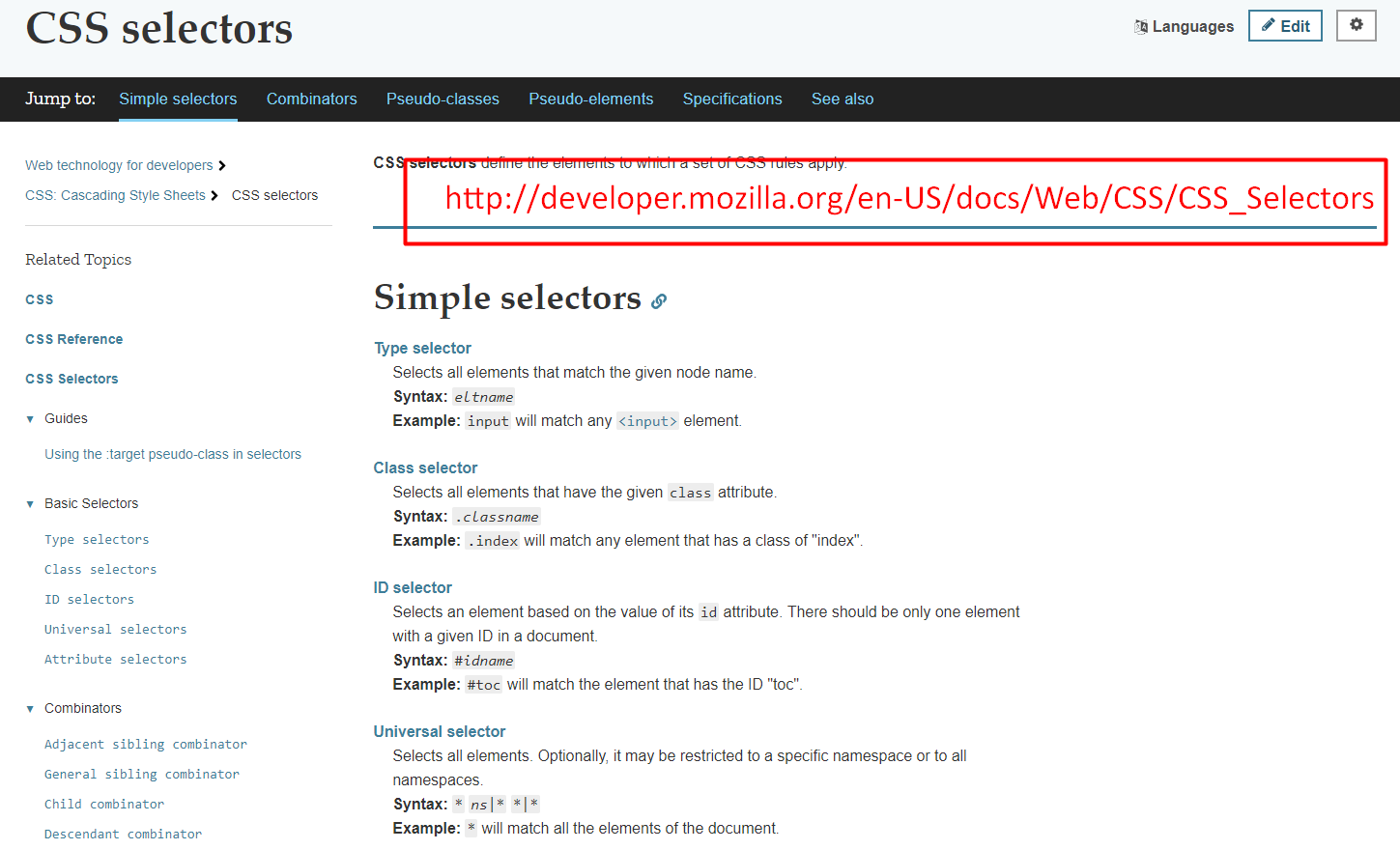
One thing we will want to understand and get better at are CSS selectors and [all the documentation is in here](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors), but we can get away with a very minimum set of selectors to be absolutely honest.



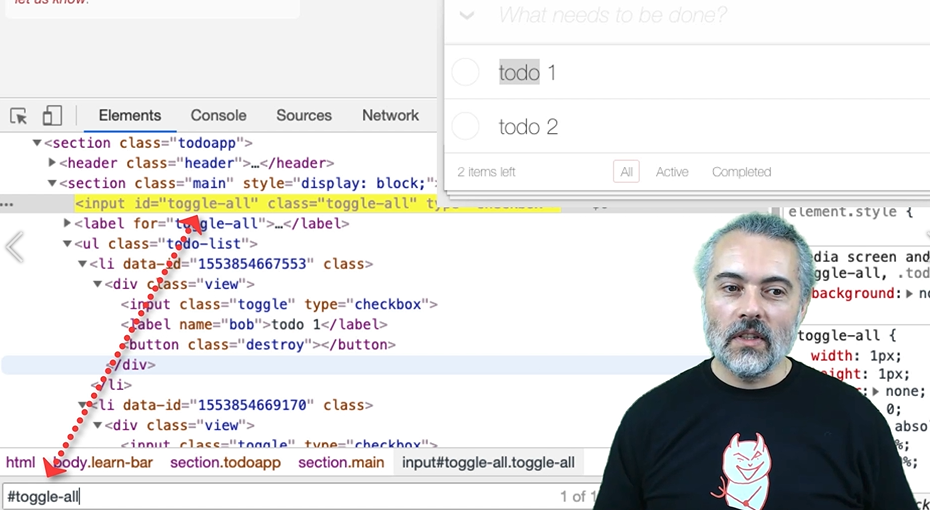
If we know how to select with a class name, we know how to select with an ID, we know how to select with a tag, which is essentially this, we type in the input name. And we know how to use the nth-child, that will probably be enough for this application.

So, let me just experiment with those CSS selectors.

[#](https://testautomationu.applitools.com/automating-in-the-browser-using-javascript/chapter3.3.html#finding-css-selectors-by-id-idname)**Finding CSS Selectors - by Id: #idname**

In the elements, if I want to find something with an Id, I just do hash [#].

So, I could do — #toggle-all — because that's got an Id.



toggle-all, it's found one thing.

I don't think anything else in this app really has an Id, but that will be useful to us, and I can find things by class.

[#](https://testautomationu.applitools.com/automating-in-the-browser-using-javascript/chapter3.3.html#finding-css-selectors-by-class-classname)**Finding CSS Selectors – by Class: .classname**

I can say, "Get me anything that has the class toggle-all." — .toggle-all

I can be more specific and say, "I only want the inputs that have got a tag with that." — inputs.toggle-all

And you can see that if I make a mistake here, it doesn't match on screen, so I'm getting the feedback that I need.

**Choosing CSS Selectors**

When I'm building code to interact with the application, I will inspect elements, then try and write CSS selectors here that match specifically the things I want to see and I will see the results in here, to make sure that when I put it in JavaScript I'm getting the right results back.

[#](https://testautomationu.applitools.com/automating-in-the-browser-using-javascript/chapter3.3.html#finding-css-selectors-by-tag-nameoftag)**Finding CSS Selectors – by Tag: name of tag**

We've seen Id, we've seen class, let's have a look at tags.

I can get all the list items —li

Now that says it's matched 13 of them. I only want to match these ones [the todos], how can I do that?

Well, I want to find an unordered list which is followed by list items — ul li

Now I've got 6, but I still only want to match 2.

So, I want to find the unordered list that has the class todo items.

What is it called? Can't remember. I shouldn't rely on my memory. I should be inspecting this in here, so, let me inspect this.

There we go. Unordered list for the classes “todo-list” — ul.todo-list li

Now, it has matched 2 items, so I have got a good query for matching that.

I could be more specific and say I don't want anything underneath this, I want specifically underneath it — ul.todo-list > li

[#](https://testautomationu.applitools.com/automating-in-the-browser-using-javascript/chapter3.3.html#finding-css-selectors-path-queries)**Finding CSS Selectors – Path Queries**

To demonstrate the path queries, I'm going to select the “destroy” button.

So, I'm going to say, "Find the unordered to-do list and find me a button that has a destroy class." — ul.todo-list button.destroy

There we go. It has found 2 of those.

If I say find me an unordered to-do list and immediately below that, find a destroy button — ul.todo-list > button.destroy — it won't find it because this has an unordered to-do list followed by a list item, followed by a div, followed by the button.

I would have to say, followed by a list item, followed by a div in order to match it — ul.todo-list > li > div > button.destroy

If I want to just match a “destroy” button anywhere under the unordered list, I just don't put in the greater than signs, and that is probably as much as you need, except the nth-child, which we saw earlier on.

[#](https://testautomationu.applitools.com/automating-in-the-browser-using-javascript/chapter3.3.html#finding-css-selectors-nth-child-1)**Finding CSS Selectors – :nth-child(1)**

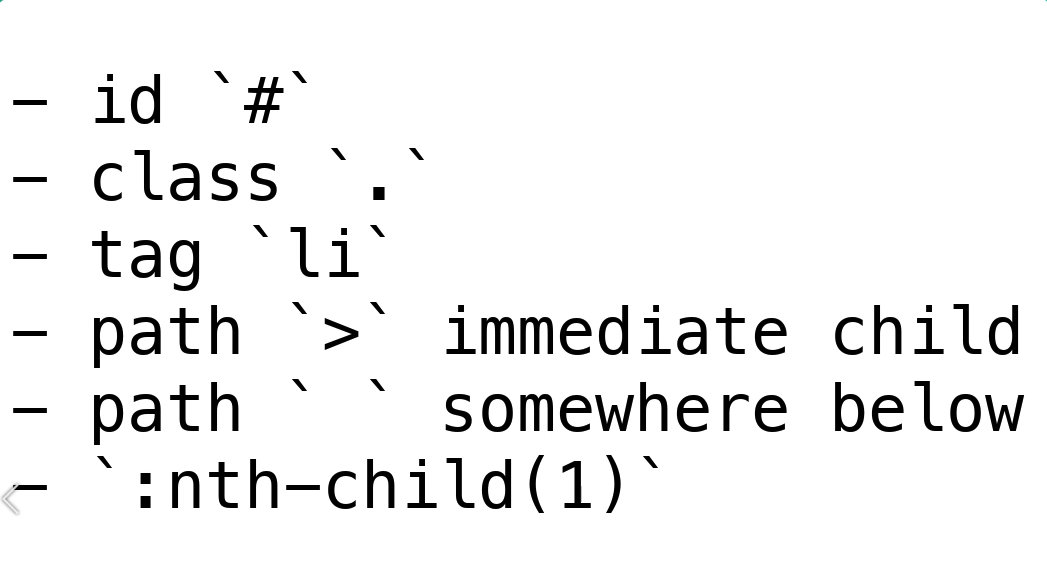
If I say unordered list, if I need the list items, got 2, and I want the first one, and I will say, "Call on nth child," and this is one, starts at 1 —

 ul.todo-list li:nth-child(1)

**Indexing for nth-child**

Very often things are 0 indexed in computing, but the nth child is 1 index. So, 1, will find the first one, 2 we'll find the second one.

**Now, I can specifically choose items on the screen, and that's all the CSS selectors we're going to need to use to automate this.**

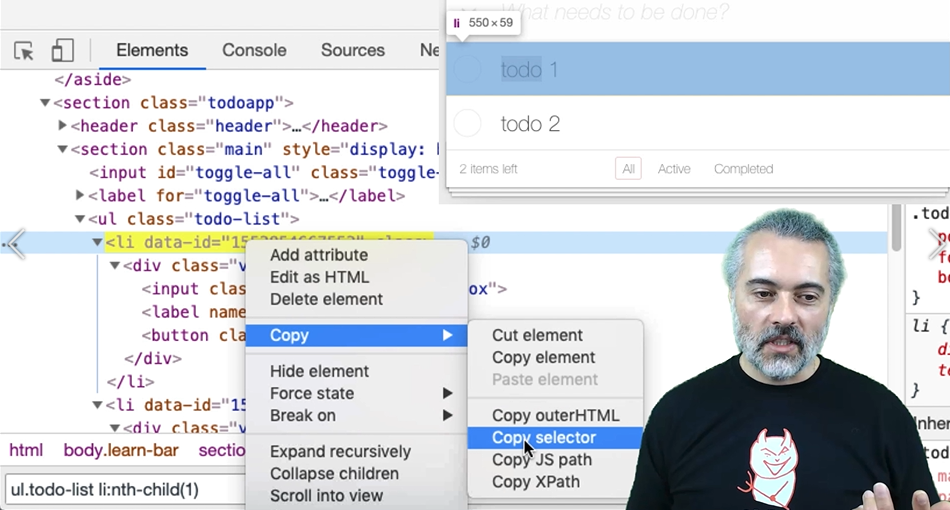


By Id, with a hash [#]. By class, with a dot [.]. By tag, which is just the tag name (ul, li, input button, that kind of thing).

And the path selectors, we can do immediately below with the greater than sign [>]. Or we can do anywhere underneath just with a space and the nth-child pseudo class for finding particular items in a list.

[#](https://testautomationu.applitools.com/automating-in-the-browser-using-javascript/chapter3.3.html#optimizing-css-selectors-through-practice)**Optimizing CSS Selectors Through Practice**

The challenge when you start out is learning how to optimize these CSS selectors, because very often we're going to use the *Copy* to help us.



So this [*Copy selector*] is going to copy a selector, and we get something that isn't specifically optimum.

We're going to have to learn how to read this and say, "I just want to start at the unordered list, and I want it to make sure it's the todo-list," so I'm going to select with the class. And we're going to have to just learn how to do that through practice, because we want to get the smallest CSS selector that matches as specific as we can in order to make our automated execution robust.

Things to practice are doing an *Inspect* on the screen and trying to figure out a CSS selector that will match the item that you have selected on. That will stand you in good stead going forward, but I'm going to show you examples of all the CSS selectors we're going to use as we go through the course.

**One advantage of practicing this way and experimenting with the application in the DOM is that when we do web automated execution being able to construct good CSS selectors is massively important.**

This is a skill that will help you out when you're doing automated execution through WebDriver or other tools, or just navigating the web and you just want to try and find something on the screen that you can't find, being able to do very specific CSS selectors, not just “find by text “in the page will help you out in the real world.

**Resources**

* [MDN CSS Selectors Reference](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors)
* [Mozilla Developer Network](https://developer.mozilla.org/en-US/docs/Web)
* [W3 Schools](https://www.w3schools.com/)