Links: <a> element with an href attribute, href is hypertext reference. Can be absolute or relative.

Internal links point to internal webpages, and are relative URLs, but we can have absolute also for internal pages.

If we do not provide a directory, the browser assumes that the file is in the same directory. It’s a good idea to provide a title for screen readers for visually impaired people.

The content between the <a> tag

[MUSIC] Links pretty much what

makes the web what it is. In this lecture, we're going to take

a look at different types of links and how you could create

them in your HTML page. So the first type of links that

we're going to take a look at are internal links. Here I am looking at the file

called links-internal.html, and it's located in

the examples/Lecture09 folder. And here we have a couple of links

we're going to take a look at. Now the way you create links

is by specifying an a element with an attribute href. And href stands for hypertext reference. The value of href can either be

a relative or an absolute URL. In our case here,

since we're discussing internal links which are links that point to internal

web pages of the where application, the links we're showing

are all relative URL links. However we could still have an even

absolute URL links as long as they point, again, to the same web

application that we're in. Since we're providing no

directory information, the browser will assume that

same-directory.html is a file that lives in the same directory

as links-internal.html. It's also a very good idea to always

specify a title attribute for the a tag. The title attribute is

used by the screen readers that help the visually impaired

people get through the web page. The content in between the opening and

closing a tags is the content you're going to see in the web page, when

it is served and on which you're going to be able to click to go to the href

that the a tag is configured with. Note the second example of our a tag. In this example, we're surrounding

a div tag with our anchor tag. In other words,

this div tag is going to be the content over our link on which

were going to be able to click. Let's go ahead and

see what this look like in the browser? Okay.

So, here is our internal links page, and we have a couple of links. One it says linking to a file in the same

directory, and the one is the DIV. That's the a tag surrounding

the div tag that says DIV linking to a file in the same directory. And if we click that, we're going

to go to this same-directory.html page which basically just provides

a link back to our original page. What's interesting to note

here is that the first link, linking to a file in the same directory, is clearly an inline tag since it's not

forcing itself to go to a new line. But yet in the second case, div linking

to a file in the same directory, we're surrounding the div

tag with our a tag. So what's going on here? Are we using block level element,

the div, inside an inline element? Well, turns out things

are little bit more interesting. Let's take a look at the a tag as

defined in the W3C specification. If you take a look closely,

the a tag is both a flow content and a phrasing content. In other words, to map it back to

the HTML four days, the a tag in the HTML5 is both an inline element and

a block level element at the same time. And this is what allows us to take the a

tag and surround a div tag inside of it. The authors of the HTML5 specification

realized that there are a lot of times where you would like to be

able to click on a whole region. And a most common occurrence of

that is on when you have a logo or some sort of a company name in the top

left corner for example, of your webpage. People expect to be able

to click on that and be able to go to the front

page of the company. Prior to HTML5,

people had to use all kinds of tricks in order to achieve that effect because

the a tag was only an inline tag, and you weren't really able to wrap

the a tag around the div tag. Next, let's take a look at

an example of an external link. Here we are in Sublime Text again and

looking at the file called links-external.html which is again located

in the examples/ Lecture09 folder. And this document basically

contains just a single link. There's really nothing super special about

external links other that their href value usually starts with http://

because it's usually the case that external links are hosted on

a different domain name than your website. However, there's one feature of the a

element that I would like to demonstrate that is quite often used in

conjunction with external links. And that is the target attribute. Target attribute,

when it's set to the value \_blank, forces the browser to open this

page in a new tab or a new window. The reason that's advantageous is because

nowadays unfortunately people have a very short attention span, and if you take them

from your website to a different website, it is a very good chance that they

will never come back to your site. And usually you don't want them

to leave your site completely. So target="\_blank" is very

useful in that regard. Let's a take a look at

this page in the browser. And here's the page in the browser. And if we click this link, Coursera

Facebook Page, it will open a new tab and will load the Facebook page

pointed to by that link. Coincidentally, this is a Facebook page I

created for the two courses I'm teaching on Coursera which is at

facebook.com/CourseraWebDev. So please like this page, and follow

it as I'll be posting some optional but interesting and relevant material. Another type of link that is

extremely important to know about is a fragment identifier. So here I'm in again in Sublime Text

in links-same-page.html. That is located in

examples/Lecture09 folder. And as you could see,

the links that have set up here all have a very specific

format in the href attribute. It's a # followed by some name like

section1, section2, and so on. Now what these links are pointing

to is a section of our page. Now you could identify a section

a couple of different ways. You can have in any tag that has

an id with that section name. Notice that the section name

does not contain the # sign. Only the link to that

section contains the # sign. That's one way to identify

a section within the page. Another way is, if we scroll down all

the way to the bottom of the page, is to create an anchor tag

with a name attribute and name the section very similarly

to the way you name a section id. The way you refer to these

sections is exactly the same, you put a # in front of

the name of the section and stick that value in the href

attribute of an anchor tag. So let's take a look at this

document in the browser. And here's the document. As you can see,

we have six different links. And if we click on any one of these links, it will take us to the section of the page

that is marked with that particular id. If we scroll all the way down,

we even have a back to the top link which basically points to

all the way to the top. Now, the top is right here, and

if you look at the code and scroll all the way up,

we have an h1 tag that has an id with top which allows us to scroll all

the way to the bottom of the page, see that link, and

jump right back to the top. What's really neat about fragment

identifiers is if you click on one and you have them in URL,

you could copy and paste this URL and send it to a friend as a bookmark. And when they paste it into their browser,

they will be taken to this page and jump straight to the section that was

pointed to by the fragment identifier. Now I am on my local hard drive, and

I'm not on a server so obviously sending this link that points to my local hard

drive is not going to do any good. But if we're on a server, this link with

the section identifier would have taken anybody anywhere, with a connection to

the internet, to the proper page and to the proper section of that page. While fragment identifiers are very useful

in order to jump to different parts of the same page, recently they have become

even more important as they're used for navigation within the SPA or

Single Page Applications. And SPA, Single Page Applications

have become extremely popular. So knowing how fragment identifiers

work is an important step towards being able to code single

page applications.