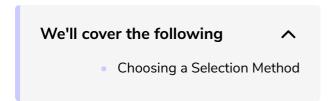
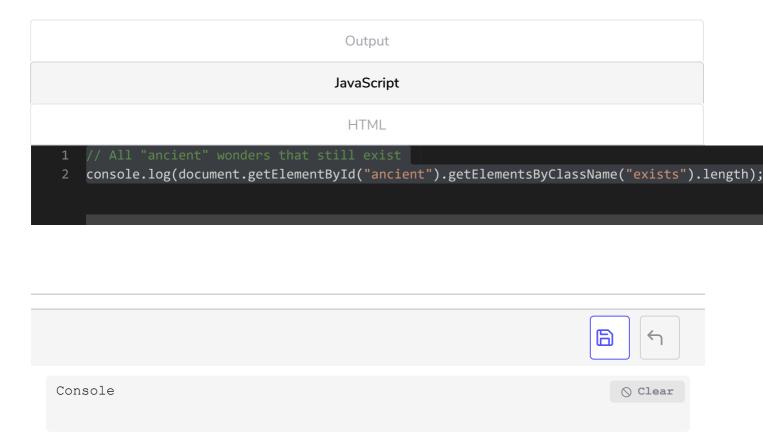
Selecting Elements via CSS Selectors

This lesson is about selection of elements through CSS selectors. Let's see how JavaScript selects the elements via CSS.



For more complex use cases, you can also use CSS selectors to access DOM elements. For example, let's say that you want to grab all the elements of wonders that are both ancient and still exist.



This syntax is a little clunky though. Let's learn two new methods that make finding elements easier. The first is querySelectorAll(), with which you can use CSS selectors to identify elements.



Check the Mozilla Developer Network for a primer on the different CSS selectors available. The second method using CSS selectors is called querySelector(). It works the same way as querySelectorAll() but only returns the first matching element. It returns null if no associated element can be found.





Choosing a Selection Method

You just discovered several ways of selecting DOM elements. How do you choose the right one? Since they use CSS selectors, querySelectorAll() and querySelector() could cover all your needs, but they might perform slower than the others. Here are the general rules of thumb that you should follow:

Numbers of Items to Get	Selection Criterion	Methods to Use
Many	By tag	<pre>getElementsByTagName ()</pre>
Many	By class	<pre>getElementsByClassNa me()</pre>
Many	Not by class or tag	querySelectorAll()
Only One	By ID	<pre>getElementById()</pre>
Only One (the first)	Not by ID	querySelector()