**JAVASCRIPT TRICKS**

1. **Go Back button:**

Browsers already have "back" buttons, so you'd better have a darn good reason for needing to put one on your page!

[#](https://css-tricks.com/snippets/javascript/go-back-button/#article-header-id-0)Input button with inline JavaScript

<input type=”button” value=”Go Back From When You Came!” onClick=”history.back(-1)” />

This is totally obtrusive, but you could fix that by only appending this button through JavaScript.

[#](https://css-tricks.com/snippets/javascript/go-back-button/#article-header-id-1)PHP

If JavaScript isn't a possibility, you could use the HTTP\_REFERER, sanitize it, and echo it out via PHP.

<?php

$url = htmlspecialchars($\_SERVER[‘HTTP\_REFERER’]);

Echo “<a href=’$url’>back</a>”; ?>

1. **1024\*768 BookMarkLet:**

The days of 800x600 screens are all but over, but most of us still try to accommodate 1024px wide screens. Hence the popularity of "960" width sites. This bookmarklet will resize the current browser window to that width and height. You know, so us web designers with giant monitors can see what it's like to be slummin' with a 1024 screen. Also, to see "the fold", if such a concept even matters anymore.

Javascript:resizeTo(1024,768)

### [**#**](https://css-tricks.com/snippets/javascript/1024x768-bookmarklet/#article-header-id-0)The Bookmarklet

[1024X768](javascript:resizeTo(1024,768))   < Drag to bookmarks bar

This doesn't seem to work at all anymore (tested in stable version of Chrome and Firefox, April 2013). It used to work only when the window was by itself, not when multiple tabs were open, but now that doesn't work anymore either.

There is some kind of way to do it, since I still regularly see resized popups, so feel free to chime in the comments if you know more.

1. **Trim First and Last Characters:**

### Remove last four characters

Var myString = “abcdefg”;

Var newString = mystring.substr(0,myString.length-4);

// newString is now “abc”

### [**#**](https://css-tricks.com/snippets/javascript/trim-firstlast-characters-in-string/#article-header-id-1)Remove first two characters

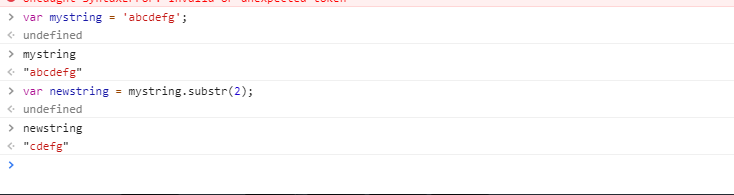
Var myString = “abcdefg”;

Var newString = myString.substr(2);

// newString is now “cdefg”

### [**#**](https://css-tricks.com/snippets/javascript/trim-firstlast-characters-in-string/#article-header-id-2)Notes

The substr function can be called on any string with two integer parameters, the second optional. If only one provided, it starts at that integer and moves to the end of the string, chopping off the start. If two parameters provided, it starts at the first number and ends at the second, chopping off the start and end as it is able.



1. **Cookie Getter/Setter:**

/\*\*

\* Gets or sets cookies

\* @param name

\* @param value (null to delete or undefined to get)

\* @param options (domain, expire (in days))

\* @return value or true

\*/

\_.cookie= function(name, value, options)

{

If(typeof value === “undefined”){

var n,v ,

cookies= document.cookie.split(“;”);

for (var i=0;i<cookie.length;i++){

n=$.trim(cookies[i].substr(0,cookies[i].indexOf(“=”)));

v=cookies[i].substr(0,cookies[i].indexOf(“=”)+1);

if(n === name){

return unescape(v);

}

}

} else {

options = options || {};

If(!value){

Value=””;

options.expires = -365;

} else {

value = escape(value);

}

if (options.expires) {

var d = new Date();

d.setdate(d.getDate() + options.expires);

value += “; expires=” +d.toUTCString();

}

if(options.domain)

{

value += “; domain=” + options.domain;

}

if(options.path) {

value += “; path=” + options.path;

}

document.cookie = name + “=” + value;

}

};

1. **Error-Free Console Logging:**

var Fb = {}; //An empty object literal for holding the function

Fb.log = function(obj, consoleMethod) {

if(window.console && window.console.firebug.replace(/^\s\s/, ‘’).replace(/\s\s\*$/, ‘’) !== ‘’) {

if (typeOf console === “string” && typeOf console[consoleMethod] === “function” ) {

console[consolemethod](obj);

} else {

console.log(obj);

}

}

}

If you leave console.log, console.info, etc messages in your JavaScript code and open the page in browser like IE then it may halt the page loading completely saying that 'console not defined' especially if your user uses a non IE8 browser.

This code snippet will allow you to leave the logging message as such in your code if you wish and your page will render properly in IE or any other browser that does not support the console messages.

#### [#](https://css-tricks.com/snippets/javascript/error-free-firebug-logging/#article-header-id-0)Usage

#### Fb.log(“This will be logged”);

#### Fb.log(“This will be displayed in console as info”,”info”);

The FB.log function accepts two parameters the first one is the "item" that you want to display in the firebug console and the second one is the firebug method that you want to use for the logging, like info, error, etc. If you omit the second parameter the result will be equivalent to console.log()

### [**#**](https://css-tricks.com/snippets/javascript/error-free-firebug-logging/#article-header-id-1)Simple log-only way

function ltc(what) {

try{

console.log(what);

}

catch(e) {}

finally {

return;

}

}

ltc(“message”);

1. **Get all Possible DOM events:**

Used in firefox

[i for(i in document)].filter(function(i){return i.substring(0,2)=='on'&&(document[i]==null||typeof document[i]=='function');})

For Chrome:

var items = Array.prototype.slice.call(

  document.querySelectorAll('\*')

  ).map(function(element) {

  var listeners = getEventListeners(element);

  return {

  element: element,

  listeners: Object.keys(listeners).map(function(k) {

  return { event: k, listeners: listeners[k] };

  })

  };

  }).filter(function(item) {

  return item.listeners.length;

  });

You'll get an array like this (Firefox 23):

["onreadystatechange", "onmouseenter", "onmouseleave", "onwheel", "oncopy", "oncut", "onpaste", "onbeforescriptexecute", "onafterscriptexecute", "onabort", "oncanplay", "oncanplaythrough", "onchange", "onclick", "oncontextmenu", "ondblclick", "ondrag", "ondragend", "ondragenter", "ondragleave", "ondragover", "ondragstart", "ondrop", "ondurationchange", "onemptied", "onended", "oninput", "oninvalid", "onkeydown", "onkeypress", "onkeyup", "onloadeddata", "onloadedmetadata", "onloadstart", "onmousedown", "onmousemove", "onmouseout", "onmouseover", "onmouseup", "onpause", "onplay", "onplaying", "onprogress", "onratechange", "onreset", "onseeked", "onseeking", "onselect", "onshow", "onstalled", "onsubmit", "onsuspend", "ontimeupdate", "onvolumechange", "onwaiting", "onmozfullscreenchange", "onmozfullscreenerror", "onmozpointerlockchange", "onmozpointerlockerror", "onblur", "onerror", "onfocus", "onload", "onscroll"]

Firefox, because it supports [array comprehension](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Predefined_Core_Objects?redirectlocale=en-US&redirectslug=JavaScript%2FGuide%2FPredefined_Core_Objects#Array_comprehensions) which that little snippet uses.

1. **Get URL and URL parts in javascript:**

JavaScript can access the current URL in parts. For this URL:

http://css-tricks.com/example/index.html?s=flexbox

* window.location.protocol = "http:"
* window.location.host = "css-tricks.com"
* window.location.pathname = "example/index.html"
* window.location.search = "?s=flexbox"

So to get the full URL path in JavaScript:

var newURL = window.location.protocol + "//" + window.location.host + "/" + window.location.pathname + window.location.search

If you need to break up up the pathname, for example, a URL like http://css-tricks.com/blah/blah/blah/index.html, you can split the string on "/" characters

var pathArray = window.location.pathname.split('/');

var pathArray = window.location.href.split('/');

OutPut:

(5) ["https:", "", "html-online.com", "articles", ""]

0: "https:"

1: ""

2: "html-online.com"

3: "articles"

4: ""

length: 5

Then access the different parts by the parts of the array, like

var secondLevelLocation = pathArray[0];

To put that pathname back together, you can stitch together the array and put the "/"'s back in:

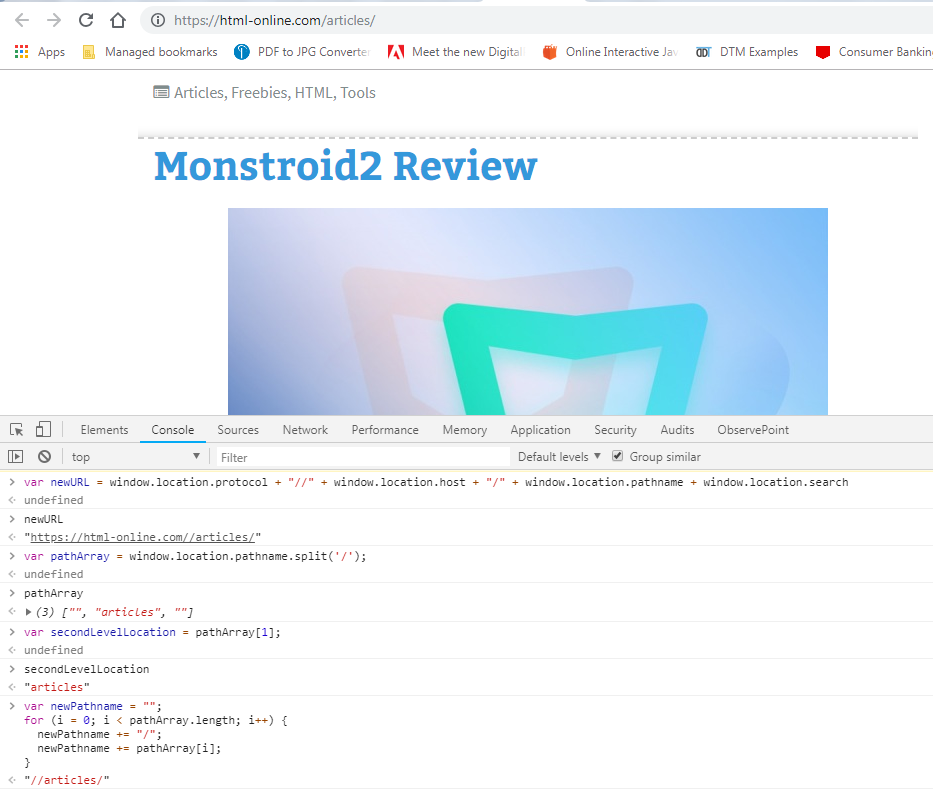
var newPathname = "";

for (i = 0; i < pathArray.length; i++) {

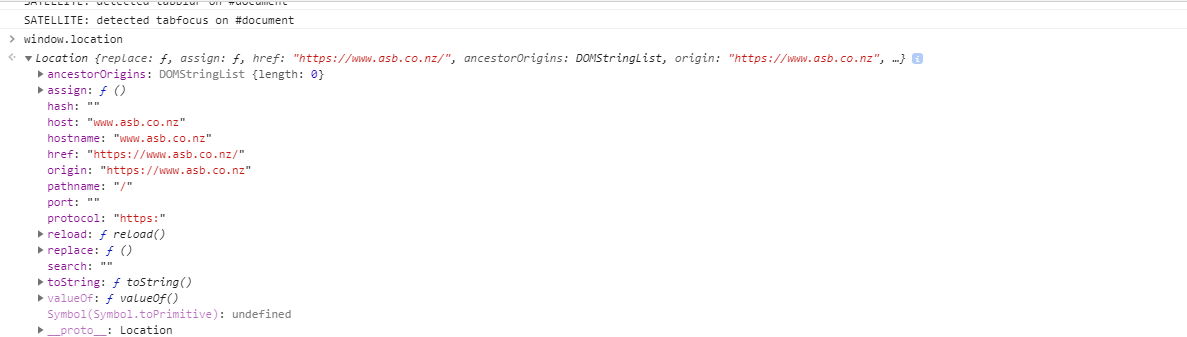
newPathname += "/";

newPathname += pathArray[i];

}



Probably the quickest way to take a peak at what you have is to put window.location in the DevTools console and see:



1. **Get URL Variables:**

function getQueryVariable(variable)

{

var query = window.location.search.substring(1);

var vars = query.split("&");

for (var i=0;i<vars.length;i++) {

var pair = vars[i].split("=");

if(pair[0] == variable){return pair[1];}

}

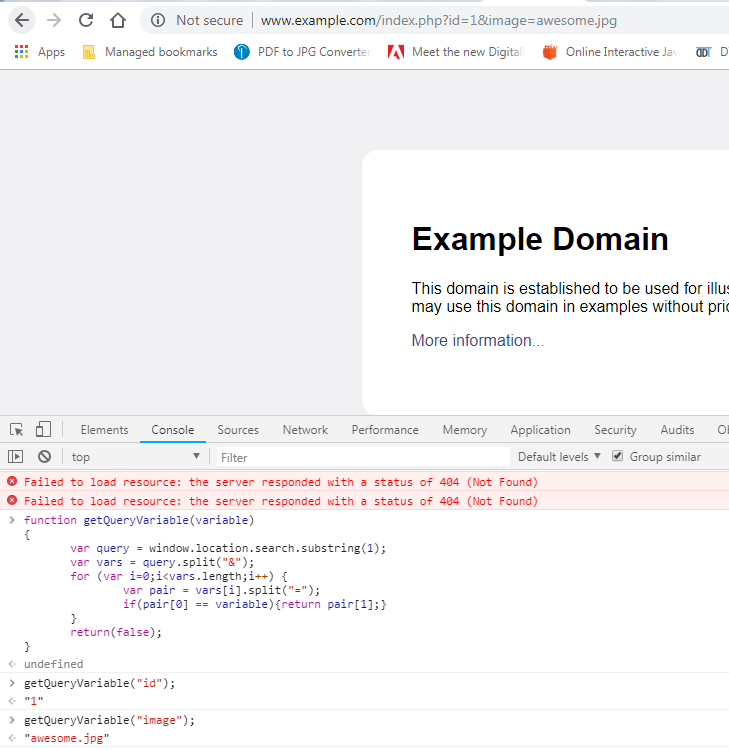
return(false);

}

### Usage

Example URL:  
http://www.example.com/index.php?id=1&image=awesome.jpg

Calling getQueryVariable("id") - would return "1".  
Calling getQueryVariable("image") - would return "awesome.jpg".



1. **Inject new CSS rules:**

If you need to change the style of an element with JavaScript, it's typically better to change a class name and have the CSS already on the page take effect and change the style. However, there are exceptions to every rule. For instance, you might want to programmatically change the pseudo-class (e.g. :hover). You can't do that through JavaScript for the same reason inline style="" attributes can't change pseudo classes.

You'll need to inject a new <style> element onto the page with the correct styles in it. Best to inject it at the bottom of the page so it overrides your CSS above it. Easy with jQuery:

function injectStyles(rule) {

var div = $("<div />", {

html: '&shy;<style>' + rule + '</style>'

}).appendTo("body");

}

$("button").on("click", function() {

injectStyles('a:hover { color: red; }');

});

1. **Remove An Element:**

For whatever reason, an element can't destroy itself in JavaScript. jQuery has a method for this, which is nice because this is how we think:

$(".remove-me").remove();

But there is no direct equivalent in JavaScript. Instead you'll need to select the parent element and use removeChild.

So if you have:

<div class="module">

<p>Stuff.</p>

<div class="remove-me">...</div>

</div>

You'll need to do:

var thingToRemove = document.querySelectorAll(".remove-me")[0];

thingToRemove.parentNode.removeChild(thingToRemove);

Or if you had a reference to an element and wanted to empty out all the elements inside it, but keep it:

mydiv = document.getElementById('empty-me');

while (mydiv.firstChild) {

mydiv.removeChild(mydiv.firstChild); }

1. **Toggle(Show/Hide) Element:**

<script type="text/javascript">

<!--

function toggle\_visibility(id) {

var e = document.getElementById(id);

if(e.style.display == 'block')

e.style.display = 'none';

else

e.style.display = 'block';

}

//-->

</script>

#Inline usage:

<a href="#" onclick="toggle\_visibility('foo');">Click here to toggle visibility of element #foo</a>

<div id="foo">This is foo</div>

1. **Remove the Last Character from a String:**

var origString = 'Happy Dance7';

var trimmedString = origString.substring(0, origString.length-1);

console.log(trimmedString);

// 'Happy Dance'