Understanding HTML

Hunter Jorgensen

Background

HTML was proposed by Tim Berners-Lee in 1989, created in 1991 and finally released in 1995.

HTML became popular with the release of HTML 4.0 in 1999

HTML 5.0 (or HTML5) was initially released in 2014 and was the standard version until last year

HTML Living Standard is now the guiding design behind HTML

Background

HTML (HyperText Markup Language) is a way to write documents that can shared over the internet

For many years, W3C (World Wide Web Consortium, run by Tim Berners-Lee (!)) managed HTML

W3C tried to create ‘releases’ of HTML (i.e., 4.0, 5.0, etc.)

After some infighting/disagreements, WHATWG (Web Hypertext Application Working Group) continued HTML development with Living Standard in 2019, but is regularly known at HTML5

Essentially, these groups propose changes to HTML and the browser have to support them (so some browsers can behave differently than others!)

Starting Simply

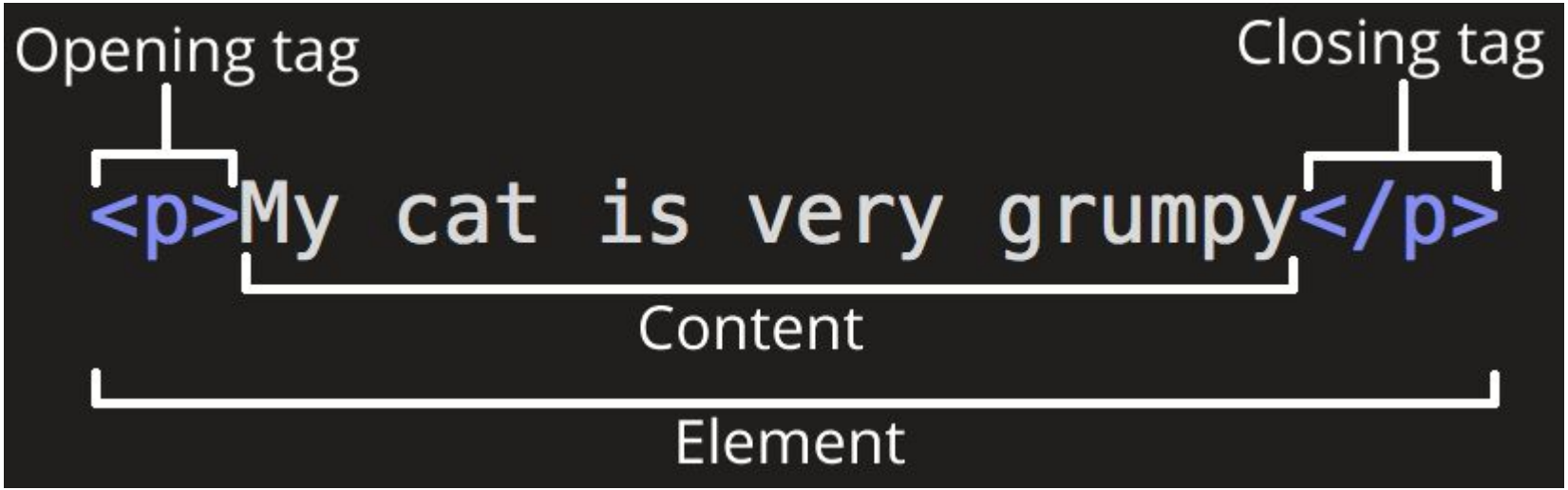
HTML is not a programming language; it is a markup language that defines the structure of a web page’s content.

It uses a series of elements to make the content appear or behave in specific ways.

The Grumpy Cat

Say we wanted a website that had the information: “My cat is grumpy.”

We can put this in between the paragraph tags (explained more later) to give it certain characteristics on a website: `<p>My cat is grumpy</p>`



Nesting

You can nest elements, but each element has to be closed in the proper order:

```
<p>My cat is <strong>grump!</strong></p>
```

Note how the tag strong element is closed, before the paragraph tag is closed

Block vs Inline

Block and Inline are overloaded terms in software development (they have different meanings in JavaScript, CSS, etc.), but in HTML they refer to elements that start on a new line (i.e., have their own block) or inline (does not start on a new line.)

```
<em>first</em><em>second</em><em>third</em>
```

```
<p>fourth</p><p>fifth</p><p>sixth</p>
```

Attributes

ALL HTML tags have attributes, which is extra information outside of the content of the element. Some attributes are universal across all tags, while some are unique, but all have some purpose:

```
<p class='cat-fact'>My cat is grumpy</p>
```

Class is a fairly common attribute (shared by many tags). Note that the ‘cat-fact’ can be written with single or double quotes or, if it’s a single string, no quotes at all.

Boolean Attributes

Some attributes are activated simply by referencing them:

```
<input type="text" disabled>

<input type="text">
```

Some your will also see:

```
<input type="text" disabled="disabled">
```

Empty Elements

Empty Elements, also known as Self Closing or Void Elements, are elements consists of a single tag modified by attributes.

```
<img
src="https://raw.githubusercontent.com/mdn/beginner-html-site/gh-pages/images/firefox-icon.png">
```

Special Characters

Literal Character	Character Reference Equivalent
<	<
>	>
“	"
‘	'
&	&
New space	

Note that all special characters start with an & and end with a ;

Spacing

All spaces in HTML are reduced down to a single space when rendering:

```
<p>My            cat is

grumpy</p>
```

You'll need to use special characters if you want additional spaces!

However, this means that you can organize HTML into many different ways.

There is one exception to this - do you know what it is?

Comments

Comments are delimited like:

```
<!-- <p>I am invisible!</p> -->
```

See also also the duck on Amazon.com’s home page!

Making a page

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My first page</title>
  </head>
  <body>
    <p>Hello world</p>
  </body>
</html>
```

Important Stuff!

`<!DOCTYPE html>` is both antiquated and highly recommended. This let's the browser know what version/format the document is in. However, if you don't set this, some browsers may infer the incorrect version of HTML, causing unexpected behavior.

`<html>...</html>` is the container of everything (except DOCTYPE) and is often referred to as the root.

<head>...</head> is content and metadata that is not usually seen on the page, but affects it (setting the page title, importing style)

`<title>My Title</title>` gives a custom header to the page.

`<meta charset="utf-8">` specifies what characters should be used on the page. Should almost always be set to 'utf-8'.

Body

<body>...</body> is where you'll be doing most of the work. It is where almost all the content, including JavaScript, is used.

There can only be one set of body tags per page.

Headings

Use headings as titles to introduce distinct sections

Headings come in 6 sizes starting with h1, h2, through h6

H1 is the largest size, h6 is the smallest

- `<h1>This is the largest heading</h1>`
- `<h2>This is the second largest heading</h2>`
- `<h6>This is the smallest heading</h6>`

P, or Paragraph

Use paragraphs to add long portions of text

Paragraphs add vertical spacing between portions of text

- <**p**>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book</**p**>
- <**p**>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book</**p**>

Div

Div, short for dividers, is probably the most common HTML tag since it just creates horizontal blocks of content.

My cat is grumpy

But my dog is nice.

div is often used to organize and arrange huge large amounts of HTML content, and other tags are often nested within div’s.

span

`<span>` is probably the second most common HTML tag after `<div>`. Unlike div, span is in-line and usually meant for small amounts of content. It is less common for there to be nested elements in here.

How will this render?

The cat jumped over the moon.But the mouse did not!

List

```
<h2>Courses Spring 2018</h2>
<ul>
  <li>CS5610</li>
  <li>CS5610</li>
  <li>CS5610</li>
</ul>
<h2>Semesters</h2>
<ol>
  <li>Spring</li>
  <li>Summer 1</li>
  <li>Summer 2</li>
  <li>Fall</li>
</ol>
```

Tables

Table Link:
https://gist.github.com/hunterjorgensen167/8b1221f5ae2a8693d37a7a07057583fa

Are created by <table>, with a <thead> defined and individual <th> elements. The <tbody> is made up of rows (<tr>) combined by individual elements, <td>
<tfoot>, <thead> and <tbody> are all optional! Table will render without them

Tables

Tables have a lot of contextual nesting that makes it a bit tricky to read sometimes. You can have, in this order:

1. an optional <caption> element,
2. zero or more <colgroup> elements,
3. an optional <thead> element,
4. either one of the following:
    - zero or more <tbody> elements
    - one or more <tr> elements
5.
6. an optional <tfoot> element

Input fields and labels

```
<label for="usernameFld">
Username
</label>
<input id="usernameFld"
type="text"
title="Username"
placeholder="alice" />
```

```
<label for="firstNameFld">
First Name</label>
<input id="firstNameFld"
     value="Alice" />
```

Date Input Fields

```
<label for="date-of-birth">
Date of Birth
</label>

<input type="date" id="date-of-birth" name=”dob” value="2011-11-22" />
```

Input Fields

Input fields are quite powerful, especially when paired with forms (which we will talk about shortly.)

They have required fields, which ensures that the input field is not empty.

Additionally, you can specify the input attribute `type`, such as *text*, *number*, *email*, *password*, and more.

Labels increase the area that a user can interact with a field.

Drop Down List

```
<select name="role">
  <option value="FACULTY">
     Faculty
  </option>
  <option value="STUDENT">
     Student
  </option>
  <option value="ADMIN">
     Admin
  </option>
</select>
```

Buttons

<button type="button"> Delete**</button>**

<button type="button"> Edit **</button>**

<button type="submit"> Update **</button>**

The button *type* attribute will override the behavior in the form! Good consideration when wanting to cancel, X-out, etc.

For this class, we will be ignoring most normal type behavior

Forms

Buttons, Inputs, etc. all come together with the <form> tag.

Form tags usually wrap around <input> and <button> tags and allow everything to be built as a single input. Specifically, the form action attribute will redirect the data to another URL:

<form action="https://google.com/">

…

</form>

Demo: https://jsbin.com/nanezufuya/edit?html,css,output

The action data can be overridden by the button/input formaction attribute.

Anchors

This is mainly used to create links to other sites, but is also used to open email/phone clients, etc.

****

Hello world!

****

What's new in HTML5?

There are a lot of new feature in HTML5 and while we won't be using them this semester, many of them are about changes in site behavior (i.e. - related to blogs, videos, music, etc.) This is the reason Adobe Flash isn't very common anymore!

`Header <header></header>`

`Footer <footer></footer>`

`Page section <section></section>`

New media type

<audio> Defines sound content
<embed> Defines a container for an external (non-HTML) application
<source> Defines multiple media resources for media elements (<video> and <audio>)
<track> Defines text tracks for media elements (<video> and <audio>)
<video> Defines video or movie

```
<video width="400" controls>
  <source src="abc.mp4" type="video/mp4">
  <source src="abc.ogg" type="video/ogg">Make sure your browser does support HTML5 video.</video>
```

Other Tags

There are almost 100 unique tags in HTML5, but I've tried to identify in the previous slides ones that will be most important. Some other cool tags to checkout out might be:

- time - to format datetimes
- blockquote - when you have a quote to show
- sup/em/strong/u/etc. - for different ways to style text
- pre and code - to help write code in HTML
- img - link to an image

And the most import tag of all!?

`<marquee>`

…. Just kidding

Debugging

HTML is permissive: many syntax or logic errors can be ignored so an unclosed element or a typo might not get caught for some time.

Example: https://gist.github.com/hunterjorgensen167/3fef3004fec5cde4ed67f0122de9e5d9

Tools:

1. DOM Inspector in your Browser
2. W3 Validator: https://validator.w3.org/

Lab: Formatting A Recipe

Let’s try to format some text:

https://gist.github.com/hunterjorgensen167/2b0dc54bd803f79fb6345062ea3451c8

If you send this to TA Nandish as an HTML file within 24 hours of class, you will get 2 EC points