



How to Code
in

HTML5

and

CSS3

Damian Wielgosik

Foreword

We live in a time when websites have become part of our everyday lives, replacing newspapers and books, and offering users a whole range of new opportunities. You probably visit at least a few of your favorite places on the Web every day, whether it's for shopping, sending messages, playing games, checking the news or looking at pictures of your friends. With the help of websites, you can have fun, make a living, and even get to know other people.

So it seems that the ability to create websites would be extremely valuable. The Internet provides us with many opportunities for development, and knowledge about how the Internet is built allows us to understand the changes taking place in modern society and economy.

But have you ever wondered how to create your own website?

If you have, I invite you to take a journey with me. We will look at websites that you know and visit every day. By analyzing them, we'll use comparisons and analogies that will help you better understand how they are built. I've been a web developer and working on the web professionally for several years and I am confident that at the end of our adventure together, we will create your first website.

Thanks to [Kvba](#) for design help; [Peter Mierzejewski](#), [Paul Czerski](#) and [Greg Kaliciak](#) for proofreading. Special thanks go to Veronica and Arnold.

CHAPTER 1

Websites and Legos

I really like [The Verge](#). It's a website where you can find interesting articles about new technologies, science and culture. On the home page, you can find dozens of feature stories and articles which tend to look similar to each other. They have a title, category, date, and image entry. It almost looks like it was built with bricks.

Let take a look at one of them:

FEATURE

Back to School 2013: The Verge buying guide

Verge Staff | August 22, 2013 11:02 am



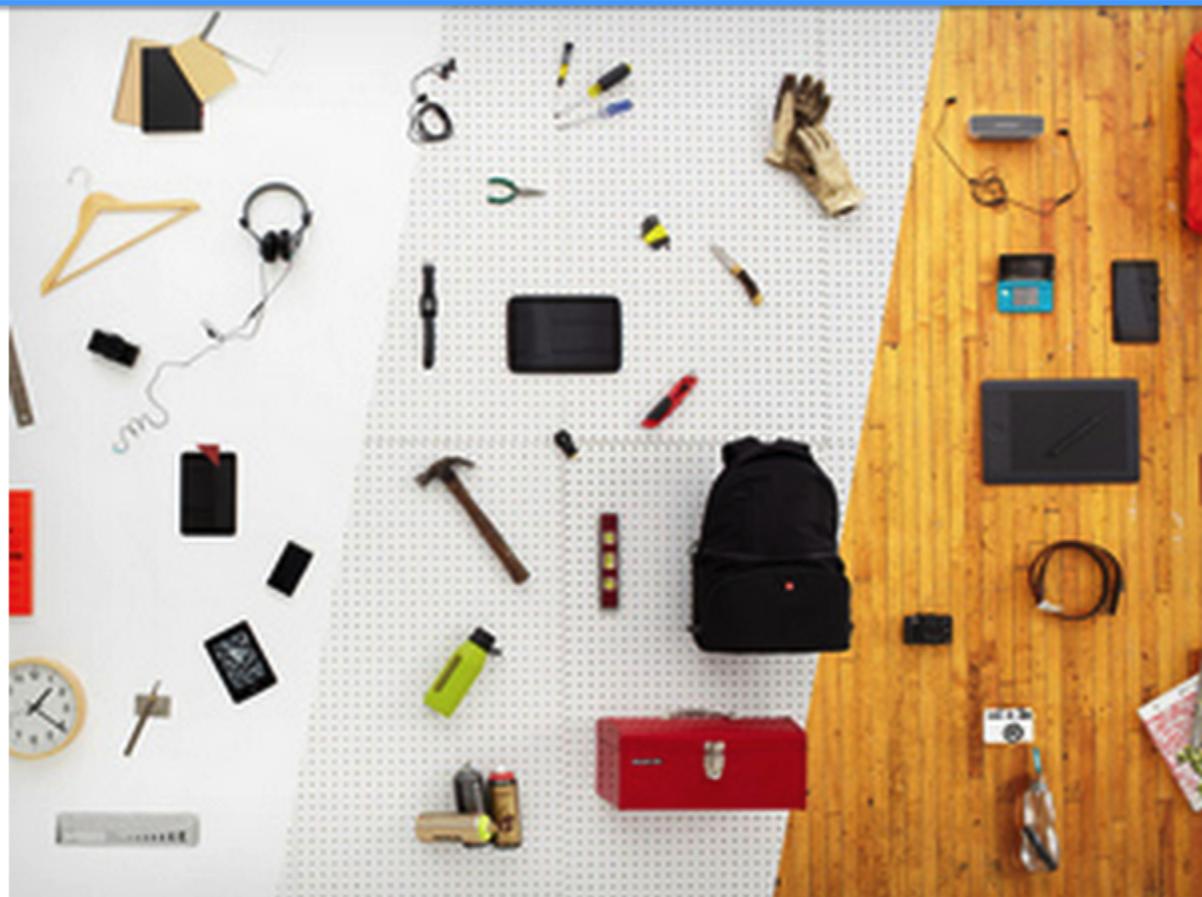
First, the bad news: summer's over. It's chillier out there every day, and soon you won't have beaches, road trips, or *kaiju* to entertain you. It's time to pull your jacket out of the closet and get to work.

Now the good news: You can make your own fun, you just need the right gear. So...

At first glance, there's nothing complicated. Let's have some fun and highlight (or block out) each element, as though we were dealing with Lego parts.

Back to School 2013: The Verge buying guide

Verge Staff | August 22, 2013 11:02 am



First, the bad news: summer's over. It's chillier out there every day, and soon you won't have beaches, road trips, or *kaiju* to entertain you. It's time to pull your jacket out of the closet and get to work.

Now the good news: You can make your own fun, you just need the right gear. So...

In total, we've got five pieces, arranged one on top of the other.

You might remember from your childhood that in order to build something specific, you need a lot of different blocks that are useful for different things. Each of them has a certain function. For example, when building a house, one type is useful for walls, and the other more useful for floors. There is no single, universal block or element from which you can make anything that comes to mind. See the figure below? This is what a large selection of lego choices should look like, right?



The same is true with websites. When you build a website, you use the different elements according to their ideal destination. In the Verge example, it seems that we

the picture. This is no different than organizing legos. We're going to keep together similar fragments of text that belong in the same group. As a formality, we'll name each of the sections based on their function in the context of the article.

Back to School 2013: The Verge buying guide

Verge Staff | August 22, 2013 11:02 am



First, the bad news: summer's over. It's chillier out there every day, and soon you won't have beaches, road trips, or *kaiju* to entertain you. It's time to pull your jacket out of the closet and get to work.

Now the good news: You can make your own fun, you just need the right gear. So...

article title

author and date

picture

paragraph

paragraph

So we've marked each element according to their semantic meaning. This is exactly the kind of behavior and logic that we can expect to see from a web browser. It's our job to tell the browser, in a way that it can understand, what each of these elements

mean and how they fit together syntactically. If this is not done, then our site will appear as a clump of single text.

You might have created such pages or articles using a text editor program like Microsoft Word or Pages. In text editors, achieving effects like "header styles" is a matter of clicking a few buttons. In other words, when we select text in a word editor and press "Header 1" we are assigning a bunch of different features in the background that tell the editor to display stuff in a specific way.

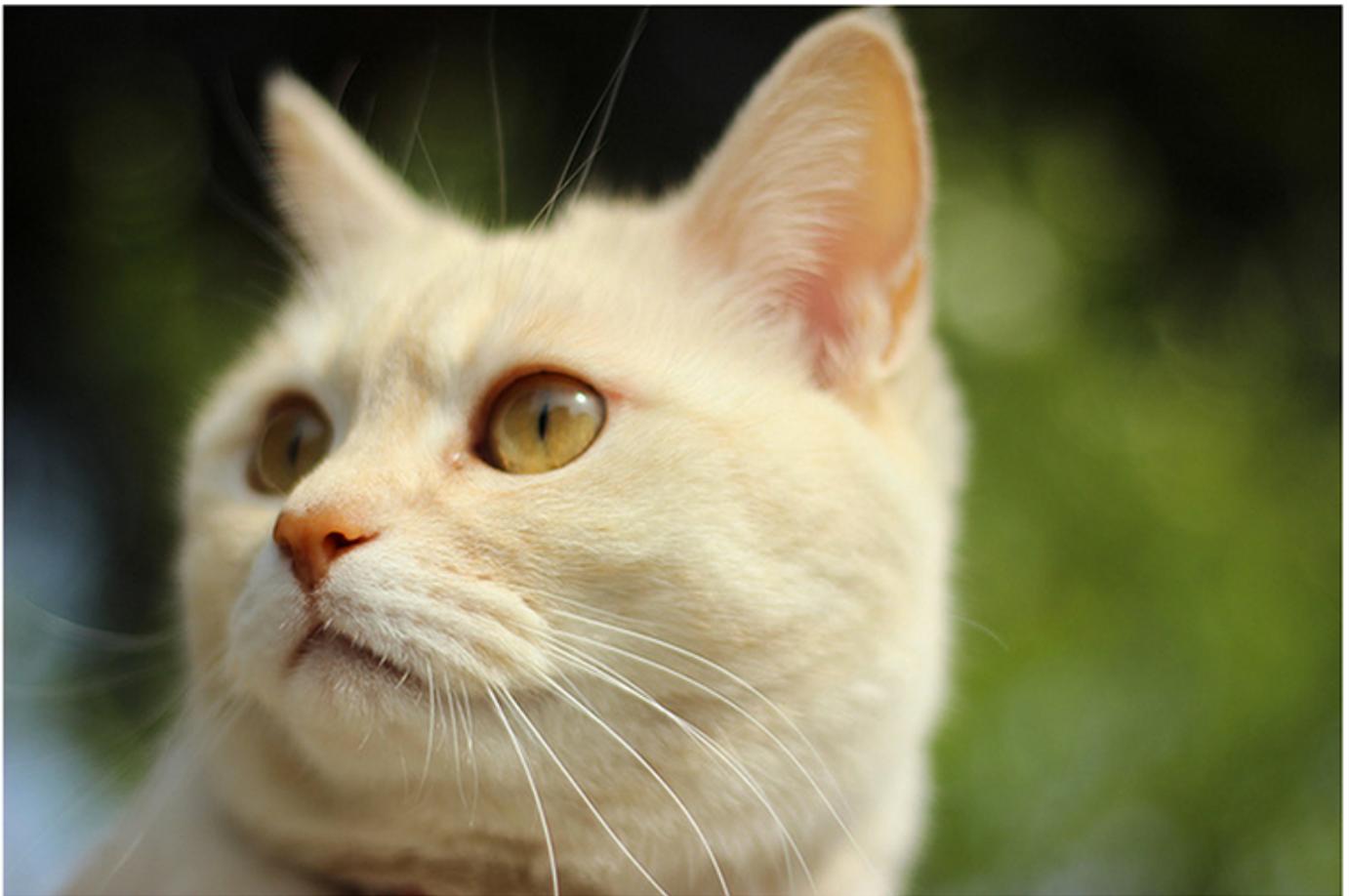
Therefore, if we wanted to recreate a page like the above example in a word processor, it would be simple and easy, at which point you could probably close this book and go do something else (Breaking Bad would be a good alternative). The problem is that we want to display this article on the web, which must be displayed in a browser, and never in a text editing program.

CHAPTER 2

Let's build our first website

Let's suppose that we want to build a webpage with a job posting. It should look something like this:

We're looking for an HTML and CSS developer



For our client, The Cat Factory, **we need a skilled web developer in HTML and CSS**. We offer a competitive salary, a bag of cat food, and toys.

Don't wait, apply now! Our crazy team is waiting for you right meow!

Before creating any kind of web page, it's a good idea to divide the content into smaller components by their importance. Let's now try to identify and highlight each element of this job posting, just as we did in the example from The Verge.

We're looking for an HTML and CSS developer



For our client, The Cat Factory, **we need a skilled web developer in HTML and CSS**. We offer a competitive salary, a bag of cat food, and toys.

Don't wait, apply now! Our crazy team is waiting for you right meow!

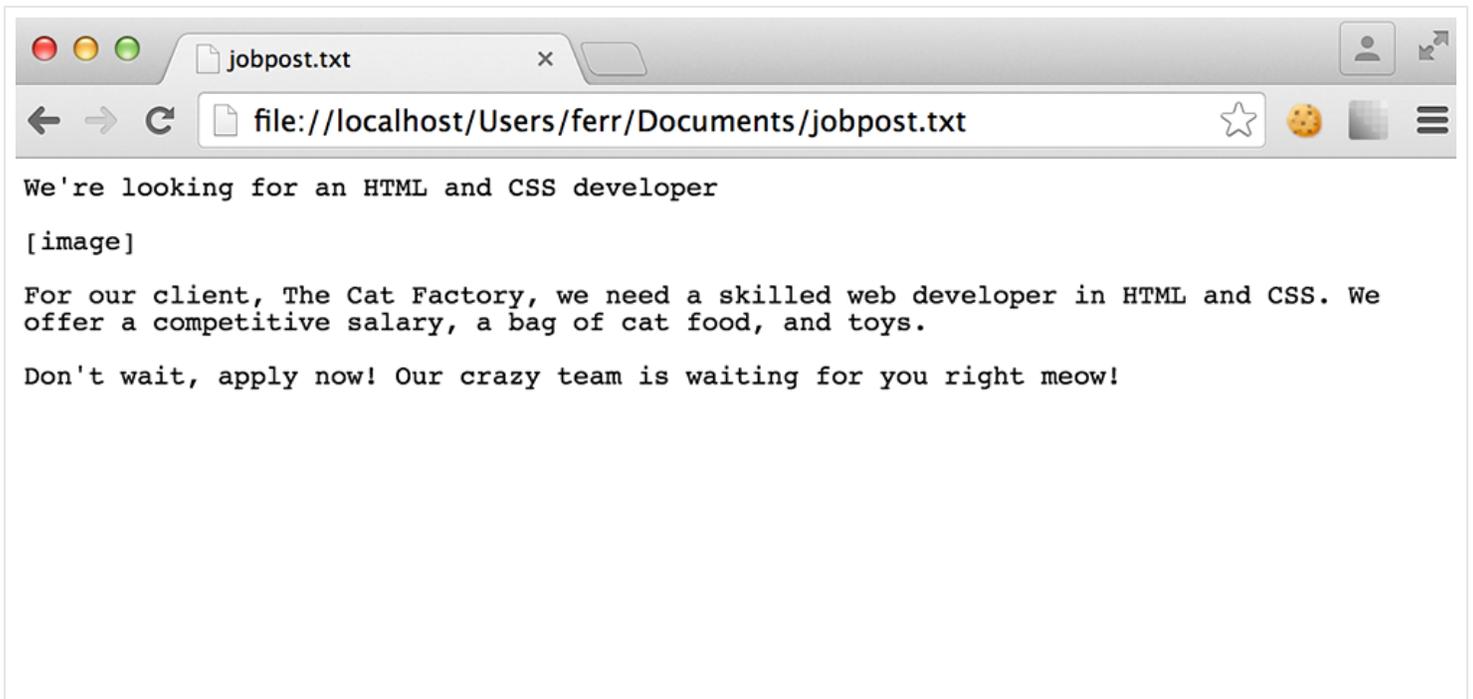
A brief analysis will help us to better understand which areas of the text should stand out in the job posting. Red indicates the headline text. Green indicates the accompanying image. And purple marks the two paragraphs (or "body") of the job posting.

Let's return for a moment to the analogy of word processors and text editing programs where web pages could be created in regular text documents. Perhaps Open, Notepad, or Word, would contain the text of the announcement, and then save as a text file. It should work right?



```
1 We're looking for an HTML and CSS developer-  
2 -  
3 [image]-  
4 -  
5 For our client, The Cat Factory, we need a skilled web developer in HTML and CSS|. We offer  
· a competitive salary, a bag of cat food, and toys.-  
6 -  
7 Don't wait, apply now! Our crazy team is waiting for you right meow!
```

If you check this in a web browser, you get the following:



```
file:///localhost/Users/ferr/Documents/jobpost.txt  
We're looking for an HTML and CSS developer  
[image]  
For our client, The Cat Factory, we need a skilled web developer in HTML and CSS. We  
offer a competitive salary, a bag of cat food, and toys.  
Don't wait, apply now! Our crazy team is waiting for you right meow!
```

However, this doesn't look like what we designed, does it? It's just a mass of text and doesn't display a picture. What now? Maybe we should try to create this post in Word or Photoshop? We don't exactly want to do this. We've made an error here that we'll soon fix. The key problem is that we've created a website with only plain text.

A web browser cannot understand how to display a page properly with only plain text — it doesn't know which part of the text should be the title or which part should be a picture. In order to display the page properly, we need to define each element by the function of the text and pass this information to the browser. We've partly done it. Let's do a little definition work and use proper syntax so browser will also understand it:

We're looking for an HTML and CSS developer



For our client, The Cat Factory, we need a skilled web developer in HTML and CSS. We offer a competitive salary, a bag of cat food, and toys.

Don't wait, apply now! Our crazy team is waiting for you right meow!

```
<h1>We're looking for an HTML and CSS developer</h1>
```

```

```

```
<p>For our client, The Cat Factory, we need a skilled web developer in HTML and CSS. We offer a competitive salary, a bag of cat food, and toys.</p>
```

```
<p>Don't wait, apply now! Our crazy team is waiting for you right meow!</p>
```

As you can see, there are special markings within the text. We'll get to these later. For now, it's best to simply try to copy them exactly as they are, from top to bottom, as if you were building with blocks, one by one.

You should get something like the following:

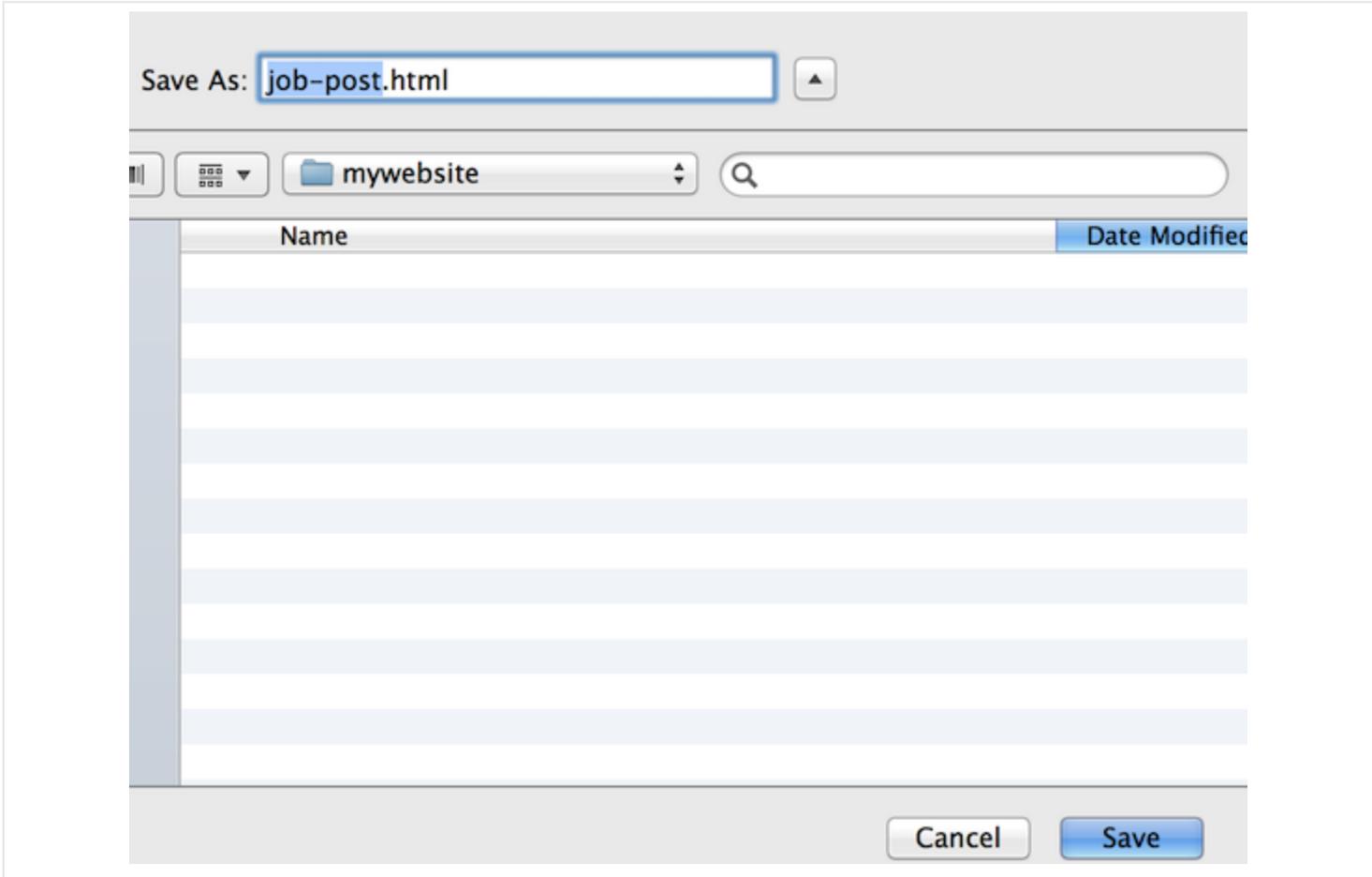
```
<h1>We're looking for an HTML and CSS developer</h1>



<p>For our client, The Cat Factory, we need a skilled web developer
in HTML and CSS. We offer a competitive salary, a bag of cat food,
and toys.</p>

<p>Don't wait, apply now! Our crazy team is waiting for you right
meow!</p>
```

Next, save the HTML file. By the way: I recommend you [Sublime Text Editor](#) which is one of the best code editors out there.



Now let's try the new file in a web browser.

We're looking for an HTML and CSS developer



For our client, The Cat Factory, we need a skilled web developer in HTML and CSS. We offer a competitive salary, a bag of cat food, and toys.

Don't wait, apply now! Our crazy team is waiting for you right meow!

It looks a bit better. Each section of the text is now formatted differently, and the picture is also displayed. This is not the final version though. The lack of color, bold text and style doesn't quite match up yet. But before we work on that, let's keep our focus on the structure of the page.

First, let's compare the plain text document version of the posting, and the one presented in HTML:

<pre>job-post.html 1 <h1>We're looking for an HTML and CSS developer</h1> 2 3 4 5 <p>For our client, The Cat Factory, we need a skilled web developer in HTML and CSS. We offer a competitive salary, a bag of cat food, and toys.</p> 6 7 <p>Don't wait, apply now! Our crazy team is waiting for you right meow!</p></pre>	<pre>We're looking for an HTML and CSS developer 1 We're looking for an HTML and CSS developer 2 3 [image] 4 5 For our client, The Cat Factory, we need a skilled web developer in HTML and CSS. We offer a competitive salary, a bag of cat food, and toys. 6 7 Don't wait, apply now! Our crazy team is waiting for you right meow!</pre>
---	---

As a result of surrounding specific text fragments with special tags, we've created a completely different, which the browser understands. Way to go!

This is called HTML, or Hypertext Markup Language, and is the primary markup language for displaying information for a web browser. In simpler terms, it's a language that uses "tags" (like <this>) to mark text, so that you can describe the text to your browser. This also includes search engines and a few other programs which we'll cover later on. HTML tags are like grammar forms in verbal communication, without them, you cannot build a coherent website, much like you cannot form correct sentences or paragraphs without grammar.

In our case, we used only a few simple tags, and below is a list of the functions that each tag performs:

- `<h1>` - headline (of the first degree)
- `` - picture element
- `<p>` - a paragraph of text

So, if you have a website with a few paragraphs of text, you surround each paragraph with the `<p>` tag. If the text is the main headline, then surround the text with `<h1>`, and so on and so forth.

All tag constructions are extremely simple and look like this:

```
<tagname>content</tagname>
```

You might have noticed that the second tag is different. This is because each tag must "close," as the saying goes in HTML jargon. The forward-slash appears in the closing tag to tell the browser that this marks the end of an element. For example, all paragraph elements end in `</p>`, while a Header 1 ends in `</h1>`.

There are also tags which do not need to be enclosed. For example image tags, ``, which are used to insert pictures. In our example above, we used it as follows:

```

```

There are several more tags that do not need to be "closed", which you will learn about later in this book.

One important realization to make about HTML tags is that, like dictionaries contain the vocabulary of a language, so too does HTML have a kind of dictionary that defines tags and describes when and where to use them. At the moment we have covered only a few, but there are many more. The international organization known as The World Wide Web Consortium (W3C) encourages best practices, standards, and compatibility across the web. You can view their definition of HTML elements here: <http://www.w3.org/TR/html-markup/elements.html>.

Also, its helpful to remember that, like natural languages, HTML evolves over time. Today, we use the fifth version of the HTML language, known as HTML5.

Returning to our example, it should look like the following in HTML code:

```
<h1>We're looking for an HTML and CSS developer</h1>
```

```

```

```
<p>For our client, The Cat Factory, we need a skilled web  
developer in HTML and CSS. We offer a competitive salary, a bag  
of cat food, and toys.</p>
```

```
<p>Don't wait, apply now! Our crazy team is waiting for you  
rightmeow!</p>
```

Notice that it contains two paragraphs of text, marked within `<p></p>` tags:

```
<p>For our client, The Cat Factory, we need a skilled web  
developer in HTML and CSS. We offer a competitive salary, a bag  
of cat food, and toys.</p>
```

```
<p>Don't wait, apply now! Our crazy team is waiting for you  
rightmeow!</p>
```

Suppose that we want the reader of our ad to pay particular attention to a certain part in a paragraph. Let's say that we wanted to mark the part of the text that says "we need a skilled web developer in HTML and CSS." How would an HTML tag let the browser know? The answer is the `` element, which will surround parts of text that should be represent strong importance for its contents.

```
<p>For our client, The Cat Factory, we need a skilled  
web developer  
in HTML and CSS</strong>. We offer a competitive salary, a bag  
of cat food,  
and toys.</p>
```

```
<p>Don't wait, apply now! Our crazy team is waiting for you  
right meow!</p>
```

Note that the `` tag surrounds the text, and also sits within the `<p>` tag. In programming terms, we would say that the `` tag is a "child" element within `<p>` because it nests within the parent. There are many tags that can nest within other tags, while many others cannot. Each tag has a specific list of possible HTML elements which it can contain (or be contained by), and you will need to check whether a certain tag is allowed or not. You can think of it like smaller blocks combining to make a bigger block piece (in this case the `` block is a component of `<p>`, which is in turn a block of the whole text).

Here is what it looks like on the web.

For our client, The Cat Factory, **we need a skilled web developer in HTML and CSS.**

Take note that this web browser displayed the fragment in bold, but this is not always a rule. It's very common that `` makes text bold, but sometimes it will be ignored. The reason for using this HTML tag in this text is a good example of how nesting works, but visual effects (like bold) can be better achieved through a language called CSS, which we will learn more about in a moment.