alias to chrome console.log

Asked 8 years, 5 months ago Active 6 years, 1 month ago Viewed 6k times

20

11

I would like to know why the follow code doesn't work in the Google Chrome:

```
// creates a xss console log

var cl = ( typeof( console ) != 'undefined' ) ? console.log : alert;
cl('teste');
```

output: Uncaught TypeError: Illegal invocation

thanks.

javascript exception google-chrome

edited Feb 27 '11 at 14:23

asked Feb 27 '11 at 14:08

augustowebd

153 1 8

I just ran across a similar problem, where I wanted to override the `console.log` method (to catch and send output). My solution was to declare the alias on the `console` object. Roughly: `console.__log = console.log;console.log = function(){__console.log.apply(console, arguments);};` . – Rob W Jan 3 '12 at 10:38

3 Answers

30

When you write `cl();` , you're calling `log` in the global context.

Chrome's `console.log` doesn't want to be called on the `window` object.

Instead, you can write

```
cl = function() { return console.log.apply(console, arguments); };
```

✓

This will call `log` in the context of `console`.

edited Mar 2 '11 at 18:40

answered Feb 27 '11 at 14:10
SLaks
710k 144 1684 1800

So, how I can do that? – augustowebd Feb 27 '11 at 14:12

Thanks Slaks, it'a works very well! – augustowebd Feb 27 '11 at 14:15

5 @augustowebd: For more about this "context" thing, I've written up a couple of blog posts that may be useful: *Mythical methods* and *You must remember* `this` – T.J. Crowder Feb 27 '11 at 14:17

4 This has the drawback of not logging the correct line number in Web Inspector. – Justin Sep 27 '11 at 2:04

27

https://groups.google.com/a/chromium.org/d/msg/chromium-bugs/gGVPJ1T-qA0/F8uSupbO2R8J

Apparently you can also defined log:

```
log = console.log.bind(console);
```

and then the line numbers also work

edited Oct 23 '12 at 12:34
Adaptabi
3,414 1 20 25

answered Oct 17 '12 at 20:23
Ben West
528 4 17

I love you, this has been driving me nuts for years – Kevin Apr 7 '13 at 21:05

Finally a solution for this! Thanks :D – M K Aug 29 '14 at 10:07

Unfortunately @SLaks answer isnt applied to IE because it uses window-object as context in console.log-method.

I would be suggest another way that doesnt depend on browser:

2

```
!window.console && (console = {});

console.debug = console.debug || $.noop;
console.info = console.info || $.noop;
console.warn = console.warn || $.noop;
console.log = console.log || $.noop;

var src = console, desc = {};
desc.prototype = src;
console = desc;

desc.log = function(message, exception) {
    var msg = message + (exception ? ' (exception: ' + exception + ')' : ''), callstack
= exception && exception.stack;
    src.log(msg);
    callstack && (src.log(callstack));
    //logErrorUrl && $.post(logErrorUrl, { message: msg + (callstack || '') }); // Send
clientside error message to serverside.
};
```

edited Jun 13 '13 at 21:24

answered May 24 '13 at 10:03

vladimir
3,235 2 18 30