



Intro to HTML and CSS

Natassja Linzau

9/15/2016



Natassja Linzau (nlinzau@doc.gov)
Lead of Data Education Initiatives
Commerce Data Service
US Department of Commerce

Data Academy questions:
DataAcademy@doc.gov



Goals

Walk away with the foundations for :

- Understanding how tags and other HTML elements structure the content of a webpage
- Being able to create a basic webpage using HTML and CSS
- Understanding how CSS interacts with HTML
- Becoming comfortable with how to use the Developer Tools in Chrome Browser



Commerce Data Academy

- A data education initiative of the Commerce Data Service (CDS).
- Launched by the CDS to offer data science, data engineering, and web development training to employees of the U.S. Department of Commerce.
- Course schedule and materials (e.g. slides, code, papers) produced for the Commerce Data Academy can be found on our website at <http://dataacademy.commerce.gov>.
- Questions? Feel free to write us at Data Academy (dataacademy@doc.gov).



HyperText Markup Language (HTML)

HyperText Markup Language (HTML)



HTML describes the structure of a webpage to your browser.



HTML Tags

```
<h1>This is a headline</h1>
<p>This is a paragraph.</p>
<p>This is a second paragraph.</p>
<p>This is a third paragraph.</p>
```

- Your computer can tell the difference between a **headline** and a bunch of **paragraphs** because each individual element has its own tag.
- You can write HTML just like normal text and save it in a file with .html at the end. HTML is the only thing that you *have to have* in order to make a website.

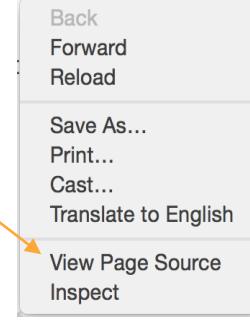


HTML supports many types of media:

Paragraph	Audio	Video	Lists
<p>	<audio>	<video>	
Articles	Image	Emphasis	Button
<article>			<button>

In order to put any one type of media in your website, you have to use its specific HTML tag. For example, videos get <video> tags, images get tags, buttons get <button> tags, lists get tags, and so on..

Try this:

1. Open up your browser and navigate to your favorite webpage.
2. Right-click (for Windows) or Control+Click (for Mac) then select “View Page Source.”
3. See if you can find any of the tags I have mentioned so far.



HTML and Cascading Style Sheets (CSS): What's the difference?

HTML is used for **meaning** and CSS is used
for **presentation**.

HTML = fancy structured content
CSS = the visual formatting of that content



Let's Get Started

- Most of the stuff on the Web is just a bunch of files stored in directories, just like the files on your own computer.
- HTML files are nothing more than simple text files.
- To get started, open your text editor such as Notepad or Atom or Sublime.



Try this:

1. Open a new text file and type the following text: Hello World! This is my first webpage.
2. Create a folder called “html” wherever you like to save files on your computer and save the file into the folder as myfirstpage.html
3. View the files online; your files don’t even have to be on the web for you to view them in a browser.
4. To view online, open a web browser such as Chrome, Firefox, Safari or Internet Explorer and in the address bar, where you usually type web addresses, type in the location of the file you just saved (for example, c:\html\myfirstpage.html) and hit return. Alternatively, go to the File menu of the browser, select Open, and browse for the file.



Tags

Although plain text is the basics of HTML, you need a bit more to make it a more polished document. Some of the “bit more” includes tags, attributes, and elements.

Tags surround content and apply meaning to it. Generally, tags have *opening* and *closing tags* (`<html></html>`).

However, not all tags have closing tags--some tags which do not wrap around content will close themselves. The line-break tag for example, looks like this : `
` - a line break doesn't hold any content so the tag sits all by its own lonesome self.

**All you need to remember for now is that, generally, all tags with content between them should be closed, in the format of: opening tag → content → closing tag.



Try this:

Return to your document and type the following:

```
<!DOCTYPE html>
<html>
<body>
```

This is my first web page.

```
</body>
</html>
```

Now save the document again, go back to the web browser and reload the page.



Attributes

Tags can also have **attributes**, which are extra bits of information related to the tags.

Attributes appear inside the opening tag and their values sit inside quotation marks.

Attributes look something like `<tag attribute="value">Pudding</tag>`. We will come across tags with attributes later.

Note: Quotation marks aren't always essential but it is a good-practice convention that is recommended for consistency and clarity.



Elements

Tags generally don't do much more than mark the beginning and end of an element. Elements are the bits that make up web pages.

An HTML **element** usually consists of a start tag and end tag, with all the content inserted in-between: <tagname>Content goes here...</tagname>.

Examples:

- Everything that is in-between (and includes) the <body> and </body> tags is the **body element**.
- Whereas "<title>" and "</title>" are tags, "<title>James Bond</title>" is a **title element**.



Nested Elements

- HTML elements can be nested (elements can contain elements).
- All HTML documents consist of nested HTML elements.
- How many elements do you count on the page you just saved?



Try this:

Return to your document and type the following:

```
<!DOCTYPE html>
<html>
<head>
    <title>My first web page</title>
</head>
<body>
```

This is my first web page.

```
</body>
</html>
```

Save this and then view it in the browser.



Try this:

```
<!DOCTYPE html>
<html>
<head>
    <title>My first web page</title>
</head>
<body>
```

This is my first web page.
How exciting!

```
</body>
</html>
```

Save and then view in the browser.



Paragraphs

The < p > tag is used for **paragraphs**. Think of the HTML content as if it were a book - with paragraphs where appropriate.

Also, if you want text to appear on different lines or, rather, if you intend there to be two distinct blocks of text (because, remember, HTML is about meaning, not presentation), you need to explicitly state that.



Try this:

```
<!DOCTYPE html>
<html>
<head>
    <title>My first web page</title>
</head>
<body>

    <p>This is my first web page.</p>
    <p>How exciting!</p>

</body>
</html>
```

Save and then view in the browser.



Emphasis

You can emphasize text in a paragraph using `` (emphasis) and `` (strong importance).

For example:

```
<p>Yes, that really <em>is</em> exciting.  
<strong>Warning:</strong> level of excitement  
may cause head to explode.</p>
```



Try this:

```
<!DOCTYPE html>
<html>
<head>
  <title>My first web page</title>
</head>
<body>
  <p>This is my first web page.</p>
  <p>How exciting!</p>
  <p>Yes, that really <em>is</em> exciting.</p>
  <p><strong>Warning:</strong> Level of excitement may cause head to
  explode.</p>
</body>
</html>
```

Save and then view in the browser.



Headings

If you have documents with genuine **headings**, then there are HTML tags specifically designed just for them.

These headings are **h1**, **h2**, **h3**, **h4**, **h5** and **h6**. **<h1></h1>** is considered to be the almighty emperor of headings and **<h6></h6>** is the “lowest heading on the totem pole.”



Try this:

```
<!DOCTYPE html>
<html>
<head>
<body>

<h1>My first web page</h1>
  <h2>About</h2>
    <p>This is a simple page put together using HTML</p>
  <h2>Purpose of this Page</h2>
    <p>To learn HTML</p>

</body>
</html>
```

Save and then view in the browser.



Lists

There are three types of list: unordered lists, ordered lists, and definition lists.

- **Unordered lists** are used for *non-sequential lists with list items usually preceded by bullets.*
- **Ordered lists** are used for *sequential lists, which are normally represented by incremental numbers.*



List Tags

- The `` tag is used to define unordered lists.
- The `` tag is used to define ordered lists.
- Inside the lists, the `` tag is used to define each list item.



Try this:

```
<!DOCTYPE html>
<html>
<head>
<body>
<h1>My first web page</h1>
  <h2>About</h2>
    <p>This is a simple page put together using HTML</p>
  <h2>Purpose of this Page</h2>
    <ul>
      <li>To learn HTML</li>
      <li>To show off</li>
      <li>Because I can</li>
    </ul>
</body>
</html>
```

Save and then view in the browser.



Try this:

```
<!DOCTYPE html>
<html>
<head>
<body>
<h1>My first web page</h1>
  <h2>About</h2>
    <p>This is a simple page put together using HTML</p>
  <h2>Purpose of this Page</h2>
    <ol>
      <li>To learn HTML</li>
      <li>To show off</li>
      <li>Because I can</li>
    </ol>
</body>
</html>
```

Save and then view in the browser.

Try this:

... (Note: Keep the code above this line in your document and just change the portion below)

```
<h2>Purpose of this Page</h2>
<ul>
  <li>To learn HTML</li>
  <li>To show off</li>
  <ol>
    <li>To my boss</li>
    <li>To my friends and acquaintances</li>
    <li>To my guinea pig</li>
  </ol>
  <li>Because I can</li>
</ul>
</body>
</html>
```

Save and then view in the browser.

Links

- What makes the Internet so special is that it all **links** together.
- The “H” and “T” in “HTML” stand for “hypertext”, which basically means a system of linked text.
- An **anchor** tag (`<a>`) is used to define a link, but you also need to add something to the anchor tag — the **destination** of the link.



Try this:

```
<!DOCTYPE html>
<html>
<head>
<title>My first web page</title>
</head>
<body>
    <h1>My first web page</h1>
    <h2>What this is</h2>
        <p>A simple page put together using HTML</p>
    <h2>Why this is</h2>
        <p>To learn HTML</p>
    <h2>Where to find this tutorial</h2>
        <p><a href="http://dataacademy.commerce.gov/previous-
            courses.html">Previous Course Materials</a></p>
</body>
</html>
```



More About Links

- Links can be **absolute**, leading to a specific URL `` or **relative** to the current page`Previous Courses`.
- A link does not have to link to another HTML file in the same directory; it can link to any file anywhere on the web.
- A link can also send a user to another part of the same page they are on, using an id attribute. You could type in the following:

```
<h2 id="courses">Courses</h2>
```

Then link to it by using something like this:

```
<a href="#courses">Go to previous courses</a>.
```



Images

Everything we've explored so far is very text-centric.

The web is not just about text; it's about images, video, and audio too.

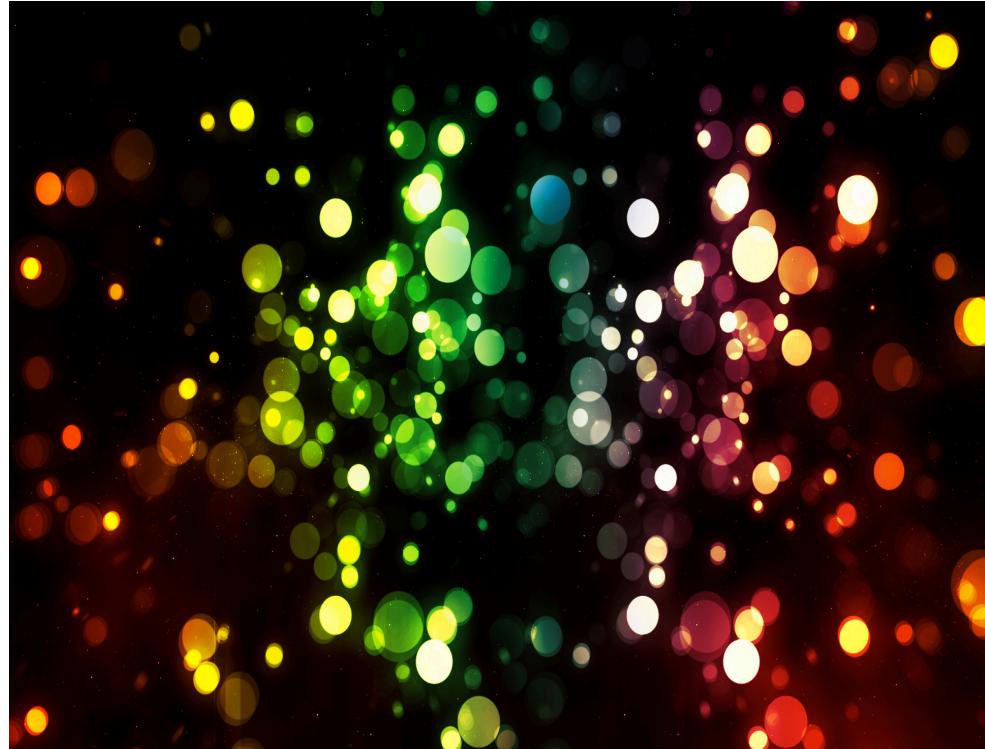




Image Tags

- The **** tag is used to put an image in an HTML document and it looks like this:

```

```

- The **src** attribute tells the browser where to find the image.
- Like the **<a>** tag, this path can be *absolute*, but is usually *relative*. For example, if you create your own image and save it as “sparkleimage.jpg” in a directory called “images” then the code would be ``



Try this:

```
<!DOCTYPE html>
<html>
<head>
<title>My first web page</title>
</head>
<body>
    <h1>My first web page</h1>
    <h2>What this is</h2>
        <p>A simple page put together using HTML</p>
    <h2>Why this is</h2>
        <p>To learn HTML</p>
    <h2>How to learn more about HTML</h2>
        <p><a href="http://www.example.com"></a></p>
</body>
</html>
```

Tables

- HTML tables are still best known for being used and abused to lay out pages.
- However, the correct use for **tables** is to do exactly what you would expect a table to do — to **structure tabular data**.
- There many tags specifically used in tables, and to fully get to grips with how they work is one of the more challenging aspects of beginning to work in HTML.

Table Tags

- The **<table>** element defines the table.
- The **<tr>** element defines a **table row**.
- The **<th>** element defines a **table header** cell.
All **<th>** elements must be enclosed
inside **<tr>** elements.
- The **<td>** element defines a **table data cell**.
As with **<th>** elements, all **<td>** elements must be
enclosed in **<tr>** tags.



Try this:

```
<!DOCTYPE html>
<html>
  <table>
    <tr>
      <th>Column 1 heading</th>
      <th>Column 2 heading</th>
      <th>Column 3 heading</th>
    </tr>
    <tr>
      <td>Row 2, cell 1</td>
      <td>Row 2, cell 2</td>
      <td>Row 2, cell 3</td>
    </tr>
    <tr>
      <td>Row 3, cell 1</td>
      <td>Row 3, cell 2</td>
      <td>Row 3, cell 3</td>
    </tr>
  </table>
</html>
```



Cascading Style Sheets (CSS)



CSS

CSS, or **Cascading Style Sheets**, is a way to style and present HTML.

Whereas the HTML is the **meaning** or **content**, the style sheet is the **presentation** of that document.

Styles have a basic format of '**property: value**' and most properties can be applied to most HTML tags.



Applying CSS

There are three ways to apply CSS to HTML: **Inline**, **internal**, and **external**.

Inline=**Inline** styles are put straight into the HTML tags using the style attribute. `<p style="color: red">text</p>`

Internal=**Embedded**, or **internal**, styles are used for the whole page. Inside the head element, the style tags surround all of the styles for the page.

External=**External** styles are used for the whole, multiple-page website. In this case, there is a **separate CSS file** that includes all the styles for the site.

Internal Styles

Internal=Embedded, or internal, styles are used for the whole page. Inside the head element, the style tags surround all of the styles for the page. For example:

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Example</title>
<style>
    p {
        color: red;
    }
    a {
        color: blue;
    }
</style> ...
```



External Styles

External=External styles are used for the whole, multiple-page website. In this case, there is a **separate CSS file** that includes all the styles for the site. This file would simply look something like:

```
p {  
    color: red;  
}  
  
a {  
    color: blue;  
}
```



Linking to External Style Sheets

If this file is saved as “style.css” in the same directory as your HTML page then it can be linked to in the HTML like this:

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Example</title>
<link rel="stylesheet" type="text/css" href="style.css"> ...
```



Try this:

1. Start a fresh new file with your text-editor and save the blank document as “style.css” in the same directory as your HTML file.
2. Change your HTML file so that it starts something like this:

```
<!DOCTYPE html>
<html>
<head>
<title>My first web page</title>
<link rel="stylesheet" href="style.css">
</head> ...
```

3. Save your HTML file and refresh your browser.



Selectors, Properties, and Values

HTML has tags; CSS has selectors.

Selectors=the names given to styles in internal and external style sheets.

Each selector has “**properties**” inside **curly brackets {}**, which take the form of words such as color, font-weight, or background-color.

A **value** is given to the property following a **colon** (NOT an “equals” sign). **Semi-colons** are used to separate the properties.



Length and Percentages

- **px** (such as font-size: 14px) is the unit for pixels.
- **em** (such as font-size: 4em) is the unit for the calculated size of a font. So “4em”, for example, is four times the current font size.
- **pt** (such as font-size: 12pt) is the unit for points, for measurements typically in printed media.
- **%** (such as width: 80%) is the unit for percentages.
- Other units include **pc** (picas), **cm** (centimeters), **mm** (millimeters) and **in** (inches).

When a value is **zero**, you do not need to specify a unit. For example, if you wanted to specify no border, it would be border: 0.



Colors

CSS colors can take the form of a predefined name, an RGB value or a hex code.

Name: Predefined color names

include aqua, black, blue, fuchsia, gray, green, etc. *Example:* red

RGB: (red/green/blue) values of various combinations between 0 and 255. 0 is the lowest level (no red, for example) and 255 is the highest level (full red, for example). These values can also be a percentage.

Example: `rgb(255,0,0)`

`rgb(100%,0%,0%)`

Hex code=The hex number starts with a hash character (#) and can be three or six digits in length.

Example: `#ff0000`

`#f00`



Try this:

Green text on a yellow background could look like this:

```
h1 {  
  color: green;  
  background-color: yellow;  
}
```

You can apply color and background properties to most HTML elements, including body, which would change the colors of the page and everything in it.

```
body {  
  font-size: 14px;  
  color: navy;  
}  
  
h1 {  
  color: #ffc;  
  background-color: #009;  
}
```

Experiment with color and background-color to find the right combination for your web page.



Text

Using CSS, you can change the size and shape of the text on a web page with a range of properties.

Font-family is the font itself, such as Times New Roman, Arial, or Verdana.

Font-size is the actual size of the font.

Font-weight indicates the thickness of the font and whether it is bold or not (common values are bold or normal)

Font-style states whether the text is italic or not. It can be font-style: italic or font-style: normal.

Text-decoration states whether the text has got a line running under, over, or through it.

Text-transform will change the case of the text (capitalize, lowercase, uppercase, none).

Text, cont'd.

- The **letter-spacing** and **word-spacing** properties are for spacing between letters or words. The value can be a length or normal.
- The **line-height** property sets the height of the lines in an element, such as a paragraph, without adjusting the size of the font. It can be a number (which specifies a multiple of the font size, so “2” will be two times the font size, for example), a length, a percentage, or normal.
- The **text-align** property will align the text inside an element to left, right, center, or justify.
- The **text-indent** property will indent the first line of a paragraph, for example, to a given length or percentage. This is a style traditionally used in print, but rarely in digital media such as the web.



Margins, Padding, and Borders

Margin is the space **outside** something.

Padding is the space **inside** something. [Picture/graphic here]

The four sides of an element can also be set individually:

margin-top, margin-right, margin-bottom, margin-left

padding-top, padding-right, padding-bottom, and padding-left

Borders can be applied to most HTML elements within the body.

To make a border around an element, all you need is border-style. The values can be solid, dotted, dashed, double, groove, ridge, inset and outset.



The Box Model

Margins, padding and borders are all part of what's known as **the Box Model**. The Box Model works like this: in the middle you have the content area (let's say an image), surrounding that you have the padding, surrounding that you have the border and surrounding that you have the margin.

[Box model here]

The Box Model can be applied to every element on the page, which gives great flexibility.



Try this:

You should already have an HTML file, with a line that we added to link the HTML file to the CSS file.

The best way to fully understand all of this is to play around with the HTML and CSS files and see what happens when you change things.

Try to apply some of the styles we discussed in this section. Change things around and see what happens. Good luck!



To learn more...

Tools

<http://www.codepen.com>

Developer Tools in Chrome Browser:

<https://developers.google.com/web/tools/chrome-devtools/?hl=en>

Free Online Learning Resources

<http://www.w3schools.com/>

<https://www.codecademy.com/>

<https://www.coursera.org/courses?languages=en&query=programming>

<https://www.edx.org/>

<https://www.khanacademy.org/>

<https://www.freecodecamp.com/>

<http://www.html5rocks.com/en/>



Questions?