Now at the moment we've been doing individual commands to interact with particular elements.

We're going to need to have the ability to loop round all of them and do stuff. One of the easiest ways to do that is with a for loop.

**Let me just show you what a for loop looks like.**

I'm going to say for and I'm going to create a variable counter.

for(var counter = 1; counter < 20; counter++){

}

I'll set it to 1 to start with. And then I'll say this loop is going to last for as long as counter is less than 20. And then, in order to increment the counter, I need to say, "counter + +." That plus plus means, add 1 to the counter. Then I need to create a set of brackets that say, "Each time you go around this loop, I want you to run the code in here."

**Creating a New Line in Console Editor**

Now you can see that I just did a new line [hitting enter] and it repeated the command. I need to do a “shift enter” to create a new line. Later on, we'll see a better editor that's built into Chrome, to help us to do this.

And I'm going to say console which is a built-in object in the browser, and it lets me write things to the console.

So, I'm going to say "console.log hello" and then I'm going to append the counter to that String.

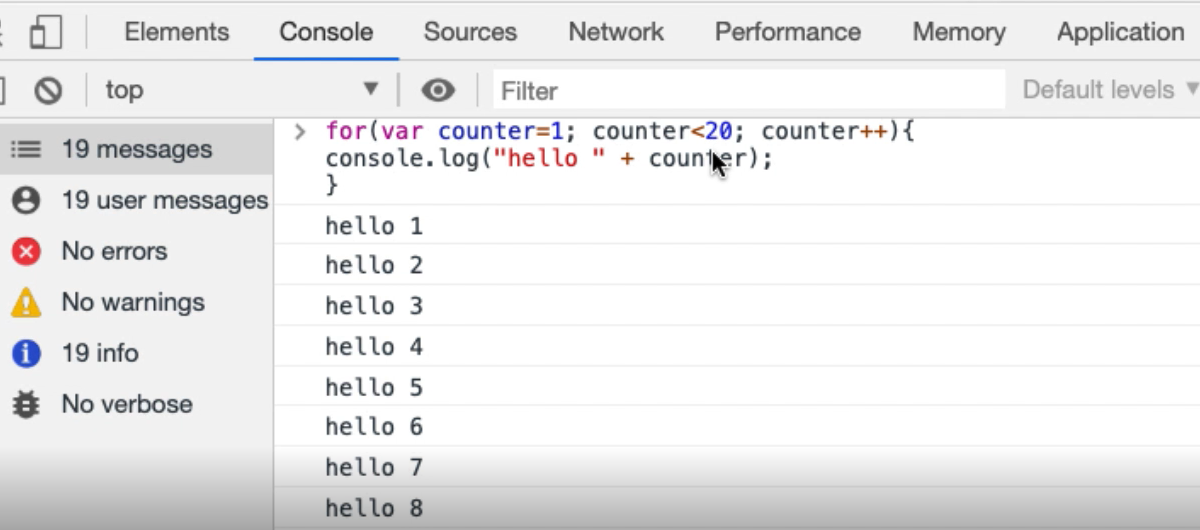
for(counter = 1; counter < 20; counter++){

console.log("hello " + counter);

}

I probably don't need a semicolon, but just by habit I put these in.

Let me run that.



So now I've done a loop there that says, "Hello, start the counter at one, then write out 'hello 1.' Then while the counter is less than 20, keep doing this, 'hello 2,' 'hello 3,' da da da da da, 'hello 19.' 19 is less than 20, yes it is, add one, 20. Is 20 less than 20? No. Okay, stop the loop."

So, this is output 1 to 20. If I wanted to output actually up to 20, less than or equal to 20, run that. Now we go, "hello 20."

That's the basics of how I create a loop — and it's a for loop using a counter variable.

**What might I want to loop round?**

Well, what I might want to do is loop round the selection of all of these todos.

Now I know I've got toggle all. I might want to loop through them all, and check the name, if the name is a certain value, then select it. Select every second one, things like that. So, I need the ability to loop round all the items here.

To do that, I need to say document — well let's see, what am I clicking on?

I'm clicking on the “toggle”. I knew that because that's already in my code here (input.toggle). So essentially, I could take this, but I don't want to querySelector because I want them all. I want querySelectorAll — get me all the “input toggles” that are under the todo-list.

document.querySelectorAll('ul.todo-list input.toggle')

That seems reasonable.

Yeah. There's five of those. 1, 2, 3, 4, 5.

The input toggle only appeared in the todo-list, so I don't actually need that “ul.todo”.

document.querySelectorAll(' input.toggle')

There we go, 5.

So that's the smallest ... in fact, that's not the smallest. I don't even need the input. I could just say, "Give me everything that is marked class toggle":

document.querySelectorAll('.toggle')

There we go, so that's the smallest CSS selector.

I might want to be more specific and say, "Just get me the inputs," but I'll live with .toggle for the moment. That seems good enough.

So .toggle returns an array.

If I use this and say, "What is the second one in that array?" There we go:

document.querySelectorAll('.toggle')[2]

I can see that this array is zero-indexed — 0, 1, 2 — which is that item [input.toggle]. 0, yep, that's the first one. 4, yep, that's the last one.

I could actually, if I had the ability to say, “find the number of things that are in there, and loop round them (Well I know how to loop round them).”

Then I could then click on that:

document.querySelectorAll('.toggle')[4].click()

**All I really need to know is, how can I find the number of things there?**

That's the length: document.querySelectorAll(".toggle").length

I can take all of these things and put them in a variable: var toggles = document.querySelectorAll(".toggle")

So, what's in “toggles”? Toggles is my list, so then I can loop over that.

for(togglepos=0; togglepos<toggles.length; togglepos++){

toggles[togglepos].click();

}

Because “toggles” is an array, I need the particular index point “togglepos”, and I just want to click on that.

So now I have the ability to select or unselect them all, which is slightly different from the “toggle all”. Remember, because toggle all says, "All will be toggled." What our loop does is it says, "Click on the actual toggle value."

So, there's some usefulness here, and I might eventually want to use this to switch them all round. But we're learning how to manipulate all of the things there in a much easier way.

Now you can see that, say, I had 300 todos, having a for loop is much easier than clicking through manually. And we're starting to see how we can use these little bits of code to support us tactically in our testing.

So, the basic pattern for a for loop — for, create a variable, while the variable is less than a particular value, increment the variable, then inside the curly brackets, do some stuff.

**Resources**

* [JavaScript for Loops](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/for)

**Quiz: +100 credits**

**1. What does `console.log` do?**

write text to the browser page

write text to the JavaScript console

write text to a file

**2. When we have a variable called `counter`, what does `counter++` mean?**

create a new variable called `counter+`

add 1 to the value stored in the `counter` variable

subtract 1 from the value stored in the `counter` variable

**3. If we have an array called `listitems` what is the syntax for finding the 5th item in the array?**

listitems[2]

listitems[5]

listitems[4]

listitems(5)

**4. If we have an array called `listitems`, how do we find the length of the array in JavaScript?**

listitems.size()

listitems.size

listitems.length

listitems.length()

listitems

Submit