HTML: HyperText Markup Language

CPIT-405 Web Applications

Khalid Alharbi, PhD

Last updated: Spring 24

Table of Contents

Introduction

- HTML (HyperText Markup Language) is the most basic building block of the web.
- HTML is a markup language, which means it uses tags `< >` to tell the browser how to display the content of a web page.
- The purpose of HTML is to describe the structure of a web page.
  - HTML defines headings, paragraphs, lists, images, links and more elements that make up a web page
- Standardized by World Wide Web Consortium (W3C)
- W3C handed HTML over to a group of browser vendors known as Web Hypertext Application Technology Working Group (WHATWG).
  - WHATWG is made up of the four major browser vendors: Apple, Google, Microsoft, and Mozilla.
- The complete HTML specification is maintained by WHATWG at https://html.spec.whatwg.org.

History of HTML (I)

- HTML was invented by Tim Berners-Lee in late 1991 at CERN, the European Organization for Nuclear Research.
- HTML version timeline:
  - 1991: The first public description of HTML is released.
  - 1993: HTML 1.0 is released.
  - 1994: The World Wide Web Consortium (W3C) was founded and led by Tim Berners-Lee as the the main international standards organization for the World Wide Web.
  - 1995: HTML 2.0 is released.
  - 1997: HTML 3.0 is released.
  - 1998: W3C had shifted priorities away from HTML and instead began working on an XML-based equivalent, called XHTML.
    - XHTML strictly enforce XML rules. If there was a syntax error, the page wouldn't load properly.
  - 1999: HTML 4.0 is released.

History of HTML (II)

- HTML version timeline (Cont.):
  - 2000: XHTML 1.0 is released as a reformulation of HTML as an XML application.
  - 2004: WHATWG was founded by individuals of Apple, the Mozilla Foundation, and Opera Software.
  - 2008: HTML 5.0 first draft appeared with better support for video, audio, and mobile devices.
  - 2012: W3C and WHATWG announced that WHATWG would maintain a living standard, while the W3C would continue with versions.
  - 2014: HTML5 became a W3C recommendation.
  - 2019: both WHATWG and W3C signed an agreement to collaborate on a single version of HTML going forward: "Living Standard".
- The W3C and WHATWG had different goals. WHATWG wanted to continue working on a Living Standard for HTML while W3C wanted to publish a finished implementation of HTML.
- WHATWG had set up a solid process in which HTML features get early support by browsers before the details of a specification are fully worked out.

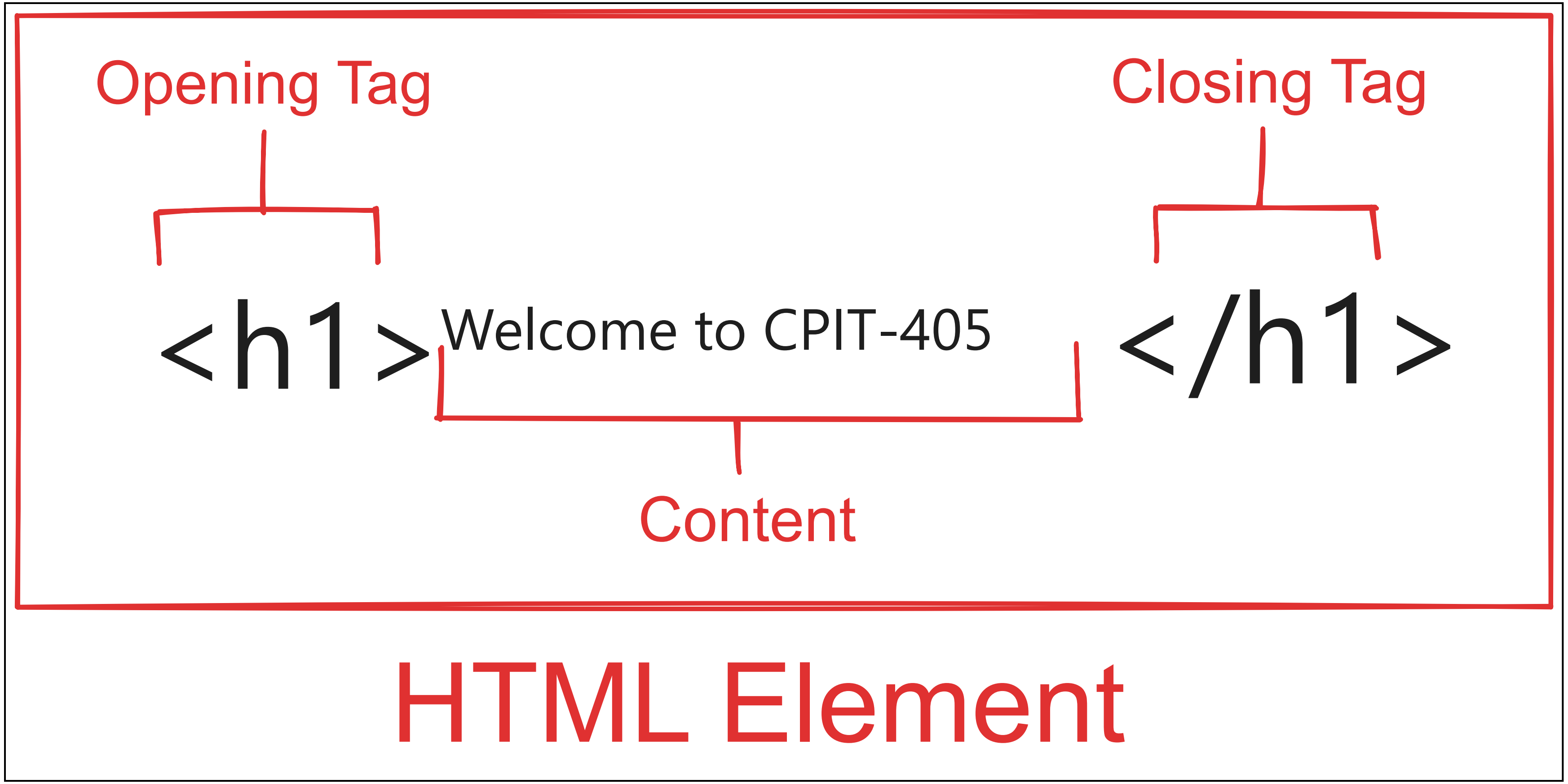
History of HTML (III)

- Interestingly, WHATWG existed due to the reason why W3C lost power over the HTML specs.
- In 2004, W3C abandoned organizational efforts on HTML in favor of the the Semantic Web, also known as Web 3.0.
  - Their efforts shifted to towards new standards that promote common data formats and exchange protocols on the Web such as XHMTL 2, RDF, OWL, etc.
- WHATWG was formed in reaction to that, rewriting HTML completely from its W3C HTML 4.0 version to make it better for web 2.0 applications.
- Now, that both groups came to an agreement, there will be no more versions of HTML.
- Although HTML is a mature language, HTML specs are constantly being proposed, discussed, updated and revised in a draft called "the living standard".
- W3C still controls a number of other important web technologies, including CSS, WebAssembly, and web performance.

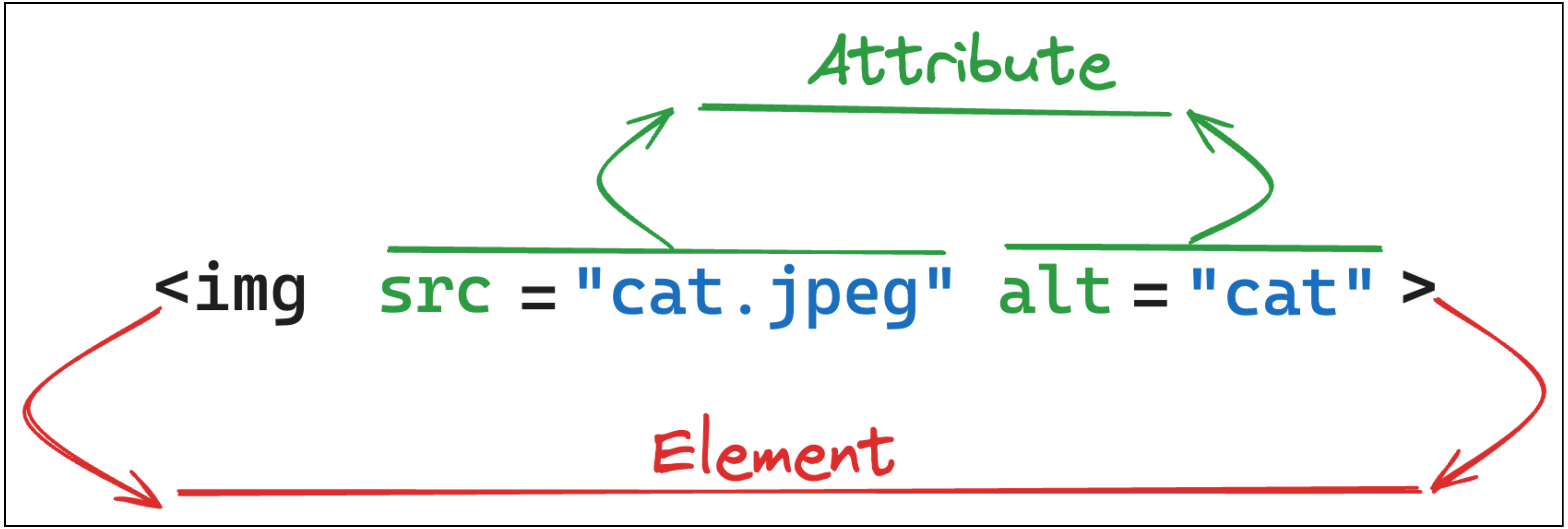
Concepts and Syntax

- The elements of HTML
- The `<!DOCTYPE`> element
- The `<head>` element
- The `<body>` element

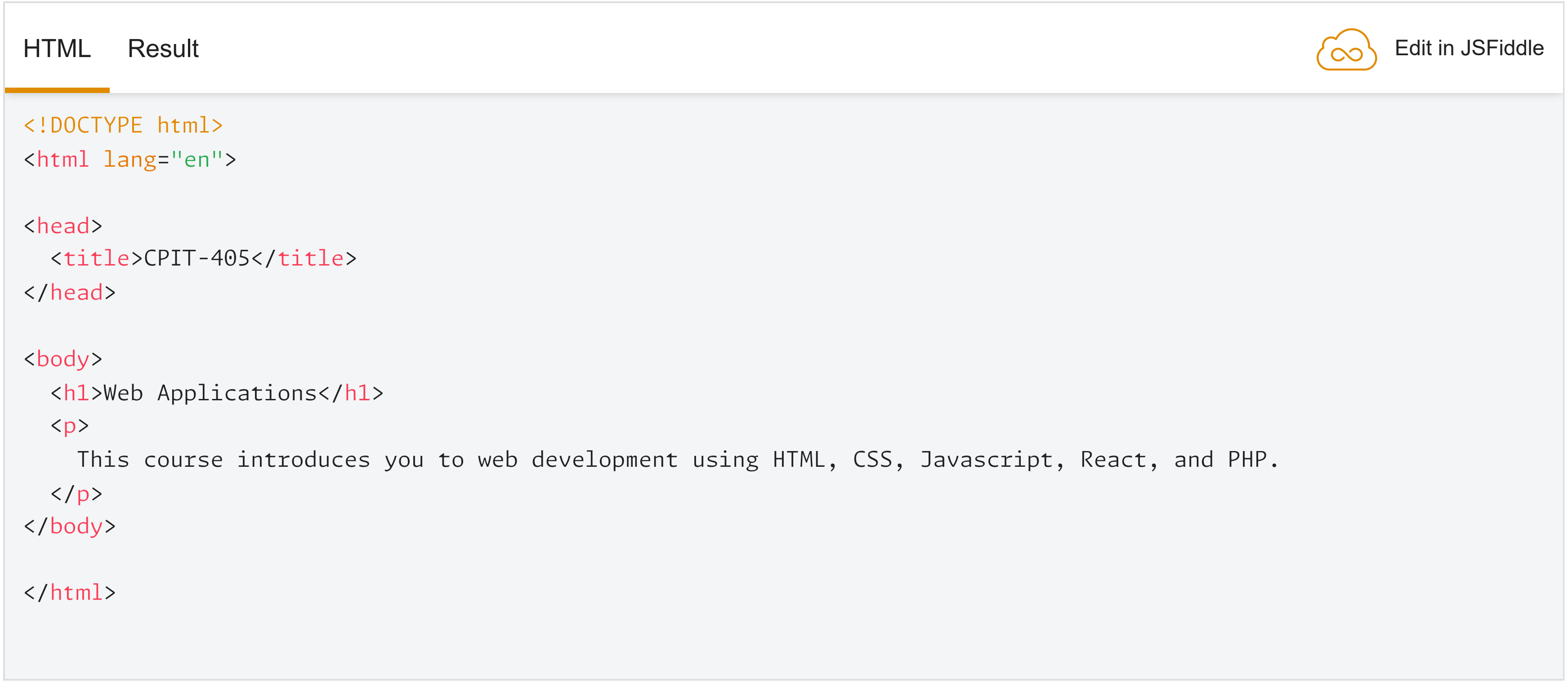
HTML Element



HTML Element's Attributes



HTML Document



```html
<!DOCTYPE html>
<html lang="en">

<head>
  <title>CPIT-405</title>
</head>

<body>
  <h1>Web Applications</h1>
  <p>
    This course introduces you to web development using HTML, CSS, Javascript, React, and PHP.
  </p>
</body>

</html>
```

- Save this as an HTML file (e.g., `index.html`) and open it up in your browser.

The `!DOCTYPE` element

```
<!DOCTYPE html>
```

- In HTML the doctype is a required preamble at the top of HTML documents to prevent the browser from switching into the so-called "quirks mode" when rendering a document.
- In the old days of the web and before W3C HTML standards, HTML documents were typically written in two versions for two major browsers: Microsoft Internet Explorer and Netscape Navigator.
- Layout engines in web browsers support three modes: quirks mode, limited-quirks mode, and no-quirks mode.
  - In **quirks mode**, layout engines emulate behavior in Navigator 4 and Internet Explorer 5.
  - In **limited-quirks mode**, there are only a very small number of quirks implemented.
  - In **no-quirks mode**, layout engines emulate the behavior described by the modern HTML and CSS specifications.
    - The `<!DOCTYPE html>` tag is used to indicate that the HTML document uses the no-quirks mode.

The `<head>` element

- The `<head>` element contains elements that won't get displayed in the web browser.
- It contains the `<meta>` element for metadata (data about the HTML document).
  - Examples include character encoding, description, and viewport size.
- It also contains other elements such as:
  - `<title>` for the document's title.
  - `<link>` for links to CSS and custom favicons.
  - `<script>` for both inserting javascript code or linking to external javascript files.

```html
<head>
    <meta charset="utf-8" />
    <meta name="author" content="Khalid Alharbi" />
    <meta name="description" content="This is CPIT 405 course website.
          This course introduces students to web development using HTML, CSS,
          Javascript, React, and PHP." />
    <link rel="icon"  type="image/vnd.microsoft.icon"  href="https://example.com/image.ico">
    <title>CPIT 405 - Learning HTML</title>
</head>
```

The `<body>` element

- The `<body>` element represents the content of an HTML document.
- There can be only one `<body>` element in a document.
- It has a number of attributes that can be used to control the appearance and behavior of the web page (e.g., `bgcolor` to set the document's background color).
  - However, it is best to avoid using style attributes in HTML and to use CSS instead.
  - Refer to the complete list of valid attributes on MDN

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document title</title>
</head>
<body>
  <p>
    The body contains all of the content
    that is displayed in the browser window.
  </p>
</body>
</html>
```

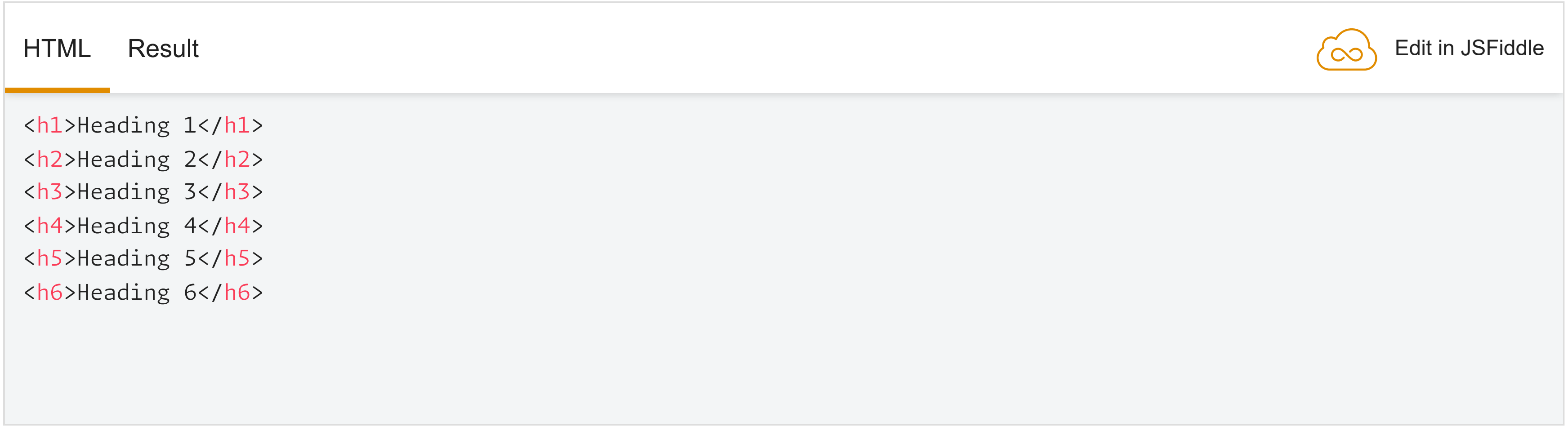
HTML elements by function

HTML elements can be grouped by function or purpose into:

- Text content: Elements that structure the content of the web page.
  - Examples: `<h1>` through `<h6>`, and `<p>`.
- Inline text elements: elements that define the meaning, structure, or style of a word or line.
  - Examples: `<a>`, `<span>`, `<strong>`, and `<em>`
- Image and multimedia: elements that define multimedia resources such as images, audio, and video.
  - Examples: `<img>`, `<video>`, `<audio>`
- List elements: elements that define list of items
  - Examples: `<ul>`, `<ol>`, and `<dl>`.
- Sectioning elements: elements that define sections of content.
  - Examples: `<header>`, `<main>`, `<nav>`, `<footer>`, `<section>`, `<article>`, and `<aside>`.
- Form elements: elements that can be used to create input forms.
  - Examples: `<form>`, `<label>`, `<input>`, `<select>`, `<textarea>`, and `<button>`.

HTML Headings `<h1>` through `<h6>`

- Heading content defines the heading of a section
- The `<h1>` to `<h6>` HTML elements represent six levels of section headings.
  - `<h1>` is the highest section level and `<h6>` is the lowest.



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

Paragraphs `<p>`, `<sup>`, `<sub>`, `<strong>`, and `<em>`

- The `<p>` element represents a paragraph.
- The `<sup>` element represents a superscript while the `<sub>` element represents a subscript.
- The `<strong>` element is used to add strong emphasis to text. It is typically displayed in bold.
  - Screen readers will pronounce the words in `<strong>` with added stress.
- The `<em>` element marks text that has stress emphasis. It is typically displayed in italic.
  - Screen readers will pronounce the words in `<em>` with an emphasis, using verbal stress.
- The `<strong>` and `<em>` tags should not be used to apply styling. CSS should be used instead.

`<a>` The anchor element for links

- The `<a>` element with its `href` attribute creates a hyperlink to a URL.
- Links are not restricted to web pages and can include email addresses, files, phone number, JavaScript, and anything else with a URL.
- The `target` attribute is used to indicate where to open the linked URL.
  - Examples: `target="_blank"` will open the link in a new tab.

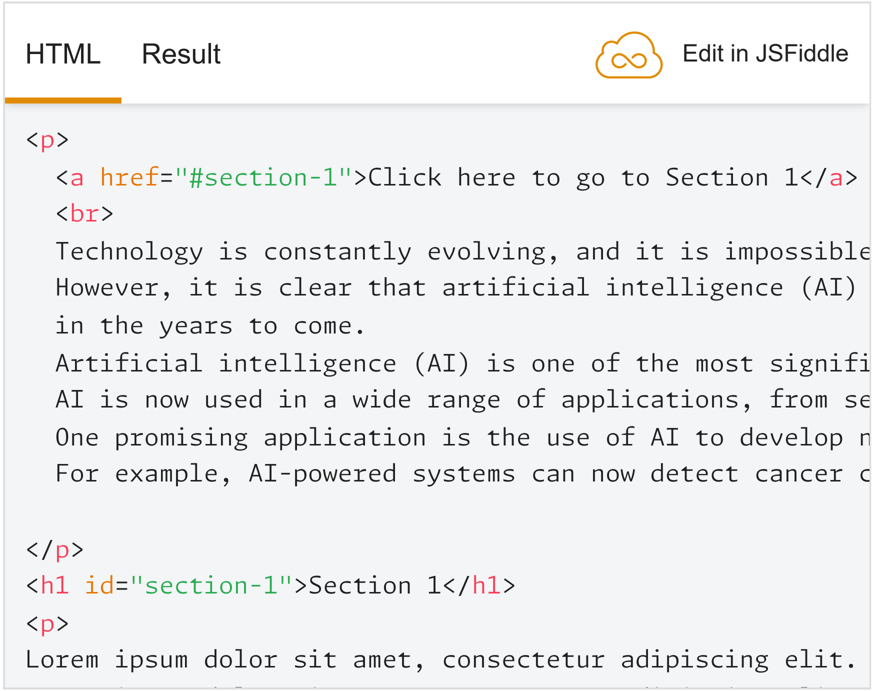
```html
<a href="example.com" target="_blank">Go to example.com</a>

<a href="example.com">Go to example.com</a>

<a href="mailto:nowhere@example.com">Send email to nowhere</a>

<a href="tel:+9668008501520">Call FedEx Express now!</a>

<a sms="sms:1411">To check your balance, text 1 to 1411</a>

<a href="./file.pdf">Download PDF file</a>

<a href="javascript:alert('An alert message!');">Show me an alert popup in JavaScript</a>
```

`<a>` The anchor element and internal links

- The anchor element, `<a>`, can be used to link to an element in the same webpage.
- This is often used to create links in a table of contents.
- Use the following steps to link to another element within the same HTML document:
  1. Add an `id` attribute to the HTML element that you want to link to (e.g., `<h2>`, `<h3>`, etc.).
  2. Add an anchor `<a>` element with an `href` attribute that points to the header element with the corresponding id attribute.

```html
<a href="#section-1">
<h1 id="section-1">Section 1</h1>
```

Demo

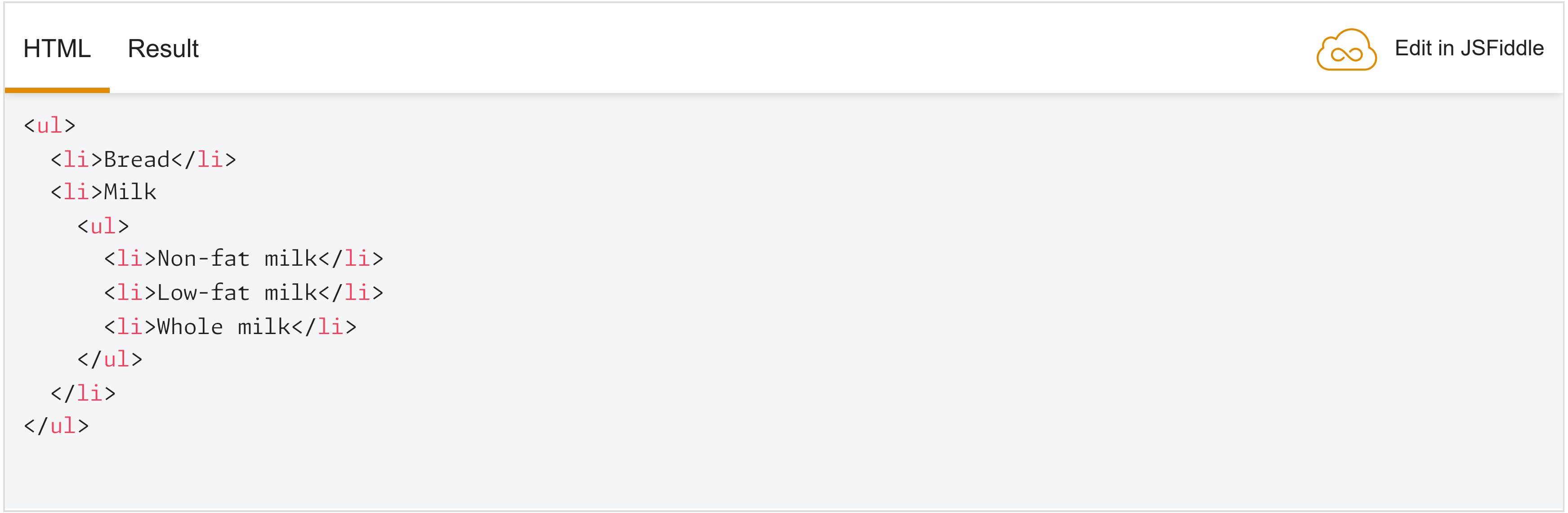


```html
<p>
  <a href="#section-1">Click here to go to Section 1</a>
  <br>
  Technology is constantly evolving, and it is impossible
  However, it is clear that artificial intelligence (AI)
  in the years to come.
  Artificial intelligence (AI) is one of the most signifi
  AI is now used in a wide range of applications, from se
  One promising application is the use of AI to develop n
  For example, AI-powered systems can now detect cancer c

</p>
<h1 id="section-1">Section 1</h1>
<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
```

List elements: `<ul>` The unordered list element

- The `<ul>` element represents an unordered list of items, typically rendered as a bulleted list.
- It may contain a list item (`<li>` element) or another list (e.g., `<ul>` or `<ol>`) nested as deeply as desired.

Demo



```html
<ul>
  <li>Bread</li>
  <li>Milk
    <ul>
      <li>Non-fat milk</li>
      <li>Low-fat milk</li>
      <li>Whole milk</li>
    </ul>
  </li>
</ul>
```

List elements: `<ol>` The ordered list element

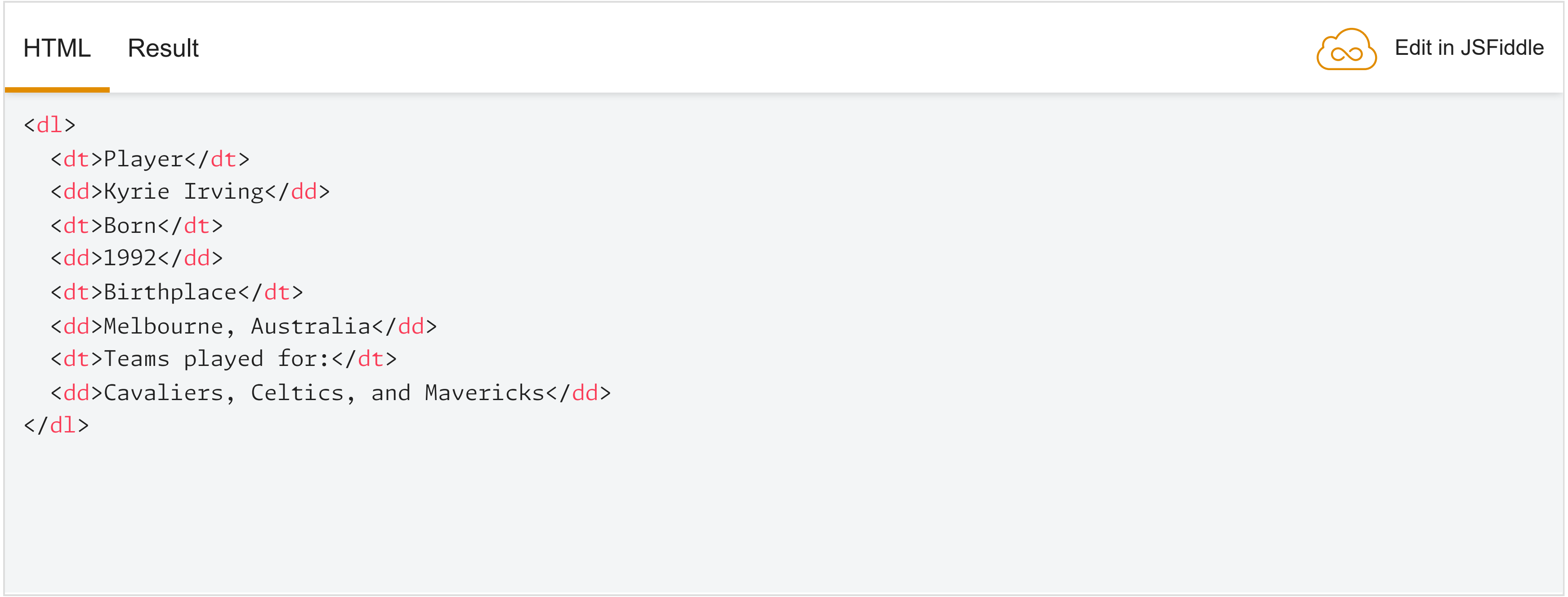
- The `<ol>` element represents an ordered list of items, typically rendered as a numbered list.
- It may contain a list item (`<li>` element) or another list (e.g., `<ul>` or `<ol>`) nested as deeply as desired.

Demo

List elements: `<dl>` The description list element

- The `<dl>` element represents a description list of items, typically rendered as a numbered list.
- It contains a list of terms (using the `<dt>` element) and descriptions (using the `<dd>` element).

Demo



```html
<dl>
  <dt>Player</dt>
  <dd>Kyrie Irving</dd>
  <dt>Born</dt>
  <dd>1992</dd>
  <dt>Birthplace</dt>
  <dd>Melbourne, Australia</dd>
  <dt>Teams played for:</dt>
  <dd>Cavaliers, Celtics, and Mavericks</dd>
</dl>
```

Multimedia elements: `<img>`, `<video>`, and `<audio>`

- Multimedia on the web includes images (`<img>`), sound/music (`<audio>`), videos/movies (`<video>`).

Demo

Table element `<table>` (I)

- The `<table>` element represents tabular data comprised of rows (`<tr>` elements), columns of headers (`<th>` elements), and cells containing data (`<td>` elements).
- It may contain a caption (`<caption>`) element that specifies the title of the table.

Demo

Table element `<table>` (II): `colspan` and `rowspan`

- The `colspan` attribute specifies the number of *columns* that a cell should span.
- The `rowspan` attribute specifies the number of *rows* that a cell should span.

Demo

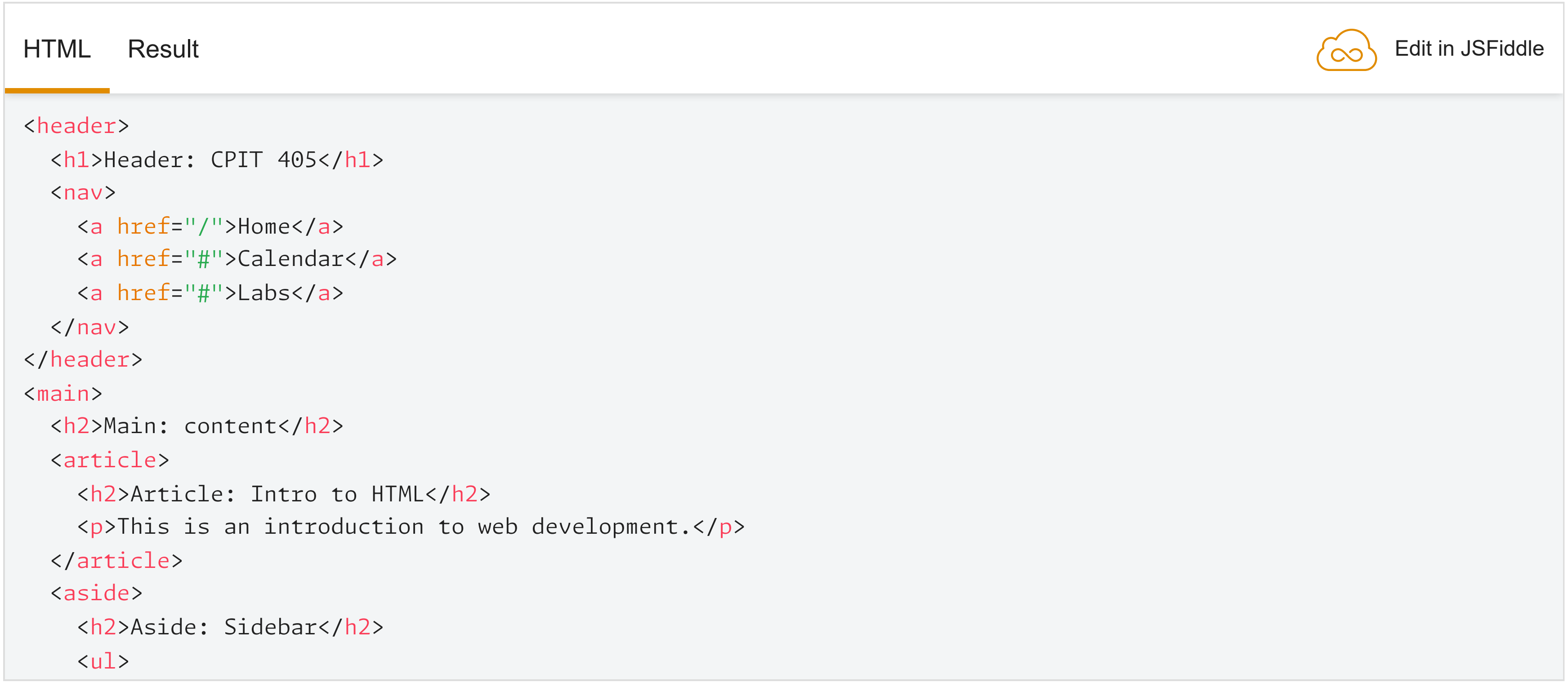
HTML Result Edit in JSFiddle

```html
<table border="1">
  <tr>
    <th rowspan="3">Day</th>
    <th colspan="3">My Class Schedule</th>
  </tr>
  <tr>
    <th colspan="2">Time</th>
    <th rowspan="2">Topic</th>
  </tr>
  <tr>
    <th>Start</th>
    <th>End</th>
  </tr>
  <tr>
    <td rowspan="2">Monday</td>
```

HTML Sectioning elements

- HTML sectioning elements are used to define the structure of a web page.
- They’re use to organize the content on a page into logical sections, such as a header, footer, main content area, and navigation bar.
- The following are the HTML sectioning elements:
    - `<header>`: Represents the header of the page.
    - `<main>`: Represents the main content area of the page.
    - `<nav>`: Represents a section of navigation links.
    - `<footer>`: Represents the footer of the page.
    - `<article>`: Represents a self-contained piece of content, such as a blog post or news article.
    - `<section>`: Represents a generic section of content.
    - `<aside>`: Represents a section of related content that is not essential to the main content of the page, such as a sidebar or footnote.

HTML Sectioning elements (Demo)



```html
<header>
  <h1>Header: CPIT 405</h1>
  <nav>
    <a href="/">Home</a>
    <a href="#">Calendar</a>
    <a href="#">Labs</a>
  </nav>
</header>
<main>
  <h2>Main: content</h2>
  <article>
    <h2>Article: Intro to HTML</h2>
    <p>This is an introduction to web development.</p>
  </article>
  <aside>
    <h2>Aside: Sidebar</h2>
    <ul>
```

The `<form>` element

- The form element is used for collecting user input.

```html
<form>
<input type="text" name="name" id="name" required />
</form>
```

Demo

HTML Result Edit in JSFiddle

```html
<form action="" method="post">
    <label for="name">Name: </label>
    <input type="text" name="name" id="name" required />
    <label for="email">Email: </label>
    <input type="email" name="email" id="email" required />
    <input type="submit" value="Subscribe" />
</form>
```

The `<form>` element: complete example

- HTML form elements include text inputs, checkboxes, dropdowns, radio buttons, textarea fields, and much more.

Demo

More on `<input>` element types

- The available HTML Input Types `<input type="">` are: `text`, `password`, `email`, `number`, `date`, `time`, `datetime-local`, `month`, `week`, `url`, `tel`, `range`, `color`, `checkbox`, `radio`, `file`, `hidden`, `submit`, `reset`, and `button`.

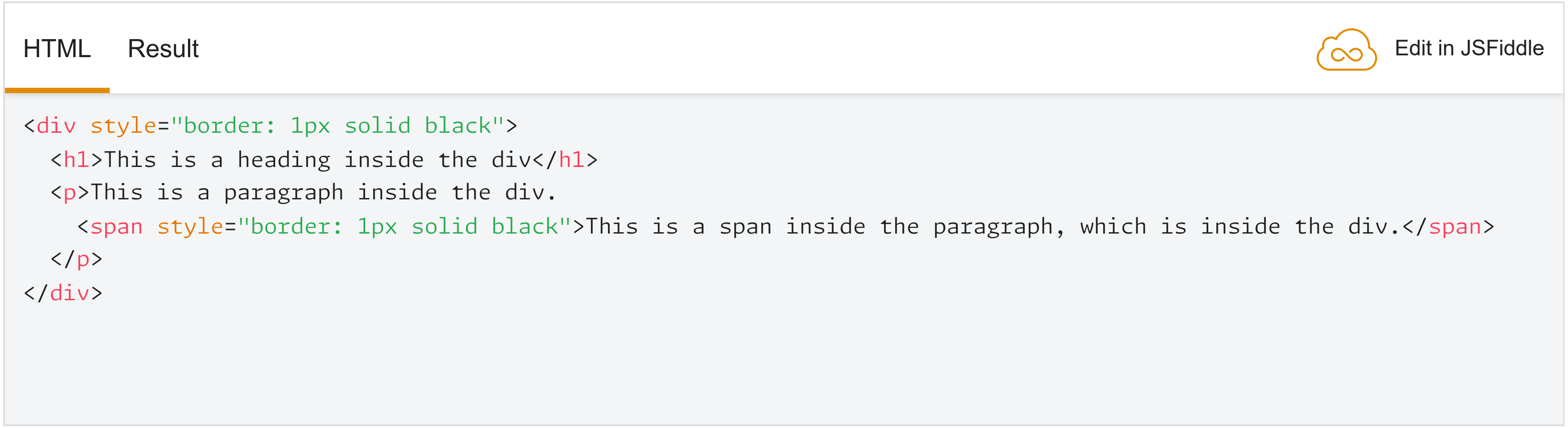
Demo

HTML Result Edit in JSFiddle

```html
<h3>HTML form example with all available input types</h3>
<form action="/action_page.php" method="post">
  <label for="name">Name:</label>
  <input type="text" name="name" id="name" placeholder="Enter your name">
  <br>

  <label for="password">Password:</label>
  <input type="password" name="password" id="password" placeholder="Enter your password">
  <br>

  <label for="email">Email address:</label>
  <input type="email" name="email" id="email" placeholder="Enter your email address">
```

The `<div>` and `<span>` elements

- The `<div>` and `<span>` elements are two of the most common HTML elements.
- The `<div>` element is a block-level element and often used to group related content together.
- The `<span>` element is an inline element and often used to format text.
- Both can be styled with CSS.
    - We'll learn about CSS next lecture!



```html
<div style="border: 1px solid black">
  <h1>This is a heading inside the div</h1>
  <p>This is a paragraph inside the div.
    <span style="border: 1px solid black">This is a span inside the paragraph, which is inside the div.</span>
  </p>
</div>
```

The `<code>`, `<pre>`, and `<samp>` elements

- These elements are used to preserve whitespace and make displaying code blocks more readable on the web.
- The `<code>` element is used for displaying text as code.
  - Typically used to display snippets of code.
- The `<pre>` element is used for displaying preformatted text.
  - Typically used to display text that needs to be preserved in its original format.
- The `<samp>` element is used for displaying sample output.
- These three elements preserve the whitespace and line breaks of the original text inside them.
- The demo is next >>

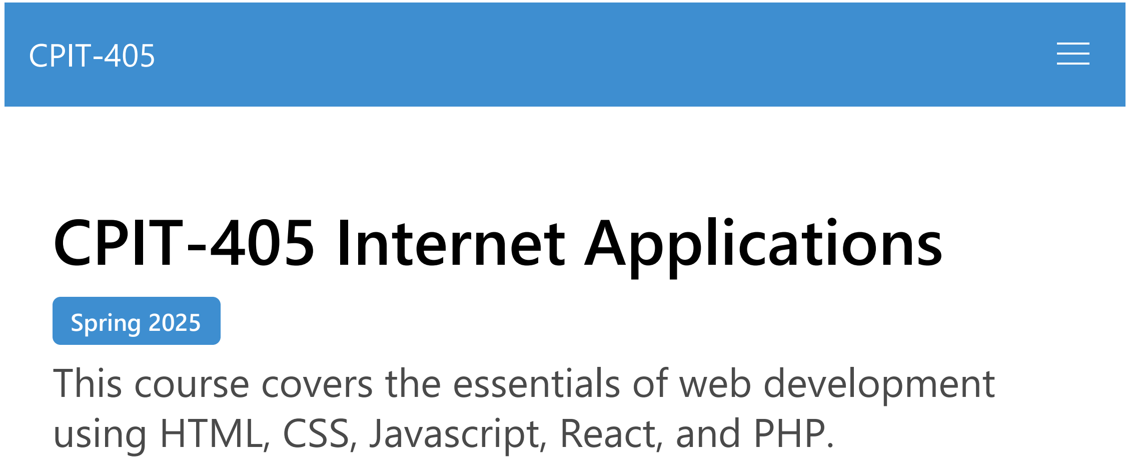
The `<code>`, `<pre>`, and `<samp>` elements

Demo

The `<iframe>` element

- The `<iframe>` element embeds another HTML document into the current document.
- This can be used to embed content from any web page, a video, or an audio file.
- The `<iframe>` element has an attribute called `src` for specifying the URL of the content to be embedded.

```
<iframe src="https://cpit405.gitlab.io" width="560" height="256"></iframe>
```


CPIT-405

CPIT-405 Internet Applications

Spring 2025

This course covers the essentials of web development using HTML, CSS, Javascript, React, and PHP.

HTML inline vs block elements

HTML elements can be grouped into two categories based on their display characteristics: block level and inline elements.

Inline elements:

- Do not start a new line
- Can't contain block elements
- Avoid nesting block elements within inline elements.
- May be used to format text within a block element.
- Examples: `<a>`, `<img>`, `<span>`, `<input>`, `<button>`, `<label>`, and `<code>`.

Block elements:

- Start a new line
- Can contain inline elements
- Cannot be nested within inline elements
- May be used to structure the content of the webpage.
- Examples: `<div>`, `<p>`, `<ul>`, `<h1>`, `<form>`, `<table>`, and `<video>`.

HTML inline vs block elements (II)

Inline elements example:

```html
<!-- INCORRECT: -->
<span><h1>Block h1 element</h1></span>

<!-- CORRECT: -->
<h1><span>Inline span element</span></h1>
```

Block level elements example:

```html
<div> This is a block element.
  <p> This is a paragraph, which is a block element.
    <span> This is an inline element.
      <a href="#"> This is a link, which is an inline element.
        <img src="image.png" alt="This is an image, which is an inline element.">
  </p>
</div>
```

HTML void elements

- HTML elements that do not have a closing tag are called *void elements*.
- We do not need to close HTML void elements.
  - The closing tag is not required, but it is allowed.
- Void elements should be self-contained and not contain any other child elements.
- Void elements are typically used to represent content that is inserted into a document inline, such as an image `<img>`, or meta data such as the `<meta>` tag.
- Examples: `<area>`, `br`, `col`, `embed`, `hr`, `img`, `input`, `link`, and `meta`.

```html
<!-- INCORRECT: -->
<img src="image.png" alt="This is an image." ></img>

<!-- CORRECT: -->
<img src="image.png" alt="This is an image." />
<img src="image.png" alt="This is an image.">
```

HTML: Wrap up and what's next

- We have seen how HTML is used to define the structure and content of a web page.
- We have learned about the following HTML topics:
  - HTML basics
  - HTML text content elements
  - HTML list and table elements
  - HTML form elements
  - HTML sectioning elements
  - HTML embedding elements
  - HTML elements that preserve whitespace
  - HTML inline vs block level elements
- You can find many helpful tutorials and articles on the MDN web documentation.
- Coming up next: Use CSS to style your web pages! 🤩