

Debugging JavaScript code using devtool Console

I - Writing messages to the devtool console

One common way to debug your code: write messages to the devtool console!

Using the `console.log(...)` instruction, you can write messages to the devtool console of your browser.

For example, let's say you added a button to your Web page, but nothing happens when you click on it. Without using the devtool console and `console.log(...)` it can be very difficult to spot your error.

Example: check that some parts of the code you wrote are executed correctly

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Write messages to the devtool console</title>
  <meta charset="utf-8"/>
  <script>
    var x = 10;

    function addXToThePage() {
      // This code is strangely never executed...
      document.body.innerHTML += "<p>The value of x = " + x + "</p>";
      x = X + 1;
    }
  </script>
</head>
<body>
  <button onclick="addXToToThePage();">Click me to display the value of the
x variable</button>
</body>
</html>
```

Normally, when the button is clicked, we should call the `addXToThePage()` function, which in turn should display the value of the variable `x` inside an HTML paragraph.

Let's check if the function is really executed. We need to add a `console.log("In the addXToThePage function");` as the first instruction of the function:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Write messages to the devtool console</title>
  <meta charset="utf-8"/>
  <script>
    var x = 10;

    function addXToThePage() {
      // THIS WILL DISPLAY A MESSAGE IN THE DEVTOOL CONSOLE
      console.log("In The addXToThePage function");

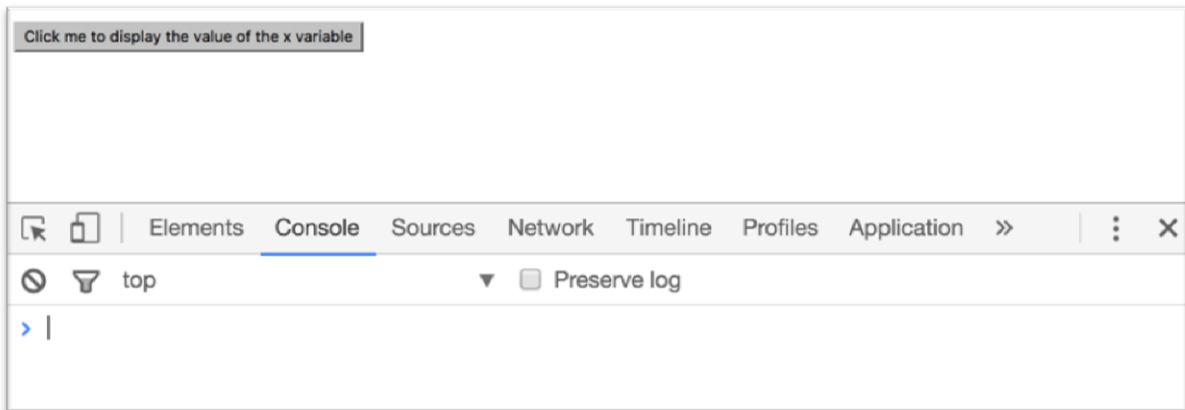
      // This code is strangely never executed...
      document.body.innerHTML += "<p>The value of x = " + x + "</p>";
      x = X + 1;
    }

  </script>
</head>
```

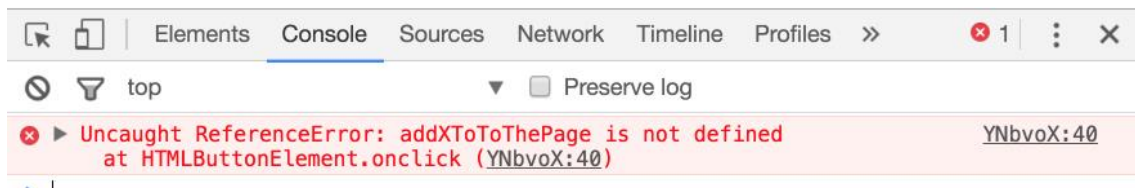
Debugging JavaScript code using devtool Console

```
<body>
  <button onclick="addXToToThePage();">Click me to display the value of the
x variable</button>
</body>
</html>
```

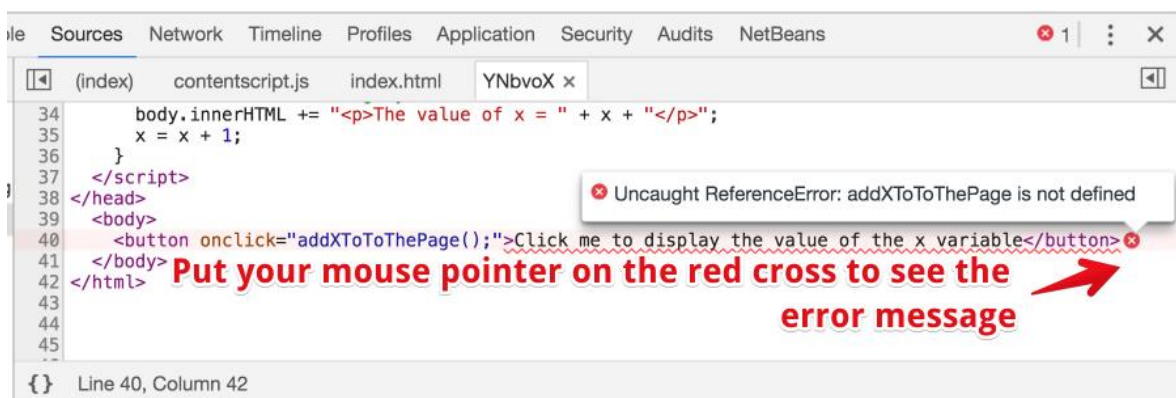
And let's execute this with the devtool console open (Chrome: CTRL + Shift + I) then the browser will just display a page with the code running:



Now, press the button! An error message appears:



Apparently the function name is wrong, addXToToThePage does not exist. We can check the source code by clicking on the YNbvoX:40 link on the right:

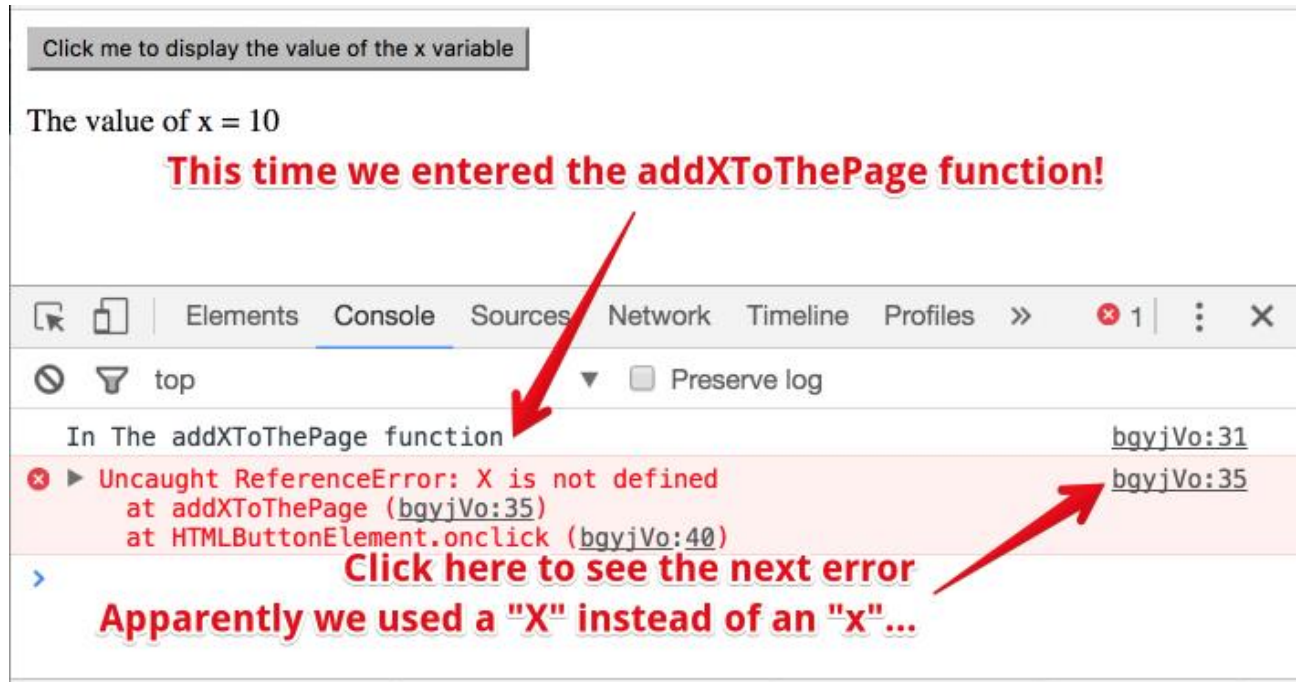


Ok, so our first error was that we typed the wrong name in the onclick="...", the name we used : AddXToToThePage has "ToTo" instead of "To". Let's fix that now and try again:

Let's try again (after we've changed the name of the function to the correct one), go in debug mode, open the devtool console and click the button.

Debugging JavaScript code using devtool Console

Here is what we get:



Ok, you understand the principle... by using `console.log(...)` with a string message as parameter, you can make the message appear in the devtool console, confirming that you have executed your code without error at least to this point...

Let's fix this "X"! We'll replace it with an "x" and everything should be ok now:

You can click the button now....

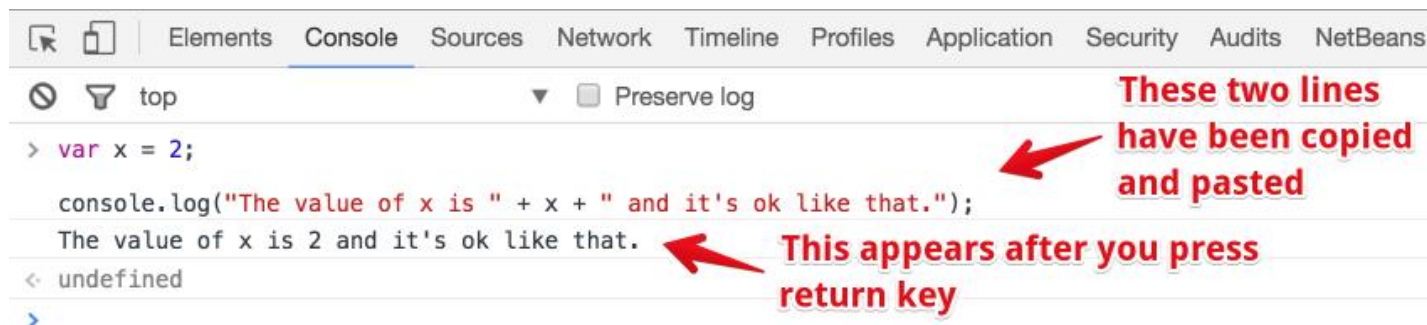
Going further with `console.log`

II - Displaying variable values

Instead of just displaying pure string messages, you can also use the "+" operator to concatenate variable values to the pure character strings. Example :

```
var x = 2;  
console.log("The value of x is " + x + " and it's ok like that.");
```

We typed this code in the devtool console to check what it does:



Debugging JavaScript code using devtool Console

You can use as many "+" as you like, the part right after the "+" will be turned into a string if the part in front is a string...

You can also use parentheses in order to display results of simple calculations:

```
var x = 2;  
var y = 4;  
console.log("x = " + x);  
console.log("y = " + y);
```

```
console.log("The value of (x+y) is " + (x+y) + " and it's ok like that.");
```



You can use any operator (+, -, /, *, etc.), you are not limited to using the "+" operator.

And of course you can use such code in your programs, not only in the devtool console :-)