ITP Skills Lab: HTML & CSS Basics

with

Ian G. Williams

iwilliams1@gradcenter.cuny.edu

October 23rd, 2023

Link to slides: https://bit.ly/htmlcss-itp-f23

(or scan QR code)

adapted from

Fall 2021 & 2022 ITP lab slides

by Zachary Mulbauer

Agenda

1. Learning outcomes

2. Introductions

3. HTML
   - Origins & concepts
   - Nuts & bolts
   - HTML practice (text, tags, elements, formatting)

4. CSS
   - Origins & concepts
   - Nuts & bolts
   - CSS practice (universal & specific stylesheet)

5. Recommendations and next steps
   - Resources
   - Websites
   - Recommended tutorials

Learning outcomes

Today we'll learn how to…

- Conceptualize hypertext, markup, and stylesheets
- Enter basic hypertext in HTML
- Add comments to HTML script
- Style the paragraph with CSS
- Use an online code editor (codepen.io)
- Create a coding environment on your computer
- Save files on your desktop

What you'll need

- Web browser with Internet access
- Access to a text editor software (Textedit or Notepad)
- Ability to create and save files on your desktop
- A little patience and curiosity

Facilitator background

- **Name:** Ian
- **Pronouns:** he/him/his
- **Program:** 3rd year student in PhD Program in Social Welfare & ITP Certificate, Program Social Media Fellow with GCDI
- **Research interests:** technology adoption and innovation in human services
- **ITP Project:** Working focused on a hypertext-based note-taking tool (Obsidian) as an autoethnographic tool
- **Tech background:** Created websites in early 2000's in HTML and CSS. Took coursework in JavaScript and D3.JS in PhD. Find coding equally frustrating and fascinating.

Introductions

- Name
- Pronouns
- Program
- 1 thing you do or would like to do with HTML and/or CSS

HTML: Overlooked and Undervalued?

image source: programminghumor.io

HTML: Anatomy

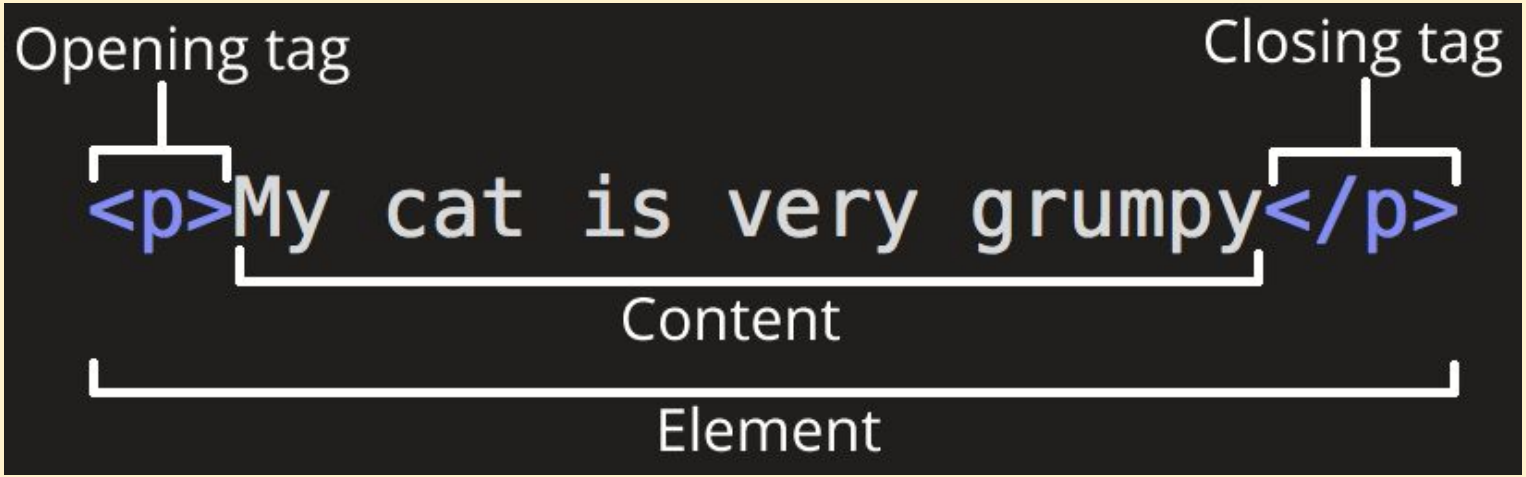


image source: MDN

HyperText Markup Language (HTML): Origins

- First proposed around 1990 by Tim Berners-Lee, a physicist at CERN, the European Organization for Nuclear Research. Current version is HTML5.
- Standard at that time for scientific publishing was Standard Generalized Markup Language (SGML).
- SGML was highly technical, creating barriers to scientific publication.
- Unlike SGML, HTML automated markup process to simplify output.
- Content could be translated into front-end (web page) display.
- Scientific innovation made online publishing (and information dissemination) much easier.
- Also made design of output responsive to user activity.

HTML: HyperText Explained

- Text that links to other documents or information without containing it
- Allows for display of materials by referencing and retrieving them
- A dictionary is an early iteration. Computational vision first articulated as Vannevar Bush's memex
- Mostly used today for underlying structure of web pages and other HTML tools

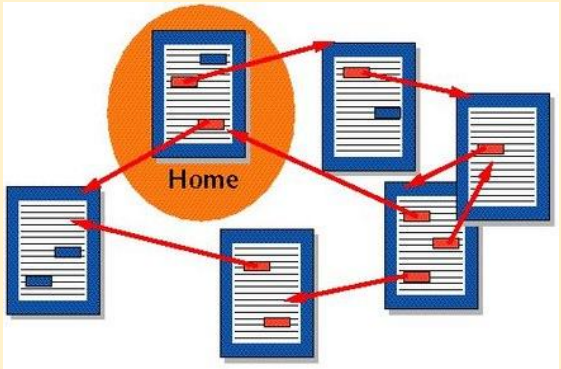


image source: Wikimedia

HTML: Markup Explained

- What is a markup language?
  - named after the practice of “marking up” or annotating print manuscripts for formatting purposes
  - instructs web browsers how to “mark up” or format discrete units of information
    - e.g. text, images, videos, lists, tables, links, etc
- HTML is the standard markup language for creating web pages; i.e. *the essential building blocks of the Web*
  - employs elements that begin & end with two tags: one opening tag ** ** and one closing tag ** **
    - tags are enclosed by two angled brackets
    - closing tags must have a forward slash (/) after the first angled bracket

HTML: Markup Process

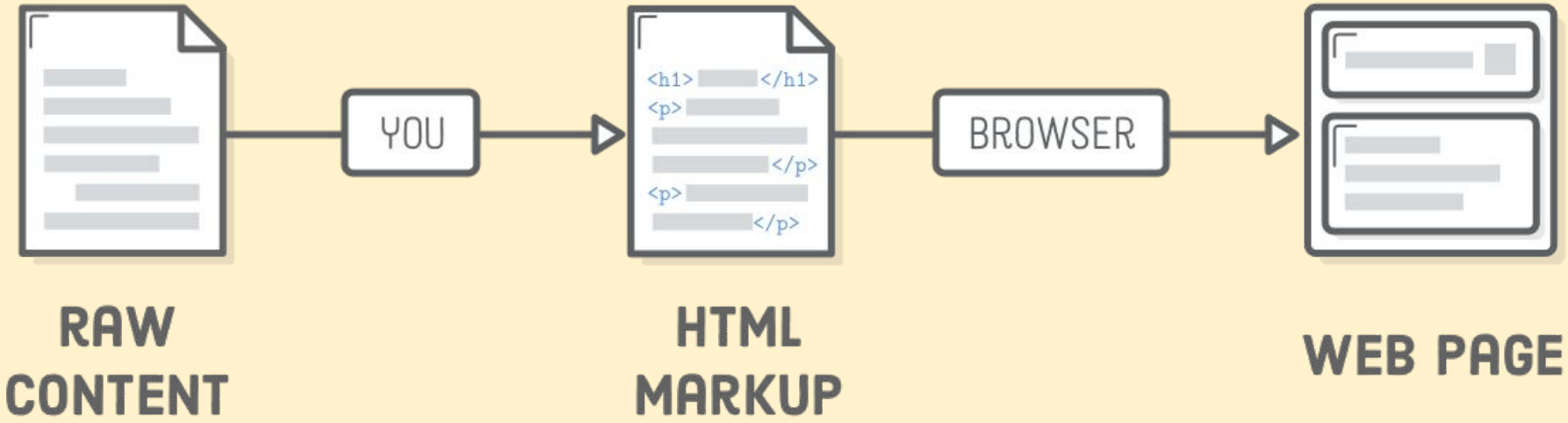


image source: Oliver James, *Interneting Is Hard*

HTML: Tags for Paragraphs and Text Styling

- Paragraphs are denoted by ** **
- Element =

```
<p> Hello world! </p>
```

  - opening tag = ** **
  - closing tag = ** **

- Italics are denoted by **<i>** or ****
- Bolded are denoted by **<b>** or ****
- Block quotes are denoted by **<blockquote>**
- Line breaks are denoted by **
**

➢ ***
*** *presents a rare case in which an HTML element does not require a closing tag*

HTML: Code Snippet: Practice!

```
<p>
I stumbled across this poem by Tracy K. Smith
this morning when reading the 9/28 issue of
<i>The New Yorker</i>:

<blockquote>
I wish it would grab me by the ankles and pull.
<br>
I wish its shadow would dance up close, closing
in.
</blockquote>

Reciting these words to myself, I was drawn back
to when I first read Smith's collection of poems,
<strong>Life on Mars</strong>, which seems to me
one of the more poignant inquiries into the final
frontier in the U.S. today.
</p>
```

I stumbled across this poem by Tracy K. Smith this morning when reading the 9/28 issue of *The New Yorker*:

> I wish it would grab me by the ankles and pull.
> I wish its shadow would dance up close, closing in.

Reciting these words to myself, I was drawn back to when I first read Smith's collection of poems, **Life on Mars**, which seems to me one of the more poignant inquiries into the final frontier in the U.S. today.

Practice with paragraph and text styling tags using the online code editor linked here: https://codepen.io/pen/

Use this text, or make your own while retaining the formatting

HTML: Tags, The Basic Structure of a Web Page

<html> = opens page for HTML code

- **<head>** = opens metadata field
- **</head>** = closes metadata field
- **<body>** = opens visible page content
- **</body>** = closes visible page content

</html> = closes page for HTML code

➢ *be sure to check that you have closed all of your brackets and tags*

Return to your code snippet (via https://codepen.io/pen/) to add these tags, and to layer any existing paragraph content between the **<body>** and **</body>** tags

HTML: Tags, Titles, and Headings

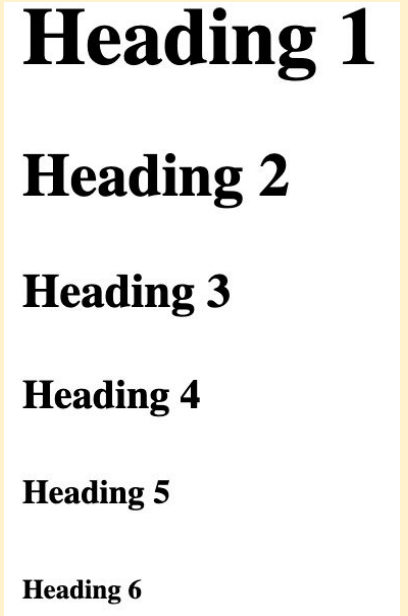
- Page titles are denoted by **<title>**
  - shows up in browser tab
  - located under **<head>** as metadata
- Heading levels span **<h1>** to **<h6>**
  - sorts headings and subheadings
  - located under **<body>** as body content

Now return to your code snippet to add a *title* and *a heading*, making sure to layer these elements correctly. Remember:

- title tags between **<head>** and **</head>**
- h1-h6 tags between **<body>** and **</body>**

Example: https://bit.ly/33u0QNx

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

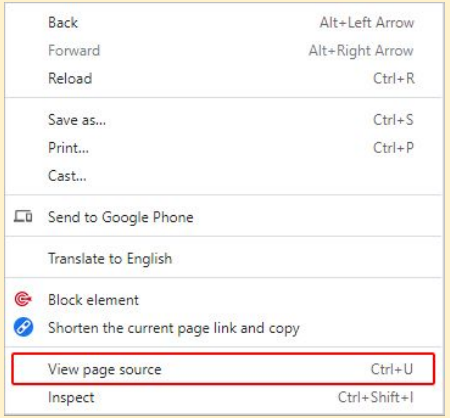


These are pre-defined within HTML; you can modify with CSS

HTML: Viewing Page Sources

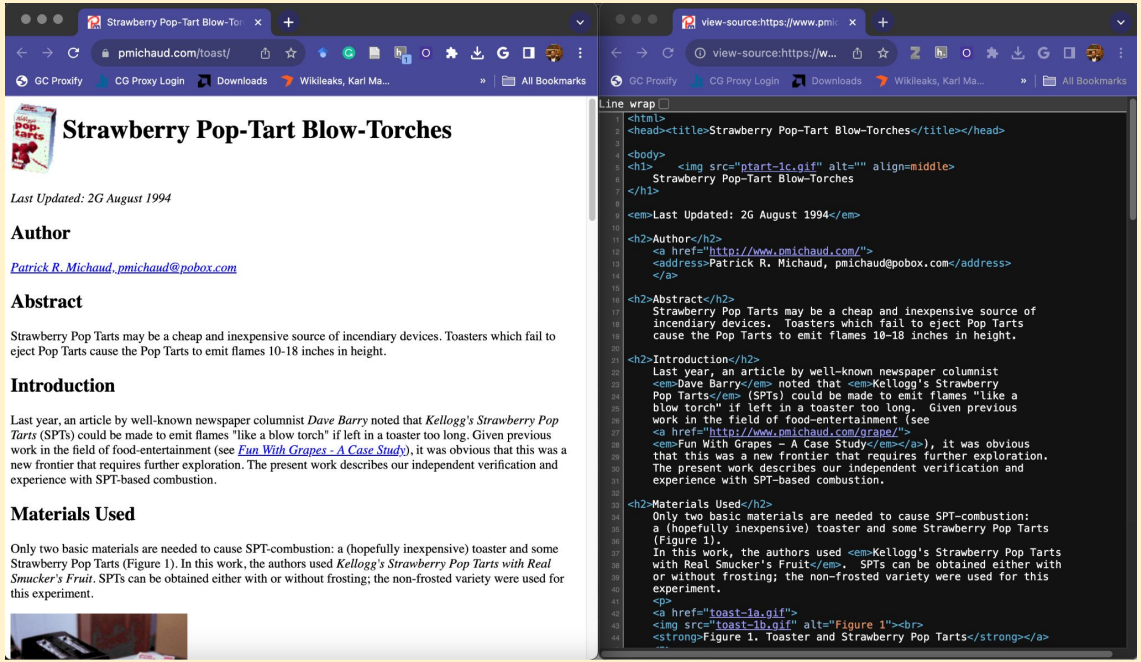
Now that you have more of a grasp on how HTML uses tags to “mark up” content on the web, take a glimpse *behind the veil* and see for yourself:

- Early-stage websites are more legible given their exclusive use of HTML
- So let’s visit an old-school website (last updated in 1994): http://www.pmichaud.com/toast/
- When you’re ready:
  i. right-click or control+click anywhere on the page
  ii. click “view page source”
  iii. try reading its HTML source code
  iv. share what stands out to you



HTML: Viewing Page Sources

It is recommended that you display the page and the source side by side, in different



HTML: Comments & Page Version Histories

- HTML code itself does not contain histories of its changes, but a website might contain an edit history log.
- An author may include notes about changes using the *comment* tag, which looks like this `<!-- -->`
- Example:

```
<!-- this is a comment -->
```

- A comment is not processed by the web browser, meaning the text contained within it is not fed as instructions for markup. Try adding one to your code snippet.

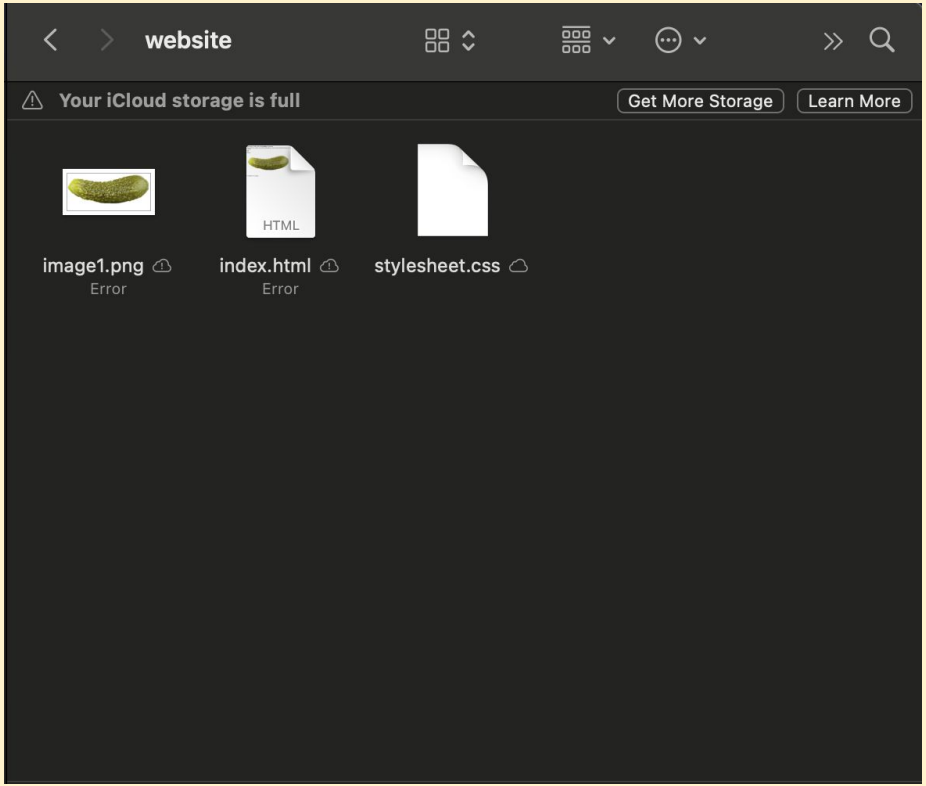
Another way to check for changes and a version history is the Internet Archive's Wayback Machine, which you can access here https://archive.org/web/

Creating a Coding Environment & Website Folder

This is an essential step to coding. Remember, HTML combines connects referenced information.

1. Minimize all your windows so only your computer desktop remains
2. Click File >> New Folder >> name that folder "**website**"
3. Open Finder >> select Applications >> open your preferred text editor
    a. e.g. TextEdit (MacOS), Notepad (Windows)
4. Click New Document (or click File >> New)
5. Click Format >> Press Make Plain Text *(this is an essential step)*
6. Copy and paste your HTML code from codepen into the new file
7. Click File >> Save >> title your new file "**index.html**"
8. Click "Save" with your "**website**" folder as the file's directory

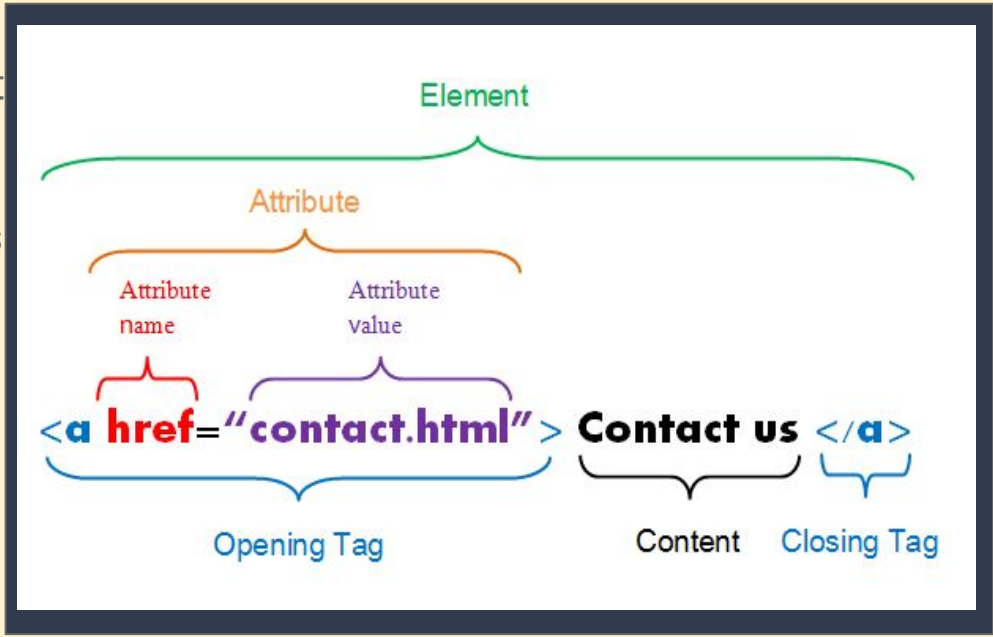
Coding Environment: Example



Simply put, it's a folder that contains everything (which may include sub-folders)

HTML: Tags for Links - Hypertext

- The **<a>** tag links users to another file in your your Website folder — but more on that in a bit
- The **href** part of the opening tag, known as the attribute name, indexes the title of your intended file, known as the attribute value
    - **href** = hyperlink reference
- The attribute must be be contained in “ “
- The example on the right yields the following hyperlink: Contact us



Working with Links in a Local Directory

- When linking to another page on your website, you need to save that other page *in the same folder* as the page in which you're embedding an intended hyperlink
- If you want to embed a link in your "**index.html**" file that yields an image file, then you would first need to save that image file in "**website**" folder where "**index.html**" lives

Let's try it out…

Linking to Images in a Local Directory

- You can call upon an image file with the **<img>** tag by using the **src** attribute to index the title of that image file: `<img src="image1.png">`
- In order to use this tag, follow these following steps:
  1. create a new document, save it to your "**website**" folder as "**image1.html**"
  2. copy/paste **<img src="image1.png">** in your "image1.html" file, *save your work*
  3. download an image file, *save it* to your "website" folder as "image1.png"
  4. return to your "**index.html**" file to embed a link to the "**image1.html**" file
     a. example: `<a href="image1.html">Link to my image</a>`
  5. save your work and open "**index.html**" in your browser to see if your link works

Cascading Style Sheets (CSS)

- HTML separated document structure and layout. Layout, by default, was often determined by the browser.
- First proposed in 1994 by Håkon Wium Lie. First version invented in 1996.
- Goal of CSS was to create a universal style selector for the Web, improve visual design.
- Sets rules for how the layout and design elements of HTML should be rendered in your browser
    - i.e. CSS dresses up web pages using “rule sets” established in a stylesheet
- Adaptable across web pages, browsers, tablets, devices, screen sizes, and more.
- Radically improved the efficiency with which developers built visually appealing websites
- CSS has a *cascading* effect in how CSS rules establish an order of precedence that instructs browsers how to style HTML elements

CSS: Syntax and Rule-Set

CSS presents a rule-set consisting primarily of *selectors* and *declaration blocks*

- *selectors* index the tag assigned to HTML element(s) that you want to style
- *declaration blocks* provide a series of declarations
- Each block is: enclosed in curly brackets
  separated by a semicolon
- Declarations include both a *property name* and a *property value* (name/value pair)

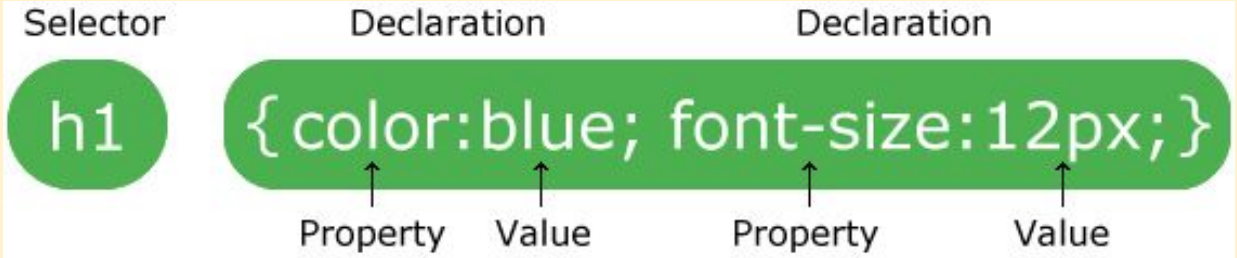


Image credit: https://www.w3schools.com/css/css_syntax.asp

CSS: Incorporating Stylesheets

- Three ways to use CSS stylesheets to style HTML files
  - Internal CSS (example)
  - Inline CSS (example)
  - External CSS (example)
    - Nested under the **<head>** element of an HTML document
    - Employs a **<link>** tag to call on the CSS stylesheet to style your HTML file
    - Example: **<link rel="stylesheet" type="text/css" href="style.css">**

<link> elements present yet another a rare case in which an HTML element does not require a closing tag

CSS: Selectors and Declaration Blocks

- *Universal selector* * = applies to everything in an HTML document

```
* {
      font-family: arial-serif;
      font-size: 12px;
 }
```

- *Type selector* = matches or targets element names; multiple selectors can be applied at once

```
         h1 {
                margin-left: 25px;
           font-size: 18px;
      color: #746E6E;
                  }
             p, blockquotes {
                    text-align: justify;
                font-family: Verdana;
                  }

```

CSS: Employing an External Stylesheet in Your Website Folder

1. Using TextEdit/Notepad, create a new file >> add one or more selectors to a declaration block, and create declarations for the heading and/or the paragraph elements of your “**`index.html`**” file
2. Save the CSS file as "**`style.css`**" to your "website" folder
3. Open “**`index.html`**” then nest the bulleted **`<link>`** element under the **`<head>`** section so your HTML files will index your CSS stylesheets as metadata
   a. **`<link rel="stylesheet" type="text/css" href="style.css">`**
4. Save your work and navigate to your browser
5. Click “File” >> “Open File” >> choose “**`index.html`**” from your “website” folder
6. Voila!

CSS: Code Snippet

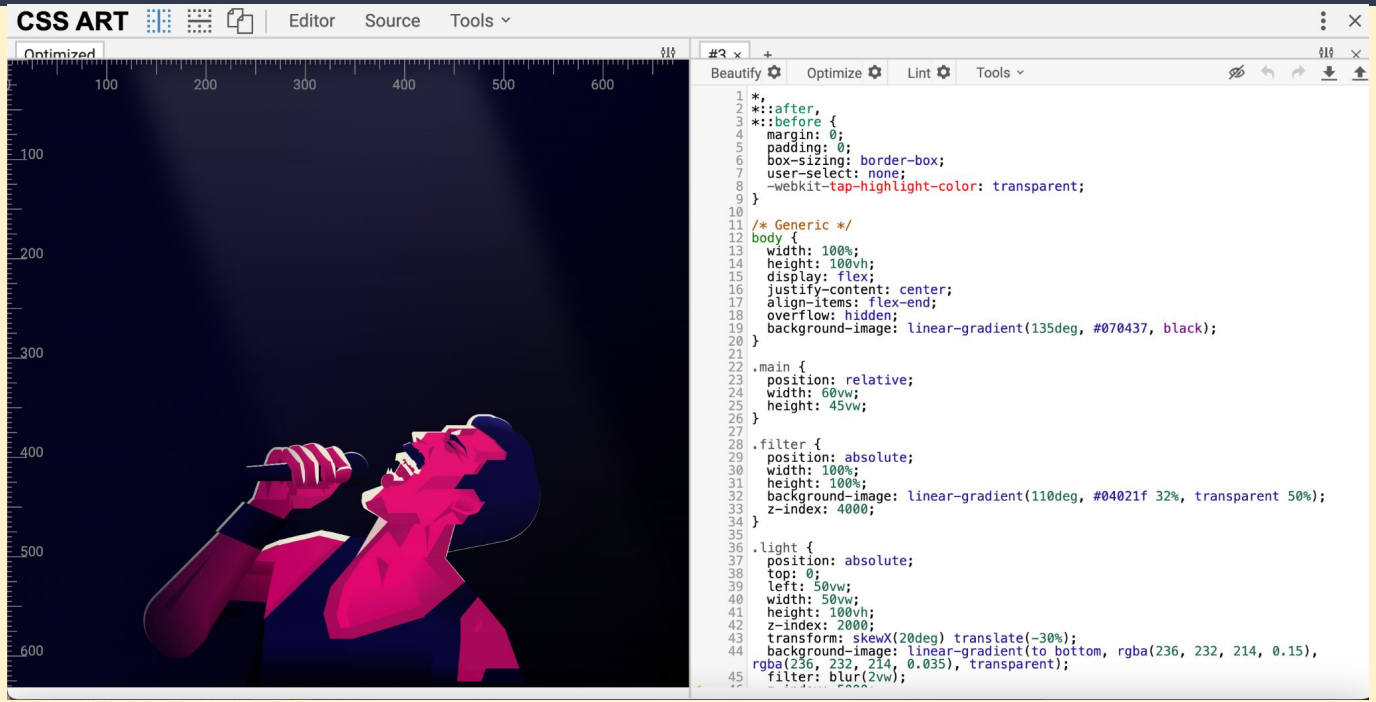
stylesheet.css

```
* {
        font-family: arial-serif;
        font-size: 12px;
        }

h1{
        margin-left: 25px
        font-size: 18px;
        color: #746E6E;
        }

p, blockquotes {
        text-align: justify;
        font-family: Verdana;
                }
```

HTML & CSS Art - Ricardo Oliva Alonso, “Freddie Mercury”



https://css-art.com/freddie-mercury-html-css/

Summing it up

- Tonight we learned…
  - A little about the histories of HTML and CSS
  - How they relate to solving technical problems, and then create their own
  - How to create a simple HTML script in *codepen.io*, format and style it, add an image, and add comments.
  - How to use CSS to create an external stylesheet for an HTML page
  - How to read a page source and create a desktop coding environment
- With these skills, you can build a really simple website
- Any lingering questions?

Moving Forward

Find a problem or a project

Build something simple & silly

Look at examples!

- HTML sites
- CSS Zen Garden
- CSS Art Gallery

Wikipedia for background reading

- HTML, CSS

Tools & Tutorials

Codepen.io

- **Online code editor**

w3schools.com

- **HTML Reference, Examples, Exercises**
- **CSS Reference, Examples, Exercises**

Interneting Is Hard: HTML & CSS

- Created by Oliver James
- Free, visually appealing
- Excellent conceptual explanations
- Good practice from current lesson

The JavaScripting Graduate Student

- Created by GC Steven Zweibel (GC librarian) and Zachary Lloyd (GC Digital Fellow)
- Incorporates coding practice in user friendly environment; self-paced
- Scaffolds from HTML & CSS to JS