HTML

HTML is a markup language!
    Resource
Syntax
Doctype
The W3C HTML validator
<Head>
    Setting primary language
    Setting character encoding
    Doc title
    Adding style
    Adding dynamic feature using JavaScript
HTML formatting
    id VS class
 container for other HTML elements
 container for some text
<class>
    Define equal styles for elements with the same class name
<id>
    Unique id defined, used by CSS and JS to perform task for the element
    For CSS: use #
    With `href` link
    For JS: use `document.getElementById()`
HTML content block template
    Useful template:

HTML is a markup language!

Resource

HTML Tutorial
https://www.tutorialspoint.com/html/index.htm

HTML 5.2
http://www.w3.org/TR/html5

HTML Standard
http://www.whatwg.org/specs/web-apps/current-work/multipage

Your Web, documented. · WebPlatform.org
http://www.webplatform.org

HTML reference
http://developer.mozilla.org/en-US/docs/Web/HTML/Reference

Syntax

```
<p lang="en">
  This is for a paragraph. <em>emphasized text</em> <strong>bold text</strong>
</p>

<h1>
  Heading 1
</h1>
```

HTML is a plain textual representation of content and its general meaning. For example:

```
<p id="example">This is a paragraph.</p>
```

The ** ** part is a marker, commonly called a *tag* that means "what follows should be considered as a paragraph". Because it is at the start of the content it affects, it is an "opening tag". Likewise, the ** ** tag indicates the end of the paragraph, and is thus a "closing tag". The opening tag, closing tag, and everything in between is called an *element*. Note: Many people use the terms “element” and “tag” interchangeably, which is incorrect. (The **id="example"** is an *attribute-value pair*; we'll come back to these later.)

In most browsers there is a “Source” or “View Source” option, commonly under the “View” menu. Try this now: go to a web site, choose this option, and spend some time looking at the HTML that makes up the page.

Doctype

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>My fabulous document</title>

   ... etc.
```

HTML	XHTML
Elements and attributes are case insensitive, e.g., <h1> is the same as <H1>.	Elements and attributes are case sensitive; they are all lowercase.
Certain elements don't need a closing tag (e.g., paragraphs,), while others (called "empty elements") shouldn't have a closing tag (e.g., images, <img>).	All elements must be explicitly closed (e.g., A paragraph). Elements without content should be closed using a slash in the start tag (e.g.,).
Attribute values may be written without being enclosed in quotes.	Attribute values must be enclosed by quotes.
Shorthand can be used for certain attributes (e.g., <input required>).	The full attribute form must be used for all attributes (e.g., <input required="required">).

The W3C HTML validator

The W3C Markup Validation Service
http://validator.w3.org

<Head>

The head is the place where most of the instructions for the browser are located and where you store extra information, called **metadata**, about the document.

- **Metadata** typically define the document title, character set, styles, links, scripts, and other meta information.
- The following tags describe metadata: `<title>`, `<style>`, `<meta>`, `<link>`, `<script>`, and `<base>`.

- The `<meta>` element is used to specify which character set is used, page description, keywords, author, and other metadata.

Setting primary language

```
<html lang="en-GB">
  ...
</html>
<head>
  ...
</head>
```

http://www.iana.org/assignments/language-subtag-registry/language-subtag-registry

Lookup tool

Language subtag lookup app
https://r12a.github.io/app-subtags/

Setting character encoding

```
<meta charset="utf-8">
```

Doc title

Appears in the browser application title bar (the bar bordering the top of the browser window)

```
<!DOCTYPE html>
<html lang="en-GB">
<head>
  <meta charset="utf-8">
  <title>I am a title example</title>
</head>
<body>
</body>
</html>
```

Adding style

```
<style type="text/css">
    body{
      background:#000;
      color:#ccc;
      font-family: helvetica, arial, sans-serif;
    }
  </style>
```

Adding dynamic feature using JavaScript

```
<script type="application/javascript">
    function leave(){
      return confirm("This will take you to another site,\n are you sure you want to go?")
    }
  </script>
```

It is much better to put your styles and scripts in external files, and import them into your HTML files where needed, so you only need to update them in one place if changes need to be made. For JavaScript, you do this using `script` elements that have no script inside them, but instead link to an external file using a `src` attribute, as seen in the code below.

```
<script type="application/javascript" src="leaving.js"></script>
```

HTML formatting

```
<!DOCTYPE html>
<html>

  <head>
    <title>My First Webpage</title>
  </head>

  <body>
    <h1> Title 1</h1>
    <p>
      This is the first paragraph
    </p>
    <h2> Title 2</h2>
    <p>
      This is the second paragraph
    </p>
    <div>for css styling...</div>
    <span>for css styling...</span>
    <p>
      This is <strong>strong</strong>.<br>
```

```
      This is <em>emphasized</em>.<br>
      This is <i>italic</i>.<br>
      This is <b>bold</b>.<br>
      This is <u>underlined</u>.<br>
      This is <strike>strike-through</strike>.<br>
      This is <del>deleted</del> <ins>inserted</ins>.<br>
      This is <mark>marked</marked>.<br>
      This is <sub>subscript</sub>.<br>
      This is <sup>superscript</sup>.<br>
      This is <big>BIG text</big>.<br>
      This is <small>small text</small>.<br>
    </p>
    <a href="http://www.google.com"> This is the Google website. </a>
    <h3> Here is an unordered list </h3>
    <ul>
      <li>Item 1</li>
      <li>Item 2</li>
    </ul>
    <h4> Here is an ordered list </h4>
    <ol>
      <li>Item 1</li>
      <li>Item 2</li>
    </ol>

  </body>

</html>
```

id VS class

The `id` attribute specifies a unique id for an HTML element (the value must be unique within the HTML document).

The id value can be used by CSS and JavaScript to perform certain tasks for a unique element with the specified id value.

In CSS, to select an element with a specific id, write a hash (#) character, followed by the id of the element:

```
<style>
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>

<h1 id="myHeader">My Header</h1>
```

An HTML element can only have one unique id that belongs to that single element, while a class name can be used by multiple elements:

```
<style>
/* Style the element with the id "myHeader" */
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}

/* Style all elements with the class name "city" */
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>

<!-- A unique element -->
<h1 id="myHeader">My Cities</h1>

<!-- Multiple similar elements -->
<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>

<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>
```

id can be used as **bookmark**

```
<h2 id="C4">Chapter 4</h2>
...
<a href="#C4">Jump to Chapter 4</a>
...
cross-file
<a href="html_demo.html#C4">Jump to Chapter 4</a>
```

get id in JavaScript

```
<script>
function displayResult() {
  document.getElementById("myHeader").innerHTML = "Have a nice day!";
}
</script>
```

container for other HTML elements

- Common attributes

- style
- class
- id

container for some text

- Common attributes
  - style
  - class
  - id
- can be used to style part of the text

<class>

Define equal styles for elements with the same class name

```
<head>
<style>
.classname{
  background-color: black;
  color: white;
  margin: 20px;
  padding: 20px
}
</style>
</head>

...

<body>

<div class="classname">
  ...
</div>

<div class="classname">
  ...
</div>

</body>
```

<id>

Unique id defined, used by CSS and JS to perform task for the element

For CSS: use

```
<style>
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>

<h1 id="myHeader">My Header</h1>
```

With `href` link

```
<h2 id="C4">Chapter 4</h2>
...
<a href="#C4">Jump to Chapter 4</a>

<!-- or even from another page -->
<a href="html_demo.html#C4">Jump to Chapter 4</a>
```

For JS: use `document.getElementById()`

```
<script>
function displayResult() {
  document.getElementById("myHeader").innerHTML = "Have a nice day!";
}
</script>
```

HTML content block template

- `extends` a layout
- **Open up** a content block

```
{% here you can write code %}

example:

{% if xxx %}
<>...</>
{% else %}
<>...</>
{% endif %}
```

```
"xxx.html"
<!DOCTYPE html>
<html>
<head>
  ...  --> everywhere else will be inherited
</head>
<body>
  {% block content %}{% endblock %} --> here is the substitution
</body>
</html>

"yyy.html" which inherits "xxx.html" as template
{% extends "xxx.html" %} --> "xxx.html" contains your template
{% block content %}
  ... --> Your content here
{% endblock content %}  --> This is a content block
```

Useful template:

Bootstrap
https://getbootstrap.com/