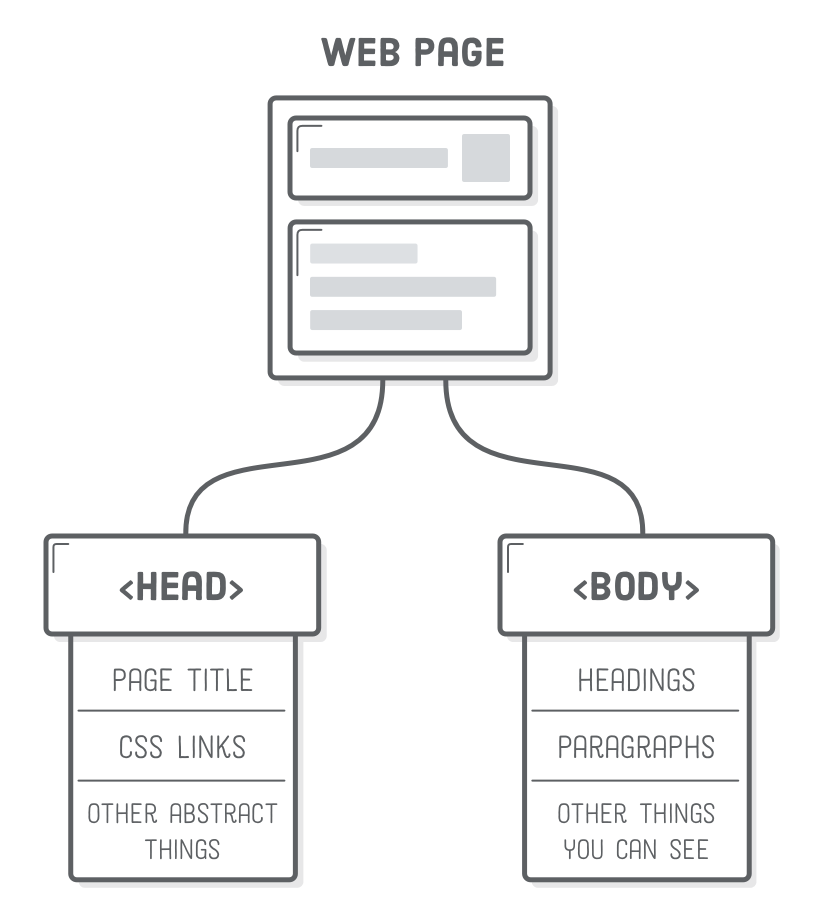
* HTML defines the content of every web page on the Internet.
* The actual <html> text is called an “opening tag”, while </html> is called a “closing tag”.
* Everything inside of these tags are considered part of the <html> “element”
* Inside of the <html> element, we have two more elements called <head> and <body>
* A web page’s head contains all of its metadata, like the page title, any CSS stylesheets
* Note that opening up our page in a web browser won’t display anything, since it has an empty <body>.



* Anything that starts with <!-- and ends with --> will be completely ignored by the browser.
* One of the most important pieces of metadata is the title of your web page, Browsers display this in the tab for your page, and Google displays it in search engine results.

**Title of your web page**

<!doctype html>

<<!DOCTYPE html>

<html>

<head>

<title>HTML is easy</title>

</head>

<body>

<!--- content--->

</body>

</html>

**Paragraph of your web page**

<!DOCTYPE html>

<html>

<head>

<title>HTML is easy</title>

</head>

<body>

<p>Basics of html</p>

</body>

</html>

* Headings are like titles, but they’re actually displayed on the page.
* HTML provides six levels of headings, and the corresponding elements are: <h1>, <h2>, <h3>, … , <h6>.
* The higher the number, the less prominent the heading.

**Headings of your web page**

<body>

<h1>Interneting Is Easy!</h1>

<p>First, we need to learn some basic HTML.</p>

</body>

* less important headings in smaller fonts

<!DOCTYPE html>

<html>

<head>

<title>Interneting Is Easy!</title>

</head>

<body>

<h1>Interneting Is Easy!</h1>

<p>First, we need to learn some basic HTML.</p>

<h2>Headings</h2>

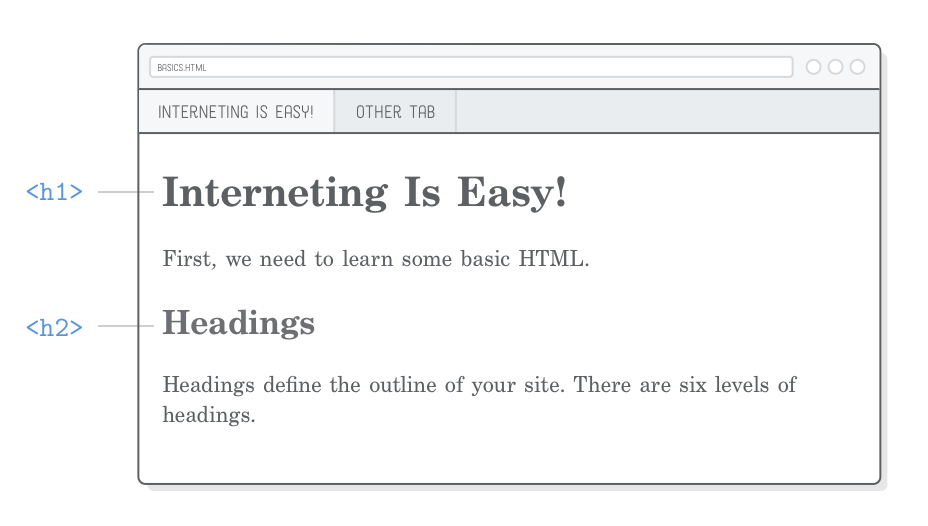
<p>Headings define the outline of your site. There are six levels of

headings.</p>

</body>

</html>

This should result in a web page that looks something like this:



* Headings are the primary way you mark up different sections of your content.
* They define the outline of your web page as both humans and search engines see it.

**Unordered list**

Wrapping content in <ul> tags, individual items in that list, you wrap them in <li> tags

<h2>Lists</h2>

<p>This is how you make an unordered list:</p>

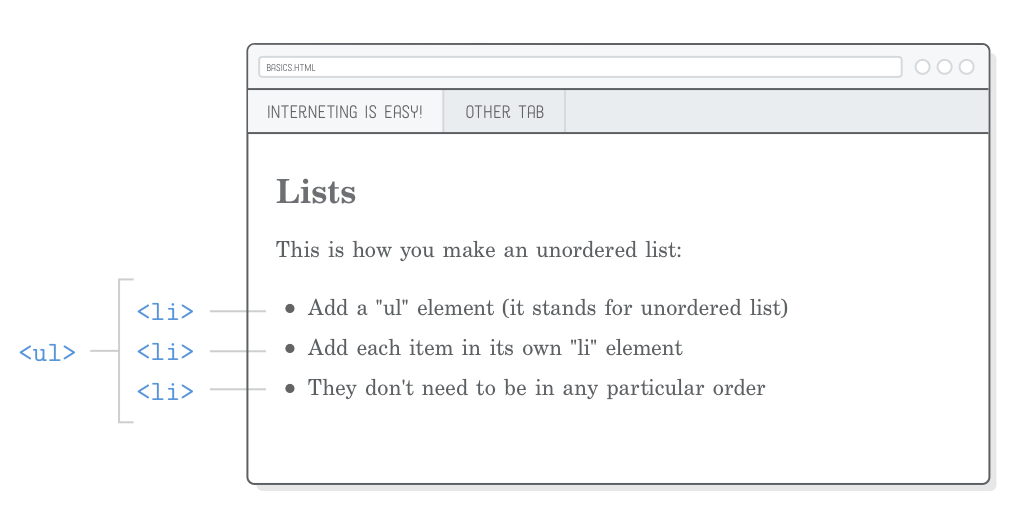
<ul>

<li>Add a "ul" element (it stands for unordered list)</li>

<li>Add each item in its own "li" element</li>

<li>They don't need to be in any particular order</li>

</ul>



**Ordered list**

p>This is what an ordered list looks like:</p>

<ol>

<li>Notice the new "ol" element wrapping everything</li>

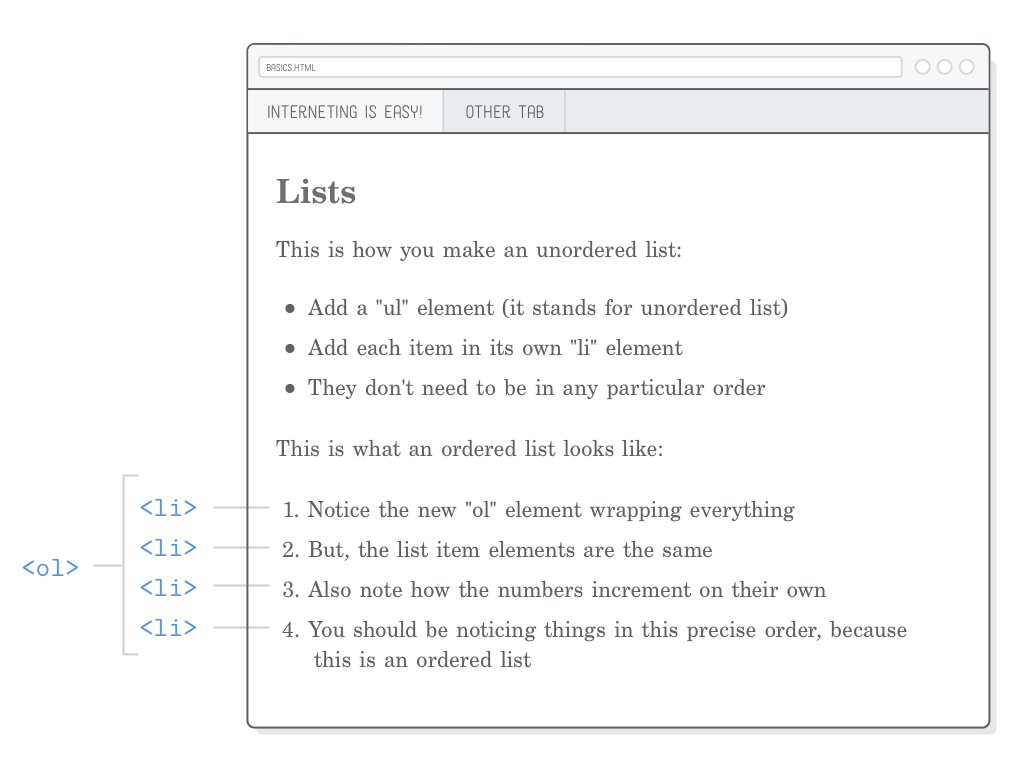
<li>But, the list item elements are the same</li>

<li>Also note how the numbers increment on their own</li>

<li>You should be noticing things is this precise order, because this is

an ordered list</li>

</ol>



Step-by-step procedures like recipes, instructions, and even tables of contents are good candidates for **ordered lists**, while <**ul**> lists are better for representing item inventories, product features, pro/con comparisons, and navigational menus.

**Emphasis**

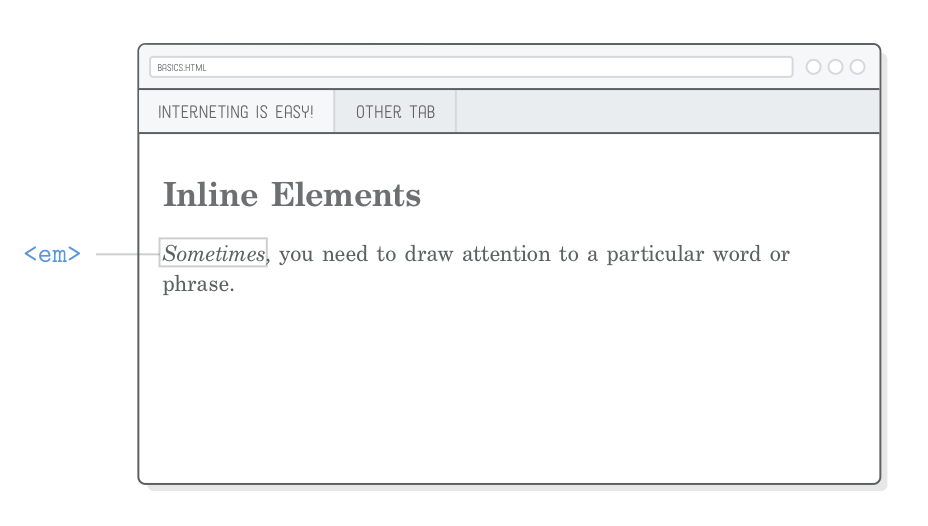
<p> is a block-level element, while <em> is an inline element.

It stands for “emphasis”, and it’s typically displayed as italicized text.

<h2>Inline Elements</h2>

<p><em>Sometimes</em>, you need to draw attention to a particular word or

phrase.</p>

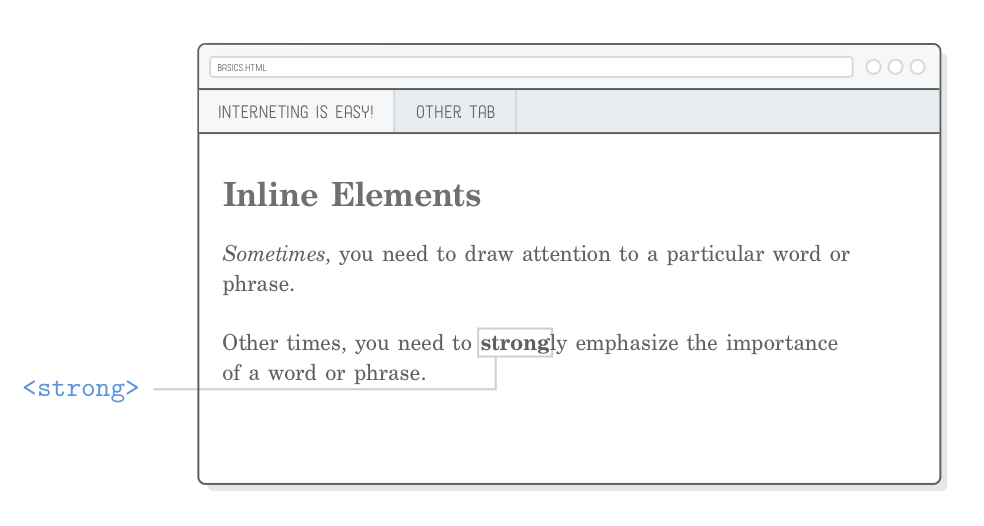


**Strong**

<strong> It’s an inline element just like <em>

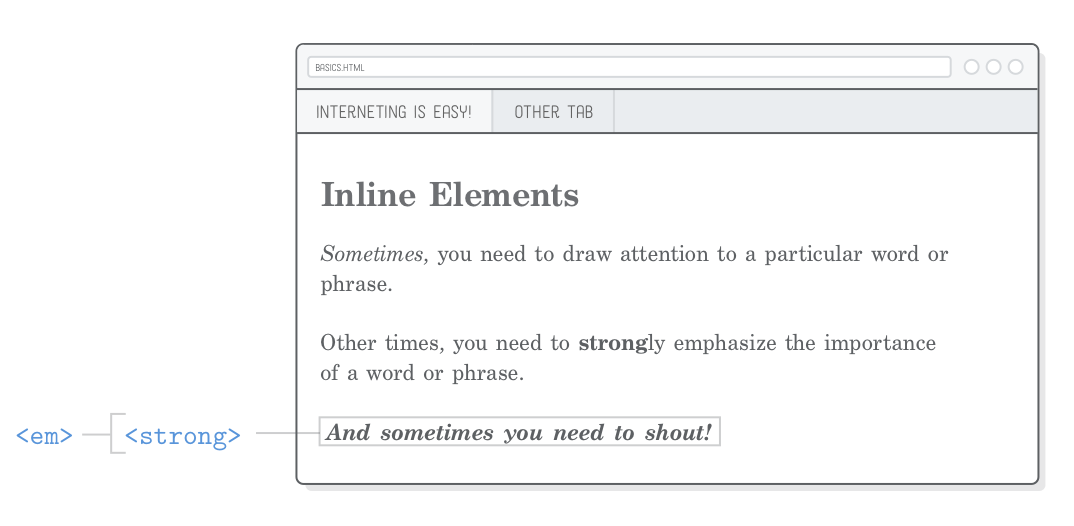
<p>Other times you need to <strong>strong</strong>ly emphasize the importance

of a word or phrase.</p>



To draw even more attention your a span of text, you can nest a <strong> element in an <em> element

<p><em><strong>And sometimes you need to shout!</strong></em></p>

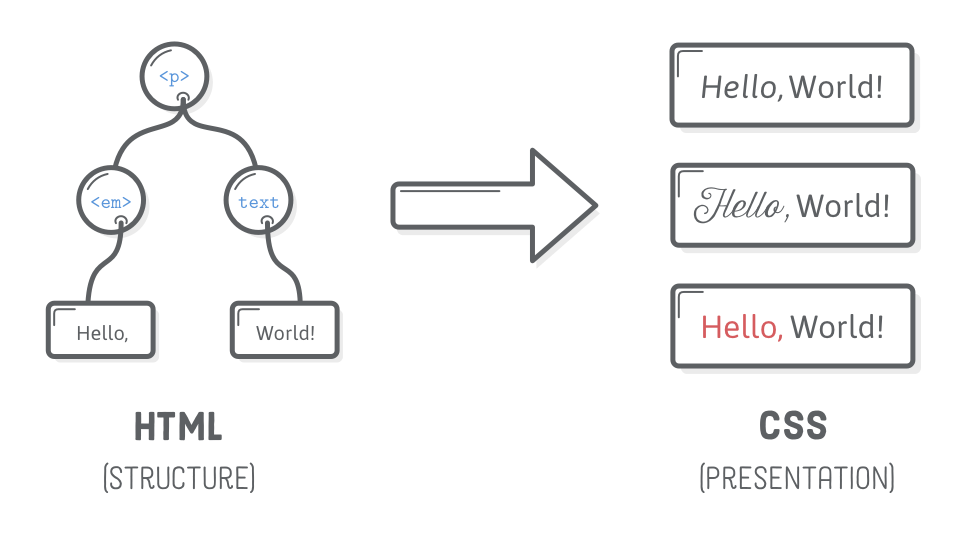


You might be wondering why we’re using the terms “emphasis” and “strong” instead of “italic” and “bold”.

That brings us to an important distinction between HTML and CSS.

HTML markup should provide semantic information about your content—not presentational information.

In other words, HTML should define the structure of your document, leaving its appearance to CSS.



**Empty elements**

Some of them can be “empty“ or “self-closing”. Line breaks and horizontal rules are the most common empty elements you’ll find.

**Horizontal rule**

The <hr/> element is a “horizontal rule”, which represents a thematic break.

<h2>Empty Elements</h2>

<p>Thanks for reading! Interneting should be getting easier now.</p>

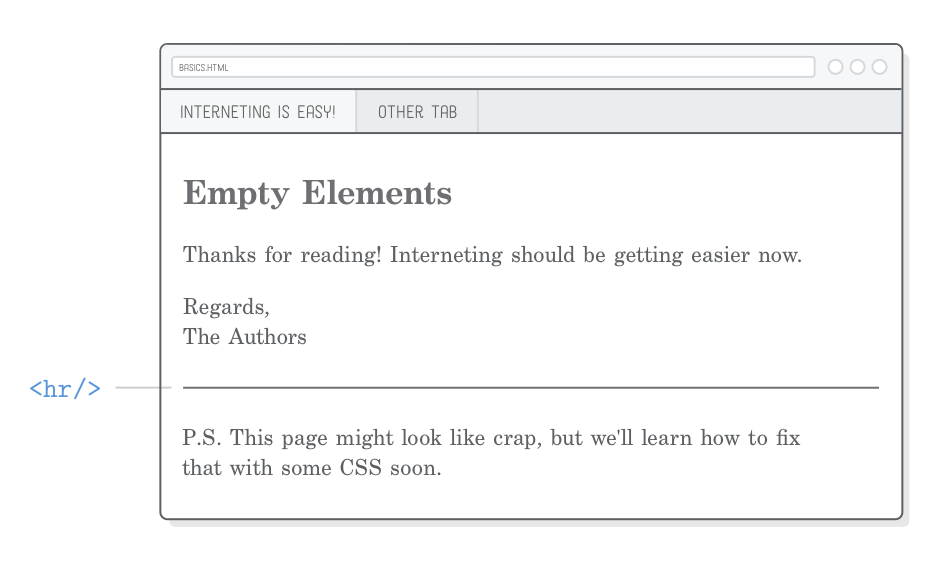
<p>Regards,<br/>

The Authors</p>

<hr/>

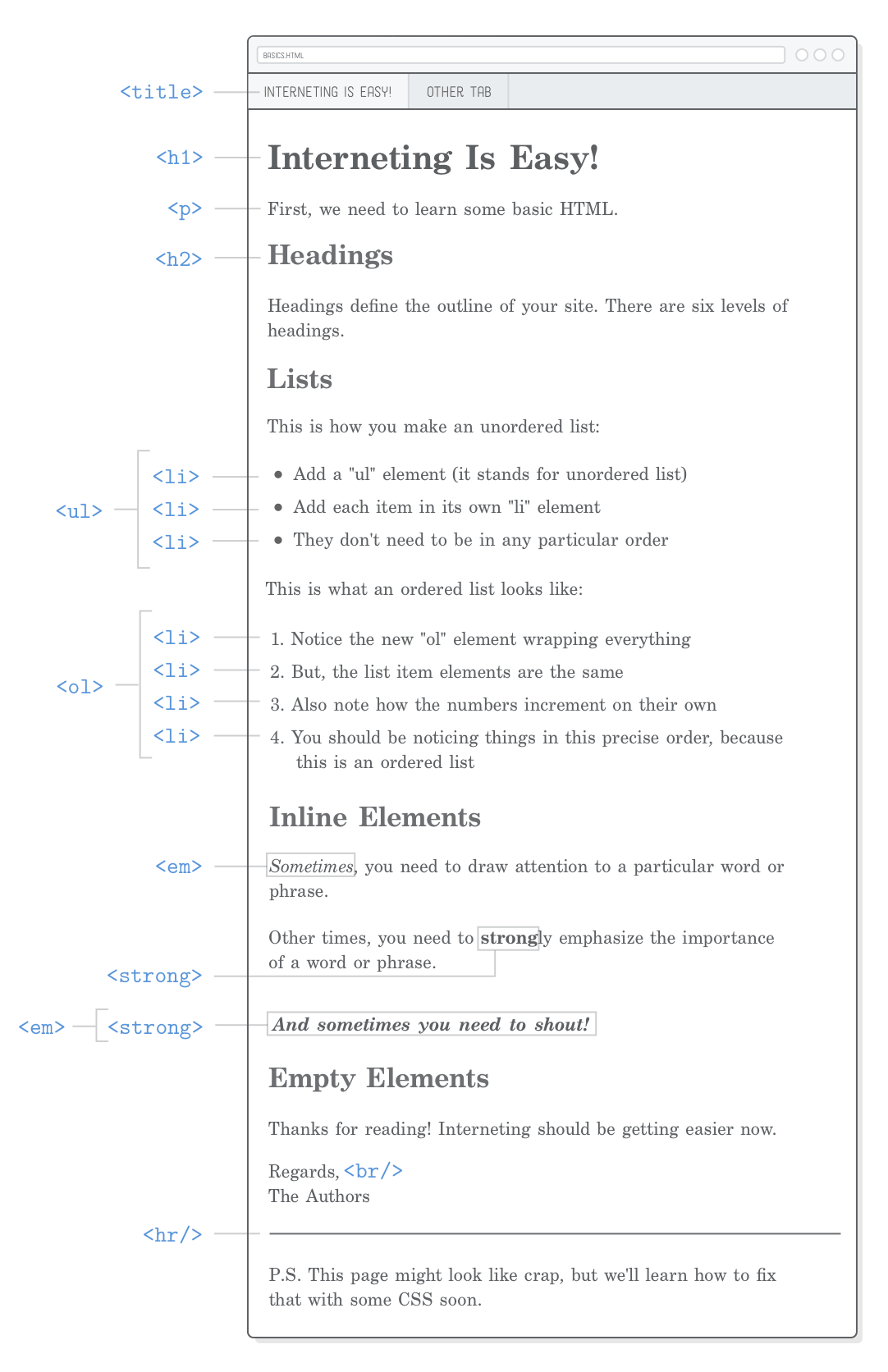
<p>P.S. This page might look like crap, but we'll fix that with some CSS

soon.</p>



Like <br/>, <hr/> should carry meaning—don’t use it when you just want to display a line for the sake of aesthetics.

For that, you’ll want to use the CSS border property, which we’ll discuss in a few chapters.



**Links and images**

Links are created with the <a> element, which stands for “anchor”.

<!DOCTYPE html>

<html>

<head>

<title>links</title>

</head>

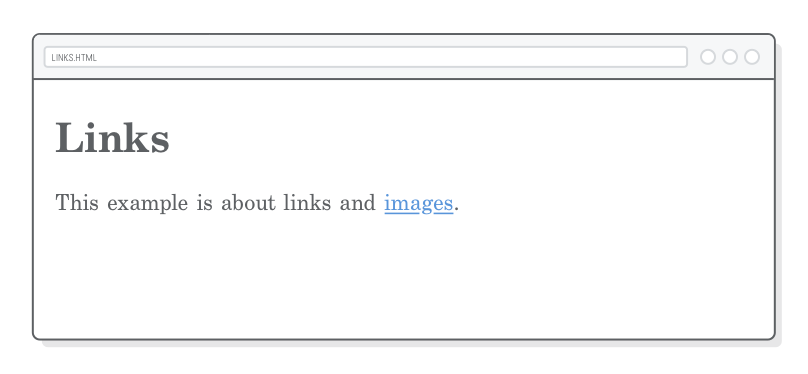
<body>

<h1>Links</h1>

<p>This example about Links and <a href="Images.html"> images </a></p>

</body>

</html>



**Absolute, relative, and root-relative links**

“Absolute” links are the most detailed way you can refer to a web resource. They start with the “scheme” (typically http:// or https://), followed by the domain name of the website, then the path of the target web page.

<li>Absolute links, like to

<a href='https://developer.mozilla.org/en-US/docs/Web/HTML'>Mozilla

Developer Network</a>, which is a very good resource for web

developers.</li>

“Relative” links point to another file in your website from the vantage point of the file you’re editing. It’s implied that the scheme and domain name are the same as the current page, so the only thing you need to supply is the path.

<li>Relative links, like to our <a href='misc/extras.html'>extras

page</a>.</li>

Each folder and file in a path is separated by a forward slash (/). So, if we were trying to get to a file that was two folders deep, we’d need a URL like this:

misc/other-folder/extras.html

#### **Parent folders**

That works for referring to files that are in the same folder or a deeper folder. What about linking to pages that are in a directory above the current file? Let’s try creating relative links to links.html and images.html from within our extras.html file. Add this to extras.html:

<p>This page is about miscellaneous HTML things, but you may

also be interested in <a href='links.html'>links</a> or

<a href='images.html'>images</a>.</p>

“Root-relative” links are similar to the previous section, but instead of being relative to the current page, they’re relative to the “root” of the entire website. For instance, if your website is hosted on our-site.com, all root-relative URLs will be relative to our-site.com.

<!DOCTYPE html>

<html>

<head>

<title>Images</title>

</head>

<body>

<p>Images <a href="links.html">links</a> and <a href="misc/Extras.html">extras</a></p>

<img src="Images/mochi.jpg" /><br>

<img src="Images/mochi.jpg" width="250"/><br>

<img src="Images/mochi.jpg" width="300" alt="images" title="images"/>

</body>

</html>

**Character set**

A “character set” is kind of like a digital alphabet for your browser.

The special characters should now render correctly. These days, UTF-8 is sort of like a universal alphabet for the Internet. Every web page you create should have this line in its <head>.

<h2>Character Sets</h2>

<p>You can use UTF-8 to count in Turkish:</p>

<ol>

<li>bir</li>

<li>iki</li>

<li>üç</li>

<li>dört</li>

<li>beş</li>

</ol>

