ual: creative computing institute

Peckham DAZ

Accessible Web Development

Session 1: HTML & CSS

Date

Course Agenda

- **Session 1:** Intro to HTML and CSS
- **Session 2:** Intro to JavaScript
- **Session 3:** DOM Manipulation (HTML, CSS, JS)
- **Session 4:** Accessible Web Development

Session Structure

- Lecture (1 hour)
- Break (15 mins)
- Labs Exercises (2 hours 30 mins)
- Debrief (15 mins)

Resources that will help you learn...

Web Design Tools

- **Figma** – A free design tool for creating website wireframes and UI prototypes.
- **Canva** – Easy-to-use for designing web page layouts and mockups.

HTML & CSS Basics

- **MDN Web Docs (HTML & CSS)** – A beginner-friendly, comprehensive guide to HTML, CSS, and JavaScript.
- **freeCodeCamp** – Hands-on exercises to learn HTML & CSS by building projects.
- **W3Schools – HTML & CSS** – Interactive tutorials with examples and quizzes.

...More Resources

Javascript

- JavaScript.info – A detailed, beginner-friendly JavaScript tutorial.
- Eloquent JavaScript (Free Book) – Teaches JavaScript with exercises and real-world examples.
- The Odin Project – Full beginner-to-advanced web development course (includes HTML, CSS, JavaScript, and more).

Accessible Web Development

- MDN Web Docs (HTML & CSS) – A beginner-friendly, comprehensive guide to HTML, CSS, and JavaScript.
- freeCodeCamp – Hands-on exercises to learn HTML & CSS by building projects.
- W3Schools – HTML & CSS – Interactive tutorials with examples and quizzes.

...More Resources

Interactive Practice & Challenges

- **Frontend Mentor** – Free real-world web development challenges.
- **CSS Battle** – Fun challenges to improve CSS skills.
- **CodePen** – Play around with HTML, CSS, and JavaScript in an online editor.
- **JSFiddle** – An interactive sandbox for experimenting with code.

ual: creative computing institute

Web Development

Background

What is Web Development?

Web development is the process of building websites and applications that run on the internet.

Key Aspects:

- **Front-end:** The part users interact with
- **Back-end:** Databases and server logic

(front-end is the restaurant, back-end is the kitchen)

Website Types:

- **Static:** Simple, fixed content (e.g., personal blog)
- **Dynamic Websites:** interactive, data-base driven (e.g., e-commerse)
- **Web Applications:** Complex sites that function like software (e.g Spotify Web)

Front-end Web Languages

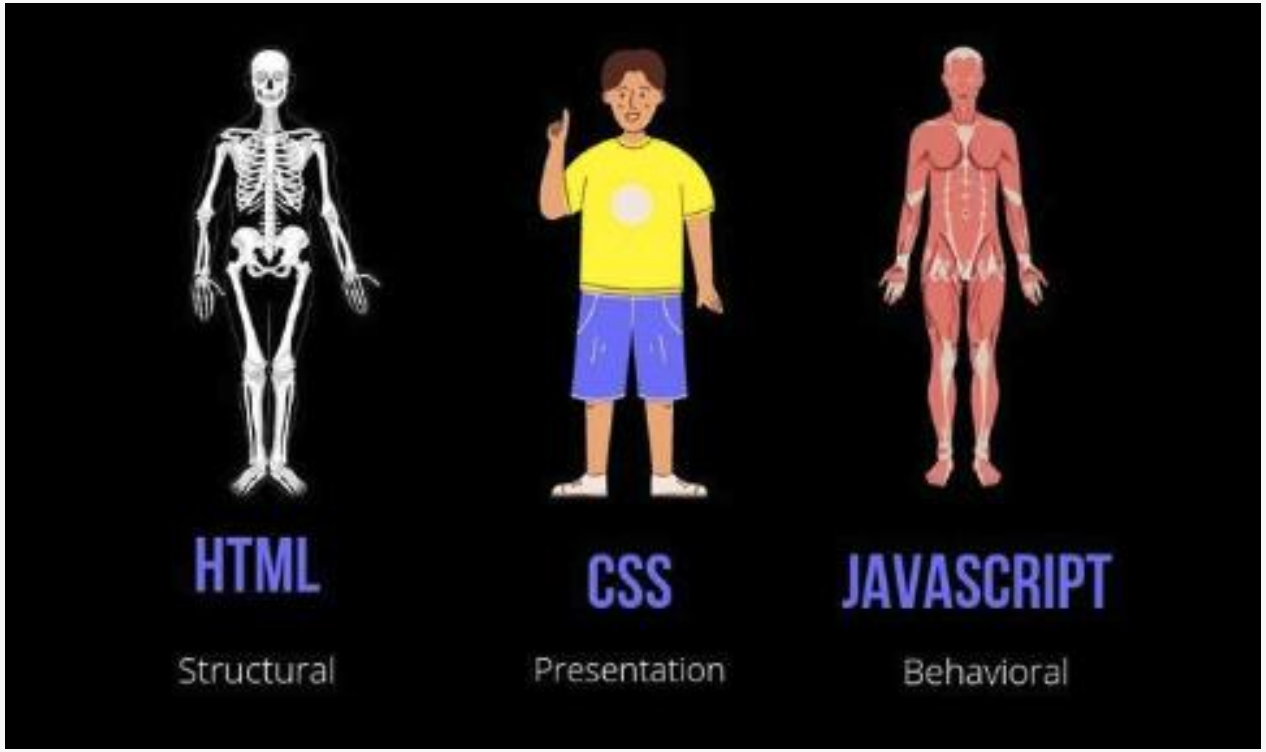
Front-end refers to the visible user interface that a user interacts with.

It’s made up of 3 languages:

HTML: The structure and layout of a page

CSS: The styling and look of a page

JAVASCRIPT: The functionality of a page



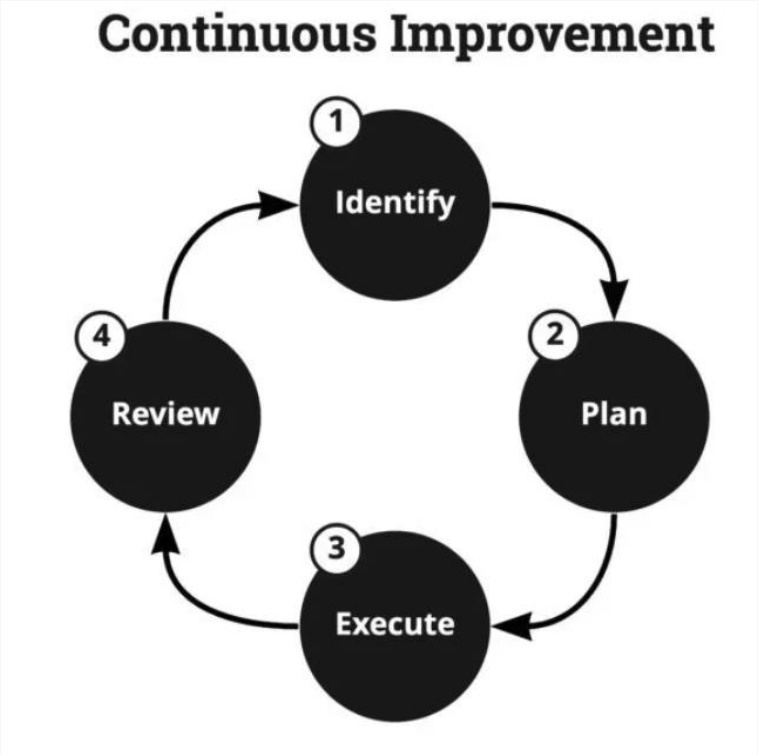
Web Development Process

Planning and Design

- Define goals and user needs
- User interface and user experience design (UI/UX)
- Iterative testing and feedback with users

Development and Deployment

- Writing HTML, CSS, JavaScript for structure & style
- Implementing interactivity with JavaScript
- Connecting to back-end if needed
- Uploading to a server



Accessible Web Development

Why Accessibility Matters?

- Ensures everyone, including people with disabilities, can access and use websites.
- Improves usability for all users.

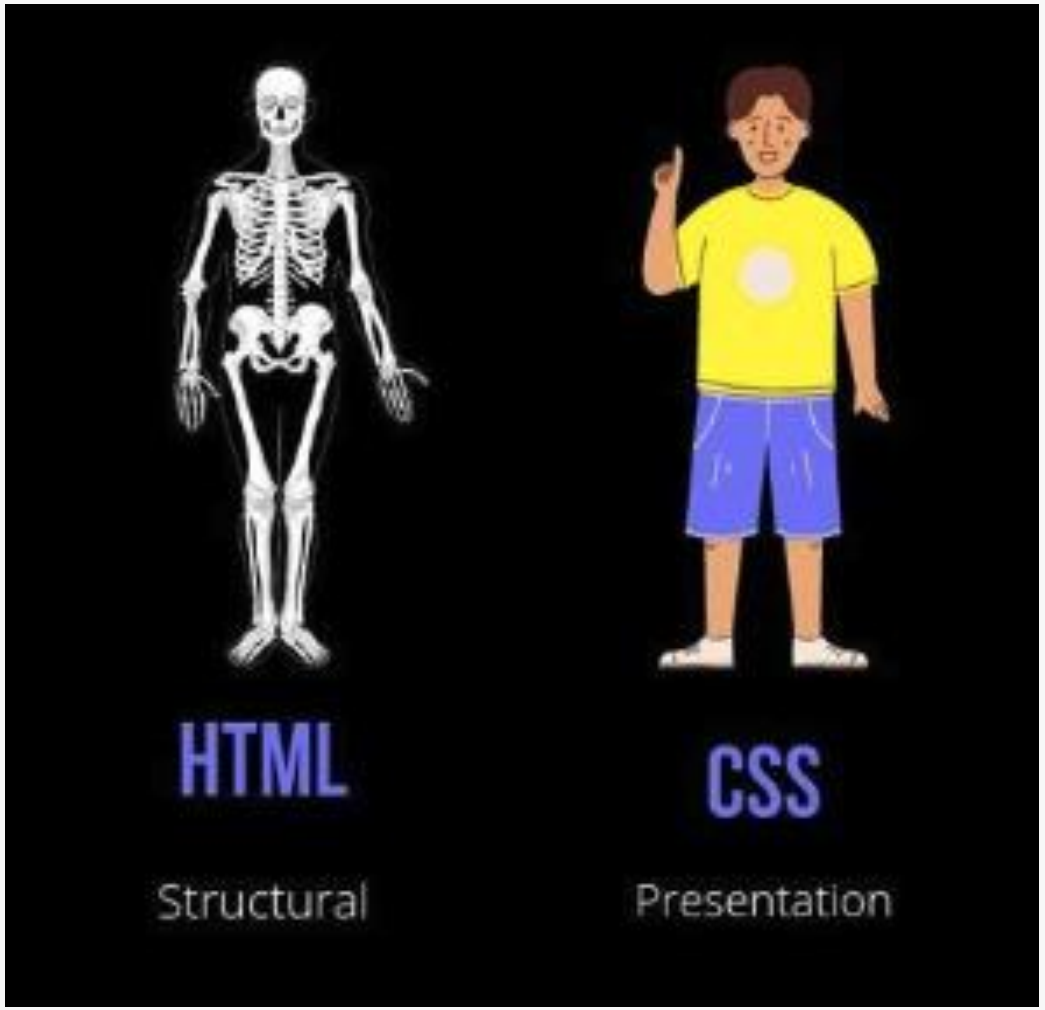
Key Accessibility Principles

- **Perceivable:** Content must be easy to see and hear (e.g., alt text for images).
- **Operable:** Users must be able to navigate easily (e.g., keyboard navigation).
- **Understandable:** Clear and simple content and UI.
- **Robust:** Compatible with different devices and assistive technologies.

ual: creative computing institute

HTML & CSS

The foundation and styling of websites

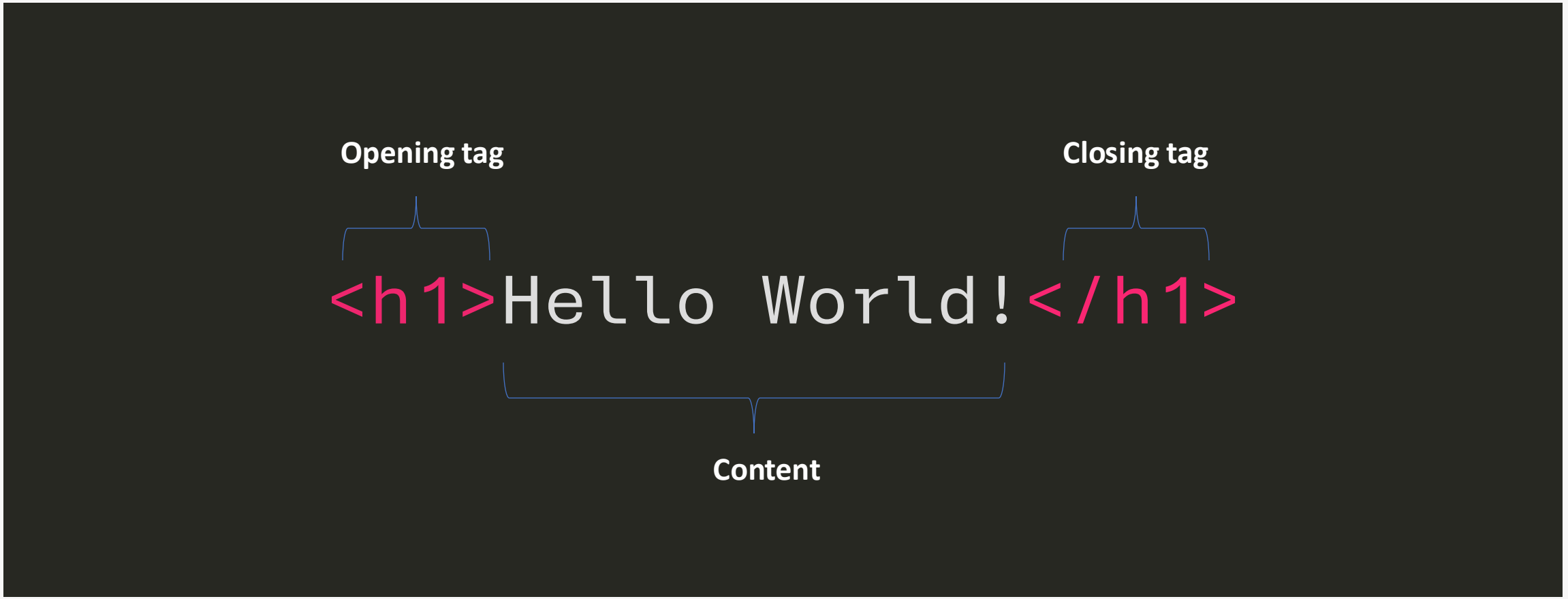

HTML
CSS
Structural
Presentation

Hyper Text Markup Language (HTML)

- Describes the structure and meaning of a webpage
- Made up of tags (also called elements)
- Denotes text and images/video
- Can be manipulated by CSS and JavaScript
- It forms the foundation of the DOM (Document Object Model)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title> This is the tab title in your browser </title>
</head>
<body>
  Everything inside here gets rendered in your browser!
</body>
</html>
```

Tags/Elements



Tags/Elements...

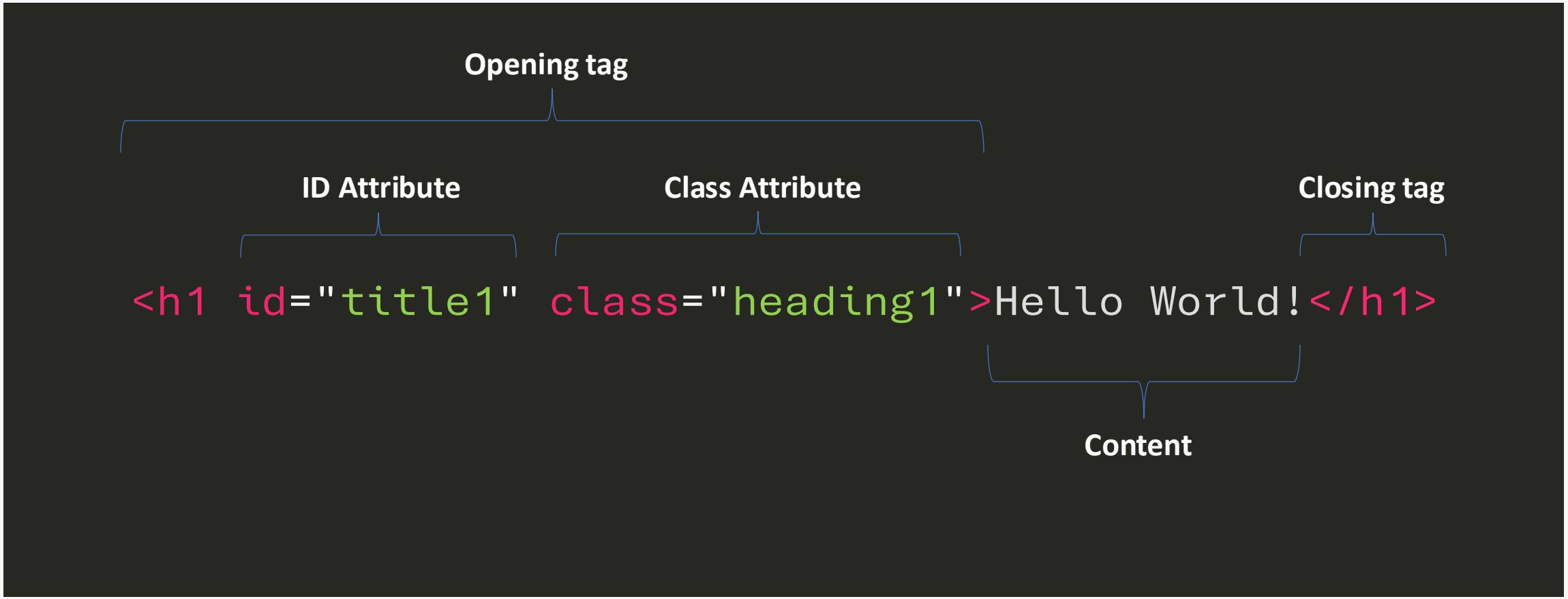
There are loads! We'll mainly be using these...

- Heading `<h1></h1>…<h6></h6>`
- Paragraph `<p></p>`
- Div `<div></div>`
- Button `<button></button>`
- Form `<form></form>`
- Inputs `<input></input>`
- Input label `<label></label>`
- Anchors (page links) `<a></a>`
- Image `<img>`
- Script `<script></script>`
- Option `<option></option>`
- Select `<select></select>`
- Textarea `<textarea></textarea>`
- Link (Stylesheet) `<link>`
- Unordered List `<ul></ul>`
- List Items `<li></li>`

HTML Tag Cheatsheet!

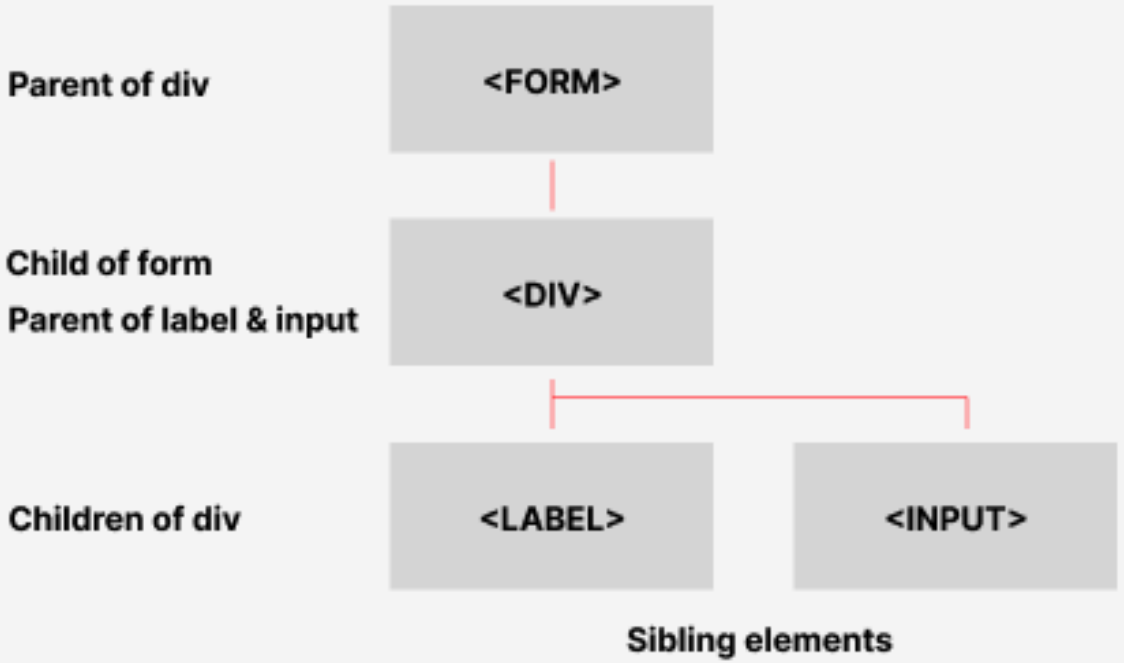
HTML Attributes

List of Attributes



Parent/child Relationship...

…Forms the DOM Family Tree



Writing Accessible HTML Code

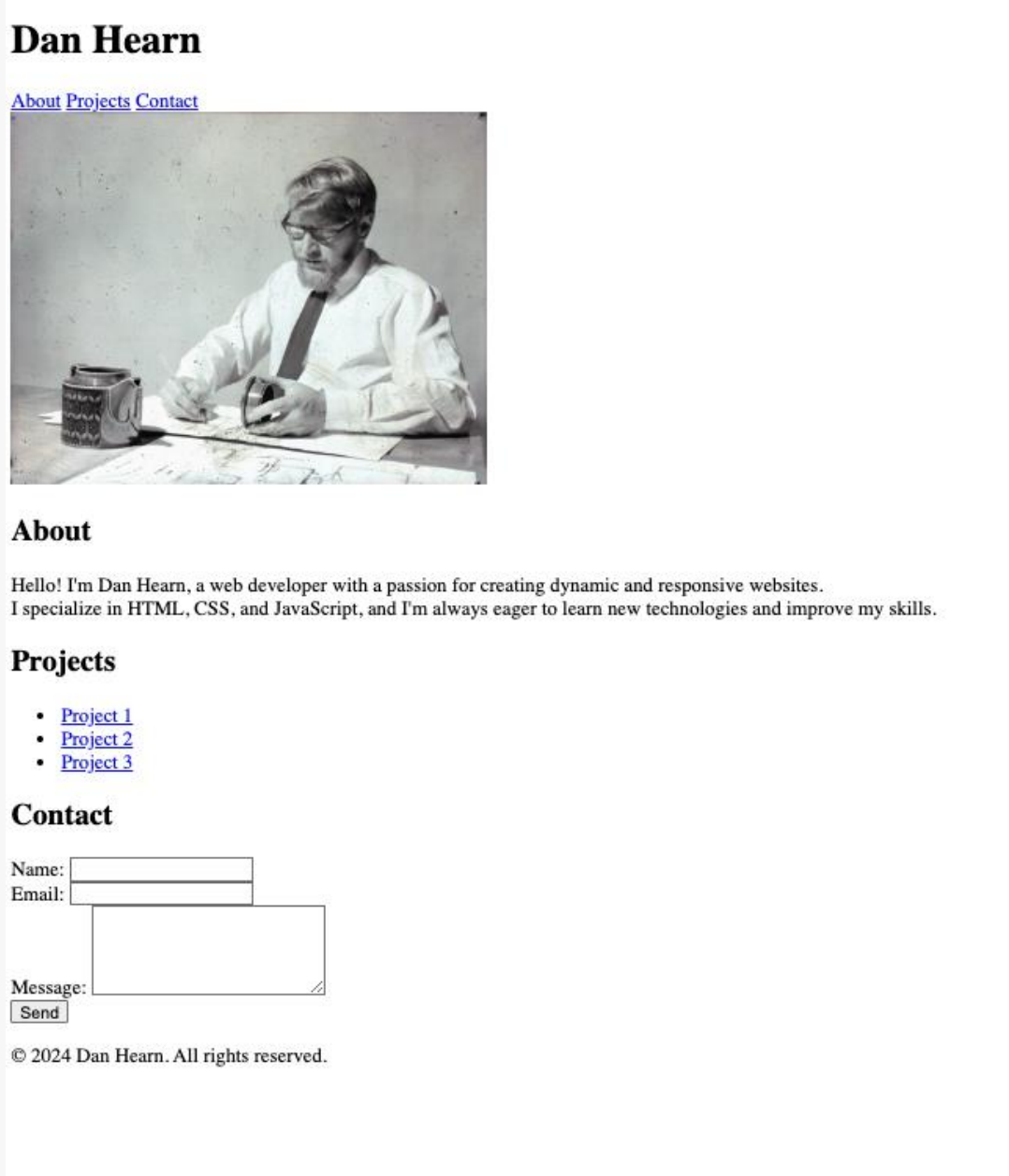
- **Use Semantic HTML:** `<button>`, `<nav>`, `<article>`, `<table>` instead of generic `<div>` and `<span>`.
- **Headings Matter:** `<h1>` to `<h6>` structure the the content of the page, helps screen readers
- **Use `alt` Attributes for Images:** Describe images for visually impaired users
- **Declare the Language:** at the top of your HTML document `<html lang="en">`
- **Use Clear, Simple Language**

Back to the 90's...

This is a raw HTML website. No CSS or Javascript has been used.

- Notice how the browser applies some default styles to certain elements.
- **What HTML Tags do you think make up this page?**

[My 90's style portfolio]



Cascading Stylesheets (CSS)

- A stylesheet language used to style HTML elements
- Can add some basic functionality to a webpage
- Used to create the 'branding style' of your webpage
- Can be used to improve accessibility and user experience.

```
button {
  background-color: #4CAF50;
  border: none;
  color: white;
}
```

CSS Selectors

Selectors allow you to target specific HTML tags to style them.

Core selectors:

- All selector: * {}
- Tag selectors: button {}
- ID selectors: #id-name {}
- Class selectors: .class-name {}

List of other selectors

```css
/* All Selector */
* {
  font-family: 'Courier New', Courier, monospace;
}
/* HTML Tag selector */
button {
  background-color: #4CAF50;
  color: white;
}
/* ID selector */
#submit-btn {
  border: 1px solid #000;
  padding: 10px;
}
/* Class Selector */
.button {
  font-size: 16px;
  cursor: pointer;
}
```

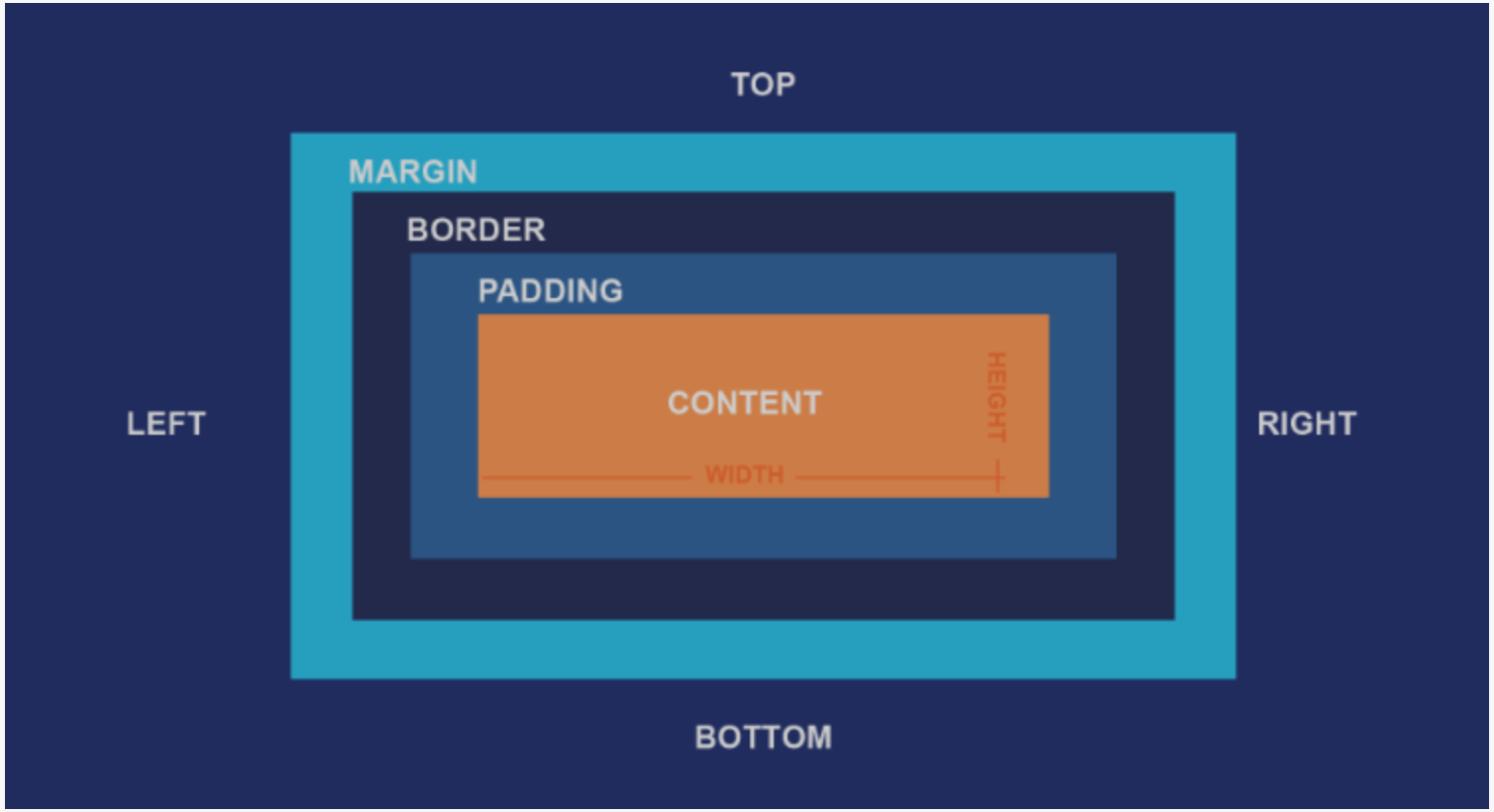
CSS Properties

- CSS properties specify how HTML elements should be displayed
- manage the positioning and spacing of elements on a webpage
- They allow for customisation of colors, fonts, and backgrounds
- Properties can change styles dynamically, improving user experience and accessibility

[There are hundreds of CSS properties]

```css
button {
  background-color: #4CAF50;
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
  border-radius: 8px;
  transition-duration: 0.4s;
}
```

Properties - Box Model



Use inspector tools to help understand this

Cascade, Specificity, and Inheritance

- **Cascade** refers to how a CSS file is read from top to bottom. If there are two identical selectors with different properties, the second selector will override the properties of the first.
- Some selectors are more **specific** than others - meaning that some can override others.
- Some properties are **inherited** from parent elements

Specificity Values

`h1` = 1

`.heading-1` = 10

`#title` = 100

```
/* What's the specificity value? */
 button.btn-class#button1 {

 }
```

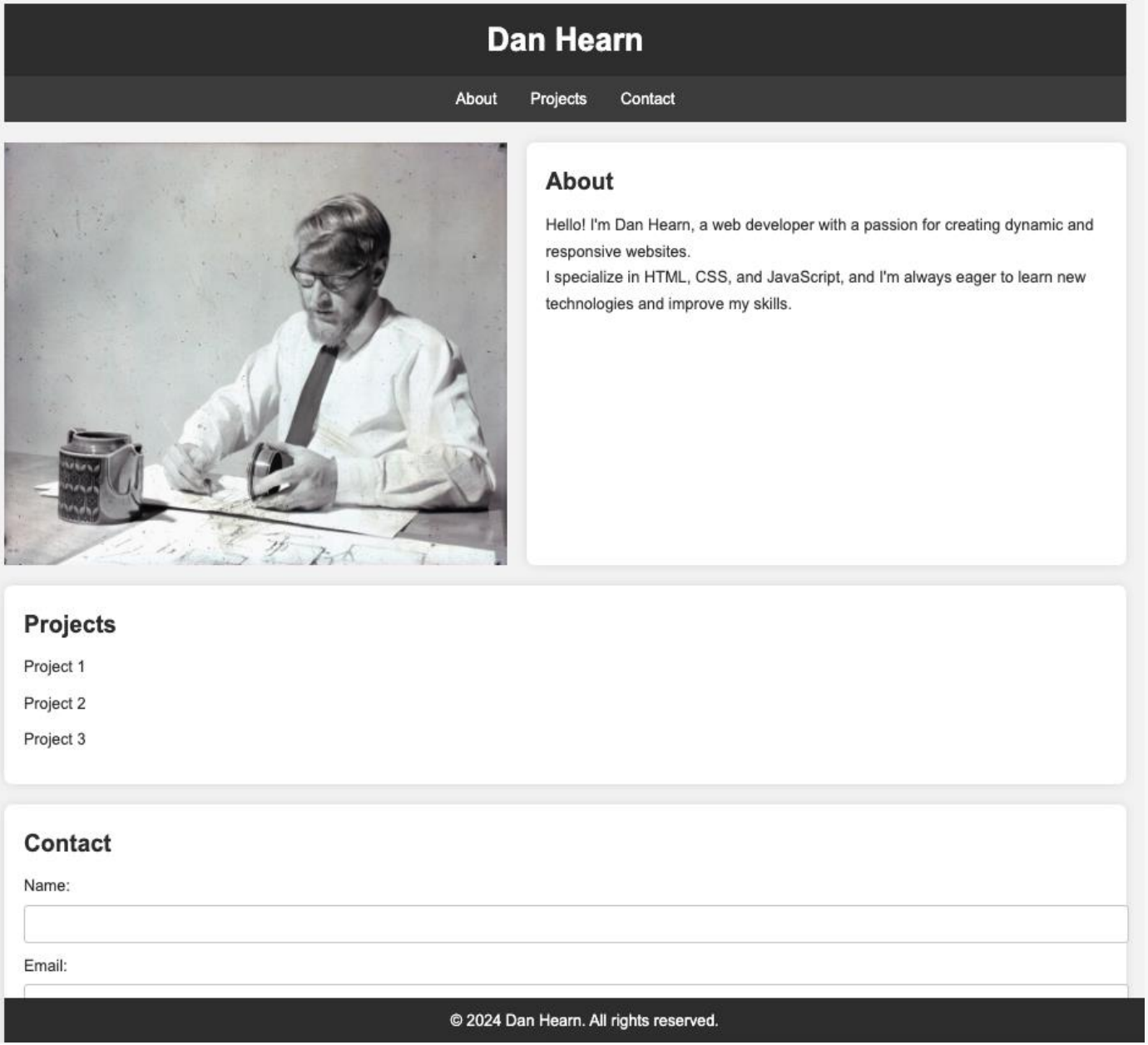
Writing Accessible CSS Code

- **Contrast:** Use high colour contrast for readability**.**
- **Semantic Colour:** Use colour to convey the meaning behind inputs e.g **red = delete, green = create**
- **Keyboard-Friendly:** Ensure focus indicators are clear and visible. Support keyboard navigation for all interactive elements.
- **Responsive Design:** Use relative units **em, rem, %** for flexible layouts. Implement media queries for different screen sizes.
- **Focus & Visibility:** Ensure focus styles are visible and not removed. Use animations and transitions carefully to avoid distraction.
- **Reduced Motion Support:** Avoid auto-playing videos and excessive movement.

HTML & CSS

- CSS can target HTML elements with selectors
- CSS applies styling and functionality to HTML elements through properties

[Portfolio with CSS]



Lab Exercises

- Can be found on the Peckham DAZ GitHub

ual: creative computing institute

Great work! ☺

@ual_cci

@ual_cci

arts.ac.uk/cci

Date