Chapter 01:

HTML Structure

HTML (**HyperText Markup Language**) is the **skeleton of a webpage**.

It organizes content (text, images, links, videos, etc.) into a **structured document** that browsers can read and display.

1. Head = the brain of the page (info about the page, but not shown to users).
2. Body = the heart of the page (everything users actually see).

html Copy code

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My First Webpage</title>
</head>
<body>
  <h1>Hello World!</h1>
  <p>This is my very first webpage.</p>
</body>
</html>
```

🔍 Breakdown of Structure

1. **<!DOCTYPE html>**
   - Tells the browser this is an HTML5 document.
   - Without it, browsers may act weird or outdated.
2. **<html lang="en">**
   - The root element — wraps all HTML code.
   - lang="en" tells search engines and screen readers the language is English.
3. **<head>**
   - Contains metadata (data about data).
   - Includes info like:
     - Character set (<meta charset="UTF-8">)
     - Responsiveness (<meta name="viewport" ...>)
     - Page title (<title>My First Webpage</title>)
     - Links to CSS, fonts, icons, etc.

☞ Students should know: **Nothing in <head> shows up on the page**.

4. **<body>**
   - The visible content of your site.
   - Contains:
     - **Headings** (<h1> to <h6>)
     - **Paragraphs** ( )
     - **Images** (<img>)
     - **Links** (<a>)
     - **Lists** (<ul>, <ol>, <li>)
     - **Sections, divs, articles, navbars**, etc.

Key Takeaways for Students

- **HTML = Skeleton** → Provides structure.
- **Head = Info** → Hidden instructions.
- **Body = Content** → Everything users interact with.
- Always start with <!DOCTYPE html> and wrap everything in <html>.
- Use semantic tags (<header>, <main>, <footer>) for clean, professional structure.

What are HTML Elements?

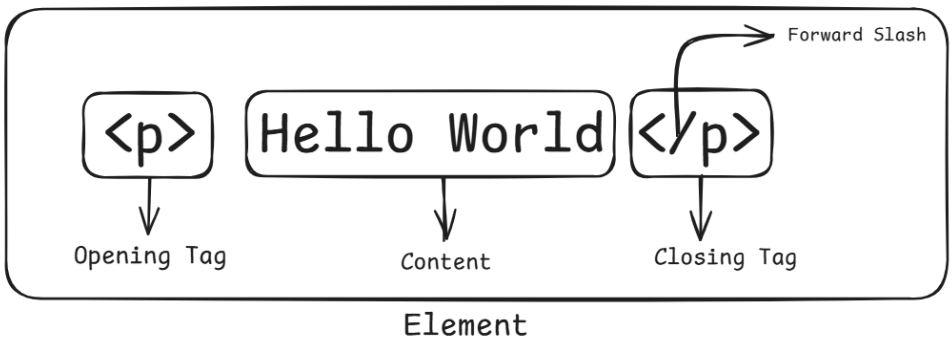
Think of **HTML elements** like little **containers (boxes)** that hold your content.

Each container tells the browser **what type of content** it is (a heading, a paragraph, a link, etc.).

Structure of an Element

Most elements come in a pair of tags**:**

- Opening Tag → ** (starts the element)**
- Closing Tag → ** (ends the element, notice the /)**
- Content → **The actual text, image, or info inside**



```html
<p>Hello, I am learning HTML!</p>
```

☞ Here:

- = opening tag
- = closing tag
- Hello, I am learning HTML! = content

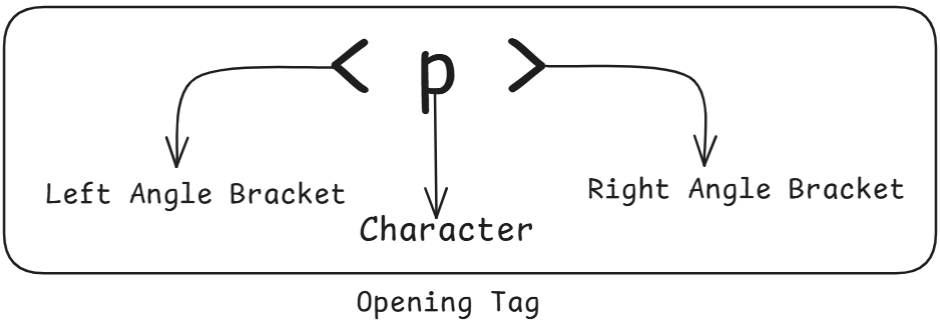
Together = one **HTML element**.

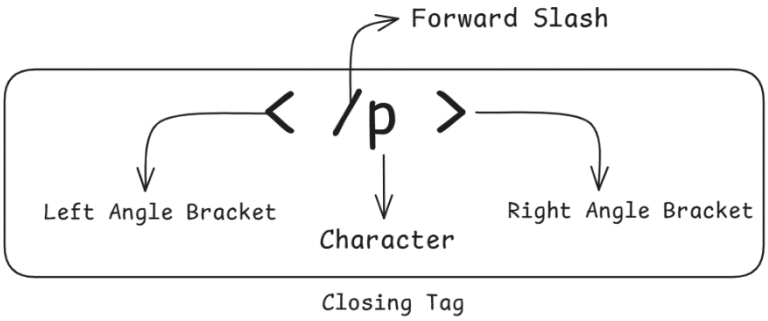
Easy Analogy

Think of HTML elements like a **sandwich** :

- The **top bread** = opening tag
- The **filling** = content (text/image)
- The **bottom bread** = closing tag

Without both breads, your sandwich (element) falls apart.





What Are Attributes?

Attributes are like **extra information** we give to an element.

They always go **inside the opening tag** and tell the browser something *more* about the element.

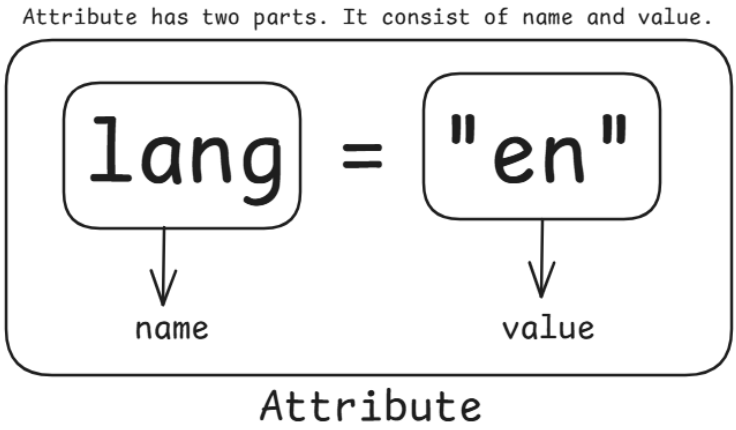
```html
<p lang="en-us">Paragraph in English</p>
```

Here:

- <p> = paragraph element
- lang="en-us" = **attribute** (tells the browser this paragraph is in US English)

📌 General Rules About Attributes

1. Attributes are always written in the **opening tag**.
2. They come in **name="value"** pairs.
   - lang = name
   - "en-us" = value
3. They give **extra meaning or behavior** to elements.



Easy Analogy (Attributes = Extra Details)

Think of an **element as a water bottle** 🧴:

- The **bottle itself** = element (<p>, <img>, <a>)
- The **label on the bottle** = attributes (brand, size, flavor, etc.)

The bottle works without the label, but the label gives you **more details** about what's inside.

1. <body> → The Stage 🎭

- Whatever you put inside <body> = shown directly in the browser window.
- This is the **main performance** — text, images, links, videos, buttons — all appear here.

Analogy:

Think of your webpage as a **theater**:

- The <body> is the **stage** where actors (content) perform.
- The audience (users) can see everything happening here.

```html
<body>
  <h1>Hello Students</h1>
  <p>This is visible on the screen!</p>
</body>
```

2. <head> → The Control Room

- The <head> doesn't show content directly to users.
- Instead, it holds **instructions and settings** for the browser.
- It can include:
  - <title> (page name on tab)
  - <meta> (info about language, description, responsiveness)
  - <link> (CSS files)
  - <script> (JavaScript files)

Analogy:

The <head> is like the **control room backstage**.

- The audience doesn't see it,
- but it controls how the show (webpage) looks and behaves.

```html
<head>
  <title>My First Page</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="style.css">
</head>
```

3. <title> → The Nameplate

- The text inside <title> appears:
  - At the top of the browser window
  - Or on the **tab** (if you have multiple tabs open)
- It also helps search engines know the page's name.

Analogy:

The <title> is like the **signboard above a shop**.

- People walking by don't see what's inside the shop (body),
- But they see the name on the board (title).

```html
<title>Student Portfolio</title>
```

Final:

- <body> = the **stage** → what users see.
- <head> = the **control room** → invisible settings.
- <title> = the **shop signboard** → tells name of the page in browser tab.

Summary of HTML Basics

- An **HTML page** is just a **text document**.
- It uses **tags** (inside < >) to give meaning to content.
- Tags = **elements** (like , <h1>, <a>).
- Most elements have **two tags**:
  - **Opening tag** → starts content
  - **Closing tag** → ends content
- **Attributes** add extra details to elements (e.g., lang="en", src="image.jpg").
- Attributes are always written as **name="value"**.
- To learn HTML, you need to:

1. Know which **tags** exist.
2. Understand what each **tag does**.
3. Learn where each **tag can be used**.

☞ In one line:

HTML = text + tags + attributes → together they build the structure and meaning of a webpage.

Chapter # 02

Text

Headings in HTML

✸ What are Headings?

Headings are like the **titles and subtitles** of a webpage. Headings are **semantic** because they **explain the role of the text** (main title, sub-section, etc.), not just make text bigger.

They help organize content and show importance.

In HTML, we have **6 levels of headings**:

html Copy code

```html
<h1>Main Heading</h1>
<h2>Sub Heading</h2>
<h3>Sub-Sub Heading</h3>
<h4>Smaller Section</h4>
<h5>Even Smaller</h5>
<h6>Tiniest Heading</h6>
```

🖼️ Example

html Copy code

```html
<h1>My Portfolio</h1>
<p>Welcome to my website!</p>

<h2>About Me</h2>
<p>Some details about myself...</p>

<h3>Education</h3>
<p>Graduated from XYZ...</p>

<h3>Skills</h3>
<p>HTML, CSS, JavaScript...</p>

<h2>Projects</h2>
<p>Here are some of my works...</p>
```

🔍 SEO Benefit

Search engines (like Google) read headings to understand **page structure**.

- <h1> tells **what the page is about** (main keyword goes here).
- <h2> and <h3> show **subtopics**.

👉 Example in search engine:

If your <h1> is **“Healthy Recipes”**, Google knows this page is about cooking/food.

Screen Reader Benefit

For visually impaired users, a **screen reader** can **jump through headings** instead of reading the whole page.

- <h1>: “Healthy Recipes”
- <h2>: “Breakfast Ideas”
- <h3>: “Oatmeal Bowl”

So they can **navigate quickly**, like scanning a book’s table of contents.

Analogy (Complete Now)

Headings = **road signs on a highway** 🚦

- <h1> = Big green sign → "Main City" (main topic)
- <h2> = Exit signs → "Neighborhoods" (sections)
- <h3> = Smaller signs → "Street names" (subsections)

Drivers (users), Google (SEO), and GPS (screen readers) **all use those signs to understand where they’re going**.

✅ So now you see:

- **Semantic meaning** → tells role of text.
- **SEO** → helps Google rank and understand content.
- **Screen readers** → make the page accessible for everyone.

📌 What is ?

- stands for **paragraph**.
- It is used to **group sentences into blocks of text**.
- Every paragraph starts with an **opening tag** and ends with a **closing tag** .

```html
<p>This is my first paragraph.</p>
<p>This is my second paragraph.</p>
```

By default:

- Each appears on a **new line**.
- Browsers automatically add a little space between paragraphs.

Easy Analogy

Think of a **notebook** 📓:

- Each new paragraph = like starting on a new line with a small gap from the previous one.
- It makes reading easier, instead of writing everything in one big block.

Semantic or Not?

✅ Yes, is a semantic element.

- Because it cleary tells the browser and developers:
  "This text is a paragraph."
- It's not just for styling; it adds meaning to the text.

Screen Reader Support

✅ Screen readers recognize as a paragraph.

- So when they read aloud, they add a pause between paragraphs.
- This helps visually impaired users understand where one thought ends and another begins.

📌 Key Points

- Use <p> for text content.
- It is semantic (adds meaning).
- Screen readers understand it and make content accessible.

1. → Bold Text

- The tag makes text **bold**.
- It is mainly **visual styling** (doesn't add special meaning).
- Used for highlighting words that should stand out, but without implying importance.

```html
<p>Today's <b>special offer</b> is available till midnight.</p>
```

Analogy:

Think of writing with a **thicker marker** ✍ — the word looks darker, stands out visually, but doesn't change the meaning of the sentence.

⚡ Semantic? → ❌ **Not semantic.**

⚡ Screen Readers? → They just read normally, no extra emphasis.

 BOLD

2. <i> → Italic Text

- The <i> tag makes text **italic** (slanted).
- Used for foreign words, thoughts, technical terms, ship names, etc.
- Unlike , <i> sometimes adds a little **semantic hint** (like "this is different from normal text").

```html
<p>He whispered, <i>I hope she understands.</i></p>
<p>My favorite ship is <i>Titanic</i>.</p>
```

Analogy:

Imagine you're **speaking in a softer or different tone** 🎭. That's what italics do — same words, but *spoken differently*.

⚡ Semantic? → Partly semantic (depends on context).

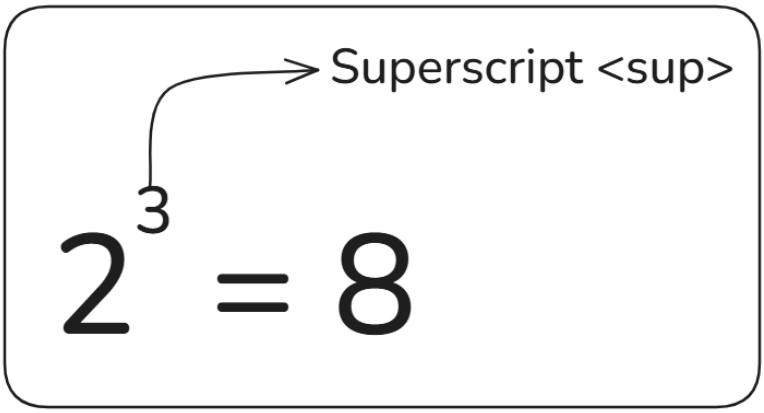
⚡ Screen Readers? → Some may read it with slight emphasis (but often just read it normally).

 Italic Text

3. <sup> → Superscript

- The <sup> tag displays text **slightly above the normal line**.
- Used for math powers, dates, footnotes.

```html
<p>2<sup>3</sup> = 8</p>
<p>Today is July 4<sup>th</sup>.</p>
```



Analogy:

Think of someone **standing on a small stool** □ in a group photo — they appear a little higher than the rest.

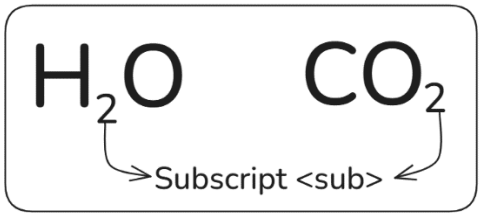
⚡ Semantic? → ✅ Yes (it tells the browser: this is superscript text).

⚡ Screen Readers? → They announce it properly, like "two to the power of three."

4. <sub> → Subscript

- The <sub> tag displays text **slightly below the normal line**.
- Used in chemical formulas, footnotes, etc.

```html
<p>H<sub>2</sub>O is water.</p>
<p>CO<sub>2</sub> is carbon dioxide.</p>
```



Analogy:

Think of someone **sitting down while others stand** — they appear a little lower than the rest.

⚡ Semantic? → ✅ Yes (it tells the browser: this is subscript text).

⚡ Screen Readers? → They read it clearly, e.g., "H two O."

White Space

🌟 What is White Space in HTML?

- **White space** means **spaces, tabs, and line breaks** in your code.
- Browsers don't care if you write **one space or ten spaces** — they show it as **just one space** on the screen.
- This behavior is called **white space collapsing**.

```html
<p>Hello       World!</p>
```

👉 What you see in the browser:

Hello World!

(Only one space is shown, even though there are many in code.)

Line Breaks in Code

```html
<p>
Hello
World!
</p>
```

👉 Browser output:

Hello World!

(Line breaks in code are also treated as single spaces.)

Indentation (for Readability)

```html
<p>
   This is a paragraph
   that is written across
   multiple lines in the code.
</p>
```

👉 Browser output:

This is a paragraph that is written across multiple lines in the code.

Even though you pressed Enter several times, the browser collapses all the extra spaces/line breaks.

<pre> → Preformatted Text

1. <pre> means **preformatted**.
2. It shows text **exactly as you typed it** — with the same spaces, line breaks, and indentation.
3. The browser doesn't collapse white space inside <pre>.

```html
<pre>
Hello       World!
Line 1
    Line 2 (indented)
Line 3
</pre>
```

👉 Browser output (exactly the same):

```scss
Hello       World!
Line 1
    Line 2 (indented)
Line 3
```

Analogy:

Think of <pre> like a photocopy machine 🖷.

- Whatever you put inside, it copies exactly — same gaps, same lines, nothing gets squished.

2.   → Non-Breaking Space

-   stands for **Non-Breaking Space**.
- It's an **HTML entity** used to add a space that the browser won't collapse.
- It adds a **space that the browser will not collapse**.
- Also prevents the text from breaking into a new line at that point.

```html
<p>Hello   World!</p>
```

☞ Browser output:

Hello World!

(Here, 3 spaces are kept because of).

Analogy:

Think of   as a glue space.

- Normal spaces = soft chalk (can fade/merge).
-   = a glued gap — it sticks and stays where you put it.

Line Breaks

1. The <br /> tag is used to **start a new line inside the same paragraph**.
2. It does not create a new paragraph — just moves the text down like pressing **Enter** in a poem.

3. It is an **empty element** (it doesn't need a closing tag).

```html
<p>The Earth<br />gets one hundred tons heavier every day<br />due to falling space dust.</p>
```

☞ **Browser result:**

The Earth

gets one hundred tons heavier every day

due to falling space dust.

Analogy

Think of <br /> like pressing **Shift + Enter** in Microsoft Word.

- You're still in the same paragraph, but the text goes to the **next line**.

🌟 Horizontal Rule <hr />

✅ Definition

- The <hr /> tag adds a **horizontal line** across the page.
- It is used to **separate topics, sections, or ideas**.
- It is also an **empty element**.

```html
<p>Venus is the only planet that rotates clockwise.</p>
<hr />
<p>Jupiter is bigger than all the other planets combined.</p>
```

☞ Browser result:

Venus is the only planet that rotates clockwise.

Jupiter is bigger than all the other planets combined.

Analogy

Think of <hr /> like the **divider line in a book or magazine**.

- When a new scene, chapter, or section starts, you often see a line to show the **change of theme**.

🌟 Empty Elements in HTML5

- Both <br /> and <hr /> are called **empty elements**.
- Why? Because they **don't wrap text** inside them like ... .
- They are written as <br /> or <hr />.

In **HTML4 and XHTML**, developers often wrote empty elements with a slash:

```html
<br />
<hr />
```

1. because XHTML followed XML rules (self-closing).
2. In **HTML5**, the slash / is **not required**.

   You can simply write:

```html
<br>
<hr>
```

- Both versions (
 and <br />) **work the same way** in modern browsers.

✅ Best Practice

- If you're writing **pure HTML5**, just use <br> and <hr>.
- If you want your code to be **XHTML-compatible** (old-school strict), you can still use <br /> and <hr />.

Easy Analogy

Think of it like **closing a door** 🚪:

- In XHTML, you always had to **lock it with a key** (/).
- In HTML5, the door **closes automatically** — no need for the key.

Empty elements like <br> and <hr> don't need a / at the end anymore.

✅ Quick Recap for Students

-
 → **Line break inside text** (like pressing Enter in a poem).
- <hr /> → **Horizontal line to separate sections** (like a divider in books).
- Both are **empty elements** → no closing tags needed.

🌟 Semantic Markup

- **Semantic markup** means using HTML tags that **add meaning** to the text, not just style.
- These elements tell the browser, screen readers, and search engines *what the text means*, not just how it looks.
- Example:
  - → shows emphasis (important words).
  - <blockquote> → shows a quotation.

👉 Browsers often style them differently (like italics or indentation), but the main purpose is meaning, not looks.

✅ Why use Semantic Markup?

1. **Accessibility** → Screen readers can read with the right tone (like stressing emphasized words).
2. **SEO** → Search engines understand your content better.
3. **Clarity** → Other developers know the purpose of the text, not just the design.

Easy Analogy

Think of semantic markup like **labels on food packaging** 🏷️:

- "Sugar-free," "Gluten-free," "Vegan" → These labels don't change the taste, but they **add meaning** about what the food is.

— Strong Importance

- is used when text is **very important**.
- By default, browsers show in **bold**.
- It means "pay special attention to this."

html Copy code

```html
<p>You must <strong>wear a helmet</strong> while riding.</p>
```

Output:

You must **wear a helmet** while riding.

Here the words show **serious importance**.

Screen Reader & SEO

- Screen readers pronounce text with **heavier stress** in the voice.
- Search engines treat text as **high-value keywords**.

Analogy

Think of like when a teacher writes something on the board in **bold red marker** ✍️.

🌟 — Emphasis

✅ Definition

- stands for **emphasis**.
- It is used when a word or phrase should be **stressed** in a sentence.
- By default, browsers display text in *italics*.

```html
<p>I said <em>maybe</em>, not definitely.</p>
```

Output:

I said *maybe*, not definitely.

Here the word **maybe** changes the meaning if emphasized.

Screen Reader & SEO

- Screen readers pronounce words with **slight stress in the voice**.
- Search engines understand that the word has **extra importance** in the context.

Analogy

Think of like when you're speaking and you **underline one word with your tone**.

👉 Example:

"I said *maybe*, not definitely."

It changes the feeling of the sentence.

- → Emphasis (like italics in speech).
- → Strong importance (like bold warnings or highlights).
- Both are semantic elements → they add meaning, not just style.

🌟 Quotations in HTML

In HTML, we use quotation elements to show text that comes from someone else's words.

- <blockquote> → for long quotes (whole sentences or paragraphs).
- <q> → for short quotes inside a sentence.
- Both can use cite="" to show the source (like a URL).

  There are two special elements for showing quotes:

1. <blockquote> → Long Quotes

✅ Definition

- <blockquote> is used for **long quotations**, usually a full sentence or paragraph.
- Browsers usually **indent** this text to make it stand out.
- You can also add a cite attribute to give the **source (URL)** of the quote.

```html
<blockquote>
  <p>Life is like riding a bicycle. To keep your balance, you must keep moving.</p>
</blockquote>
```

Browser Output:

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Quam consequatur

> Life is like riding a bicycle. To keep your balance, you must keep moving.

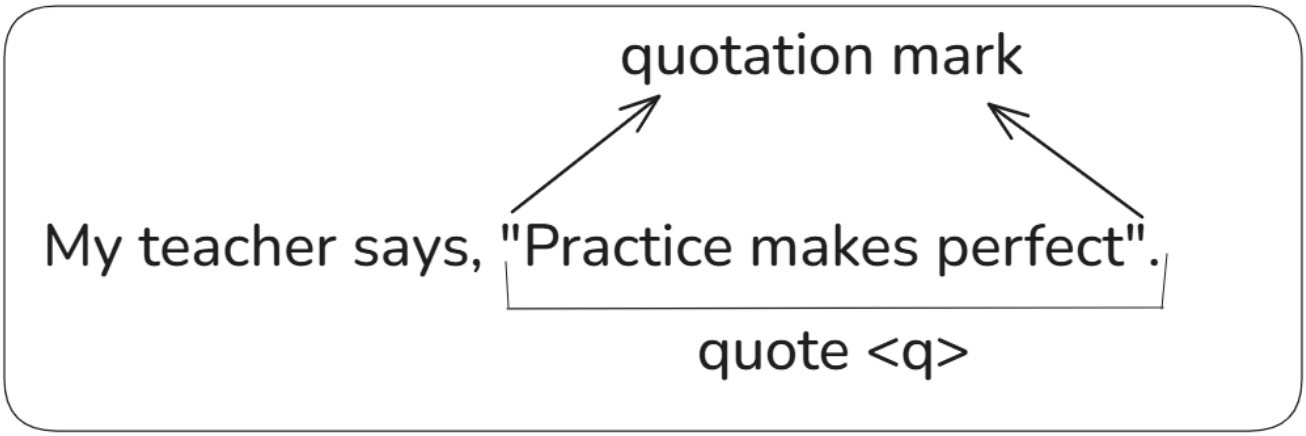
Lorem ipsum dolor sit amet consectetur, adipisicing elit. Corporis eos

<q>

- Use it for **short quotes** inside a sentence.
- Browser will **add quotation marks** automatically.

```html
<p>My teacher says, <q>Practice makes perfect.</q></p>
```

My teacher says, “Practice makes perfect.”



Easy Analogy

- <blockquote> = like writing a **whole quote on a new line** in your notebook.
- <q> = like adding **quotation marks** in the middle of your sentence.

What is the cite Attribute in <blockquote>?

- The cite attribute is used in <blockquote> (or <q>) to give the **source (reference)** of the quote.
- Its value is usually a **URL (link)** to the place where the quote originally came from (like a book page, website, or article).
- Important: The cite link is **not shown directly on the webpage**. It’s invisible for users unless you also write the source manually in text.
- But search engines, screen readers, and browsers can use it to **understand where the quote comes from**.

html Copy code

```html
<blockquote cite="https://en.wikipedia.org/wiki/Albert_Einstein">
  <p>Life is like riding a bicycle. To keep your balance, you must keep moving.</p>
</blockquote>
```

Easy Analogy

Think of cite like the **footnote or reference** in your school essay:

- You write a nice quote in your paper.
- Then at the bottom, you write in small letters → *Source: Wikipedia*.
- Readers might not always check it, but teachers (and Google 😊) love that you gave credit.

✅ In short:

cite = **gives credit to the original source of the quote.**

What is <abbr> in HTML?

- The <abbr> element is used when you write an **abbreviation** or **acronym** (short form).
- With it, you add a title **attribute** to show the **full meaning** of that short form.
- In **HTML5**, both abbreviations and acronyms are written with <abbr> (the old <acronym> tag is no longer used).

```html
<p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>
```

Copy code

👉 On the webpage, you'll see:

The WHO was founded in 1948.

But when you hover your mouse on **WHO**, a tooltip will show:

World Health Organization

Easy Analogy

Think of <abbr> like when you're texting your friend:

- You write **"LOL"** (short form).
- But when someone doesn't know, you explain: "It means *Laugh Out Loud*."
- <abbr> does the same thing — it shows both the short form (LOL) and its **hidden full form**.

✅ In short:

<abbr> **= used for short forms, with the full meaning inside** title**.**

1. <cite> (Citations)

- <cite> is used when you **reference a piece of work** → like a **book, movie, research paper, website, or article**.
- It tells the browser (and search engines) that this text is the **title of a creative work**.
- By default, browsers show <cite> in *italics*.
- Not for people's names in HTML5 (but was okay in HTML4).

```html
<p>My favorite novel is <cite>Pride and Prejudice</cite> by Jane Austen.</p>
<p>The movie <cite>Inception</cite> changed the way people saw dreams.</p>
```

🎭 Real-Life Analogy:

Think of <cite> like **putting a book or movie title in italics** when you write an essay.

Just like in school essays we write:

"I got this idea from Hamlet by Shakespeare."

Here, *Hamlet* is the **cite** part.

<dfn> — Definition

👉 **Purpose:** Use <dfn> the *first time* you explain a **new term or concept** in your document.

📌 It marks that word/phrase as being **defined**.

📌 Some browsers style it in italics, some don't.

```html
<p><dfn>Artificial Intelligence</dfn> is the simulation of human intelligence in machines.</p>
<p>The <dfn>Blockchain</dfn> is a decentralized digital ledger used to record transactions.</p>
```

Browser Output:

Artificial Intelligence is the simulation of human intelligence in machines.

The *Blockchain* is a decentralized digital ledger used to record transactions.

🎭 Real-Life Analogy:

Imagine you're a teacher.

The **first time** you introduce a difficult word to your students, you underline it or highlight it:

"Photosynthesis (definition here) is how plants make their food."

That highlighted word = <dfn>

Author Details

<address> — Contact Information Tag

The <address> tag is used to show contact details of the author/owner of the page or article.

It doesn't mean just "home address" — it can be:

- Physical address (street, city, country 🏠)
- Phone number 📱
- Email address 📧
- Website link 🔗

📌 By default, browsers usually display it in italics.

```html
<footer>
  <p>Written by John Doe</p>
  <address>
    Contact me at: <a href="mailto:johndoe@example.com">johndoe@example.com</a><br>
    Phone: +1 234 567 8900<br>
    123 Web Street, Internet City, USA
  </address>
</footer>
```

🎭 Real-Life Analogy:

Think of <address> like the **contact info section on a business card**.

When you meet someone, they hand you a card with:

- Name
- Email
- Phone number
- Office address

That's exactly what <address> does inside a webpage — it's the "business card" of the page's author.

The **main purpose** of <address> in HTML5 is to hold **contact information about the author/owner of that webpage or article**.

💻 Example (❌ Wrong use):

```html
<p>Pizza Hut is located at:</p>
<address>123 Main Road, Karachi, Pakistan</address>
```

you can use <address> for your own address/contact details, but **not for random locations**.

◆ When NOT to use <address>

If you are writing about:

- A pizza shop location 🍕
- A tourist place 🗽
- A random person's home address 🏠

👉 Then you should **not** use <address>. In such cases, just use a normal <p> or .

<ins> — Inserted Text

👉 Used when **something new is added** to the document.

📌 Browsers usually show it as **underlined**.

```html
<p>This book costs <ins>$20</ins> now.</p>
```

This book costs $20 now.

<del> — Deleted Text

👉 Used when **something is removed/edited out** from the document.

📌 Browsers usually show it with a **strike-through line**.

```html
<p>This book costs <del>$25</del>.</p>
```

This book costs ~~$25~~.

Think of it like editing a notebook:

- When you cut out something with a line through it → <del> ✍
- When you underline a new correction → <ins> ✅

For example, in your diary you might write:

~~I will wake up at 10am~~ ~~will wake up at 10am~~

I will wake up at 7am

That's exactly <del> and <ins> in HTML.

```html
<p>
  Old price: <del>$50</del>
  New price: <ins>$35</ins>
</p>
```

Old price: ~~$50~~ New price: $35

👉 These tags are super useful in **articles, blogs, or e-commerce sites** where you want to show edits, corrections, or discounts.

<s> — Strikethrough (No Longer Accurate / Relevant)

👉 The <s> element is used when something is no longer true, valid, or relevant, but you don't want to delete it from the document.

📌 By default, browsers show it with a line through the text.

⚠️ Important: <s> is different from <del>.

- <del> = something was removed (like an edit in a document).
- <s> = the info is still there but it's just outdated or incorrect.

```html
<p>The shop is open from <s>10am to 6pm</s> now it opens 24/7.</p>
```

The shop is open from ~~10am to 6pm~~ now it opens 24/7.

🎭 Real-Life Analogy:

Think of a price tag in a store:

- Old price is crossed out → ~~$100~~
- New price is written next to it → $80

That's <s>. You're saying *"this is no longer valid, but I'm keeping it for reference."*

```html
<p>Winter Sale! <s>$120</s> $90 only.</p>
```

Winter Sale! ~~$120~~ $90 only.

- <s> = shows outdated or irrelevant info.
- <del> = shows deleted/removed text.

<del> Example (Editing Text / Document Updates)

```html
<p>Our meeting is scheduled on <del>Monday</del> Wednesday.</p>
```

☞ Output:

Our meeting is scheduled on ~~Monday~~~~Monday~~ Wednesday.

(Here Monday was deleted, replaced with Wednesday.)

<s> Example (Outdated Info but Still Visible)

```html
<p>We are located at <s>123 Old Street</s> now at 456 New Avenue.</p>
```

We are located at ~~123 Old Street~~ ~~123 Old Street123 Old Street~~ now at 456 New Avenue.

(Old address is outdated but kept for reference.)

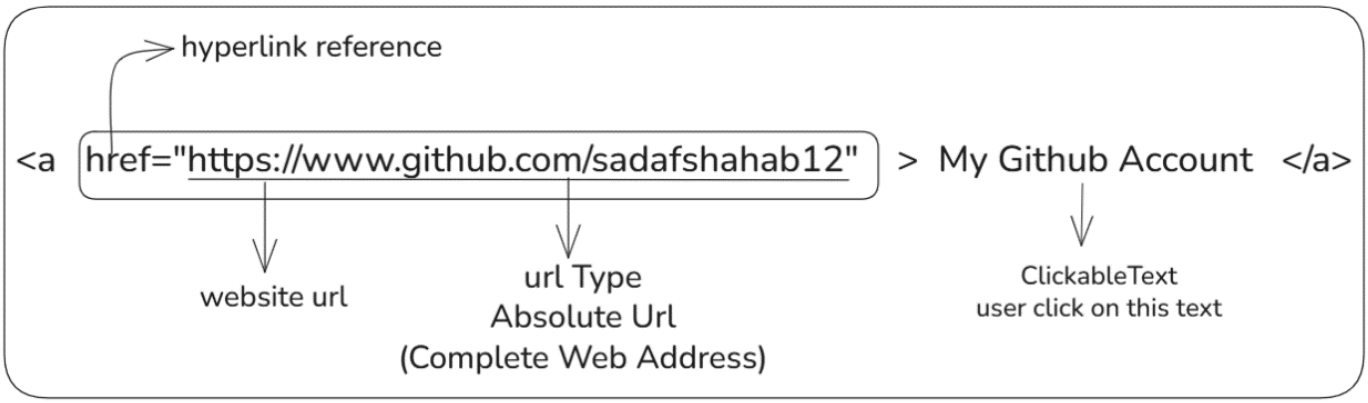
Chapter 03 Links

◆ What is a Link?

A **link (hyperlink)** is text or an image that, when clicked, takes you to another place — another webpage, another section of the same page, or even opens your email app.

☞ Links are made using the <a> tag in HTML.

Writing Links



<a> Tag (Anchor Element) in HTML

- <a> is used to create links on a webpage.
- It needs an attribute called href (hyperlink reference).
- The value of href is the URL (address) of the page or resource you want to open.

```html
<a href="https://www.google.com">Go to Google</a>
```

👉 Takes you from your website to Google.

Output:

Go to Google (blue, underlined by default → clickable).

- If you are linking to **another website**, you must use the **full address** (absolute URL) like: https://example.com.
- By default, browsers show links as **blue and underlined**.

```html
<a href="https://www.github.com/sadafshahab12">My Github Account</a>
```

Before visit the link:

My Github Account

After visit the link:

My Github Account

✅ **Easy one-liner:**

<a> tag makes clickable links, and the href attribute tells the browser **where to go** when the link is clicked.

◆ Types of Links in HTML

1. Link from one website to another 🌍

```html
<a href="https://www.google.com">Go to Google</a>
```

👉 Takes you from your website to Google.

◆ Absolute URL

🔹 URL (Uniform Resource Locator)

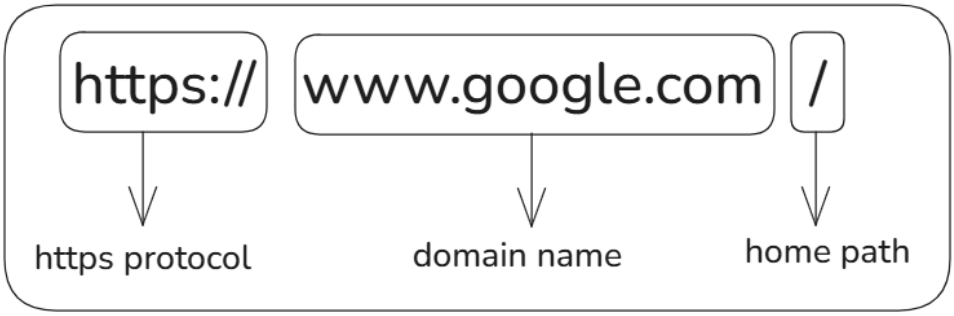
👉 A URL is just the **address of a webpage**.

Example: https://www.google.com

It’s like the **home address** of a house — if you type it into the browser, you go directly to that page.

☞ An **absolute URL** is the **full web address**, starting with:

- protocol (http:// or https://)
- domain name (like example.com)
- and sometimes the path to a page or file



🎭 Real-Life Analogy:

Think of it like **writing a full address on an envelope**:

"123 Street, Karachi, Pakistan"

No matter where you are in the world, the letter will reach that exact place.

Same with an absolute URL → it works from **anywhere on the internet**, because it points to the **complete location**.

Same with an absolute URL → it works from **anywhere on the internet**, because it points to the **complete location**.

✅ Absolute Url Definition:

An **absolute URL** is the full and complete address of a webpage on the internet. It always starts with http:// or https:// and includes the website’s domain.

Relative URL

- A **relative URL** is a **shortcut link** to another page **within the same website**.
- You **don’t need to type the full domain name** (https://example.com) — just the path to the page.

- Very useful when working **locally** on your computer or in a small website folder structure.

You can link them like this:

```html
<ul>
  <li><a href="index.html">Home</a></li>
  <li><a href="about.html">About</a></li>
  <li><a href="contact.html">Contact</a></li>
</ul>
```

✅ Clicking any of these links will go to the correct page without writing the full web address.

```html
<a href="about.html">About Us</a>
```

👉 Opens another page (about.html) within the same website.

🎭 Real-Life Analogy

Think of it like directions inside a building**:**

- Absolute URL = giving someone the **full street address of your building**
- Relative URL = saying **"Go to the 2nd room on the left"** inside the same building

Both get you there, but the relative one is shorter and easier when you're already "inside" the website.

Linking to a Part of the Same Page (Anchor Links)

- Sometimes a page is long.

  You might want to:

- Jump from the top table of contents to a section below
- Go back from the middle of the page to the top

  How to do it:

  1. Give the target element an id.

     Example: <h2 id="projects">My Projects</h2>
  2. Use an <a> tag with href="#idvalue" to jump to that section.

Example: Go to Projects

3. Link to a part of the same page (jump link / anchor) 🎯

```html
<a href="#contact">Go to Contact Section</a>

<!-- Later in the same page -->
<h2 id="contact">Contact Us</h2>
```

👉 Jumps directly to the contact section of the same page.

```html
<!-- Top of the page: Table of Contents -->
<nav>
  <a href="#about">About Me</a> |
  <a href="#projects">Projects</a> |
  <a href="#contact">Contact</a>
</nav>

<!-- Sections -->
<h2 id="about">About Me</h2>
<p>Hi! I'm Sadaf, a MERN Stack Developer...</p>

<h2 id="projects">Projects</h2>
<p>Here are some of my projects...</p>

<h2 id="contact">Contact</h2>
<p>Email me at <a href="mailto:sadafshahab07@gmail.com">sadafshahab07@gmail.com</a></p>

<!-- Link back to top -->
<a href="#top">Back to Top</a>
```

🎭 Easy Analogy

Think of it like **a map with pins**:

- Each section of the page is a pin (id)
- Clicking a link is like saying **"take me to this pin directly"** instead of scrolling manually

✅ **Easy definition for students:**

Use the id attribute to **mark a section** and href="#idvalue" to **jump to it** — it's called an **anchor link**.

4. Link that opens in a new browser tab/window

```html
<a href="https://example.com" target="_blank">Visit Example</a>
```

👉 Opens in a new tab/window.

Email Links (mailto:) in HTML

- Use <a> tag with href="mailto:sadafshahab07@gmail.com"
- Clicking the link **opens your email program** with the "To" field already filled.
- Looks like a normal clickable link on the webpage.

```html
<a href="mailto:sadafshahab07@gmail.com">Send Email to Sadaf</a>
```

5. Link that starts email 📧

```html
<a href="mailto:sadafshahab07@gmail.com">Send Email to Sadaf</a>
```

👉 Opens email app with the "To" address already filled in.

🎭 Easy Analogy

Think of it like a **pre-addressed letter**:

You just click, and your email is ready to send — no need to type the address manually.

✅ **Easy definition:**

An **email link** uses mailto: in the href so users can click and start an email to a specific address automatically.

🎭 Link Real-Life Analogy

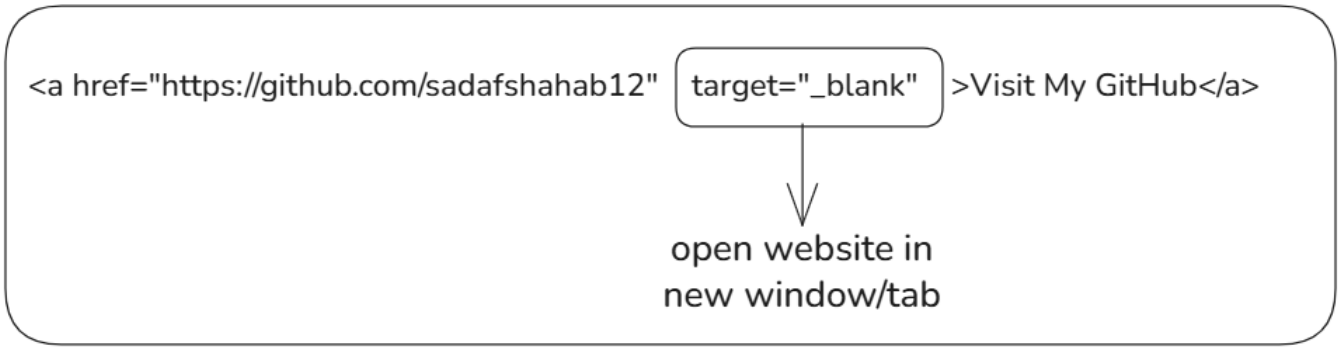
Think of links as **roads and doors** in a city:

- From one city to another = website to website.
- From one street to another in the same city = one page to another page on the same site.
- From one room to another in the same building = jump link in the same page.
- Special doors = opening a new window or starting an email.

◆ Opening Links in a New Window

- Use the target attribute in the <a> tag.
- Set it to _blank → Link
- This makes the link **open in a new browser tab or window**.

```html
<a href="https://www.github.com/sadafshahab12" target="_blank">Visit My GitHub</a>
```



Clicking it → opens Google in a **new tab**, keeping your current page open.

🎭 Another Analogy

It's like **opening a new window on your computer**:

- You start working on a new window (the link)
- Your original window (your current webpage) stays open
- You can switch back and forth without losing anything

✅ Tips:

- Use it mainly for **external websites**.
- Let users know the link will open a new tab — it's polite!

Easy definition:

target="_blank" makes a link open in a **new window or tab** so users don't leave your page.

Linking to a Specific Part of Another Page

- Sometimes you want to **jump to a section** on **a different page**, not the current one.
- The other page must have an id **attribute** for the section you want to go to.
- The **syntax** is:

```ini
href="URL-of-page#id-of-section"
```

💻 Example:

Suppose we have **another page** called about.html and it has a section like this:

```html
<h2 id="team">Our Team</h2>
<p>Meet our amazing developers...</p>
```

Now, from your **homepage** you can link directly to that section:

```html
<a href="about.html#team">Meet Our Team</a>
```

Clicking this link → opens about.html and **jumps directly to the "Our Team" section**.

🎭 Easy Analogy

Think of it like **sending someone to a specific chapter in a book** instead of just giving them the book.

- Book = the page
- Chapter = the section with an id
- Link with #id = "Go straight to this chapter!"

✅ **Easy definition for students:**

To link to a section on another page, use the page URL + # + the id of the target section.

◆ Links in HTML — Easy Summary

1. **Links are made with** <a>
   - This is called the **anchor tag**.
2. href **tells the browser where to go**
   - Can be another page, a section of a page, or even an email address.

3. **Relative links are better for your own site**
   - Shorter and easier to manage than full URLs.
4. **Email links**
   - Use mailto: in href to open an email program automatically.
5. **Target specific sections with** id
   - Use id on any element and href="#idvalue" to jump to that section.

Chapter 04

Images

◆ Why We Use Images on Websites

- **Visual appeal:** Images make a page **look attractive** and **professional**.
- **Communication:** Pictures can explain ideas faster than text (like diagrams or charts).
- **Branding:** Logos and icons help identify your website or business.
- **Illustration:** Photos, drawings, or screenshots make content easier to understand.

```html
<img src="logo.png" alt="Company Logo">
```

- src → the file path of the image
- alt → text shown if the image doesn't load or for screen readers

🎯 Things to Consider

1. **Image format** – JPG, PNG, SVG, GIF
2. **Image size** – not too big, so pages load fast
3. **Optimization** – compress images for faster loading

🎭 Easy Analogy

Think of images like **decorations in a classroom**:

- A blank wall looks boring → add posters, charts, or photos to make it appealing and informative.

✅ Easy definition:

Images on websites are used to **make content attractive, communicate ideas faster, and represent the brand visually**.

◆ Choosing Images for Your Website

Why Images Matter

- **Set the tone quickly:** A picture can give the feel of your website **faster than words**.
- **Make a site engaging:** Good images make your website look **professional and attractive**.

- **Consistency:** If you show several images together (like team members or products), use a **simple, uniform background** so they look neat.

💡 Tips for Choosing Images

Images should:

1. **Be relevant** – match the topic or content
2. **Convey information** – explain ideas without too much text
3. **Convey the right mood** – fun, serious, professional, etc.
4. **Be instantly recognizable** – people should understand it at a glance
5. **Fit the color palette** – match your site's style and colors

⚠ Important Notes

- **Copyright matters:** Don't take images from other websites without permission.
- **Stock photos:** You can buy professional images from sites like:
  - www.istockphoto.com
  - www.gettyimages.com
  - www.veer.com
  - www.sxc.hu
  - www.fotolia.com

🎭 Easy Analogy

Think of your website like a **poster or magazine cover**:

- The right pictures grab attention and tell the story instantly.
- Bad or random images can make it look messy and confusing.

✅ Easy definition for students:

Images should **match your content, look good together, and help people understand or feel your message quickly**.

◆ Adding Images with <img> Tag

- <img> is used to show an image on a webpage.
- It is an **empty element** → no closing tag is needed (<img>).

◆ Required Attributes

1. src – Source of the image (where the browser can find it)
   - Usually a **relative URL** if the image is in your website folder.

```html
<img src="images/logo.png">
```

2. alt – Alternative text for the image
   - Describes the image for visually impaired users or if the image doesn't load.

```html
<img src="images/logo.png" alt="Company Logo">
```

◆ Optional Attribute

1. title – Extra info about the image

- Shows as a tooltip when hovering

```html
<img src="images/logo.png" alt="Company Logo" title="Our Company Logo">
```

💻 Full Example

```html
<h2>My Website Logo</h2>
<img src="images/logo.png" alt="Company Logo" title="Our Company Logo">
<p>Welcome to our website!</p>
```

If the image is **just decoration** and has no meaning:

```html
<img src="images/decorative-line.png" alt="">
```

🎭 Easy Analogy

Think of <img> like **putting a photo in a book**:

- src → where the photo comes from (folder or album)
- alt → a caption describing it for someone who can't see it
- title → a note you write on the photo for extra info

✅ Easy definition:

<img> shows pictures on a webpage. Always use src for the image file, alt for a description, and optionally title for extra info.

Setting Height and Width of Images

- height → sets the **height of the image** in pixels
- width → sets the **width of the image** in pixels

Why It's Important

- Images can be slow to load.
- Specifying size reserves space on the page, so the text doesn't jump around while the image is loading.
- Helps the page look neat and organized.

```
<img
  src="https://natureconservancy-h.assetsadobe.com/is/image/content/dam/tnc/nature/en/photos/s/h/shutterstock_1512536354.jpg?
  crop=0%2C317%2C2664%2C1465&wid=1300&hei=715&scl=2.0492307692307694"
  alt="butterfly"
  width="400"
  height="200"
/>
```

🎯 Easy Analogy

Think of it like **reserving a frame for a photo on a wall**:

- You know exactly how much space the photo will take
- Even if the photo isn't ready yet, the wall looks neat

✅ Extra Tip:

Nowadays, it's better to control image size with **CSS** instead of HTML attributes, because it's more flexible.

◆ Where to Place Images in HTML

Key Idea:

- **Where you put an** <img> **tag** in your code affects **how it looks on the page**.

3 Common Placements:

Before a paragraph

```html
<img src="bird.gif" alt="Bird">
<p>Text here...</p>
```

- Because <p> is a **block element** → always starts on a new line.

At the start of a paragraph

```html
<p><img src="bird.gif" alt="Bird">Text here...</p>
```

- Text **aligns next to the bottom of the image**.
- Image and text flow together → <img> is an **inline element**.

In the middle of a paragraph

```html
<p>Text before <img src="bird.gif" alt="Bird"> text after.</p>
```

- Image appears **between words**. Text flows around it.

🎯 Simple Analogy

- **Block elements** → like a **new paragraph on a page**, always starts on a new line.
- **Inline elements** → like **words in a sentence**, they flow together.
- <img> is **inline**, but if it comes before a block element (like <p>), the block element moves to the next line.

◆ Aligning Images Horizontally in HTML

Key Idea:

- Older HTML used the align **attribute** on <img> to position images.
- **Values:**
  - left → image on the left, text flows on the right
  - right → image on the right, text flows on the left
- **Note:** align is **removed in HTML5** → now we use **CSS** for alignment.

💻 Old Example (for reference)

html Copy code

```html
<p><img src="bird.gif" alt="Bird" width="100" height="100" align="left">
There are around 10,000 living species of birds that inhabit different ecosystems...</p>

<p><img src="bird.gif" alt="Bird" width="100" height="100" align="right">
There are around 10,000 living species of birds that inhabit different ecosystems...</p>
```

- First image → left, text wraps to the right
- Second image → right, text wraps to the left

There are around 10,000 living species of birds that inhabit different ecosystems... Lorem ipsum dolor sit amet consectetur adipisicing elit. Tempore, enim facere nisi omnis praesentium expedita nostrum doloremque officia accusantium asperiores ea, porro incidunt, explicabo dignissimos non facilis eligendi corrupti voluptatibus!

There are around 10,000 living species of birds that inhabit different ecosystems... Lorem ipsum dolor sit amet consectetur adipisicing elit. Inventore veritatis eaque assumenda, perspiciatis rerum adipisci cupiditate recusandae, iure itaque placeat quas sit. Praesentium voluptates magnam iusto distinctio explicabo consequatur suscipit?

🎯 Simple Analogy

Think of it like **placing a photo on a page in a magazine**:

- Left-aligned photo → text wraps nicely on the right
- Right-aligned photo → text wraps on the left
- Makes the page look **neater than just putting text next to a floating image**

✅ Important Note:

- Modern websites **use CSS** (float, margin, padding) instead of align
- This allows more control over spacing and alignment

◆ Aligning Images Vertically (Old HTML)

Key Idea:

- The **old** align **attribute** could also **control vertical alignment** of images with text.
- **Values for vertical alignment:**
  1. top → first line of text aligns with the **top of the image**
  2. middle → first line aligns with the **middle of the image**
  3. bottom → first line aligns with the **bottom of the image**

```
<p><img src="bird.gif" alt="Bird" width="100" height="100" align="top">Text starts aligned with top of image.</p>

<p><img src="bird.gif" alt="Bird" width="100" height="100" align="middle">Text starts aligned with middle of image.</p>

<p><img src="bird.gif" alt="Bird" width="100" height="100" align="bottom">Text starts aligned with bottom of image.</p>
```


Text starts aligned with top of image.


Text starts aligned with middle of image.


Text starts aligned with bottom of image.

This changes **where the text starts next to the image** vertically.

🎯 Simple Analogy

Think of text as **riding next to a picture on a bus**:

- top → text sits at the top edge of the picture
- middle → text sits in the middle of the picture
- bottom → text sits at the bottom edge of the picture

✅ Important Notes (New vs Old)

- **Old:** align="top|middle|bottom" → works in older HTML
- **New (HTML5):** Use **CSS** to control vertical alignment → more flexible and recommended
    - Example with CSS: vertical-align: top;, middle;, bottom;
- **Text wrapping:** If you want **all text to wrap around the image**, use **CSS float** instead of relying on align.

◆ Easy Takeaway

- Vertical alignment using align is **old method** → mostly seen in older websites
- Modern websites use **CSS float + vertical-align** for better control
- Horizontal alignment (left/right) and vertical alignment (top/middle/bottom) are **now handled by CSS**

◆ Three Rules for Creating Website Images

1. **Right Format**
    - Use **JPEG, PNG, or GIF**
    - Wrong format → image looks blurry or slows down the page
2. **Right Size**
    - Save the image at the **exact width and height** you want on the page
    - Too small → stretched and blurry
    - Too big → slow to load
3. **Correct Resolution**
    - Web images = 72 pixels per inch (ppi)

- Higher resolution → bigger file, slower loading
- No need for super high resolution for web

🎯 Easy Analogy

Think of images like **posters for a wall**:

- Choose the **right type of paper** → format
- Make it the **right size** for the wall → width & height
- Don't make it **too detailed for the wall** → resolution

✅ Takeaway:

Follow these rules so your website images **look sharp, fit perfectly, and load quickly**.

◆ Tools to Edit & Save Images

- Use **image editing software** to make sure images have the **right size, format, and resolution**.
- **Adobe Photoshop** → most poCr for professionals
- **Photoshop Elements** → cheaper version, good for beginners

✅ **Easy takeaway:** Use editing tools to **prepare images correctly** for your website.

HTML5 <figure> and <figcaption>

- <figure> → Think of this as a photo frame. You put your picture (or diagram, chart, etc.) inside it. It's a container for your image and its description.
- <figcaption> → This is the caption under the frame. It tells people what the picture is about.

Real-life analogy:

Imagine you have a photo album. Each photo has a little note under it explaining what's happening in the picture.

- The photo = <img>
- The photo frame = <figure>
- The note under the photo = <figcaption>

```html
<figure>
  <img
    src="https://assets3.thrillist.com/v1/image/2559819/1200x630/flatten;crop_down;webp=auto;jpeg_quality=70"
    alt="Two sea otters holding hands"
    width="200"
    height="100"
  />
  <figcaption>
    Sea otters hold hands while sleeping so they don't drift apart.
  </figcaption>
</figure>
```

Sea otters hold hands while sleeping so they don't drift apart.

✅ **Here's what's happening:**

- <figure> wraps the image and its caption.
- <figcaption> tells us about the image.
- You can have more than one image inside a <figure> if they share the same caption.

Summary IMAGES

- The <img> element is used to add images to a web page.
- You must always specify a src attribute to indicate the source of an image and an alt attribute to describe the content of an image.
- You should save images at the size you will be using them on the web page and in the appropriate format.
- Photographs are best saved as JPEGs; illustrations or logos that use flat colors are better saved as GIFs.

Chapter 05

Lists

HTML Lists

Lists are used when you want to show a **group of items** in a neat order.

HTML has **three types of lists**:

1. Ordered List (**<ol>**)

- **What it is:** A list where items are **numbered automatically** by the browser.
- **HTML tag:** → means **Ordered List**
- **List items:** Each item goes inside <li> tags → **List Item**
- **When to use:** Steps in a recipe, instructions, or anything where **order matters**.
- **Analogy:** Think of **steps in a dance tutorial** – Step 1 → Step 2 → Step 3.

```html
<ol>
  <li>Wake up</li>
  <li>Brush teeth</li>
  <li>Have breakfast</li>
</ol>
```

1. Wake up
2. Brush teeth
3. Have breakfast

Easy Real-Life Analogy

Think of it like **steps in a recipe or instruction manual**.

- Step 1: Mix ingredients
- Step 2: Bake
- Step 3: Serve

Order matters, so you **number the steps** → just like <ol> does automatically.

2. Unordered List (<ul>)

- **What it is:** A list where items are **shown with bullet points** instead of numbers.
- **HTML tag:** → stands for **Unordered List**
- **List items:** Each item is inside <li> tags → **List Item**
- **When to use:** Shopping lists, to-do lists, or anything where **order doesn't matter**.
- **Analogy:** Think of a **grocery list** – milk, eggs, bread – the order doesn't matter.
- You can **change bullet style** (circle, square, diamond) using the type attribute or better with CSS (list-style-type).

html Copy code

```html
<ul>
  <li>Milk</li>
  <li>Eggs</li>
  <li>Bread</li>
</ul>
```

- Milk
- Eggs
- Bread

Another Example:

html Copy code

```html
<ul type="circle">
  <li>1kg King Edward potatoes</li>
  <li>100ml milk</li>
  <li>50g salted butter</li>
  <li>Freshly grated nutmeg</li>
  <li>Salt and pepper to taste</li>
</ul>
```

- 1kg King Edward potatoes
- 100ml milk
- 50g salted butter
- Freshly grated nutmeg
- Salt and pepper to taste

Easy Real-Life Analogy

Think of it like a **shopping list**.

- You don't care about order, you just need **all items listed**:
  - Milk
  - Eggs
  - Bread
  - Butter

The bullets show each item clearly → just like <ul> does.

Definition Lists (<dl>)

- What it is: A special kind of list used to show terms and their definitions.
- HTML tag: <dl> → stands for Definition List
- Inside <dl>:
  - <dt> → Definition Term → the word or term you want to explain
  - <dd> → Definition Description → the meaning or explanation of that term

Key Points:

1. <dl> wraps the whole list.
2. Each term goes in <dt> and each definition goes in <dd>.
3. You can have:
   - One term → one definition
   - One term → multiple definitions
   - Multiple terms → same definition

Why it's useful:

It's perfect for glossaries, vocab lists, FAQs, or explaining concepts on a webpage.

html Copy code

```html
<dl>
  <dt>HTML</dt>
  <dd>HyperText Markup Language, used to create web pages.</dd>

  <dt>CSS</dt>
  <dd>Cascading Style Sheets, used to style web pages.</dd>

  <dt>JavaScript</dt>
  <dd>A programming language used to make web pages interactive.</dd>
</dl>
```

HTML
HyperText Markup Language, used to create web pages.
CSS
Cascading Style Sheets, used to style web pages.
JavaScript
A programming language used to make web pages interactive.

Real-Life Analogy

Think of it like a **mini dictionary**:

- **Term:** Apple
- **Definition:** A round fruit that is usually red, green, or yellow.

Or like a **flashcard for students**:

- Front of card = term (<dt>), back of card = explanation (<dd>).

It's perfect for learning because each word is **directly linked to its meaning**, just like in a dictionary or glossary.

Nested Lists

- **What it is:** A **list inside another list**.
- **Why use it:** When you want to **group items under a main item**.
- **How it works:**

- Put a <ul> or <ol> **inside an** .
- Browsers automatically **indent the sublist** and may change bullet styles for <ul>.

html Copy code

```html
<ul>
  <li>Fruits
    <ul>
      <li>Apple</li>
      <li>Banana</li>
      <li>Mango</li>
    </ul>
  </li>
  <li>Vegetables
    <ul>
      <li>Carrot</li>
      <li>Spinach</li>
      <li>Broccoli</li>
    </ul>
  </li>
</ul>
```

- Fruits
  - Apple
  - Banana
  - Mango
- Vegetables
  - Carrot
  - Spinach
  - Broccoli

1. Fruits
   1. Mango
   2. Orange
2. Vegetables
   1. Cabbage
   2. Capsicum
      1. Green Capsicum
      2. Yellow Capsicum
      3. Red Capsicum

Real-Life Analogy

Think of it like a **folder on your computer**:

- Main folder = (e.g., “Fruits”)
- Subfolder inside it = (e.g., “Apple, Banana, Mango”)

Or like a **menu in a restaurant**:

- Main item = “Drinks”
- Sub-items = “Tea, Coffee, Juice”

Nested lists help you **organize things clearly** under categories.

Summary: Lists in HTML

1. There are 3 types of lists in HTML:
    - Ordered list (<ol>) → uses numbers
    - Unordered list (<ul>) → uses bullets
    - Definition list (<dl>) → shows terms with their meanings
2. You can also put one list inside another → this is called a nested list.

✅ **One-Line Memory Trick for Students:**

- Numbers =
- Bullets =
- Dictionary = <dl>