

# Introduction to HTML

Essentials of Web Design

CSCI 1210

[Print to PDF](#)

# Tools & Hypertext Markup Language

---

# Software Tools

---

How do we create websites?

There are several 'software tools' that we'll need this semester to create our web pages/sites

None of them are particularly special, but all are needed

Later in this lecture, we'll show you a website that will let you pick the applications you want (all we need, except one) and download/install them all at once

...which is nice

How do we create websites?

So what do you need to make a web page?

A text editor

That's it

For a couple of years, I used Notepad for making web pages, before I discovered Microsoft FrontPage - but that's another story

After all, as we'll soon see, an HTML page is just a text file

How do we create websites?

A text editor:









# Brackets

This semester we'll be using the Brackets editor for our coding

Developed and maintained by Adobe (but free, believe it or not)

Lots of nice features

I think of it as a 'lightweight IDE' (Integrated Development Environ

Available from  [brackets.io](https://brackets.io)



But you said we need several tools? What gives?

Well, yeah

But to make a simple page, we just need an editor

What if we want to view the page?

What if we want to add pictures?

What if we want other people to see our page?

What if we want to share our source code with others?

To view a page

We'll need a browser

As you know, a web browser is an application used to, well,  
browse the web

There are many flavors

You're likely most familiar with Internet Explorer -  
Ewwwwww! 🗑️



All of the development for this site was done using Google Chrome 👍

Much of it was tested with Firefox as well 👍

Collectively, we do not like IE 👍

What about pictures?

Though Photoshop is probably the best photo editor, we'll be using GIMP

Gnu Image Manipulation Program

Open source photo editor

Almost as good as PS (some argue better)

Plenty good enough for our needs, as we'll see



What about sharing (publishing)?

This is where it sometimes gets a little confusing for students

To 'publish' our finished pages, we have to upload them to a server account

You all (I hope!) already have accounts on the class web server

We will be using an application called FileZilla to transfer/upload our work to the server

FileZilla uses an Internet technology called the File Transfer Protocol to move files from one computer to another (more on that later)



What about sharing source code and files?

A lot of times, we want to be able to conveniently share and distribute projects

To do so, we typically create a compressed archive of the files, called a **.zip** file

A lot of your lab activities are going to start with one or several files that are partially completed

You'll download the archive to your working directory and extract (or 'unzip') the files from the archive

Probably the most popular application for this is called 7-Zip





## Ninite

Now, you could go to the websites for all of these applications, download their respective install files, and run each

That would work. But there's an easier way

There's a site named  [ninite.com](https://ninite.com)

On  [ninite.com](https://ninite.com), you can select any number of open-source applications, download them as an executable bundle, and install them all with one double-click

On  [ninite.com](https://ninite.com), find and select Chrome (if you don't already have it), FileZilla, GIMP, 7-Zip, and any other application you think you might find useful. Click 'Get Your Ninite' to download the bundle

Except Brackets. You still have to download the install file from  [brackets.io](https://brackets.io)



That's it!

So that's it for the software applications we will need this semester

You may have found some others that you'll find useful in the future

If, at the end of the semester, you decide you won't be using them anymore (I doubt it!), you can always uninstall them

There're a bunch that I use all the time



HTML

---

What is HTML?

HTML

Hypertext Markup Language

"Language of the Web"

Set of 'tags' or 'elements' designed to create the structure of a web page

HTML documents

"plain text" documents with the elements embedded for formatting

What is HTML?

Basic HTML pages require no compiling

Compiling? What the heck is 'compiling?'

Basic HTML documents' file extension is typically either `.htm` or `.html`

## .htm vs .html

So what's the difference?

Basically, just the letter 'l' 😊

The reason for the .htm extension dates back to when certain (\*cough, cough...MS-DOS\*) Operating Systems only allowed for 3 character extensions

In modern times, both .htm and .html behave, display, and act the exact same way

It is best practice to choose one or the other and stick with it

For this class – we will use .html

## History of HTML

Originally, HTML was based on a language called Standard Generalized Mark-up Language (SGML). SGML is a standard that is defined for a markup language for documents

SGML helps to define elements that indicate which portion of the document is a paragraph, sentence, bolded, etc


It is also hideously complicated

# History of HTML

HTML 1 was first released in 1991

Current accepted standard of HTML is HTML 5

The final revision and adoption of the standard was in October of 2014

The  **World Wide Web Consortium (W3C)** is the governing body that determines standards for Web technologies

## History of HTML

All current browsers (latest versions of Internet Explorer, Chrome, Opera, Firefox, Safari, etc) support HTML5



## HTML Tags

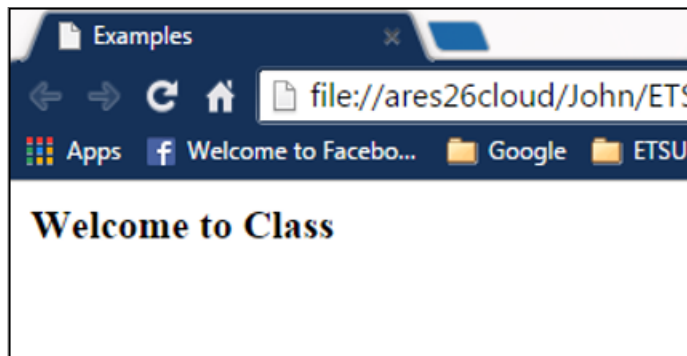
Example: Take the text, “Welcome to Class”

If, in the source document, we wrap that with the bold (<strong></strong>) tags, thus:

<strong>Welcome to Class</strong>

```
<body>  
  <strong>Welcome to Class</strong>  
</body>
```

this is what the browser displays:



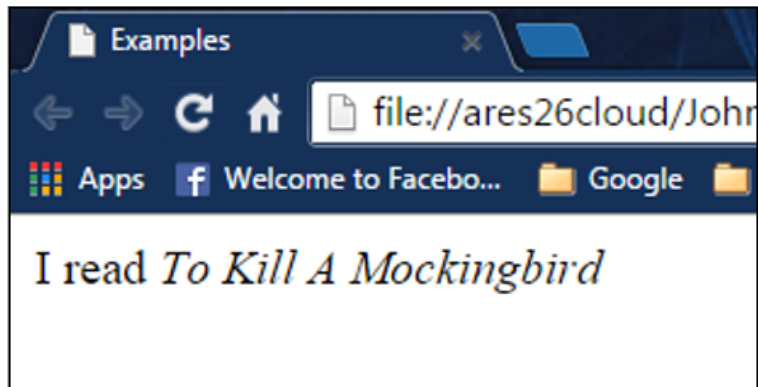


## HTML Tags

Some tags exist in pairs and are called **container** or **normal** tags

```
<body>  
  I read <em>To Kill a Mockingbird</em>  
</body>
```

the browser displays:

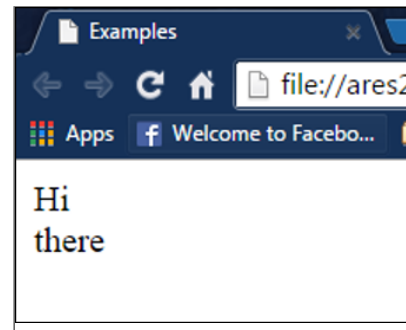


# HTML Tags

Some tags aren't paired. They're called **standalone** or **void** tags

Example: Hi <br> there (notice no ending tag)

```
<body>  
  Hi <br> there  
</body>
```



## HTML Tags

`<strong>` vs. `<b>`; `<em>` vs. `<i>`

Appearance-wise, `<strong>` and `<b>` make text bold; `<em>` and `<i>` italicize text

The difference lies in adaptive technology

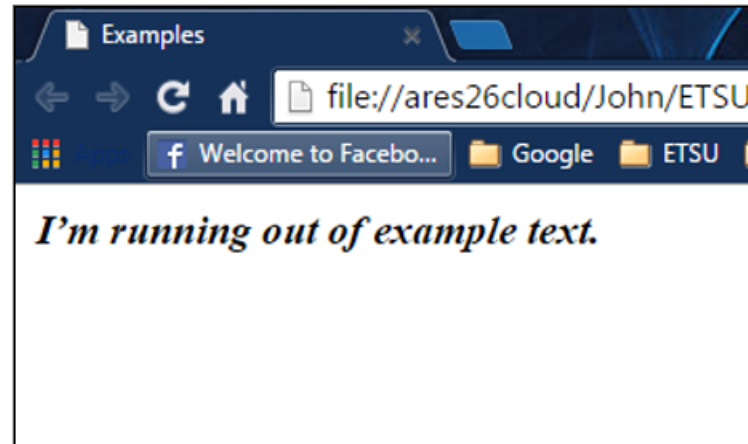
`<strong>` and `<em>` modify the playback voices for screen readers

Sometimes, you may just want to modify text's appearance without the emphasis on the reader's 'voice'

When we talk about CSS, we'll learn that we can (should) use CSS when all we want to do is modify the appearance of text content

## Nesting Tags

```
<body>  
  <strong><em>I'm running out of example text.</em></strong>  
</body>
```



## Nesting Tags

last on, first off: The last tag open should be the first tag closed

Incorrect:

```
<body>  
  <strong>this is bold<em> this is bold italic</strong></em>  
</body>
```

Correct:

```
<body>  
  <strong>this is bold<em> this is bold italic</em></strong>  
</body>
```

## HTML Tags

While HTML5 tags are not case sensitive (i.e. `<HTML>` is the same as `<html>` and `<hTmL>`), conventional and best practice states that HTML tags and attributes should be in lowercase

This is a surprisingly big issue in this class - you have to unlearn some old typing habits

Using all-lowercase is one of them

# Attributes

Many HTML tags include **attributes**

Attributes provide additional information about the tag to the browser

Attributes are included inside of the opening tag

With each attribute, you can assign a value with the equals sign followed by the value in quotation marks


Some attributes are required for the element to work; some are optional

## Attribute Examples

```
<body>
  <tag attribute="value" attribute2="value2">...</tag> <!-- syntax -->

  Optional:
  <p class="md" id="p1">...</p> <!-- the classes and ids aren't necessary
    for the paragraph to display -->
  <span class="mono">...</span>

  Required:
  <a href="http://csci1210.com">CSCI 1210</a> <!-- a link has to have an address (href)
-->
  <img src='myImage.jpg' alt='My Image'> <!-- the browser has to know which
    image (src) to display -->
</body>
```

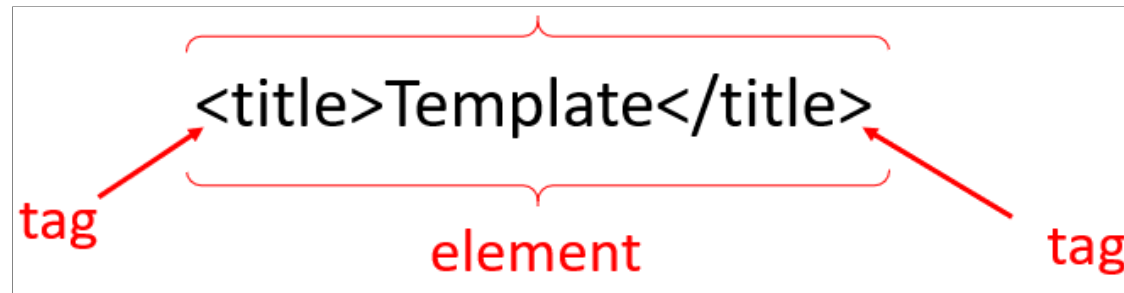
 Note: 'href' means 'hypertext reference,' a fancy way of saying 'address.'  
'src' means 'source'



## HTML: Tags or Elements?

Sometimes people will use the terms “tag” and “element” interchangeably

More correct: an element consists of both tags and the content between them



# HTML 5 Document Structure

DOCTYPE declaration – must be the first thing on the page! The <!DOCTYPE> declaration is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title></title>
6  </head>
7  <body>
8      CONTENT GOES HERE
9  </body>
10 </html>
```

## HTML 5 Document Structure

To begin an HTML document we start with the `<!DOCTYPE html>` declaration

This declaration is not a tag. Rather it is an indication to the browser that this document is a specific version of HTML

In this case `<!DOCTYPE html>` signifies that this is an HTML5 document

HTML 5: <!DOCTYPE html>

XHTML (the old standard):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

As you can see, HTML 5 (in many ways) was a huge leap forward in developer-friendliness

# HTML 5 Document Structure

The `<html>` container open tag. The `lang="en"` attribute/value pair tells browsers loading this page that the language on this page is English.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title></title>
6  </head>
7  <body>
8      CONTENT GOES HERE
9  </body>
10 </html>
```

After the Document Type Declaration, we enter in our first element – the `<html>...</html>` element

This element is used to signify to the browser that everything between the opening and closing tags is contained within the HTML element

I think of it as the “html container”

# HTML 5 Document Structure

The <head> tag. Interestingly, the HTML5 specification now makes this tag optional. Since the vast majority of existing web pages use a head section, in this class, we will also

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title></title>
6  </head>
7  <body>
8      CONTENT GOES HERE
9  </body>
10 </html>
```

## The Head Element

Inside of the `<html>...</html>` element there are two sections

The first section is `<head>...</head>`

Identifies descriptors for the document

This can include scripts, stylesheets, meta information, and the title of the document



# HTML 5 Document Structure

Meta element(s). Contain information about the document, but don't display when the page loads

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title></title>
6  </head>
7  <body>
8      CONTENT GOES HERE
9  </body>
10 </html>
```

## HTML 5 Document Structure

<meta> elements: provide metadata about the HTML document

'meta' information is information describing the document that is embedded inside the document

Does not display on the page, but can be parsed by other applications, for example, search engines

## HTML 5 Document Structure

```
<body>  
  <head>  
    <meta charset='utf-8'>  
    <title>Document Title</title>  
  </head>  
</body>
```

In the above example, meta is using the attribute `charset` with the value `utf-8`

This is the character encoding of the document

UTF-8 is the web standard (utf-8 == Universal Character Set + Transformation Format—8-byte.  
1,112,064 characters are defined by UTF-8)

This declaration is required for an HTML 5 document to be valid

## Other meta elements

```
<meta name='author' content='Joe Dokes'>  
<meta name='description' content='template page'>  
<meta name='keywords' content='template, starting page'>  
<meta name='last-modified' content='2020-07-18'>  
<meta http-equiv='refresh' content='30'>
```

These are called, in the computing world, **key/value** pairs

On the first line, for example, 'name' is a key and 'author' is its value

# HTML 5 Document Structure

The title element. All of the pages we create in this class will have titles.

```
1 <!DOCTYPE html>
2 ▼ <html lang="en">
3 ▼ <head>
4     <meta charset="utf-8">
5     <title></title>
6 </head>
7 ▼ <body>
8     CONTENT GOES HERE
9 </body>
10 </html>
```

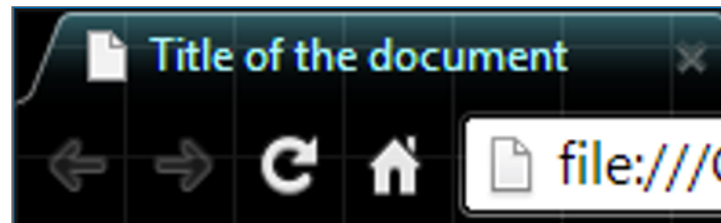
## HTML 5 Document Structure

The <title> element: title of the document

Not necessarily the file name or any titles that are used in the document

Always nested inside of the head element

Displays in the browser's tab or top bar



# HTML 5 Document Structure

Closing head tag. The head element is a container element, so it has to have a closing tag

```
1  <!DOCTYPE html>
2  ▼ <html lang="en">
3  ▼ <head>
4      <meta charset="utf-8">
5      <title></title>
6  </head>
7  ▼ <body>
8      CONTENT GOES HERE
9  </body>
10 </html>
```

# HTML 5 Document Structure

The <body> tag. This is where the displayed content will be located

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title></title>
6  </head>
7  <body>
8      CONTENT GOES HERE
9  </body>
10 </html>
```



## HTML 5 Document Structure

The second section nested in the `<html>` 'container' of the document is the `<body>...</body>` element

This section is where various elements, images, tables, forms, text, etc. can be placed to be displayed on the screen when the page is rendered by the browser

We'll see, moving forward, that a large number of elements can be embedded in the `<body>` element, including headings, paragraphs, images, tables, audio, video, JavaScript, and others

# Body Element

```
<body>  
  CONTENT GOES HERE TO BE DISPLAYED ON THE SCREEN  
  Can include text, pictures, video, tables, styling code, Javascript, etc.  
</body>
```

The body's content consists of nested elements

Nesting is an important concept

It is how we create the document's structure

## Body Element

The closing body and html tag. Again, `<body>` and `<html>` are container tags, and so they have to be paired with closing tags

```
1  <!DOCTYPE html>
2  ▼ <html lang="en">
3  ▼ <head>
4      <meta charset="utf-8">
5      <title></title>
6  </head>
7  ▼ <body>
8      CONTENT GOES HERE
9  </body>
10 </html>
```

## About the HTML Document

Text that is not inside of html tags is rendered as “default format” by the browser

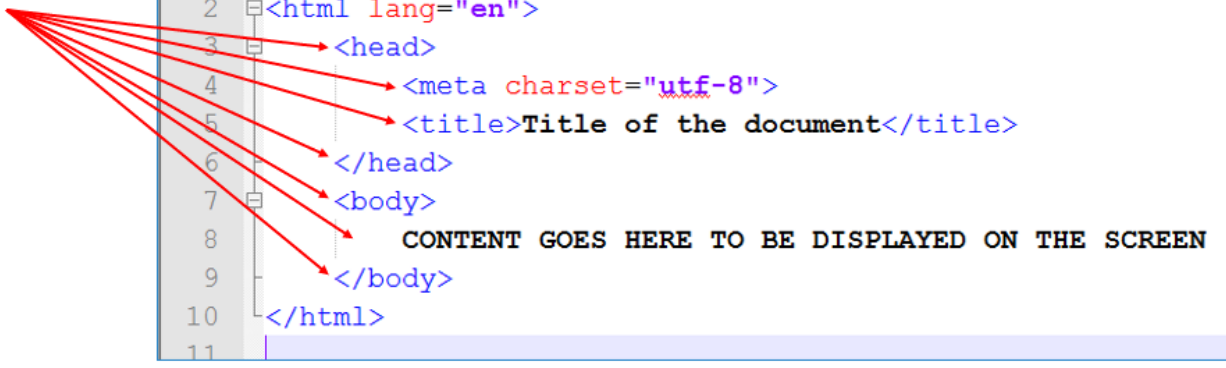
Spacing:

Multiple spaces, soft and hard returns, and tabs are ignored by the browser

Whitespace makes the code easier to read, but is ignored by browsers

## Parent / Child Relationship

Notice the indentation



```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <title>Title of the document</title>
6   </head>
7   <body>
8     CONTENT GOES HERE TO BE DISPLAYED ON THE SCREEN
9   </body>
10 </html>
11
```

The indentation of text is intentional and **required**. It helps to visually identify what is contained within tags that are the next level to the left (their parent elements)

The indentation causes the elements of the document to take the form of what is referred to as a parent/child relationship. Indented elements are the children of elements that are not indented, or are indented less

## Parent / Child Relationship

Notice the indentation

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <title>Title of the document</title>
6   </head>
7   <body>
8     CONTENT GOES HERE TO BE DISPLAYED ON THE SCREEN
9   </body>
10 </html>
11
```

Thus, in this example, the `<head>` and `<body>` elements are children of the `<html>` element, while the `<head>` element is the parent of the `<meta>` and `<title>` elements

# Comments

```
<!-- this is a comment -->
```

```
<!-- Comments - documenting code - are important for readability and code maintenance -->
```

```
<!-- Comments may be single-lined -->
```

```
<!--  
    ... or multiline  
-->
```

Comments are important

You should always code with maintenance in mind

**Someone** will eventually have to maintain/update/change the code

# Comments

```
<!-- this is a comment -->
```

```
<!-- Comments - documenting code - are important for readability and code maintenance -->
```

```
<!-- Comments may be single-lined -->
```

```
<!--  
    ... or multiline  
-->
```

Note the space between the second dash and first letter of the comment (believe it or not, it's important)



```
<!-- this is an example of a comment -->
```



# Comments

```
<!-- this is a comment -->
```

```
<!-- Comments - documenting code - are important for readability and code maintenance -->
```

```
<!-- Comments may be single-lined -->
```

```
<!--  
    ... or multiline  
-->
```

Comments begin with `<!--` and end with `-->`

## Comments

Not displayed to the screen (generally speaking)

Comments are sent to the browser

### **DO NOT STORE PERSONAL INFORMATION IN COMMENTS**

Passwords, credit card numbers, etc

Can be seen by viewing the page's source

Useful for noting design information and future support of the page

## Comments

For the purposes of this class, every document you submit should have the following comments in the <head> section of your document:

```
<!--  
  Name:           Joe Dokes  
  Course:        CSCI 1210  
  Assignment:    Lab 1  
  Due Date:      2020-05-20  
  Purpose:       The purpose of this Lab is to gain  
                 a better understanding of basic HTML  
-->
```

This isn't carved in stone - what you put into your comment block depends on the document

It's important that your documentation includes enough information to make it easier to maintain

## HTML Terms

**Tags** - define the structure of the document:

```
<a href="http://www.cs.etsu.edu">Dept of Computing</a>
```

**Elements** - includes everything from the opening tag to the closing tag:

```
<a href="http://www.cs.etsu.edu">Dept of Computing</a>
```

**Attributes** - provide additional information about the tag:

```
<a href="http://www.cs.etsu.edu">Dept of Computing</a>
```

## Next Week

Read:


pp 41-48

pp 63-72

pp 75-94

pp 95-124





JON DUCKETT

Questions?

## Lecture Quiz

1. What is HTML's responsibility in web design?

A. Document structure

B. Styling

C. Security

D. Dynamic behavior



## Lecture Quiz

2. What is the name of agency that maintains standards for web technologies?

A. ICANN

B. IEEE

C. W3C

D. CERN

## Lecture Quiz

3. Which is correct?

A. <HTML>

B. <HtmL>

C. <hTML>

D. <html>

## Lecture Quiz

4. What can be included with an element to provide additional information about it to the browser?

- A. Attribute
- B. Closing tag
- C. CSS
- D. Meta tag

## Lecture Quiz

5. Which characters are used in HTML to denote a comment?

A. `// ... //`

B. `<!-- ... -->`

C. `/* ... */`

D. `## ...`

## Lecture Quiz

6. Which element holds the part of a web page that is displayed in the browser?

A. `<!DOCTYPE html>`

B. `<container>`

C. `<head>`

D. `<body>`

## Lecture Quiz

7. What is the official character set of the web?

- A. UTF-8
- B. ASCII
- C. ANSI
- D. ISO-32

## Lecture Quiz

8. What is the current HTML standard?

- A. XHTML
- B. HTML 1
- C. HTML 5
- D. HTML 4.01

## Lecture Quiz

9. Which HTML element acts like a typewriter carriage return?

A. `<br>`

B. `\n`

C. `<p>`

D. CRLF



## Lecture Quiz

10. HTML elements cannot be nested

A. True

B. False