HTML
HTML

• HTML (Hypertext Markup Language) is a fundamental component of every website.

• HTML is a language, but it is not a programming language. It lacks concepts of variables, logic, functions, and the like.

• Rather, it is a markup language, using angle-bracket enclosed tags to semantically define the structure of a web page, causing the plain text inside of sets of tags to be interpreted by web browsers in different ways.
<html>
  <head>
    <title>Hello, world</title>
  </head>
  <body>
    World, hello
  </body>
</html>
<html>
  <head>
    <title>Hello, world</title>
  </head>
  <body>
    World, hello
  </body>
</html>
Hello, world
World, hello
<html>
<head>
    <title>Hello, world</title>
</head>
<body>
    World, hello
</body>
</html>
<html>
  <head>
    <title>Hello, world</title>
  </head>
  <body>
    World, hello
  </body>
</html>
<html>
<head>
<title>Hello, world</title>
</head>
<body>
World, hello
</body>
</html>
Notice how the markup allows us to convey extra information about the text we’ve written.

There are over 100 HTML tags, and lots of great resources online to find them. We won’t cover them all here.

Another interesting way to learn about HTML tags is to view the source of a website you frequent by opening up your browser of choice’s developer tools.
**HTML**

- **Common HTML tags**
  - `<b>`, `</b>`
    - Text between these tags will be rendered in **boldface** by the browser.
  - `<i>`, `</i>`
    - Text between these tags will be rendered in *italics* by the browser.
  - `<u>`, `</u>`
    - Text between these tags will be rendered **underlined** by the browser.
• **Common HTML tags**

  • `<p>, </p>`
    • Text between these tags will be rendered as a paragraph by the browser, with space above and below.

  • `<hX>, </hX>`
    • $X = 1, 2, 3, 4, 5, or 6$
    • Text between these tags will be rendered as an $X$-level section header.
• Common HTML tags

• `<ul>`, `</ul>`
  • Demarcate the beginning and end of an unordered (bulleted) list.

• `<ol>`, `</ol>`
  • Demarcate the beginning and end of an ordered (numbered) list.

• `<li>`, `</li>`
  • Demarcate list items with an ordered or unordered list.
HTML

• **Common HTML tags**
  
  • `<table>`, `</table>`
    • Demarcate the beginning and end of a table definition.

  • `<tr>`, `</tr>`
    • Demarcate the beginning and end of a row within a table.

  • `<td>`, `</td>`
    • Demarcate the beginning and end of a column within a row within a table.
Common HTML tags

- `<form>`, `</form>`
  - Demarcate the beginning and end of an HTML form.

- `<div>`, `</div>`
  - Demarcate the beginning and end of an arbitrary HTML page division.

- `<input name=X type=Y />`
  - Define a field within an HTML form. X is a unique identifier for that field, Y is what type of data it accepts.
HTML

- Common HTML tags

- `<form>, </form>`
  - Demarcate the beginning and end of an HTML form.

- `<div>, </div>`
  - Demarcate the beginning and end of an arbitrary HTML page division.

- `<input name=X type=Y />`
  - Define a field within an HTML form. X is a unique identifier for that field, Y is what type of data it accepts.
• Common HTML tags

• `<a href=X>, </a>`
  • Creates a hyperlink to web page X, with the text between the tags rendered and functional as the link text.

• `<img src=X ... />`
  • Another self-closing tag for displaying an image located at X, with possible additional attributes (such as specifying width and height).

• `<!DOCTYPE html>`
  • Specific to HTML5, lets the browser know that’s the standard you’re using.
• **HTML**

• **Common HTML tags**

  • `<! --, -->`
    • Demarcate the beginning and end of an HTML comment.

• Beyond the tags as explained here, each can also have multiple *attributes* that slightly modify the tag.
  • For example, you can usually add an `id=X` attribute, to uniquely identify a tag within an overall page.
HTML

• It is important that the HTML you write be well-formed. Every tag you open should be closed (unless it is a self-closing tag), and tags should be closed in reverse order of when they were opened.

• Unlike C, your HTML will not necessarily fail with syntax errors if not well-formed, so it’s up to you to be vigilant.

• Because it can be an arduous task to investigate this, be sure to use online HTML validators to help!