[SOUND]

So what is HTML? What is the term HTML even mean and

how does ti relate to the web? In a minute we'll talk about the big

picture of three technologies that drive the web. HTML, CSS and JavaScript but

where does HTML fit? In other words, precisely what role

does HTML play in web development? As we go along, you will see that understanding the

answers to these pretty simple questions will actually help you to make correct

coding decisions down the line. Well first of all, so

what does HTML stand for? Well it stands for,

Hyper Text Markup Language. So let's go over each one of these words

and find out what exactly they mean. So first of all, hypertext. Well hypertext is text which

contains links to other texts and that's basically the entire web, right? One document points to another document

which points to a bunch of other documents and it grows on and on and on. Sometimes they link back to

the original document and it becomes one gigantic web and obviously

it's not just about text nowadays, hypermedia is plays a huge

role in the web today. You can watch videos, listen to music and really hypermedia is just

an extension of hypertext. The next term is Markup. So markup means to mark something up,

to annotate. So for example,

it's really all about content. On the Web it's all about content. So if you have some content for example if

you have a document you're writing called, Why I Love This Course, which is I'm sure is going to be

the title of your next blog post. That's the content and you want to

annotate the content to tell the browser, or to tell some other machine out there, some other program out there, what this

content is and that's what HTML does. So HTML goes and

surround that, surrounds and wraps that content in some

markup language like tags. So you can see here

Why I Love This Course is wrapped around with a title tag which

kind of tells us and some other software out there that

this is the title of this document. Which brings me to the first big point,

HTML is human readable. So these tags look like instructions for

a document structure. Structure, you don't need to run it

through some interpreter in order to understand the output

structure of this document. It's very clear what it is. The last word in hypertext markup language

is language and language basically implies that it has its own syntax meaning there's

a right and a wrong way to code it. So for instance in the example on

the left the closing div tag appears after the closing h1 tag. So it's closing order is

switched which causes parsing and potentially rendering errors. So the html is language. You could actually code it up in a way

such that something is wrong and something will break. Html also has it's own

semantics which means tag names can mean something either

to machines or to humans. Okay.

So let's talk about the three technologies that drive the web. Each one has its own

distinctive purpose and all three of them fit

very nicely together. Let's start with HTML. HTML provides the structure

which means what components does the HTML document have? For example, it can have one heading,

two paragraphs, and a footer. Note that that does not tell you anything

about how these components are visually laid out, what they look like, what

color they are, what font size they are. It only tells you what are the components. Like in a house if you have three rooms

and a kitchen you don't really know what color the kitchen is,

what color the walls of the kitchen. The only thing you know is what

components make up the house. The color and style is the role of CSS. So colors, layouts, font style, font sizes, in other words any

stylistic types of things. So for example if we have a heading in our

HTML document, it will tell us what color, what font size it is. So that's the job of CSS and the third, JavaScript, it's job is to provide

behavior, provide functionality. It adds functionality to the page, so for example what happens when HTML document

finishes loading into the browser? Or what happens if I click

on one of the headings? That's the job of JavaScript

to provide that behavior. Let's take a look at a quick example

of these concepts in action. Okay, so the example that we're going to

take a look at is located in the examples lecture zero one folder, and the file

we're going to look at is called structure dash only dash before dot HTML and

it's a simple HTML file. There's nothing but some HTML and

the dummy text data here, dummy text, and it's got a title,

it's got a heading here. Couple of paragraphs and

this has a footer, where in the footer we have

the tech support email address. For the poor soul whose going to try

to contact this technical support. Joking obviously. Okay, so, basically, nothing more than

there's just a HTML and some dummy text and we're also going to take a look at

structure-only-after.html where it's the exact same HTML document but

we inserted some styles into it and some behavior as well. So for example, the heading,

we made it green and we center aligned it. The second paragraph has

some margin applied to it. The footer here has a paragraph

with our text support email, and if we click in the text support email

is going to pop up and alert for us. Okay, so let's take a look at both

of these documents in the browser. Okay so here's the first document. There is nothing special about it. It's got that title we talked about,

a couple of paragraphs and the tech support email at

the bottom in the footer and if I click on it nothing actually happens

but if we look at document number two we see that HTML is Structure Only

heading is now green and a much bigger title and we see that the

position of it has moved to the center. And we also see that the second paragraph

is in the center and it's much bigger font compared to before and if we click on

the email address that we talked about, we see an alert gets popped up with

the message emailing us is useless, which sounds just about right for most

people's experiences with tech support. So the thing to notice here is that the

difference between the two HTML document are the up styles and

the behavior applied but the structure of both HTML

documents is exactly the same. So in summary, we spoke about HTML and the

fact that its job is to annotate content and define document structure and

obviously, as any language, it has a right and wrong syntax, which

you'll have to learn in order to code it. And the three Core Web Technologies,

HTML, CSS, and Javascript work very well together because

each one of them has its own distinct and precise role and they kind of

don't really step on each other. The work very nicely together. Next we'll talk about a bit of

a relevant history of the HTML standard, and I do mean relevant.