bite-sized, kid-friendly explanations, step by step:

1. **HTML Elements** → grouped into *Block-level* and *Inline* (with do's & don'ts).
2. **CSS** → line-by-line breakdown, plain English.
3. **JavaScript** → starting with **DOM (Document Object Model)**, explained with fun analogies.
4. **Mermaid Diagram** → visualizing the DOM tree of our portfolio page.

🪜 Part 1 – HTML Elements

Think of **HTML** as the **skeleton** of your website.
Each **element** is like a **body part** → head, arms, legs, eyes, etc.

🔹 **Block-Level Elements** (big blocks that start on a new line)

They act like **big containers or boxes**.
Examples:

- `<div>` → Generic box (like a cardboard box you can put anything inside).
- `<p>` → Paragraph (for text).
- `<h1>` → Big heading (like the title of a book).
- `<h2>, <h3>…<h6>` → Smaller headings.
- `<ul>` → Unordered list (bullet points).
- `<ol>` → Ordered list (numbered).
- `<li>` → List item (inside lists).
- `<section>` → A chapter in your website.
- `<article>` → A self-contained story (like a news article).
- `<header>` → Top section (like the cover of a book).
- `<footer>` → Bottom section (like a signature).
- `<nav>` → Navigation menu (like a table of contents).
- `<form>` → Box for filling details.
- `<table>` → Table/grid of data.
- `<blockquote>` → Quoted text.

✅ **Do:** Use them to organize your page into neat sections.
❌ **Don't:** Use `<div>` for everything — choose the correct element for meaning.

🔹 Inline Elements (small things that sit inside text)

Think of them like **stickers inside your notebook**.
Examples:

- `<a>` → Link (like a shortcut to another page).
- `<img>` → Image.
- `<span>` → Small container for styling (like highlighting a word).
- `<strong>` → Bold text (important).
- `<em>` → Italics (emphasis).
- `<br>` → Line break (new line).
- `<button>` → Clickable button.
- `<input>` → Small text box for typing.
- `<label>` → Label for inputs.
- `<code>` → For showing code.

✅ **Do:** Use inline elements for *inside text or content*.
❌ **Don't:** Put block-level elements inside inline ones (e.g., don't put a `<div>` inside a `<span>`).

🎨 Part 2 – CSS Explained Line by Line

Think of **CSS** as the **clothing + makeup** of your website.
Each rule says:
👉 **"Find this element, and give it this style."**

Example CSS rule:

```
body {
  font-family: Arial, sans-serif;
  background: #f9f9f9;
  color: #333;
```

```
  line-height: 1.6;
}
```

Explanation for kids:

- `body { ... }` → We're styling the whole body of the website.
- `font-family: Arial, sans-serif;` → Change the handwriting (text style).
- `background: #f9f9f9;` → Paint the background light grey.
- `color: #333;` → Make the text dark grey (easier on eyes than pure black).
- `line-height: 1.6;` → Add breathing space between lines.

Another example:

```
nav {
  background: #222;
  color: white;
  padding: 1rem 2rem;
  display: flex;
  justify-content: space-between;
}
```

Kid-friendly breakdown:

- `background: #222;` → Paint the navbar dark grey.
- `color: white;` → Make the text white.
- `padding: 1rem 2rem;` → Add cushion (space) inside.
- `display: flex;` → Arrange items neatly in a row.
- `justify-content: space-between;` → Push logo to the left, links to the right.

Tip for Bootcamp:
👉 Show **before and after** applying each CSS rule live. Kids love to *see* the magic happen.

⚡ Part 3 – JavaScript & DOM

JavaScript is like the **brain + remote control**.

- **HTML** = skeleton.
- **CSS** = clothes.
- **JavaScript** = brain that makes it walk, talk, and react.

🔹 The DOM (Document Object Model)

Think of the **DOM** like a **family tree** of your webpage.

- The **document** is the root (grandparent).
- Inside it, you have children (like `<html>`, `<head>`, `<body>`).
- Each element has children of its own (like `<h1>`, `<p>`, `<img>`).

🖼️ Mermaid Diagram (DOM Tree Example)

```
Error parsing Mermaid diagram!

Cannot read properties of null (reading 'getBoundingClientRect')
```

🔹 JavaScript Example

```javascript
const contactBtn = document.getElementById("contactBtn");

contactBtn.addEventListener("click", function() {
  alert("Thanks for visiting my portfolio! 🚀");
});
```

Kid-Friendly Explanation:

- `document.getElementById("contactBtn")` → “Go to the DOM tree and find the button with the name `contactBtn`.”
- `.addEventListener("click", ...)` → “When someone clicks this button, wake up and do something.”
- `alert("...")` → “Shout a message on the screen.”

✅ Wrap-Up for Students

- **HTML = Structure (Skeleton)**
  → Blocks vs Inline elements, do’s & don’ts.
- **CSS = Style (Clothes/Makeup)**
  → Rules that change colors, fonts, layouts.
- **JavaScript = Brain**
  → Controls interactivity via the DOM.

Engagement Qs:

- “If HTML is the skeleton, and CSS is the clothes, what do you think JavaScript is?”
- “What would happen if we had HTML only, no CSS?”