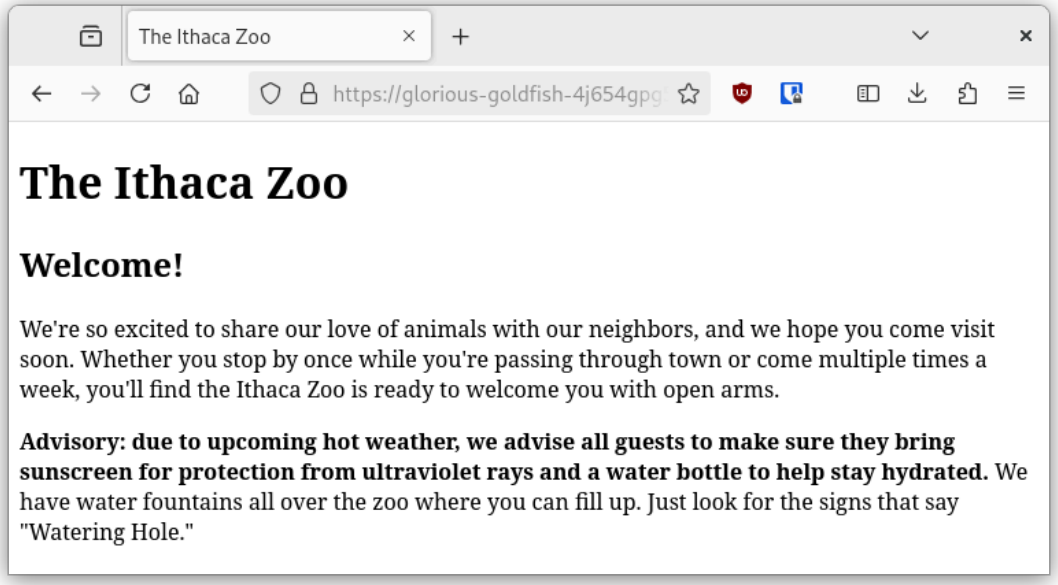
HTML provides **structure** to content with `<tags>`.

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

Opening Tag: `<p>`
Closing Tag: `</p>`

Review: HTML *Default Styling*

The browser will *style* your content by default. **Ignore it!**
(It means nothing!)



Demo: Default Styling

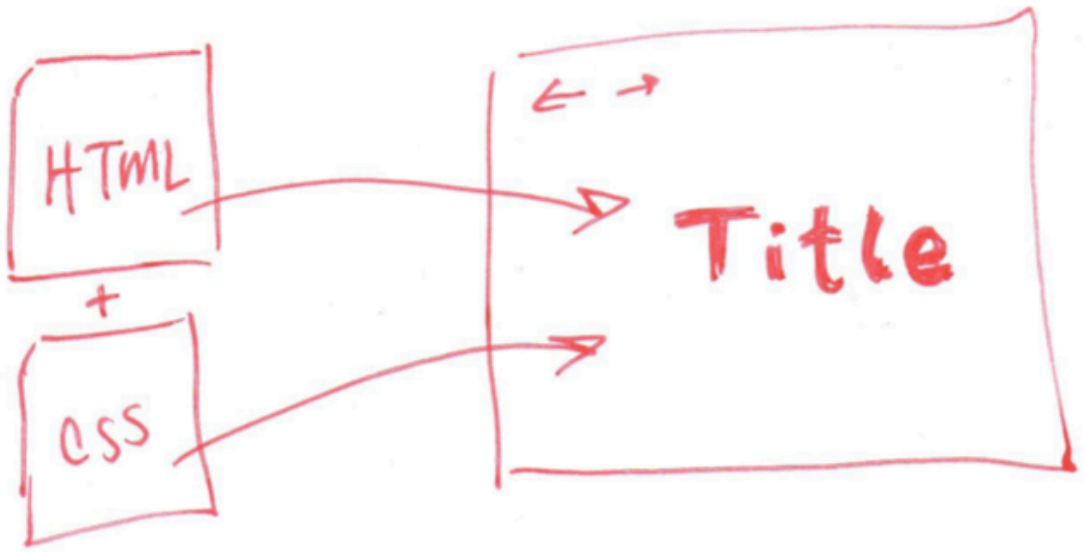
```
<h1>The Ithaca Zoo</h1>
<section>
  <h2>Welcome!</h2>
  <p>
    We're so excited to share our love of animals with our neighbors,
    and we hope you come visit soon. Whether you stop by once while
    you're passing through town or come multiple times a week, you'll
    find the Ithaca Zoo is ready to welcome you with open arms.
  </p>
  <p>
    <strong>Advisory: due to upcoming hot weather, we advise all guests
    to make sure they bring sunscreen for protection from ultraviolet
    rays and a water bottle to help stay hydrated.</strong> We have
    water fountains all over the zoo where you can fill up. Just look
    for the signs that say "Watering Hole."
  </p>
</section>
```

Using CSS to Style Structured Content

How do Browsers Style Structured Content?

HTML defines structure.

CSS defines the style.



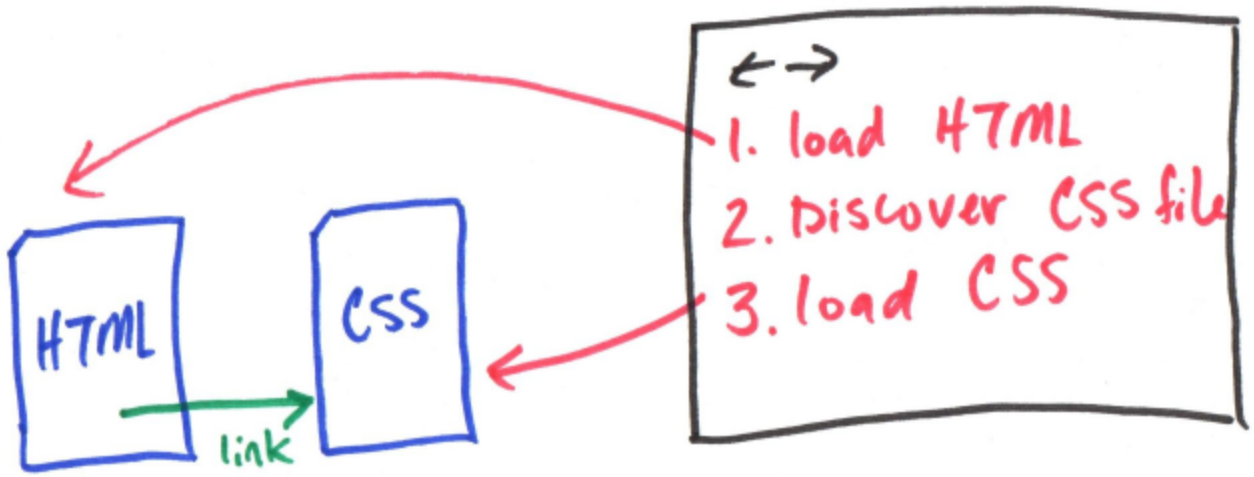
CSS: The Browser's Styling Language

CSS: **C**ascading **S**tyle **S**heets

CSS, is a language to describe the **presentation** of (HTML) documents.

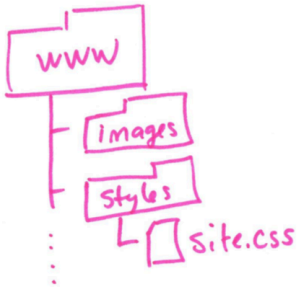
CSS is a ~~programming~~ **styling** language.

How Browsers Load & Render a Page

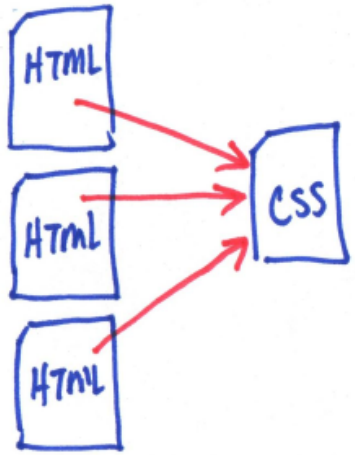


How to: Adding CSS to a Website

1. Create a new CSS file, `styles/site.css`, in the **styles** folder.



2. Link the *same* CSS file in **all** HTML files.




```
<head>
    <link rel="stylesheet"
          type="text/css"
          href="styles/site.css">
```

How to: Adding CSS to a Website

3. Code styling rules to style HTML **elements**.

```
h2 {
  font-family: sans-serif;
  font-size: 36pt;
}
```

Reflection: What does this styling rule do?

All `<h2>` elements, in every HTML file, will be 36 point sans font.

4. Validate your CSS (Same as HTML validation)

CSS Gotcha: Use *Only* External CSS

Don’t mix structure and style. Only use **external** CCS styling.

External Styling

```
<link rel="stylesheet" type="text/css" href="styles/site.css" />`
```

Internal & Inline Styling

Never add style in the HTML. Avoid the HTML `<style>` element (internal styling) or the HTML `style=""` attribute (inline styling).

Activity: Styling the Ithaca Zoo's Website

1. Open the activity repository.
2. Create a `styles/site.css` file.
   (Create a `styles` folder at the **root** of the repository, if necessary.)
3. Link to the CSS file in `index.html`

```
<link rel="stylesheet" type="text/css" href="styles/site.css">
```

4. Set the background color of the `<body>` element to `#000000`.
5. Use the color picker to select a *better* background color for the `<body>` element.
6. Validate your HTML **and** CSS files.

Activity: CSS Validation Practice

1. Open the **CSS** file in Codespace.
2. Click the **W3C validation** button.
3. Correct any errors in the Problems tab.
   (Fix only errors; warnings and infos are okay.)

CSS Styling Best Practices

1. Use the same CSS file to style all webpages.

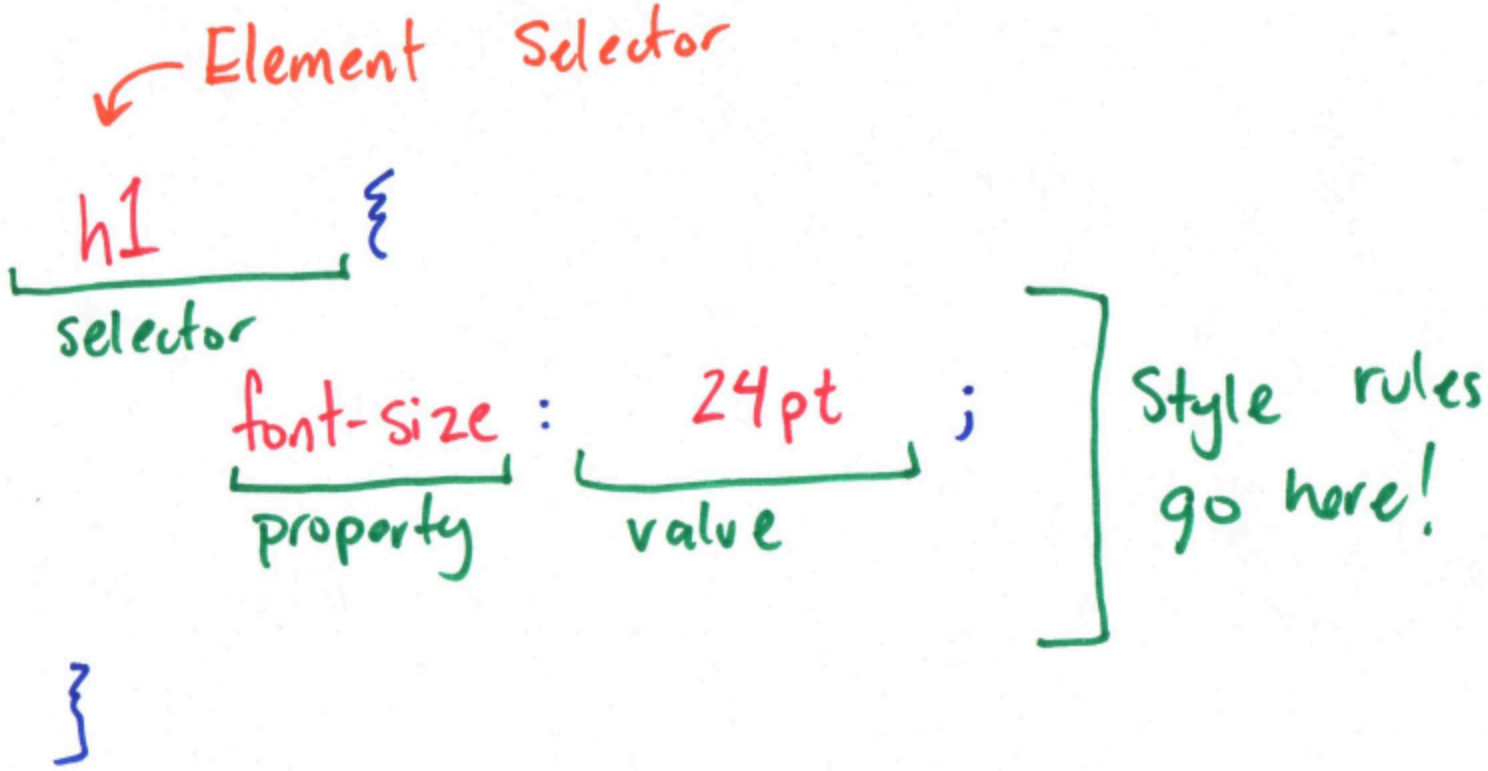
   Do **not** create a separate CSS file for each webpage.

2. Validate your CSS.

Tip: Not sure what CSS properties exists?
https://developer.mozilla.org/en-US/docs/Web/CSS/Reference

CSS Styling Rules

Anatomy of A CSS Styling Rule



style rule = property value pair

The Element Selector

Use the HTML **tag name** (e.g. `h2`, `p`, etc.) as the CSS selector.

```css
h2 {
  color: red;
}

p {
  color: blue;
}
```

Activity: Element Selector Practice

Complete problems **#1 and #2** on the **handout**.

You may style it any way you like. Reflect on the properties you read about in the class preparation.

Hint: `color`, `background-color`, `font-size`, `font-family`

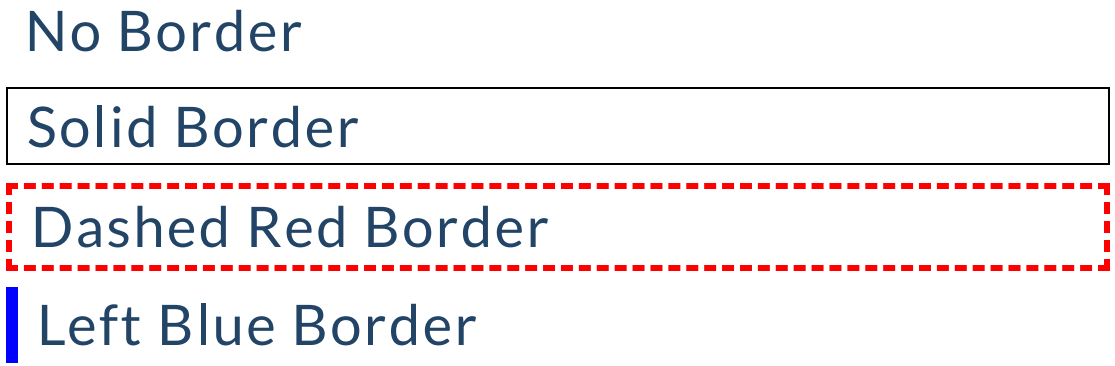
Common CSS Properties for Styling Elements

Shape

Shape: Borders

Use borders to emphasize or deemphasize the shape of an element.

```
border: none;
border: 1px solid #000000;
border: 3px dashed #ff0000;
border-left: 6px solid #0000ff;
```



Tip: Avoid too many borders; hard edges attract attention.

Shape: Border Radius

Soften element edges or change shape (ellipse).

```css
border-radius: 0;
border-radius: 12px;
border-radius: 50%;
```

Activity: CSS Shape Property Practice

Add a CSS rule to style all `<section>` elements (**#3 on handout**):

- Add a border to the left side of each section.

Example:

> The Red Panda Meet and Greet was the most fun my kids had all summer! - John Smith Sr.

Color

Color

Set element foreground (text; `color`) and background colors (`background-color`).

Accessibility Warning: High **contrast** between the foreground and background color is required for accessibility.

Contrast improves usability and accessibility.

```
background-color: #826C9E;
color: #ffffff;
```

Low contrast colors are hard to read or may even *vibrate*.

```
background-color: #ff0000;
color: #00ff00;
```

CSS Colors

(Preferred) HEX Color (use Codespace's color picker)

`RRGGBB` – **R**ed, **G**reen, **B**lue; `00` – off, `ff` – on

```
color: #ff0000; /* red */
color: #00ff00; /* green */
color: #0000ff; /* blue */
```

Named Color (see reference documentation)

```
color: red;
color: black;
```

Typography

Typography: Font Styling

```
font-weight: bold;
font-style: italic;
text-decoration: underline;
font-variant: small-caps;
text-transform: uppercase;
font-size: 48pt;
```

Bold

Italic

Underline

Small Caps

UPPERCASE

48pt

Tip: If the content should be in all CAPS, **don't** type it that way (remember accessibility), ***style* it that way!**

Typography (Fonts)

Always use Generic Font Families

```
font-family: serif;
font-family: sans-serif;
font-family: monospace;
```

Generic Serif

Generic Sans

Generic Monospace

Gotcha! Generic font family will always work in the browser; named fonts do not work if the user does not have them installed on their computer!

Best Practice – Generic Font Family Fallback for Named Fonts

Named fonts (e.g. Arial, Helvetica, Montserrat, etc.) may not be installed on your users' browsers. When using named fonts, **always** include a generic font family as an **fallback** option.

```
font-family: 'Lucida Grande', 'Arial', sans-serif;
```

Example:

This text is Lucida Grande, Arial, or generic sans serif fallback

If Lucida Grande ins't installed, the browser tries to use Arial, if Arial isn't installed the browser uses the fallback: `sans-serif`.

CSS Best Practices & Gotchas

Group CSS Rules for the Same Selector

```css
p {
  font-family: sans-serif;
}
p {
  font-size: 1em;
}
p {
  font-weight: bold;
}
```

```css
p {
  font-family: sans-serif;
  font-size: 1em;
  font-weight: bold;
}
```

The Last CSS *Property* Takes Precedence†

```css
p {
  color: purple;
  font-size: 36pt;
}

p {
  font-style: italic;
  color: green;
}

p {
  color: gray;
}
```

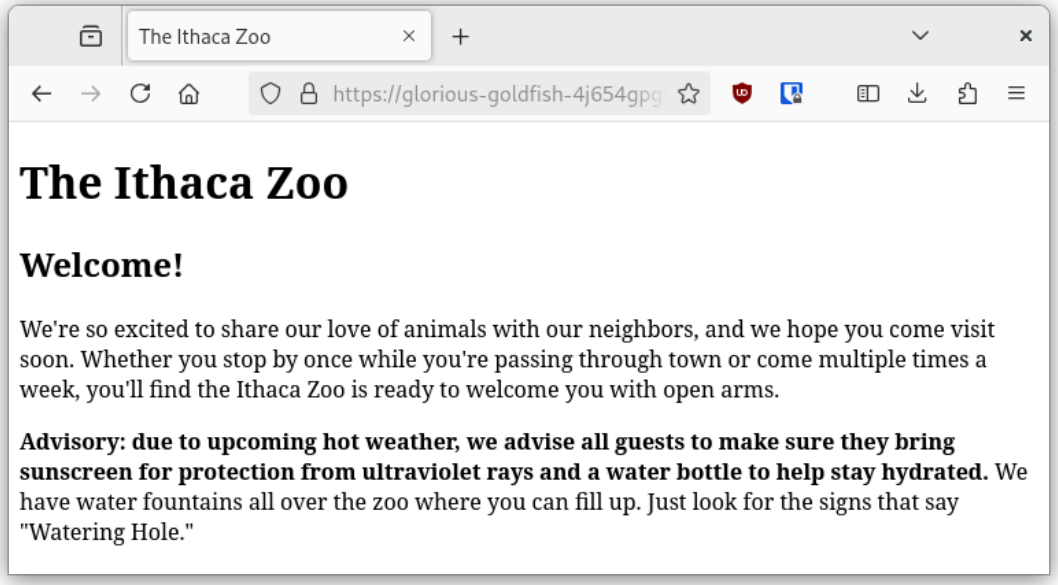
Example paragraph text. Observe this paragraph is 36pt, gray, and italic.

The last rule, for a selector, **overrides** the previous rules for the *same* property.†

† CSS precedence is more complicated than this, but for now, just remember that the last property takes precedence.

Override the Default Styling

The browser will *style* your content by default, use CSS to **override** it!




```css
h1 {
  font-weight: normal;
  font-size: 1em;
}
h2 {
  font-weight: normal;
  font-size: 1em;
}
strong {
  font-weight: normal;
}
```

Gotcha: `<strong>` ≠ bold

`<strong>` = strong importance, seriousness, or urgency

Override the default styling to work with your website's theme!

```
<strong>overridden strongly stated</strong>
```

```
strong {
  color: #ff0000;
  text-decoration: underline;
  font-weight: normal;
}
```

overridden strongly stated

Styling Only *Some* Elements

Descendant Combinator Selector

Selects **all** elements that are descendants of a given element.

```
ELEMENT_NAME DESCENDANT_NAME { ... }

figure img {
  border-radius: 10px;
}
```

Gotcha: Observe the **space** between the element names. It's **required**.

```
<picture>
  <img src="images/mcgraw-tower.jpg alt="">
</picture>

<figure>
  <img src="images/mcgraw-tower.jpg alt="">
</figure>
```

`figure img` will **only** style the `<img>` elements inside `<figure>` elements.

Activity: CSS Selector Practice

Complete **#4** today's activity sheet.

Descendant Combinator Selector

```
ELEMENT_NAME DESCENDANT_NAME {
  /* CSS property value pairs */
}
```

What's Next

Deadline: Homework 1, due Monday, 8/31/2026, 11:59pm
Deadline: Class 4 Preparation, due Tuesday, 9/1/2026, 11:59pm