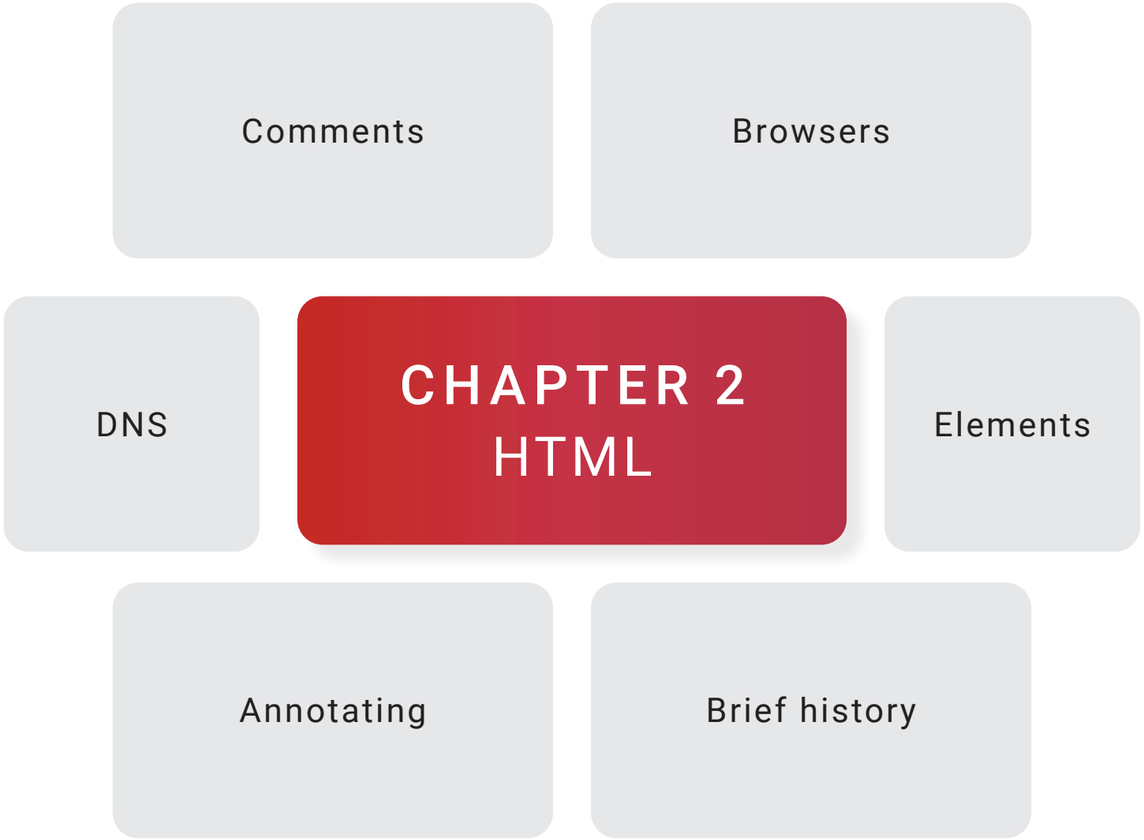
HTML + CSS Visualized

A visual Approach to Learning Web Design

Romaine Raffington

TL;DR Summary : Websites

- A website is a collection of interconnected web pages
- Websites are typically built using HTML, CSS, and JavaScript
- There are different types of websites such as blogs, and ecommerce sites
- Websites can be static, dynamic or a combination of both
- A website's files and content are stored on web servers
- A domain name is the unique address of a website
- A browser is an application used to access and navigate the web
- Browsers read files from a web server to render websites
- The Domain Name System (DNS) translates human-readable addresses into numerical Internet Protocol (IP) addresses
- HTTPS is an encrypted communication protocol used to secure data
- A local website is hosted and tested on local hardware



PREVIEW ONLY: This book is currently under development. The final content, design, and layout may differ from this draft.

A laptop showcasing HTML code

Hypertext markup language

Hypertext Markup Language (HTML) is a standardized markup system used to structure and create content for the web. HTML is used to build websites, design emails, develop mobile apps, and much more.

It structures and defines content using elements. We use HTML elements to create paragraphs, images, links, and other content within an annotated document that digital environments can understand and interpret.

With an HTML file, we tell browsers and other digital environments what our content is and how it should be treated.

Hyper Text Markup Language

H T M L

HyperText

Text that contains a link. Hypertext plays a vital role in HTML.

Markup

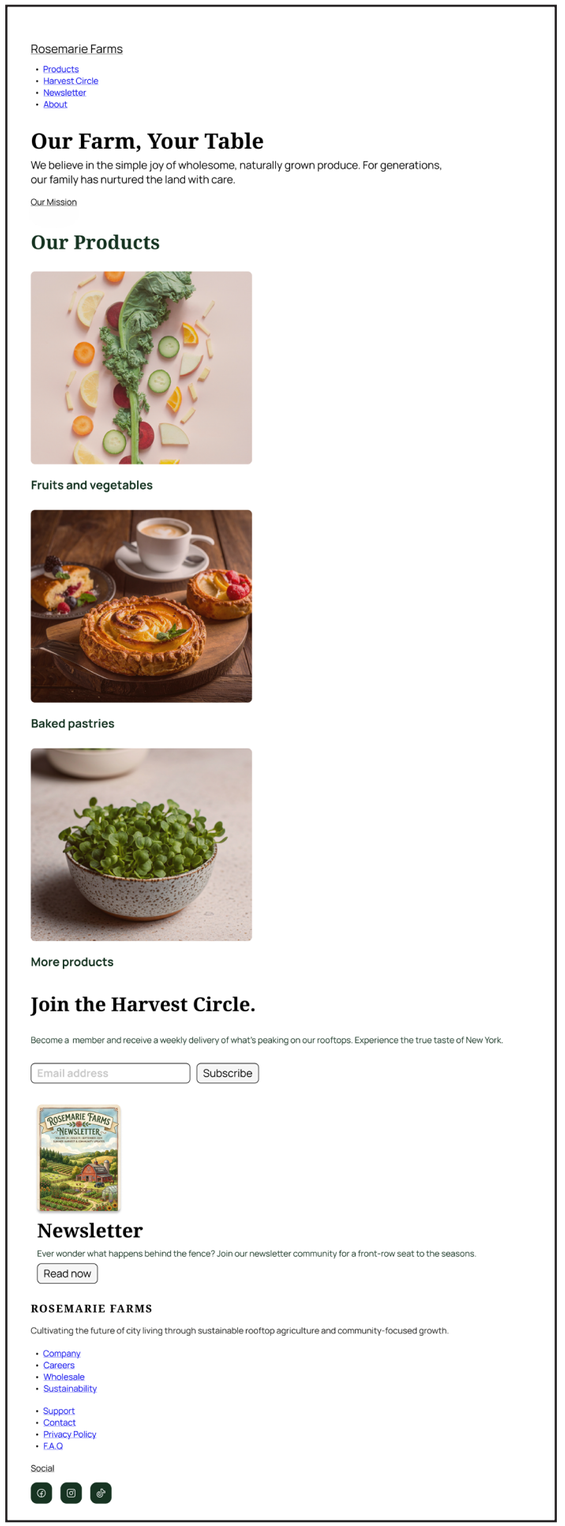
We mark up content using special tags, known as elements, to indicate their purpose.

Language

HTML is a defined system with a standardized set of rules.

Web framing

HTML provides the "scaffolding" of the web, acting as the base upon which all visual styles and interactive elements are built.



Rosemarie Farms

- Products
- Harvest Circle
- Newsletter
- About

Our Farm, Your Table

We believe in the simple joy of wholesome, naturally grown produce. For generations, our family has nurtured the land with care.

Our Mission

Our Products

Fruits and vegetables

Baked pastries

More products

Join the Harvest Circle.

Become a member and receive a weekly delivery of what's peaking on our rooftops. Experience the true taste of New York.

Email address Subscribe

Newsletter

Ever wonder what happens behind the fence? Join our newsletter community for a front-row seat to the seasons.

Read now

ROSEMARIE FARMS

Cultivating the future of city living through sustainable rooftop agriculture and community-focused growth.

- Company
- Careers
- Wholesale
- Sustainability

- Support
- Contact
- Privacy Policy
- F.A.Q

Social

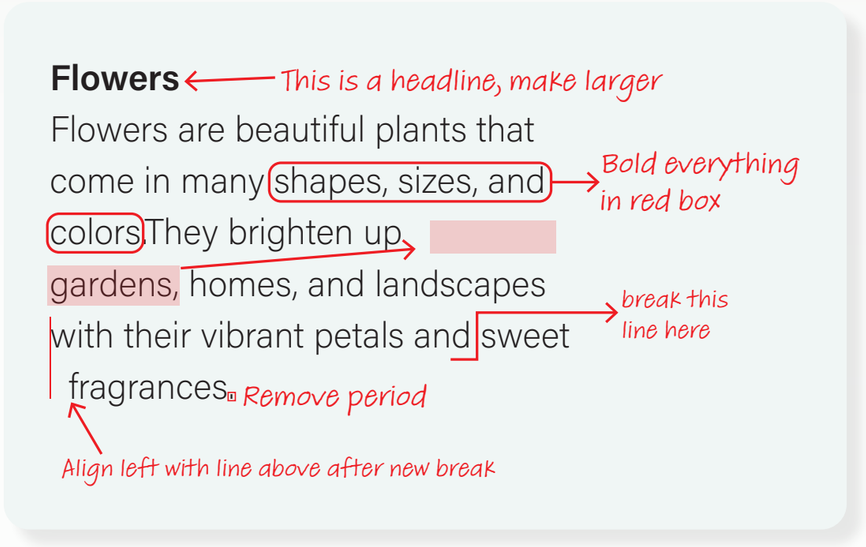
A website built using only HTML.

PREVIEW ONLY: This book is currently under development. The final content, design, and layout may differ from this draft.

Annotating content with HTML

Annotating involves adding notes, comments, or explanations to content to provide additional context or to emphasize key information.

HTML serves as a system for annotating web content, enabling browsers and other digital environments to interpret and render content as intended.



An annotated paragraph

Annotated instructions

HTML markup is used to add context to content, much like a teacher would when reviewing a student's work.

However, while teacher annotations are typically proposed changes, HTML markup provides instructions that ensure content is correctly recognized and rendered as intended.

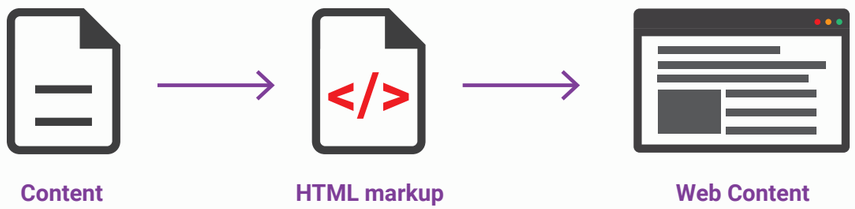
For a browser to display information correctly, it needs clear instructions on what that material is and how to treat it. Web content is structured using HTML elements (defined by tags) to ensure proper rendering.

```HTML
<p>This is HTML code, content is being annotated.</p>
```

Enclosing the text within a paragraph element instructs the browser to interpret that content as a paragraph. Without the <p> tag, the browser lacks the necessary instructions to render the content correctly.

Proper HTML structure requires that all content be contained within elements.

Overview of the HTML process



HTML tags transform content into a format compatible with the internet.

Building block for the web

Much like using styles in a word processor to establish document hierarchy, we use HTML elements to structure web content logically.

With elements, we are

- Defining structure, not appearance
- Adding semantic meaning to content

We have provided examples from a word processor below, along with the corresponding HTML code required to create the same structure for the web.

New York City

The City of Dreams and Drive

New York City is a global phenomenon, a vibrant tapestry woven from countless cultures, towering ambition, and relentless energy.

Text editor

```
1. <h1>New York City</h1>
2. <h2>The City of Dreams and Drive</h2>
3. <p>New York City is a global phenomenon, a vibrant
4. tapestry woven from countless cultures, towering
5. ambition, and relentless energy.</p>
```

HTML

Like a box of chocolate

Take a look at the box of chocolates to the right. Can you tell which one is **Midnight Bloom** and which is **Salted Caramel Kiss**?

We can take a guess, but it is impossible to know for sure without actually taking a bite out of one. Because the chocolates are not labeled, identifying them becomes a challenge.

The chocolate is the content, and we want content to be 'known and shown' in the best light. For the web, this is where HTML comes into play.

Using HTML is like adding labels and descriptions to the chocolates. This way, browsers interacting with our box of chocolates know exactly what is inside and can make decisions based on users' preferences.

PREVIEW ONLY: This book is currently under development. The final content, design, and layout may differ from this draft.

More chocolate

Let's take a look at the chocolates again, but this time with **labels and descriptions** to tell us what we are looking at.

Midnight Bloom
A dark chocolate ganache infused with lavender and a hint of honey, enrobed in rich 85% dark chocolate.

Zesty Sunset
A bright and tangy passion fruit caramel swirled into creamy milk chocolate, encased in a vibrant chocolate shell.

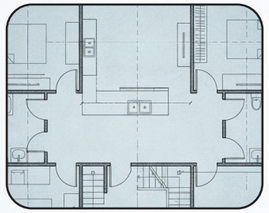
Midnight Bloom
Light and airy pistachio marzipan covered in creamy white chocolate, offering a delicate nutty sweetness.

Salted Caramel Kiss
A classic buttery, flowing salted caramel center, dipped in milk chocolate.

Mix Chocolate Cloud
A heavenly vanilla bean white chocolate ganache, incredibly smooth and aromatic, enrobed in a shell of creamy milk chocolate.

Now, picking between a Salted Caramel Kiss and a Midnight Bloom is a lot easier. We can identify each chocolate and feel confident in our choice.

HTML is like...



The blueprint for a home

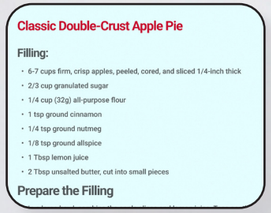
It defines the rooms and structure of the home, but not the furniture or the style.

A Label on a moving box

It describes the contents inside of the box.

Raw material

It provides the materials required to build the finished product.



A recipe

It provides the ingredients and the recipe, but it does not define how the final dish is plated or styled.

PREVIEW ONLY: This book is currently under development. The final content, design, and layout may differ from this draft.

Writing and editing code

HTML is authored using specialized tools. When writing code, coders typically choose between a lightweight code editor and a full-featured Integrated Development Environment (IDE).

Code editors

Code editors are software applications built for writing and editing source code. They function like enhanced text editors, providing features specifically designed to help us write code more efficiently.

Features of code editors

Fast & lightweight

Code editors are small apps that are quick to load and use.

Syntax highlighting

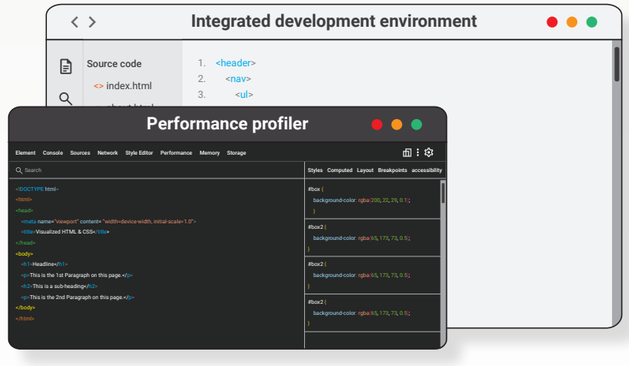
Color is used to visually distinguish parts of the code.

Auto complete

Code editors provide predictive text and suggestions.

Integrated development environment (IDE)

An IDE is a comprehensive software suite that bundles all the essential tools a developer needs to write, compile, debug, and deploy software into a single integrated application.



Features of IDEs

Compilers & Interpreters

Directly translate code into executable programs for platforms.

Refactoring Tools

Tools used to adjust variables, extract methods, and restructure code.

Advanced debugger

Step through code, inspect variables, set breakpoints, and identify issues in real-time.

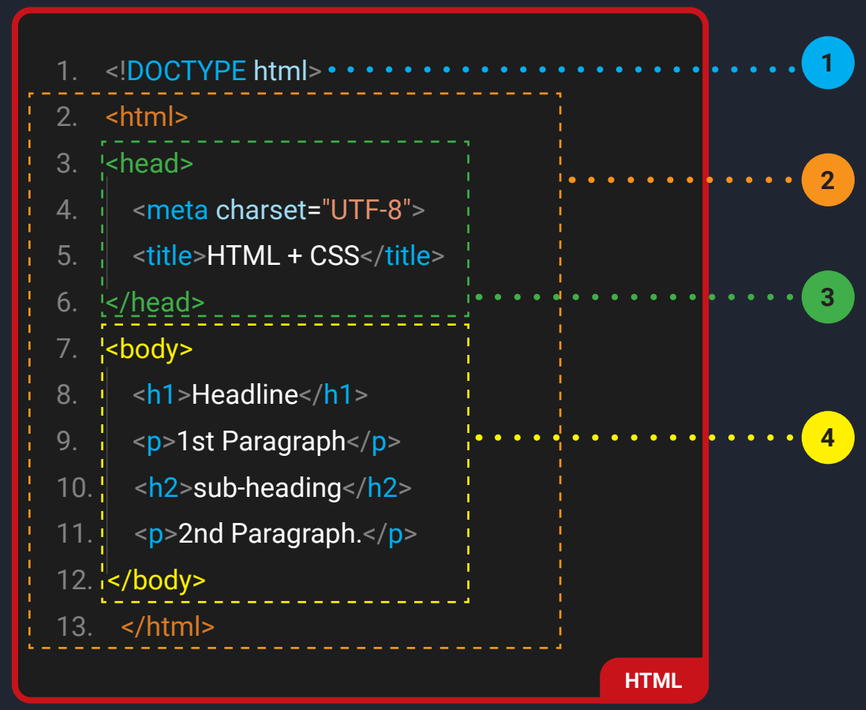
An analogy

A code editor is like a kitchen knife: it's sharp, lightweight, and perfect for chopping ingredients.

An IDE is like a fully stocked kitchen: you still have your knives, but you also have an oven, measuring tools, timers, and appliances to help you cook the entire meal.

HTML document structure

An HTML file, also known as a document, follows a specific structure that includes essential parts such as the DOCTYPE, <html>, <head>, and <body>. These components form the fundamental building blocks of a website.



1 DOCTYPE

The Document Type Declaration (DOCTYPE) is a declaration at the top of every HTML file that tells the browser which version of HTML is being used.

2 html

The <html> element is the root level of the document. All other elements must be nested between the opening <html> and closing </html> tags.

Any content placed outside of these tags is excluded from the document structure and may not be rendered correctly by the browser.

3 head

The <head> element holds the metadata for the document. This information is not visible to the user, but it is essential for browsers, search engines, and other web services to process and display the page correctly.

4 body

The body tag defines the content of the page that is visible to the user. It contains all the elements that will be displayed in the browser, such as text, images, links, and more.

PREVIEW ONLY: This book is currently under development. The final content, design, and layout may differ from this draft.

Brief history of HTML

HTML has evolved significantly since its inception in 1993; it has grown into a powerful tool capable of creating rich, interactive experiences.

HTML 1, 1993

A collection of elements that introduced the basic system for structuring and linking documents.

HTML 2, 1995

Added more elements and attribute to the language. Standardizing features like forms, frames and tables.

HTML 3.2, 1997

While HTML 3.0 never became an official standard, HTML 3.2 introduced scripting, styling, enhanced tables, and new form elements.

HTML 4.0.1, 1999

Version 4.0.1 refined HTML standards, prioritizing accessibility and improved CSS integration.

HTML 5, 2014

A landmark update that introduced new semantic elements and native multimedia support, marking the transition of HTML into a Living Standard.

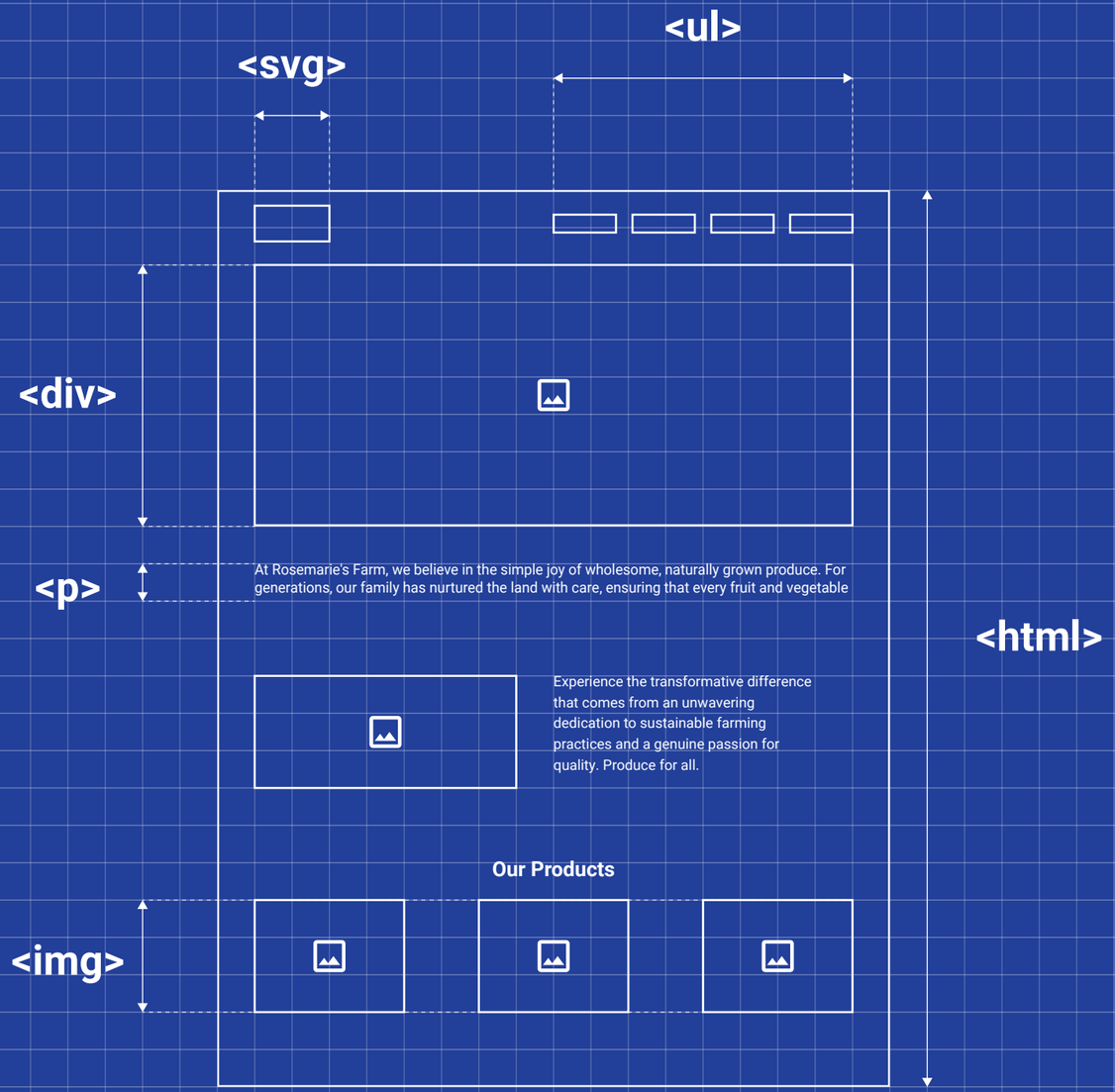
A living standard

Historically, HTML evolved through major, numbered leaps, where each release bundled years of new features into one large version.

However, as the web's pace accelerated, this slow release cycle could no longer keep up. To ensure the language evolves as fast as the internet itself, HTML transitioned into a "Living Standard.

Instead of waiting years for a massive "HTML6," the specification is now updated continuously, allowing for constant innovation and real-time improvements. The last major numbered release was HTML5.

PREVIEW ONLY: This book is currently under development. The final content, design, and layout may differ from this draft.



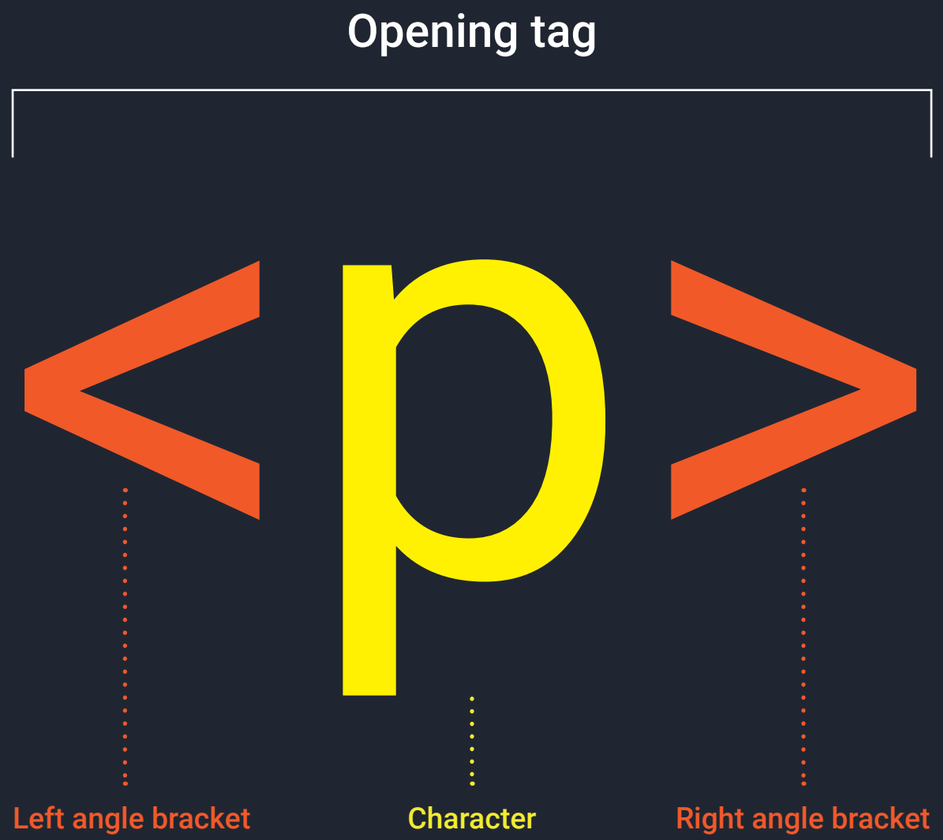
Elements

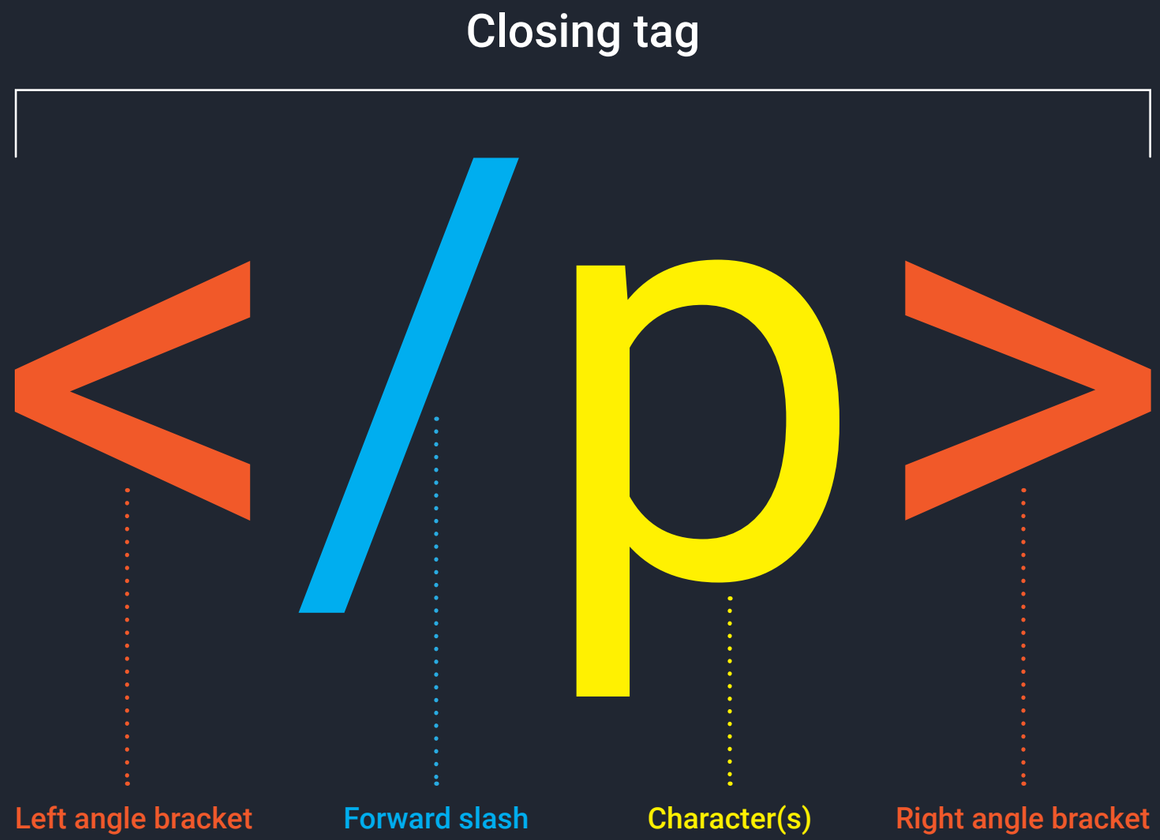
Elements are the fundamental building blocks of web documents. By wrapping content within tags, we define its purpose and determine how it should behave and display within the document structure.

Overview of elements

- Act as containers for content
- Consist of an opening tag, the content, and a closing tag
- Often referred to as "tags"
- Contain attributes that extend their functionality

HTML offers a diverse range of elements to handle virtually any type of content imaginable, including images, videos, audio, and code snippets, providing a standardized way to structure the modern web.





Opening tag

This highlights the start of an element. The opening tag begins with a left angle bracket and ends with a right angle bracket.

Closing tag

The closing tag indicates the end of an element. A closing tag starts with a left angle bracket followed by a slash, and ends with a right angle bracket.

Brackets

Angle brackets are used to indicate the start and end of an element.

Forward slash

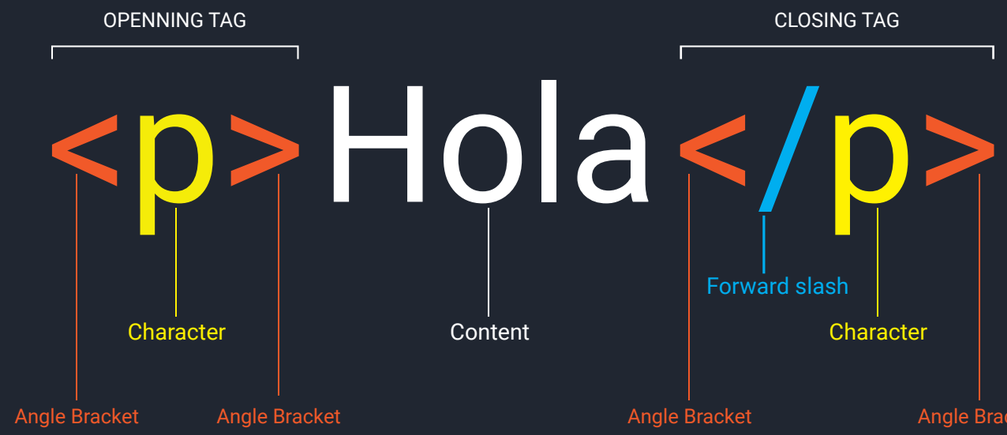
A forward slash is used within the closing tag to distinguish it from the opening tag.

Characters

Characters are enclosed within angle brackets to define the element.

PREVIEW ONLY: This book is currently under development. The final content, design, and layout may differ from this draft.

Element anatomy (*contin...*)



Example of a paragraph element with content.

Content

The content of an element is placed between its opening and closing tags; the content can be text, other elements, or a combination of both.

Attributes

Elements have attributes that are used for enhancing their functionality, accessibility, and interactivity in a document. Attributes can provide additional information about elements, control their behavior, and enable the integration of other web technology such as CSS.

Attributes are always included in the opening tag of an element.

Attribute syntax

```
<element attribute="value">Content</element>
```

```html
<h1 id="header">Welcome!</h1>
<p id="intro">Hola, Hello! Amigo</p>
<a href="https://www.google.com">google</a>
<img src="image.png" alt="Image of an apple">
<p class="greeting">Hello! Friend</p>
<p class="greeting">Hola! Amigo</p>
```

HTML

Attributes examples above (id, href, src)

Attributes types:

Global

Attributes can be used on any element.

Specific

These attributes are specific to particular types of elements.

Data

Custom attributes for storing information that can be accessed later.

Event

Code snippet that runs in response to events that occurs.

Global attributes

Global attributes are universally available for use on all elements. These attributes can influence behavior, styling, and interactivity.

Common global attributes

class

Assigns a group identifier to elements, allowing multiple elements to share style and functionality.

id

Assigns a unique identifier to an element, that cannot be shared with other elements.

title

Displays a short description in a tool-tip when the user hovers over the element.

accesskey

Allows the assignment of a keyboard shortcut to an element.

hidden

Determines whether an element should be shown or concealed.

style

Allow us to embed CSS styles within HTML document.

spellcheck

Determines whether a browser's built-in spell checker should review an element's content for spelling errors.

data-*

Store custom data within an element.

```html
<h1 id="header" class="greet">Welcome!</h1>
<p id="intro">Hola, Hello! Amigo</p>
<a id="lnkOut" href="https://www.google.com">google</a>
<img id="HDimage" src="image.png" alt="Image of an apple">
<div title="A greeting">Hello</div>
<p class="greet"><span id="glow">Hello!</span> Friend</p>
```

HTML

Elements with global attributes.

As shown above, elements often use several global attributes at once. A standard case is an element defined by both an *id* and a *class*.

PREVIEW ONLY: This book is currently under development. The final content, design, and layout may differ from this draft.

Nesting

Nesting is the process of embedding one element within another to create a structural hierarchy for content, similar to placing boxes inside larger boxes. Nesting elements is a common practice when creating websites.

When elements are nested, a parent-child relationship is formed

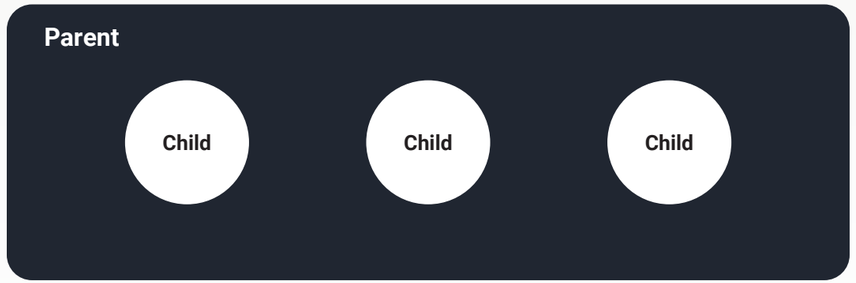


Fig. A

Fig. A illustrates the concept of nesting with white circles placed inside a larger, dark rectangular shape. The dark rectangle becomes the **parent** element, and the white circles are its **children**.

Below, we example of nesting elements in code.

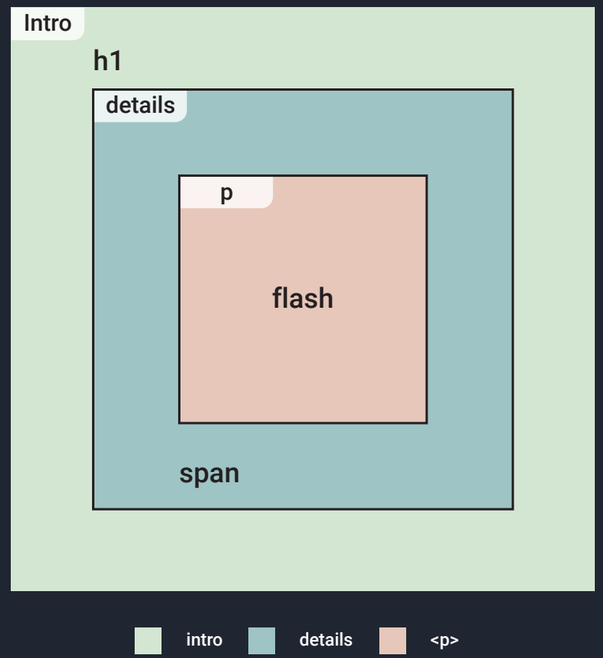
```html
<div id="intro"> <!-- start of intro -->
    <h1>Rosemarie's Berries</h1>
    <div id="details"> <!-- start of detail -->
        <p>The <span id="flash">best</span> berries in NYC.</p>
        <span>$19.99 per container</span>
    </div> <!-- end of detail -->
</div> <!-- end of intro -->
```

HTML

Fig. B

In Fig. B, we have elements nested inside a with the ID of "intro". Nested within "intro", we have an <h1> and a with the ID of "details".

Nested inside "details", we have a <p> and a element. The <p> element has a nested element with the ID of "flash".



Visual representation of Fig. B

Each color area represents a parent element. We have elements nested within other elements. An element is identified by its unique ID if it has one; otherwise, it's identified by its element type.

PREVIEW ONLY: This book is currently under development. The final content, design, and layout may differ from this draft.

Nesting (*contin...*)

```
1. <head>
2.     <meta charset="UTF-8">
3.     <title>Rosemarie Farm</title>
4.     <link rel="stylesheet" href="styles.css">
5. </head>
```

Fig. A

Root element

As the root element, <html> is the only element not nested within a parent.

It serves as the ultimate parent for all other elements in the document.

Siblings

Elements that share the same parent.

In *Fig. A*, <meta>, <title> and <link> are siblings.

Adjacent siblings

Elements that follow one another directly in the source code.

<meta>, <title> are adjacent siblings.

<title> and <link> are adjacent siblings.

Siblings

```
1. <h1 id="legal">Our Farm, Your Table</h1>
2. <p>We believe in the simple joy of wholesome, naturally grown produce.
3. For generations, our family has nurtured the land with care.</p>
4. <a href="mission.html">Our Mission</a>
```

Nested children

```
1. <section id="home">
2.     <div class="hero-content">
3.         <h1 id="legal">Our Farm, Your Table</h1>
4.         <p>We believe in the simple joy of wholesome, naturally
5.         grown produce. For generations, our family has nurtured the land with
6.         care.</p>
7.         <a href="mission.html">Our Mission</a>
8.     </div>
9. <section>
```

This book and many code editors use indentation and vertical lines to visually link opening and closing tags, making nested structures easier to navigate.

HTML comments

A comment is a section of code that is ignored by the browser and is not displayed on the rendered page. Comments are a way to add notes, explanations, or temporarily disable code within an HTML document.

Single line comment

```
<!-- This is a comment -->
<h1 id="header">Welcome!</h1>
<p id="greeting">Hola, Hello! Amigo</p>
<!-- This is also a comment  -->
```

Multi-line comment

```
<!--
<h1 id="header">Welcome!</h1>
<p id="greeting">Hola, Hello! Amigo</p>
-->
```

Code is disabled by being commented out.

Key features

Multi-line support

Comments can span multiply lines.

Hidden on web pages

Comments are visible in the source code but not on the web page.

Excellent for debugging

By enclosing code in a comment, we can disable that code.

Anatomy of a comment

Brackets

Brackets are used to indicate the beginning and ending of a comment.

Exclamation

The exclamation point at the beginning of an opening tag distinguishes it from a closing tag.

Hyphens

Both opening and closing tag includes double hyphens.

Content

The content of a comment is placed between the tags.

Example: HTML comments

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>HTML & CSS Visualized </title>
</head>
<body>
    <!-- Main Navigation section -->
    <nav>
      <ul id="nav-items">
            <li><a href="index.html">Home</a></li>
            <li><a href="about.html">About</a></li>
            <li><a href="book.html">Get Book</a></li>
            <li><a href="resources.html">Resources</a></li>
      </ul>
    </nav>

    <!-- Main content below -->
    <main>
      <h1> Rosemarie's Farm</h1>
      <h2>The freshest fruits and veggies.</h2>

      <!-- Confirm this section
      <h2>The best fruits in New York City</h2>
      <p>Find out why we have the best fruits in the city</p>
      -->
    </main>
```



```
    <!-- Articles section -->
    <article>
      <p> Whether you're visiting for a weekend harvest, picking up a basket
      of seasonal favorites, or simply enjoying the atmosphere, Rosemarie
      Farm offers an authentic farm-to-table experience that locals and
      visitors alike call the best in NYC </p>
    </article>

    <!-- footer content -->
    <!-- Expand footer in future updates, more content coming -->
    <footer>
      <ul id="footer-items">
            <li><a href="story.html">Our story</a></li>
            <li><a href="products.html">Products</a></li>
            <li><a href="locations.html">Locations</a></li>
            <li><a href="resources.html">Resources</a></li> <!-- remove -->
      </ul>
    </footer>
</body>
</html>
```

Void elements

Elements that cannot have a closing tag, content, or nested elements.

They consist solely of an opening tag.

Similar to other elements, void elements can have attributes that are included in their opening tag.

```html
<hr>
<br>
<area>
<img src="image.png">
```

Examples of void elements

Some commonly used void elements are

- <area>
- <base>
-
- <embed>
- <hr>
- <img>
- <input>
- <link>
- <meta>
- <source>

Self-closing tags

XHTML, an earlier version of HTML, was designed with stricter and more consistent syntax rules.

One key requirement was that **all elements must be properly closed**, leading to the introduction of self-closing tags.

```html
<hr />
<br />
<area />
<img src="image.png" />
```

XHTML self-closing tags

Self-closing and void elements.

In HTML5, forward slashes are ignored when present inside an opening tag.

Since the HTML parser ignores the forward slashes, elements using self-closing syntax are treated as void elements.

Adding a slash to void tags is optional and purely aesthetic in HTML5."

<area>

Void element (HTML)

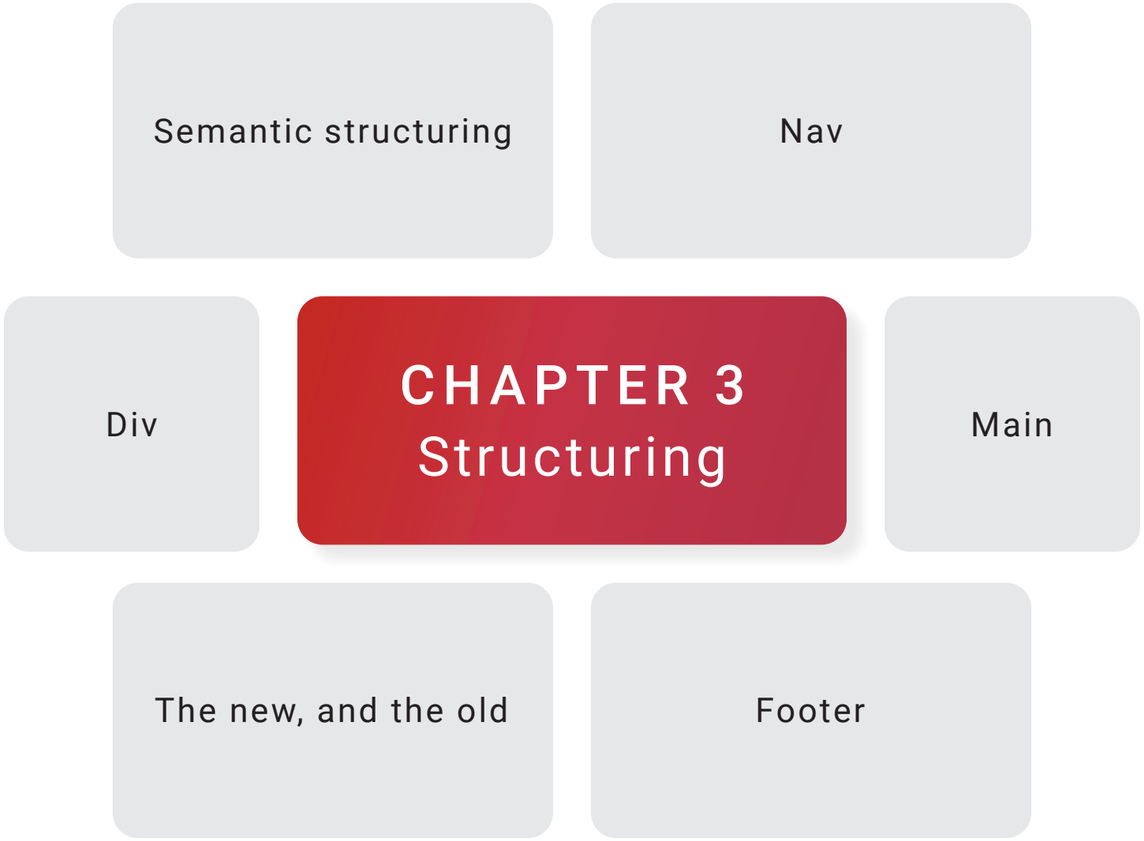
<area />

Self closing tag (XHTML)

PREVIEW ONLY: This book is currently under development. The final content, design, and layout may differ from this draft.

TL;DR Summary : HTML

- Hypertext Markup Language (HTML) is a standardized markup system
- Use HTML to create websites, emails, mobile apps, and more
- HTML is used to annotate content for the web
- Elements are used to define and structure web documents
- We use code editors and IDEs to write code
- The DOCTYPE declaration, and the <html>, <head>, and <body> tags are essential to HTML structure
- The first standardized version of HTML was released in 1993
- Any computer can be used as a server
- Elements act as containers for content
- Elements are composed of an opening tag, content, and a closing tag
- Elements can have attributes that expand their functionality
- Elements can be nested in other elements to create advanced structures
- We can add comments to code to provide context and reminders
- Void elements cannot have a closing tag, content, or nested elements
- Self-closing tags are treated as void elements



PREVIEW ONLY: This book is currently under development. The final content, design, and layout may differ from this draft.