





[MUSIC] At the core of HTML is the HTML tag. So it's pretty important to understand

what HTML tag consists of, and how to properly, syntactically properly,

code up an HTML tag. Usually HTML tags have an opening and

a closing tag. And they surround some content. In this case, the tag p, which stands for

paragraph, is communicating to us that the content in the gray

area should be treated as a paragraph. Now technically speaking,

p by itself is called an element. And together with the angle

brackets it's called a tag. But the truth is the term is used so

interchangeably, that whenever somebody refers to an element p, or a tag p, they

really could mean one and the same thing. Now most HTML tags have a closing tag

that matches its opening tag but not all. For example, the br and hr tag,

br stands for line break, and hr stands for horizontal rule,

only have an opening tag. They don't have a closing tag at all. Now every HTML element can

have predefined attributes. So we will learn some of the most common

ones as we progress in the course, but here's what you need to know

about tag attributes in general. Attribute is a name value pair

that is kind of a meta data about the element itself that

it's being applied to. So in this example, we are assigning

myId as the value of the id attribute. Each attribute has its own rules for

the meaning of its value. So for example, id attribute,

being assigned as an example, has to be unique within the scope

of the entire HTML document. In other words, no other element of any

kind in the webpage is allowed to have its id attribute equal to the string myID. If there's another element

with the same value for id that would mean that

the webpage contains invalid HTML which can potentially break some style

lingo, even functionality of the page. Now let me go over some of

the basic spacing rules. No space is allowed to exist between

the opening bracket and the tag name. And likewise, space is not allowed

between the opening bracket and the foreword slash of the closing tag. However, you must have at least one

space between the tag itself and any of its attributes, and space is allowed

everywhere else and is simply ignored. So if you have extra space after

the p element in the opening tag, or if you have extra space between

the equal sign of the attribute name, attribute value, or you have other spaces, even return characters, all of that

is completely ignored by the browser. One more rule, attributes can only

be specified on the opening tag, so you can't specify an attribute

on the closing tag. Now let's talk about the value

of attributes for a minute. In HTML5, enclosing the value of

the attribute in quotes is technically, actually not required

in all circumstances. But never the less, it's best

practice to always surround the value of the attribute in either single or

double quotes. It doesn't matter whether you

use single or double quotes. They really are equivalent in HTML. A more interesting case arises when the value of the attribute

itself contains quotes. So the actual value has

quotes as part of its value. In this type of situation,

the only thing you have to watch out for is making sure you close the quote

in opposite order of opening them. So if the last quote was a single quote,

it must be closed first. Which quotes you start with

doesn't make any difference. You could start with double quotes,

as we have in the example here, or you could start with single quotes. You could also nest these as many times

as you want as long as you close them, obviously in the proper order, but

in practice it's kind of very rare that you have to have more

than two levels of quotes. Any more than that, and your code

just becomes messy and hard to read. Now if you've dealt with previous

versions of HTML, especially XHTML, you might be familiar with

the idea of a self-closing tag. A self-closing tag is basically

an XML type of shorthand notation for a tag that happens to not

to contain any content. For example, if we wanted to have

a placeholder in our HTML document for some other content that we'll dynamically

perhaps insert at some later point. That would be one example

of when we need to have an HTML tag that has no

content to begin with. However in HTML5,

the tag is a little bit more semantic. So if the HTML5 tag is capable, meaning it can possibly ever have

some content, it is illegal for that tag to be self closing,

even if there is no content at the moment. So instead, you have to provide and

opening and the closing tags without any space

in-between them to signify and to communicate the lack of content

as surrounded by this tag. So in summary,

we went over the anatomy of HTML tag. The opening, the closing tags, the fact that it can attributes

describing what the tag itself is. Remember that you can use double and

single quotes interchangeably. Just make sure if there's actual

quotes in the value of an attribute, you have to interchange them in

the same order that you opened them. You have to all close them

in the opposite order. And we also went over how to specify a tag

without any content inside HTML5, for which the rules are a little bit different

than for the previous versions of HTML. Next, we're going to talk about how to

create the basic HTML document structure.