ASCII-based plain text are being replaced with Unicode as a universal character encoding.

**Visual Studio Code**

**[[](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/75480891520923)](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/75480891520923)**

**[Visual Studio Code Logo](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/75480891520923)**

*Available for Windows, Mac, Linux*

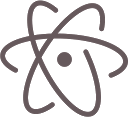
**Cost**: Free

**Style**: GUI

[**Website**](https://code.visualstudio.com/)

Visual Studio Code is a source code editor developed by Microsoft that includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring.

**Atom Editor**

**[[](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/75480891520923)](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/75480891520923)**

**[Atom Editor Logo](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/75480891520923)**

*Available for Windows, Mac, Linux*

**Cost**: Free

**Style**: GUI

[**Website**](https://atom.io/)

Atom is a free and open-source text and source code editor for macOS, Linux, and Microsoft Windows with support for plug-ins written in Node.js, and embedded Git Control, developed by GitHub.

Both Atom and Visual Studio Code are hugely popular and the majority of web developers that you'll meet will probably be using one or the other. Both feature multiple cursors and they share many of the same keyboard shortcuts. Also much like Visual Studio Code, Atom has a rich ecosystem of packages that you can use to customize your editor to your heart's content.

**Emacs**

**[[](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/75480891520923)](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/75480891520923)**

**[Emacs Logo](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/75480891520923)**

*Available for Windows, Mac, Linux*

**Cost**: Free

**Style**: Command-Line or GUI

[**Website**](https://www.gnu.org/software/emacs/) or **[Aquamacs](http://aquamacs.org/" \t "_blank)** (for OS X)

Emacs is an open source text editor that's been around since the 1970s. Along with Vim, it's one of the most popular Linux text editors.

Emacs is often described as an operating system because even in a clean install there are several included applications that you wouldn't expect inside a text editor, like a news reader, several calculators, a number of games, file encryption/decryption, and a package manager for plugins written in [**Emacs Lisp**](https://en.wikipedia.org/wiki/Emacs_Lisp).

All Emacs commands exist in the same namespace so it's not uncommon to differentiate commands by having chains of keystrokes like C-x C-f (Control - x, Control - f) to open a file. Because of the incredible customizability, it's among the editors with the steepest learning curves. You will most likely want to customize it: installing plugins, trying them, testing for conflicts, uninstalling the ones that have conflicts, and repeating.

**Vi/Vim**

**[[](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/75480891520923)](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/75480891520923)**

**[Vim Logo](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/75480891520923)**

*Available for Windows, Mac, Linux*

**Cost**: Free and open source

**Style**: Command Line or GUI

[**Website**](http://www.vim.org/index.php)

Vim, or Vi IMproved is the other text editor in the [**Unix Editor Wars**](https://en.wikipedia.org/wiki/Editor_war).

Vim runs anywhere that standard C can run and is often in the base install for most Linux and non-Windows systems including Mac OS X. It also offers a fairly robust [**tutorial**](http://www.openvim.com/) to learn how to use it. Learn it once and you can use it everywhere.

*For experienced students (everyone else, it's ok if you don't understand this right now):* Vim relies on modes, or scopes, when certain commands are applicable. In the command mode, the user can move around a file or execute commands. For instance, in insert mode, you can edit a file. While you are creating a HTML file (and are in HTML mode), you might be able to expand html:5 into the boilerplate for an empty HTML file.

## Browsers

Websites run in browsers, which means you need some browsers to run websites!

Your users could be using **any** browser to visit the websites that you build for them. There are slight differences in the way that different browsers will render your websites, which means that you'll need to test your sites in all of them.

Now, you don't need to download every browser right now, but I want to make sure that you install the two most popular cross-platform browsers: **Google Chrome** and **Mozilla Firefox**.

Other browsers, like Apple Safari and Microsoft Edge, are fantastic as well and you should continue to browse and test your sites with them. However, both are only available on a single operating system and neither have quite as robust a set of tools for developers (ie. "Developer Tools" or "DevTools") as Chrome and Firefox.

In most of the lessons on Udacity, you'll see us using Chrome. Chrome has an amazingly powerful set of Developer Tools that you'll be using to learn more about your websites than you can imagine. You'll learn more about Developer Tools in later lessons.

(That's not to disparage Firefox, which also has an awesome set of Developer Tools.)

Right now, I want you to install Chrome and Firefox if you haven't already.

[**Install Chrome**](https://www.google.com/chrome/browser/desktop/)

[**Install Firefox**](https://www.mozilla.org/)

Don't worry, you won't be required to use this workflow for every quiz right away. We will often provide you with the Udacity in-classroom code editor for many of the quizzes you'll see in this and following lessons. However, it's worth getting comfortable with your text editor and browser because you'll eventually be working exclusively with them.

Check out the [**MDN HTML Element Reference page**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element).

Once you're there, search for bold and emphasis. You can search text on a page using keyboard shortcuts cmd + F for Mac, or Ctrl + F for Windows!

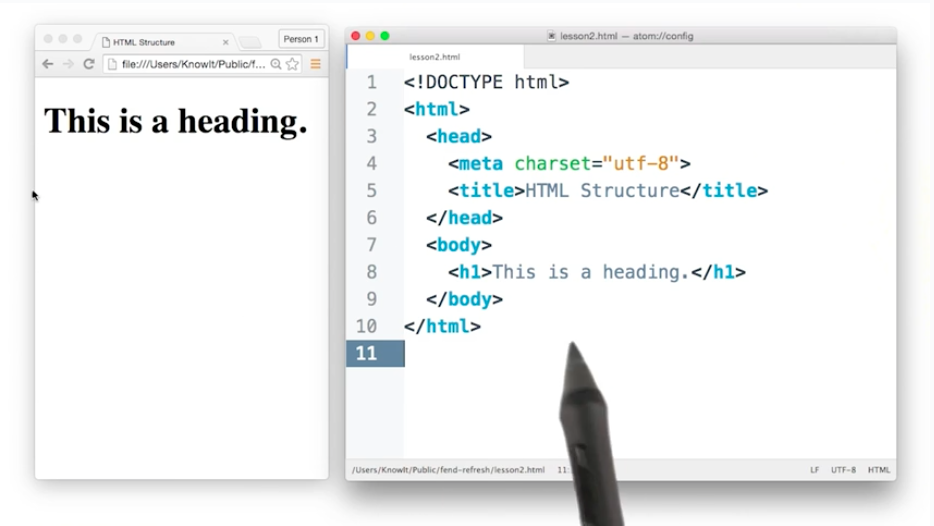
<p><strong>This text should be bold.</strong></p>

<p><i>And this text should have emphasis (italics).</i></p>

# HTML Structure Part 2

Congratulations! You created your first HTML elements, set up your development environment, streamlined your workflow, and even organized elements on a page using a standard HTML tree structure.

Now, let's take a moment for a little trip down memory lane. Remember [**this clip**](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/74229205900923) from the beginning of the lesson?

**[[](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/4a788a8e-2d58-4ba3-9469-c1728c410dea)](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/4a788a8e-2d58-4ba3-9469-c1728c410dea)**

**[Screenshot from HTML Structure video in beginning of lesson](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/4a788a8e-2d58-4ba3-9469-c1728c410dea)**

Did it seem odd that it took 10 lines of HTML to create a webpage that only displayed four words? At this point, you know that <h1>This is a heading.</h1> is responsible for displaying the heading on the page, but what about the rest of the code?

# The HTML Document

Every HTML document you create or load is derived from this basic format:

You can think of it as a template. And, following this template will help ensure that the page is displayed as the developer (you) intended. It not only says **what** should be displayed, but also includes relevant information that tells the browser **how** to display it.

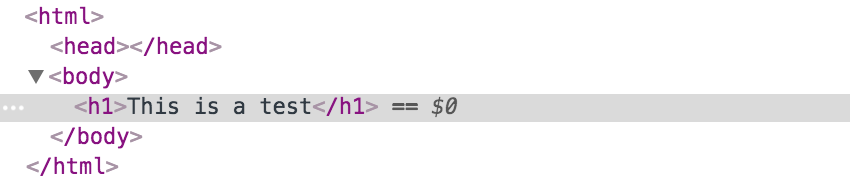
This template can be broken down into 3 parts:

1. DOCTYPE: Describes the type of HTML. While there are technically different types, for 99.999% of the HTML you'll write, you’ll likely be fine with <!DOCTYPE html>.
2. <head>: Describes meta information about the site, such as the title, and provides links to scripts and stylesheets the site needs to render and behave correctly.
3. <body>: Describes the actual content of the site that users will see.

Omitting some of this information doesn't necessarily mean that the page won't be displayed. In fact, your browser will assume certain parts of the template exist even if you accidentally leave them out. Take this line of HTML for example:

<h1>This is a heading</h1>

If you create an HTML file with only this line, open the file in any modern browser, and inspect the page with developer tools, you’ll see that certain parts of the basic HTML document format were assumed:

**[[](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/4a788a8e-2d58-4ba3-9469-c1728c410dea)](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/4a788a8e-2d58-4ba3-9469-c1728c410dea)**

**[The view from the Elements panel in Developer Tools when you omit all but <h1>...</h1>. (Notice that an empty <head> has been created for you.)](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/4a788a8e-2d58-4ba3-9469-c1728c410dea)**

That being said, this is not guaranteed behavior. Older browsers can be unpredictable, and you won’t know what browser your visitors will decide to use. It’s good practice to include all the basic parts of the template so that you aren’t relying on the guesswork of browsers to display your sites correctly.

### HTML Doctypes

An HTML document will usually start with a type declaration (which is not a tag, so it should not have a closing tag). The declaration helps the browser determine what type of HTML document it’s trying to parse and display.

If you’ve ever looked at an [**older website**](http://www.3riversstadium.com/index2.html) using dev tools, you might have noticed a doctype that looks like this:

**<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">**

(Triggers Standards mode but specifies an older form of validation.)

Or maybe you didn’t see a doctype at all?

<html>

…

</html>

(Triggers “Quirks” mode. This is bad.)

But newer websites (and your websites!) will have a declaration that looks like this:

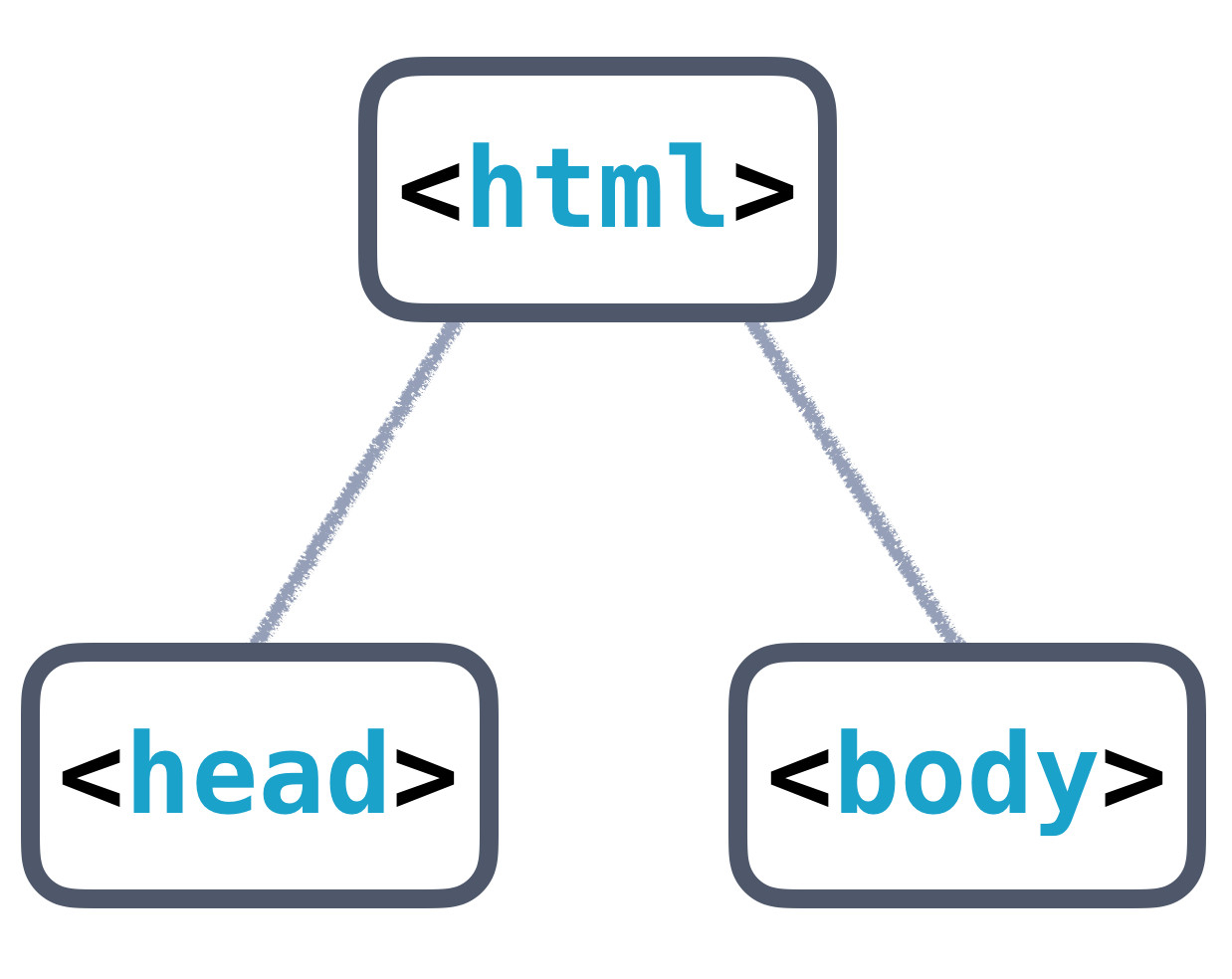
**<!DOCTYPE html>**

(Triggers Standards mode with all updated features.) 😊

Browsers look for this doctype declaration to determine which **rendering mode** to use to render the site. Generally, newer sites follow standard HTML specifications. The current standard HTML specification is called HTML5 (which is what you're learning!). On the other hand, older sites, created before HTML standards really existed, might use a different rendering mode that imitates the behavior of older browsers.

If you are interested in reading more about doctype declarations and different rendering modes, you can read about them [**here**](https://developer.mozilla.org/en-US/docs/Quirks_Mode_and_Standards_Mode).

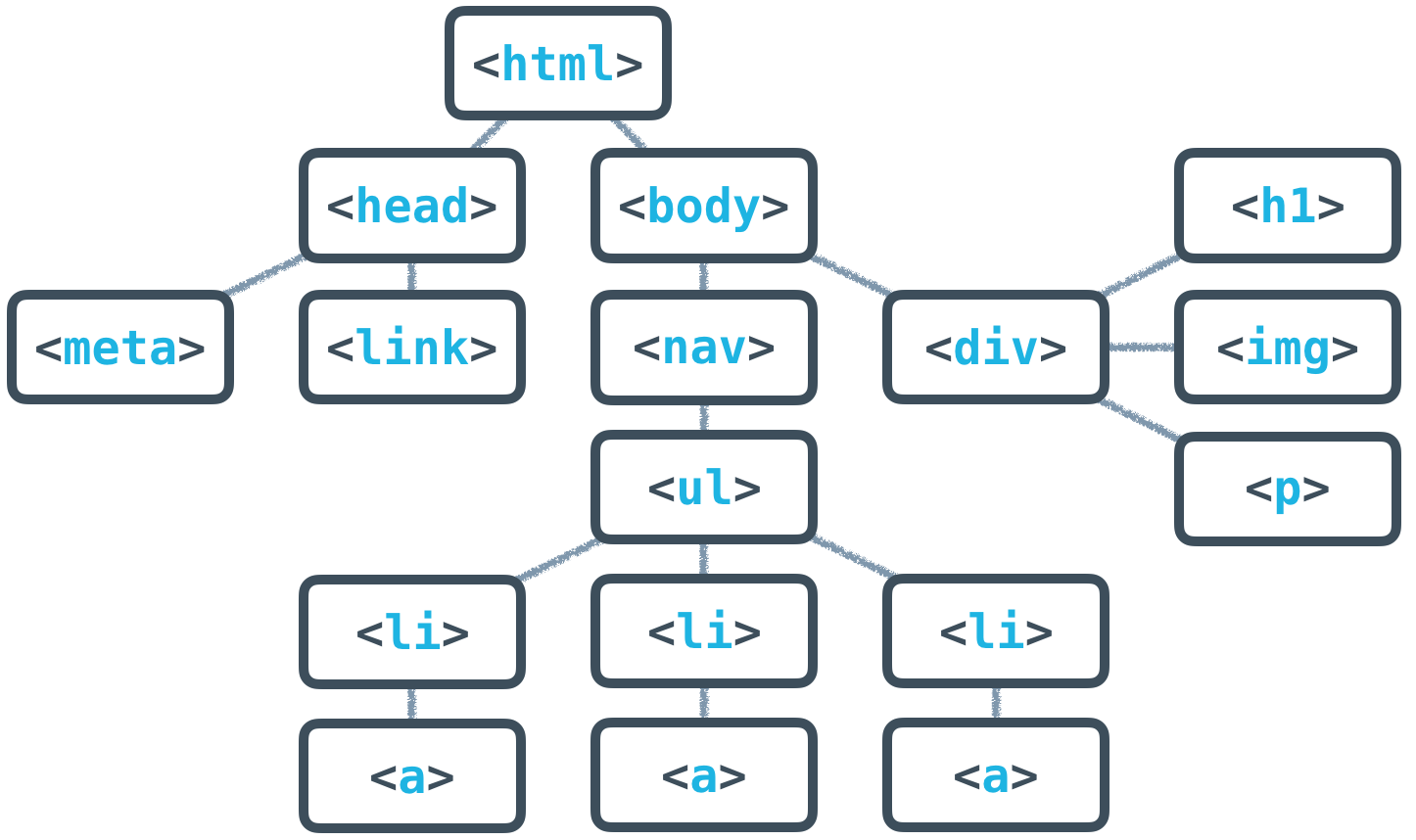
Once you’ve declared the doctype, the next part of your HTML document is the <html> tag, which tells the browser that everything enclosed inside the <html> ... </html> should be parsed as HTML. Then you have the two main sections of your HTML document: <head> and <body>

**[[](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/5dbe066a-c5a4-47e2-bc1b-9e56f2e05b7e)](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/5dbe066a-c5a4-47e2-bc1b-9e56f2e05b7e)**

**[Basic HTML Tree Structure](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/5dbe066a-c5a4-47e2-bc1b-9e56f2e05b7e)**

### <head> and <body>

The <head> will contain general information and metadata about the page, while the <body> will contain the content that will be displayed on the page. Here’s an example tree structure for a full HTML document:

**[[](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/5dbe066a-c5a4-47e2-bc1b-9e56f2e05b7e)](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/5dbe066a-c5a4-47e2-bc1b-9e56f2e05b7e)**

**[Full HTML Tree Structure](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/5dbe066a-c5a4-47e2-bc1b-9e56f2e05b7e)**

All of the HTML syntax that you’ve learned in this lesson will help you create the **content** of the page, which is always contained inside the <body> tags. The <body> is always visible.

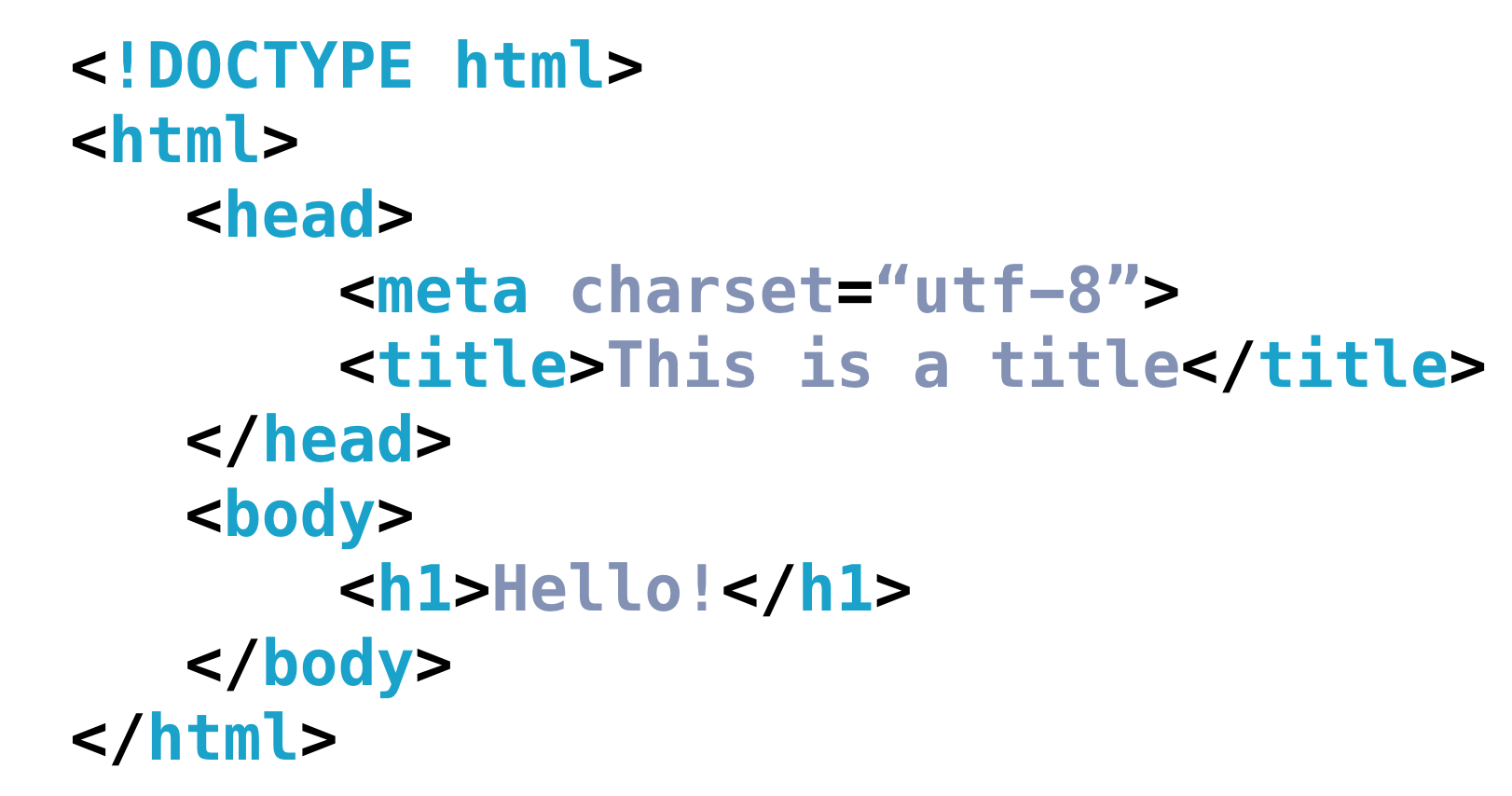
The <head>, on the other hand, is never visible, but the information in it describes the page and links to other files the browser needs to render the website correctly. For instance, the <head> is responsible for:

* the document’s title (the text that shows up in browser tabs): <title>About Me</title>.
* associated CSS files (for style): <link rel="stylesheet" type="text/css" href="style.css">.
* associated JavaScript files (multipurpose scripts to change rendering and behavior): <script src="animations.js"></script>.
* the charset being used (the text's [**encoding**](https://en.wikipedia.org/wiki/Character_encoding)): <meta charset="utf-8">.
* keywords, authors, and descriptions (often useful for [**SEO**](https://en.wikipedia.org/wiki/Search_engine_optimization)): <meta name="description" content="This is what my website is all about!">.
* … and more!

At this point, just focus on these two tags:

* <title>About Me</title>
* <meta charset="utf-8">

<meta charset="utf-8"> is pretty standard, and will allow your website to display any [**Unicode character**](http://unicode-table.com/en/). ([**Read more on how UTF-8 works here**](https://en.wikipedia.org/wiki/UTF-8).) <title> will define the title of the document and will be displayed in the tab of the browser window when a user visits the page.

**[[](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/5dbe066a-c5a4-47e2-bc1b-9e56f2e05b7e)](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/5dbe066a-c5a4-47e2-bc1b-9e56f2e05b7e)**

**[Full HTML Template](https://classroom.udacity.com/nanodegrees/nd001/parts/aabd87fc-6edf-4818-a911-15dfa9cabcb8/modules/4d4626ff-de39-4af7-9cf1-a26774ee022c/lessons/6987421963/concepts/5dbe066a-c5a4-47e2-bc1b-9e56f2e05b7e)**

# HTML Validators

This might seem like a lot to remember, but thankfully, there are tools out there to help you. Much like how the Udacity Feedback Extension tells you when you've met all the requirements for a particular project, [**HTML validators**](https://validator.w3.org/) analyze your website and verify that you're writing valid HTML.

I want you to try one out now!

# <https://validator.w3.org/#validate_by_input>