



JavaScript In-Browser Tools

Another way that you can work with JavaScript files or with JavaScript in general is through the in-browser JavaScript development tools. For example, Internet Explorer has a Developer Toolbar, Firefox has an Inspector Console and a very popular plugin called Firebug, and Chrome has a Developer Tools console.

I'm going to show you a little bit about the Chrome Developer Tools console, but the process is similar in other browsers.

Using the Chrome Console

Create a standard HTML file with a button and a script, which will print a message to the console whenever we click the button.

```
<html>
<head>
  <title>Introduction to Programming in JavaScript - Lesson 1.4</title>
</head>
<body>
  <button id="say-hi">Say Hi</button>
  <script>
    document.getElementById("say-hi").addEventListener("click", function() {
      console.log("Hi");
    });
  </script>
</body>
</html>
```

Now open the file in the browser and click on the button. Nothing happens, but we know that that `console.log` is there in the script. So how do we get to that? To open Chrome's Developer Console, you need to go to the View Menu, go to the Developer Sub Menu and choose Developer Tools.

View the Chrome Console

The console will open at the bottom of the screen. It lets us view the source of the entire page and interact with the elements in a number of exciting ways that are very useful for debugging.

The Console Tab

If you click on Console tab, you'll see that a JavaScript console opens up. Up at the top of this console, you will see that the word "hi" is here. If you click the button again, you'll see the number 2 show up in front of the word. That's because that particular command has been received twice.

On the console, you can also see a little caret, that you might recognize from our tutorial on how to use NodeJS in the terminal. You can do all of the same things here that you can do in NodeJS' terminal.

For example, I can type `2+2`, and it'll be interpreted as 4. Or, I can do `console.log('hello')` and receive the "hello" word in the console. Having easy access to an interactive JavaScript interpreter in the browser is very convenient, and it can help you if you want to try testing out a few examples or experimenting with JavaScript. Especially, if you're in the middle of working with a web page.