

HTML

S22_CA-CS-804-A

Web Application Development

[Robert W. Sebesta, “Programming the World Wide Web”]
[W3 Web Schools]

Anatomy of (Basic) Website

Your content + HTML + CSS = Your website
 structure presentation

A website is a way to present your content to the world, using HTML and CSS to present that content and make them look good

HTML: HyperText Markup Language

- Language for describing **structure** of a document
- An HTML file is a text file containing small **markup** tags (or elements)
- The markup tags tell the web browser **how** to display the page
- An HTML file denotes **hierarchy** of elements
- An HTML file can be created using a simple text editor, HTML editor, or IDE

HTML History

- Late 1980s: Tim Berners-Lee created first HTML version
- 1995: HTML 2.0 Published as standard with RFC 1866
- 1997: HTML 4.0 Standardized most modern HTML element with W3C recommendation
 - Encouraged use of CSS for styling elements over HTML attributes
- 2000: XHTML 1.0
 - Imposed stricter rules on HTML format
 - E.g., elements needed closing tag, attribute names in lowercase
- 2014: HTML5 published as W3C recommendation
 - New features for capturing more **semantic** information and **declarative** description of behavior
 - E.g., input constraints, new tags that explain **purpose** of content
 - Important changes to DOM

[<https://en.wikipedia.org/wiki/HTML>]

HTML Elements

tagname

<p lang="en-us" > This is a paragraph in English </p>

attribute name value content to be formatted

Start a paragraph element

Opening tag begins an HTML element.

Opening tags must have a corresponding closing tag.

Set the language to English

HTML attributes are name/value pairs that provide additional information about the contents of an element.

End a paragraph element

Closing tag ends and HTML element. All content between the tags and the tags themselves compromise an HTML element.

Each tag has a “start tag,” “end tag,” and some content in between, and optional attributes

HTML Elements

```
<input type="text" name="myUserName" />
```



Begin and end
input element

Some HTML tags can be
self closing, including a
built-in closing tag

Content Types

Type	Description	Example
Metadata	Content hosted in the head of an HTML document. Doesn't appear in the web page but is used to describe a webpage and its relationships to other external resources	<code><meta name=</code>
Flow	Text and all elements that can appear as content in the body of an HTML document	<code><body> <h1>Heading</h1> <p>Some content...</p> </body></code>
Sectioning	Used to structure the content of a web page and to help with layout	<code><section class= Some content ... </section></code>

Content Types (2)

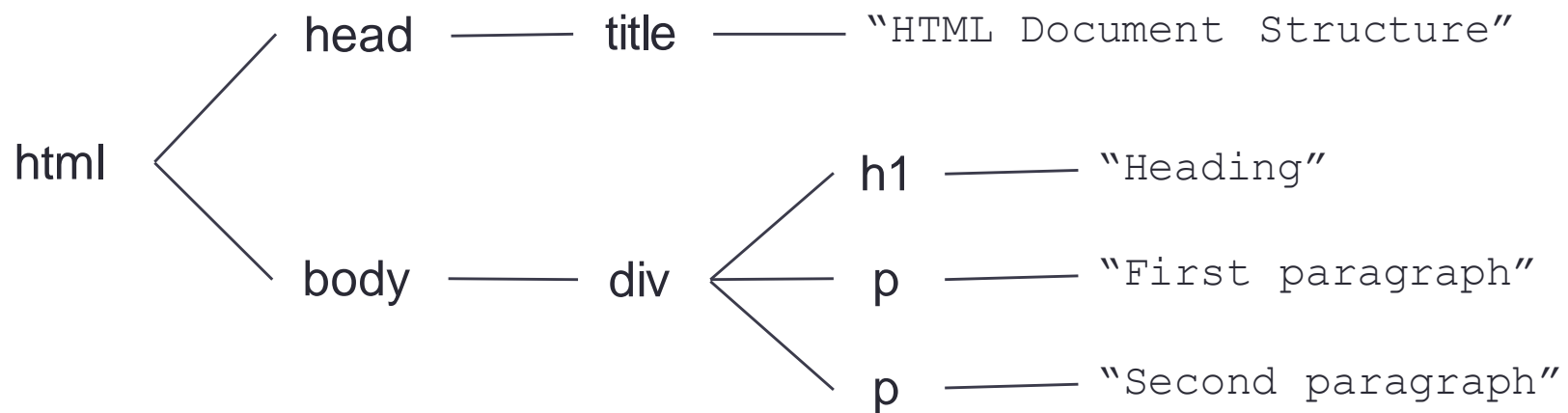
Type	Description	Example
Phrasing	Elements for marking up content within a paragraph element such as text and typography	<pre><p> Emphasized text and some normal text </p></pre>
Heading	Elements used to define the headings of a section of an HTML document. The elements <code>h1-6</code> represent headings with <code>h1</code> having the highest ranking	<pre><h1>Main heading</h1> <h2>Sub-heading</h2></pre>
Embedded	Embedded content includes media, such as video, audio, and images	<pre></pre>
Interactive	Elements that a user can interact with such as media elements with controls, form inputs, buttons, and links	<pre><input type="text" name="username" required /> <input type="password" name="pwd" required /></pre>

A Starter HTML Document

<p>Use HTML5 standard mode</p>	<pre><!DOCTYPE html> <html> <head> <meta charset="UTF-8"> <title>Insert title here</title> <!-- this is a comment --> <!-- anything in the head section is not rendered on the screen --> </head> <body> <!-- anything in the body section is rendered on the screen --> <p lang="en-us">This is a paragraph in English</p> Visit W3Schools! </body> </html></pre>	<p>HTML content</p>
<p>Interpret bytes as UTF-8 characters Includes both ASCII and international characters</p>		<p>Header Information about the page</p>
	<p>Title Used by browser for title bar or tab</p>	
	<p>Document content</p>	

HTML Document Structure

```
<html>
  <head>
    <title>HTML Document Structure</title>
  </head>
  <body>
    <div>
      <h1>Heading</h1>
      <p>First paragraph</p>
      <p>Second paragraph</p>
    </div>
  </body>
</html>
```



Text

```

1 <!doctype html>
2 <html>
3 <head></head>
4 <body>
5
6 <h1>Level 1 Heading</h1>
7 <h2>Level 2 Heading</h2>
8 <h3>Level 3 Heading</h3>
9 <h4>Level 4 Heading</h4>
10 <h5>Level 5 Heading</h5>
11 <h6>Level 6 Heading</h6>
12
13 Text can be made bold with <b>
14 or <sup>super</sup>
15 White space collapses to a single space
16 two more spaces
17 the markup to use is <br>
18 organization.
19 Space can be added with <br />
20 <br />
21 <br />New line
22
23 <p>A paragraph of text
24 that form a paragraph.
25 By default, text wraps
26 on a new line.
27 </p>
28
29 <hr />
30 Text can also be centered with <div style="text-align: center;">
31
32 </body>
33 </html>
34

```

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

Text can be made **bold** and *<i>italic</i>*, or ^{^{super}} and _{_{sub}} scripts. White space collapsing removes all sequences of two more spaces and line breaks, allowing the markup to use tabs and whitespace for organization.

Space can be added with ` `; ` `; ` `;

`
`New line can be added with `
`;

<p>A paragraph consists of one or more sentences that form a self-contained unit of discourse. By default, a browser will show each paragraph on a new line.

</p>

Text can also be offset with horizontal rules.

</body>

</html>

Level 1 Heading

Level 2 Heading

Level 3 Heading

Level 4 Heading

Level 5 Heading

Level 6 Heading

Text can be made **bold** and *italic*, or ^{super} and _{sub} scripts. White space collapsing removes all sequences of two more spaces and line breaks, allowing the markup to use tabs and whitespace for organization. Space can be added with

New line can be added with `< br />`

A paragraph consists of one or more sentences that form a self-contained unit of discourse. By default, a browser will show each paragraph on a new line.

Text can also be offset with horizontal rules.

Semantic Markup

- Tags that can be used to denote the **meaning** of specific content
- Examples:

<code></code>	An element that has importance
<code><blockquote></code>	An element that is a long quote
<code><q></code>	A short quote inline in paragraph
<code><abbr></code>	Abbreviation
<code><cite></code>	Reference to a work
<code><dfn></code>	The definition of a term
<code><address></code>	Contact information
<code><ins></code>	Content that is inserted
<code></code>	Content that is deleted

Links

```
1 <!doctype html>
2 <html>
3 <head>
4   <title>Example: links</title>
5 </head>
6 <body>
7   <a href="http://google.com">Absolute link</a>
8   <br />
9   <a href="schedule.html">Relative URL</a>
10  <br />
11  <a href="mailto:someone@example.com">Send email</a>
12  <br />
13  <a href="http://www.google.com" target="_blank">Open in new window</a>
14  <br />
15  <a href="#idName">Navigate to HTML anchor idName</a>
16 </body>
17 </html>
```

[Absolute link](#)

[Relative URL](#)

[Send email](mailto:someone@example.com)

[Open in new window](http://www.google.com)

[Navigate to HTML anchor idName](#)

Images, Audio, and Video

- HTML include standard support for , <audio>, and <video>
- Use an alt attribute to make images accessible

```

```

```
alt="thumb-up"
```

```
width="30"/>
```



- Common file formats
 - Images: .png, .gif, .jpg
 - Audio: .mp3
 - Video: .mp4

Video

- Important attributes for <video>

- src - location of video
- autoplay - tells browser to start play
- controls - show the default controls
- loop - loop the video
- muted - mutes the audio from the video



```
1 <!doctype html>
2 <html>
3 <head>
4   <title>Example: Video</title>
5 </head>
6 <body>
7   <video width="400" controls>
8     <source src="mov_bbb.mp4" type="video/mp4">
9     Your browser does not support HTML5 video.
10  </video>
11
12  <p>
13    Video courtesy of
14    <a href="https://www.bigbuckbunny.org/" target="_blank">
15      Big Buck Bunny</a>.
16  </p>
17 </body>
18 </html>
```

Tables

```
1 <!doctype html>
2 <html>
3 <head>
4   <title>Example: Table</title>
5 </head>
6 <body>
7   <table border="2" cellspacing="2" bgcolor="lightyellow" width="70%"
8     align="center">
9     <tr>
10       <th>&nbsp;</th>
11       <th>Monday</th>
12       <th>Tuesday</th>
13       <th>Thursday</th>
14     </tr>
15     <tr>
16       <th>1pm-2pm</th>
17       <td rowspan="2">Intro to Programming</td>
18       <td>Calculus</td>
19       <td>&nbsp;</td>
20     </tr>
21     <tr>
22       <th>2pm-3pm</th>
23       <td>&nbsp;</td> <!-- why &nbsp; here ? -->
24       <td>Physics</td>
25     </tr>
26   </table>
27 </body>
28 </html>
```

	Monday	Tuesday	Thursday
1pm-2pm	Intro to	Calculus	
2pm-3pm	Programming		Physics

rowspan

Forms

Send form data to
formHandler.php

Action attribute should be omitted if not using form to
submit data

```
<form action="formHandler.php" method="post">
  Username: <input type="text" name="username" value="" /> <br />
  Password: <input type="password" name="pwd" /> <br />
  <input type="submit" value="Submit" />
</form>
```

Username:

Password:

Transfer method

Method attribute specifies how
data is transmitted to server.
Method="get" sends data
appended to URL.
Method="post" sends data as an
HTML document

- Elements in a form are submitted to the server.
- A form may (or may not) have controls.

Controls

[illegible]

Text input:

Password input:

Search input:

Text area:

Initial text

Checkbox: ☒ Checked ☐ Unchecked

Drop down list box:

Option2

Multiple select Box:

Option1
Option2
Option3

File input box: No file selected.

Image button:

Button:

Range input:

Specialized Controls

- A **date** input appears differently depending on browser support

```
<input type="date" />
```

Firefox

Chrome

dd/mm/yyyy

April 2017

Mon	Tue	Wed	Thu	Fri	Sat	Sun
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Specialized Controls

- A **time** input appears differently depending on browser support

```
<input type="time" />
```

Firefox

A simple, empty rectangular text input field in Firefox.

Chrome

A time input control in Chrome showing the time 23:38, with a close button (X) and up/down arrow buttons.

- A number input can set restrictions on what numbers are accepted

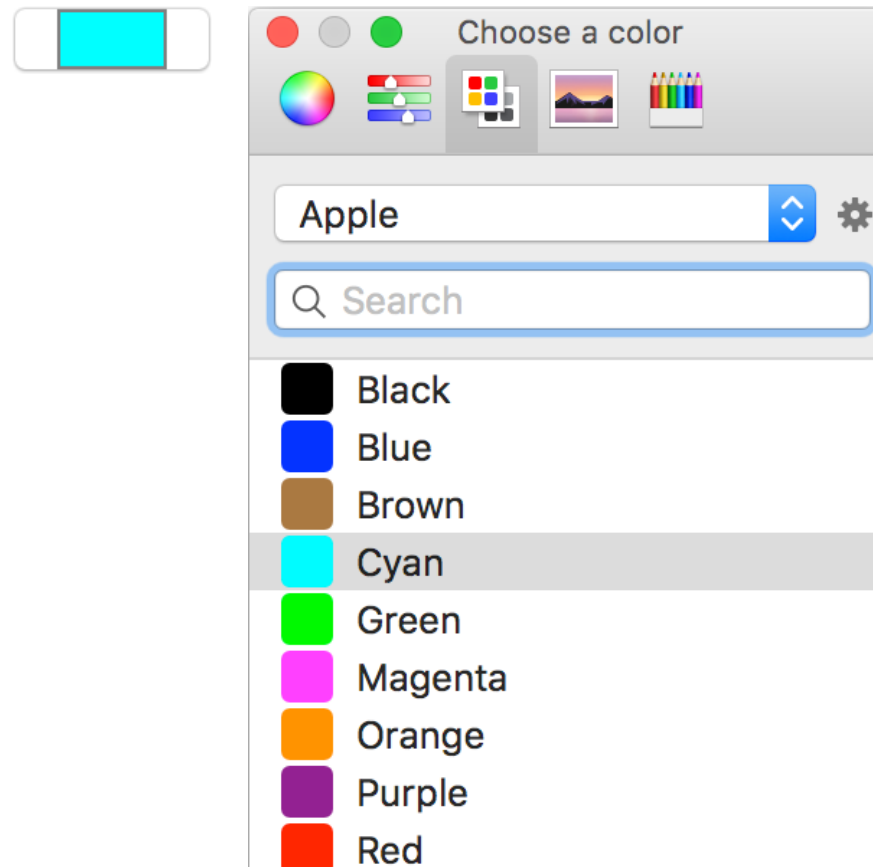
```
<input type="number" min="1" max="12" />
```

A number input control showing the value 1, with up and down arrow buttons on the right side.

Specialized Controls

- A **color** input is used for input fields that should contain a color

```
<input type="color" name="favcolor" value="#00ffff">
```



Labeling Inputs

- Suggestion can be placed **inside** input element

`<p>Input box: <input type="text" placeholder="Enter keyword" /></p>`

Input box:

- The suggestion disappears after user types

Input box:

- Label can be attached to an input

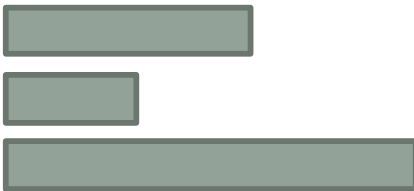
`<p><label>
 Label on input box: <input type="text" placeholder="Enter keyword" />
 </label>
</p>`

Label on input box:

Block vs. Inline Elements

Block elements

- Appear on a new line
- Example:
 - `<h1>`
 - `<p>`
 - ``
 - `<table>`
 - `<form>`
 - `<div>`



Inline elements

- Appear on the same line
- Example:
 - `<a>`
 - ``
 - ``
 - `<input>`
 - ``
 - ``



Validating Inputs

- Displays errors on invalid input **immediately**, making it easier to fix errors

- Check that input is a valid **email**

```
<label>Email: <input type="email" /></label>
```

Email:

- Check that input is a valid **URL**

```
<label>URL: <input type="url" /></label>
```

URL:

- Constrain input to be at most **max length**

```
<label>Enter a username up to 10 characters:  
  <input type="text" maxlength="10" />  
</label>
```

Enter a username up to 10 characters:

Validating Inputs

- Check that input matches **regex pattern**

```
<label>Would you like coffee or tea?  
  <input type="text" pattern="coffee|tea" />  
</label>
```

Would you like coffee or tea?

- **Prevent** all edits

```
<label>Readonly text:  
  <input type="text" readonly />  
</label>
```

Readonly text:

Grouping Elements

- Creates a **parent** or **container** element and a set of **child** elements
- Enables group to be styled together
- Can use any block or inline element or **generic** element
 - **<div>** :generic block element
 - **** :generic inline element

Grouping Elements

- Semantic layout elements are block elements that associated meaning with group (useful for CSS selectors)

Some popular semantic layout elements are
`<header>`,
`<footer>`,
`<nav>`,
`<article>`,
`<aside>`,
`<section>`,
`<figcaption>`

```
<!doctype html>
<html>
<head>
  <title>Example: Grouping Elements</title>
</head>
<body>

  <header>
    <h1>How to Get an A+</h1>
    <nav>...</nav>
  </header>
  <article>
    <section>
      <h3>Practice</h3>
      <p>When there are practice problems, ...</p>
    </section>
    <aside>
      <h4>Useful Links</h4>
      <a href="http://www.pythontutor.com/javascript.html">Javascript Tutor</a>
    </aside>
  </article>

</body>
</html>
```

HTML Style

- Tags
 - Use lowercase for names
 - Use indentation to reflect hierarchy
 - Always close tags
- Use attribute name="value" format for attributes
- Use blank lines to break up documents into closely connected regions
- Use comments to describe purpose of regions

HTML Best Practices

- Use specialized controls or input validation where applicable
- Always include elements of HTML starter document
- Use label or placeholder for labeling controls
- Use alt to make images accessible

HTML Layout - 1

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Template</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
    box-sizing: border-box;
}

body {
    font-family: Arial, Helvetica, sans-serif;
}

/* Style the header */
header {
    background-color: #666;
    padding: 30px;
    text-align: center;
    font-size: 35px;
    color: white;
}

/* Create two columns/boxes that floats next to each other */
nav {
    float: left;
    width: 30%;
    height: 300px; /* only for demonstration, should be removed */
    background: #ccc;
    padding: 20px;
}
```

HTML Layout - 2

```
/* Style the list inside the menu */
nav ul {
    list-style-type: none;
    padding: 0;
}

article {
    float: left;
    padding: 20px;
    width: 70%;
    background-color: #f1f1f1;
    height: 300px; /* only for demonstration, should be removed */
}

/* Clear floats after the columns */
section::after {
    content: "";
    display: table;
    clear: both;
}

/* Style the footer */
footer {
    background-color: #777;
    padding: 10px;
    text-align: center;
    color: white;
}
```

HTML Layout - 3

```
/* Responsive layout - makes the two columns/boxes stack on top of each other
instead of next to each other, on small screens */
```

```
@media (max-width: 600px) {
  nav, article {
    width: 100%;
    height: auto;
  }
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>CSS Layout Float</h2>
```

```
<p>Resize the browser window to see the responsive effect (you will learn more
about this in our next chapter - HTML Responsive.)</p>
```

```
<header>
```

```
  <h2>Cities</h2>
```

```
</header>
```

```
<section>
```

```
  <nav>
```

```
    <ul>
```

```
      <li><a href="#">London</a></li>
```

```
      <li><a href="#">Paris</a></li>
```

```
      <li><a href="#">Tokyo</a></li>
```

```
    </ul>
```

```
  </nav>
```


HTML Layout - 4

```
<article>
  <h1>London</h1>
  <p>London is the capital city of England. It is the most populous city in the
United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
  <p>Standing on the River Thames, London has been a major settlement for two
millennia, its history going back to its founding by the Romans, who named it
Londinium.</p>
</article>
</section>

<footer>
  <p>Footer</p>
</footer>

</body>
</html>
```

HTML Layout - 5

Cities

[London](#)
[Paris](#)
[Tokyo](#)

London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

Footer

HTML responsive



responsive.txt

The Walk

Transport

History

Gallery

The Walk

The walk from Monterosso to Riomaggiore will take you approximately two hours, give or take an hour depending on the weather conditions and your physical shape.



What?

Cinque Terre comprises five villages: Monterosso, Vernazza, Corniglia, Manarola, and Riomaggiore.

Where?

On the northwest coast of the Italian Riviera, north of the city La Spezia.

Price?

The Walk is free!