

What does "javascript:void(0)" mean?

Asked 9 years, 11 months ago Active 7 months ago Viewed 1.7m times

▲ `login`

1233

I've seen such `href` s many times, but I don't know what exactly that means.



javascript

void



353

edited Oct 3 '17 at 13:07

asked Aug 18 '09 at 5:20



Ondra Žižka

21.8k 26 150 220



omg

53.7k 131 257 335

- 8 `javascript:` is one of the many URI schemes: en.wikipedia.org/wiki/URI_scheme, like `data:` . – [Ciro Santilli](#) 新疆改造中心996ICU六四事件 Aug 23 '14 at 21:05
- 3 You can use just `href="javascript:"` for the same purpose. As stated in the answer for [this question](#), the `void(0)` part was originally intended for early versions of browsers where `javascript:` URI handling was different. But now I couldn't even find a version where the shorthand wouldn't work, at least IE7 handles this correctly. – [user](#) Apr 5 '15 at 3:24
- 1 I've also seen `href="javascript://"`, is this better? – [StarWars](#) Dec 30 '16 at 13:43
- `href="javascript://"` is not working for me `void(0)` works perfectly. – [sandip](#) Jun 3 '17 at 8:20

15 Answers



937



The `void` operator evaluates the given expression and then returns `undefined` .

The `void` operator is often used merely to obtain the `undefined` primitive value, usually using “`void(0)`” (which is equivalent to “`void 0`”). In these cases, the global variable [undefined](#) can be used instead (assuming it has not been assigned to a non-default value).

An explanation is provided here: [void operator](#).



The reason you'd want to do this with the `href` of a link is that normally, a `javascript:` URL will redirect the browser to a plain text version of the result of evaluating that JavaScript. But if the result is `undefined`, then the browser stays on the same page. `void(0)` is just a short and simple script that evaluates to `undefined`.

edited Sep 28 '17 at 1:58



Yona Appletree

3,854 5 27 37

answered Aug 18 '09 at 5:23



rahul

155k 44 208 250

-
- 8 what does it mean when href is given a "undefined primitive value"? – [omg](#) Aug 18 '09 at 5:32
-
- 5 "normally a javascript: url will redirect the browser to a plain text version of the result of evaluating that javascript. " Can you make an example here? I've never seen such usage. – [omg](#) Aug 18 '09 at 5:40
-
- 76 An example of what phoenix is talking about is `DO IT NOW! `. If `dosomething` returns `false`, then clicking the link will simply cause the browser to exit the page and display `"false"`. However... `DO IT NOW! ` avoids the problem. Go ahead and paste `javascript: 1+1;` into your browsers address bar. The browser should display `"2"` – [Breton](#) Aug 18 '09 at 5:50
-
- 8 Because `void` is a unary operator. `Void` is not a value, nor is it a function. It needs a value to operate on to its right, or it will throw an error. – [Breton](#) Aug 18 '09 at 5:59
-
- 13 try looking in the error console? It definitely throws a syntax error. It's invalid javascript. Douglas crockford recommends staying away from `void` because of the unary operator/function/value confusion is too costly to deal with. – [Breton](#) Aug 18 '09 at 6:05
-



In addition to the technical answer, `javascript:void` means the author is Doing It Wrong.

412



There is no good reason to use a `javascript:` pseudo-URL(*). In practice it will cause confusion or errors should anyone try things like 'bookmark link', 'open link in a new tab', and so on. This happens quite a lot now people have got used to middle-click-for-new-tab: it looks like a link, you want to read it in a new tab, but it turns out to be not a real link at all, and gives unwanted results like a blank page or a JS error when middle-clicked.

`` is a common alternative which might arguably be less bad. However you must remember to `return false` from your `onclick` event handler to prevent the link being followed and scrolling up to the top of the page.

In some cases there may be an actual useful place to point the link to. For example if you have a control you can click on that opens up a previously-hidden `<div id="foo">`, it makes some sense to use `` to link to it. Or if there is a non-JavaScript way of doing the same thing (for example, `'thispage.php?show=foo'` that sets `foo` visible to begin with), you can link to that.

Otherwise, if a link points only to some script, it is not really a link and should not be marked up as such. The usual approach would be to add the `onclick` to a ``, `<div>`, or an `<a>` without an `href` and style it in some way to make it clear you can click on it. This is what StackOverflow [did at the time of writing; now it uses `href="#"`].

The disadvantage of this is that you lose keyboard control, since you can't tab onto a span/div/bare-a or activate it with space. Whether this is actually a disadvantage depends on what sort of action the element is intended to take. You can, with some effort, attempt to mimic the keyboard interactability by adding a `tabIndex` to the element, and listening for a Space keypress. But it's never going to 100% reproduce the real browser behaviour, not least because different browsers can respond to the keyboard differently (not to mention non-visual browsers).

If you really want an element that isn't a link but which can be activated as normal by mouse or keyboard, what you want is a `<button type="button">` (or `<input type="button">` is just as good, for simple textual contents). You can always use CSS to restyle it so it looks more like a link than a button, if you want. But since it behaves like a button, that's how really you should mark it up.

(*: in site authoring, anyway. Obviously they are useful for bookmarklets. `javascript:` pseudo-URLs are a conceptual bizarreness: a locator that doesn't point to a location, but instead calls active code inside the current location. They have caused massive security problems for both browsers and webapps, and should never have been invented by Netscape.)

edited Jun 19 '14 at 16:19



Ciro Santilli 新疆改造中心996ICU六四事件
164k 36 623 511

answered Aug 18 '09 at 10:50



bobince
456k 91 591 781

-
- 7 In addition to excellent post by @bobince: I've done some research a couple of months ago on cross-browser keyboard navigability of `href` s, including quirks and side effects; some of you might find it useful: [jakub-g.github.com/accessibility/onclick](https://github.com/jakub-g/accessibility/onclick) – [jakub.g](#) Feb 12 '13 at 15:33 ✎
-
- 2 @ThinkBonobo: SO has changed at some point since 2009! Updated. – [bobince](#) May 16 '14 at 12:49
-
- 54 This is an opinion and doesn't answer the question. `void(0)` is needed in many cases; `"#"` is a hack that brings with it a whole host of problems (it would not work in the app I'm writing, that brought me to this page). – [felwithe](#) Mar 24 '15 at 3:46
-
- 10 I agree with @felwithe. Why force others to "Do it A Particular Way"? In 15 years of programming I am yet to see how the motto "it should always be done this way" doesn't lead people to a mess of their own making – [Steven de Salas](#) Jul 24 '15 at 2:48 ✎
-
- 2 Using the fragment-id is a bad idea from a UX perspective as it causes the document to jump to the top of the page unless `preventDefault` is used. Please don't do it in the case of an anchor being used as a button on a form. – [Josh Haldas](#) Mar 11 '17 at 14:17
-



It means it'll do nothing. It's an attempt to have the link not 'navigate' anywhere. But it's not the right way.

116

You should actually just `return false` in the `onclick` event, like so:



```
<a href="#" onclick="return false;">hello</a>
```

Typically it's used if the link is doing some 'JavaScript-y' thing. Like posting an AJAX form, or swapping an image, or whatever. In that case you just make whatever function is being called return `false`.

To make your website completely awesome, however, generally you'll include a link that does the same action, if the person browsing it chooses not to run JavaScript.

```
<a href="backup_page_displaying_image.aspx"
  onclick="return coolImageDisplayFunction();">hello</a>
```

edited Sep 9 '13 at 15:04



Rory O'Kane

19.6k 8 67 110

answered Aug 18 '09 at 5:21



Noon Silk

48.4k 6 78 98

-
- 28 no no - return false will stop the default behavior, so the # will never appear – [Magnar](#) Aug 18 '09 at 5:43
-
- 21 the javascript: url protocol is a defacto standard, not a real standard. So the href="#" onclick="return false;" is standards compliant while href="javascript:void(0)" is not, because there is no official standard that specifies what that should do. – [Breton](#) Aug 18 '09 at 5:55
-
- 9 On top of that, Douglas Crockford doesn't like void, so jslint will complain about it. Basically, since void is an operator, and not a value, it's confusing as hell, and spawns many questions such as this one. Better to avoid it altogether. haha. – [Breton](#) Aug 18 '09 at 5:57
-
- 4 Brandon: see brenton's responses. The way I recommend is the most supported and as I said in the second part of my post, in a 'proper' site you won't ever even use '#', because you'll be providing fallback systems to handle a lack of javascript. – [Noon Silk](#) Aug 18 '09 at 6:00
-
- 15 +1 for including the *completely awesome* example. Even if you have no static HTML fall-back for what you're doing in JavaScript, you can always do something like ` .` – [Grant Wagner](#) Aug 18 '09 at 18:29
-



There is a HUGE difference in the behaviour of "#" vs javascript:void

67

"#" scrolls you to the TOP of the page while "javascript:void(0);" does not.



This is very important if you are coding dynamic pages. the user does not want to go back to top just because he clicked a link on the page.

answered Aug 18 '09 at 8:25



Salvin Francis

2,682 4 30 42

- 24 @Salvin: The *scroll-to-top-of-page* behavior can be suppressed by returning `false` to the event handler: `onclick="doSomething();return false;"`, or if `doSomething()` returns `false`, you can use `onclick="return doSomething();"`. – [Grant Wagner](#) Aug 18 '09 at 18:25
- 34 @GrantWagner - Or, 5 years later, `e.preventDefault()`. – [trysis](#) Jul 26 '14 at 0:47
- 1 You might want to edit/delete this answer since `"#"` does *not* scroll to the top when you return `false`. – [Navin](#) Oct 30 '15 at 6:06
- 3 @Navin you are correct, but that is now an outdated practice. The comment by trysis is considered the correct way now. – [Tim Seguire](#) Nov 12 '15 at 15:15



57



It is a very popular method of adding JavaScript functions to HTML links.
For example: the `[Print]` links that you see on many webpages are written like this:

```
<a href="javascript:void(0)" onclick="callPrintFunction()">Print</a>
```

Why do we need `href` while `onclick` alone can get the job done? Because when users hover over the text 'Print' when there's no `href`, the cursor will change to a caret (I) instead of a pointer (). Only having `href` on an `a` tag validates it as a hyperlink.

An alternative to `href="javascript:void(0);"`, is the use of `href="#"`. This alternative doesn't require JavaScript to be turned on in the user's browser, so it is more compatible.

edited Dec 19 '18 at 23:36



[Gust van de Wal](#)
3,317 14 38

answered Jan 14 '14 at 7:17



[Huy - Vuong Do Thanh](#)
688 6 8

- 4 neither does anything useful if javascript is turned off. – [Jasen](#) Mar 9 '16 at 22:40
- 8 You don't need `href` to get the pointing hand cursor; all it takes is a bit of CSS. – [John Montgomery](#) Jul 4 '17 at 3:02
- 1 Why would you not put the JavaScript function in the href instead the onclick? – [Sid](#) Apr 28 at 15:23

I agree with @Sid - if you're using it to trigger a javascript function then `` is cleaner (although it should probably be a `button` rather than an `a nchor` if it doesn't actually take you anywhere). – [DaveMongoose](#) May 9 at 12:25

`href="#"` can lead to nasty surprises - like aborted xhr requests, that happen to be called on a click to that link. I recently had a hard time debugging a website which aborted oidc login requested, if the user happened to be in an address that wasn't the root of the site. `# href` caused it to reload the address before xhr request got completed. – [JustAMartin](#) Jul 23 at 14:56

▲ You should always have an href on your **a** tags. Calling a JavaScript function that returns 'undefined' will do just fine. So will linking to '#'.

43 Anchor tags in Internet Explorer 6 without an href do not get the `a:hover` style applied.

▼ Yes, it is terrible and a minor crime against humanity, but then again so is Internet Explorer 6 in general.

I hope this helps.

Internet Explorer 6 is actually a major crime against humanity.

edited Jan 8 '18 at 14:21



Peter Mortensen

14.3k 19 88 116

answered Aug 18 '09 at 5:25



jscharf

5,041 2 17 15

▲ It's worth mentioning that you'll sometimes see `void 0` when checking for undefined, simply because it requires less characters.

20 For example:

▼ `something === undefined`

vs.

`something === void 0`

Some minification methods replace undefined with `void 0` for this reason.

edited Jan 16 '17 at 11:28



John Weisz

13.7k 5 49 91

answered Nov 10 '16 at 6:32



Squall

531 5 8

7 One notable example is TypeScript ([live example](#)), which compiles default parameter values to checks against `void 0`. The 3 character difference adds up quickly when lots of methods are using default param values. – John Weisz Jan 16 '17 at 11:26

1 "Some minification methods replace undefined with `void 0` for this reason." Finally I understand it! Thanks @squall for the thorough answer. – Ahmed Mahmoud Nov 21 '18 at 12:10

`void` is an operator that is used to return a `undefined` value so the browser will not be able to load a new page.

16

Web browsers will try and take whatever is used as a URL and load it unless it is a JavaScript function that returns null. For example, if we click a link like this:

```
<a href="javascript: alert('Hello World')">Click Me</a>
```

then an alert message will show up without loading a new page, and that is because `alert` is a function that returns a null value. This means that when the browser attempts to load a new page it sees null and has nothing to load.

An important thing to note about the void operator is that it requires a value and cannot be used by itself. We should use it like this:

```
<a href="javascript: void(0)">I am a useless link</a>
```

edited Jan 8 '18 at 14:22



Peter Mortensen

14.3k 19 88 116

answered Mar 15 '12 at 7:34



Amr

2,775 3 34 48

1 I have seen people use `javascript:null` instead of `void`... but that's a problem. Chrome `null` works, in Firefox, it tries to load the page `null`. Glad you updated. Interesting bug. – [Gavin Pickin](#) Dec 28 '16 at 5:08

I found some other uses in a codebase like the `javascript:null` but with `javascript:null()` which is `undefined`, so that works. – [Gavin Pickin](#) Dec 28 '16 at 5:11

1 So basically it's like jquery's `preventDefault` and return `false`? – [Robert Rocha](#) Oct 27 '17 at 14:22

To understand this concept one should first understand the void operator in JavaScript.

15

The syntax for the void operator is: `void «expr»` which evaluates `expr` and returns `undefined`.

If you implement void as a function, it looks as follows:

```
function myVoid(expr) {
    return undefined;
}
```

This void operator has one important usage that is - discarding the result of an expression.

In some situations, it is important to return undefined as opposed to the result of an expression. Then void can be used to discard that result. One such situation involves javascript: URLs, which should be avoided for links, but are useful for bookmarklets. When you visit one of those URLs, many browsers replace the current document with the result of evaluating the URLs "content", but only if the result isn't undefined. Hence, if you want to open a new window without changing the currently displayed content, you can do the following:

```
javascript:void window.open("http://example.com/")
```

edited Jan 8 '18 at 14:25



Peter Mortensen

14.3k 19 88 116

answered Aug 2 '16 at 9:16



Gopal Yadav

278 2 8

2 Thanks for clarifying exactly what the argument to 'void' is for! It wasn't clear in the other answers, only that "void takes an argument". – [dbeachy1](#) Apr 13 '17 at 18:05

Good answer but one detail, the void implementation will be something like: `function myVoid(expr) { expr(); return undefined; }` You forgot add **expr()**; – [Juanma Menendez](#) Nov 18 '18 at 4:30 ✎

@Juanma Menendez: not true. `expr` is already evaluated when `myVoid()` called (the *result* of that expression is passed as parameter) – [Udo G](#) Jan 6 at 22:49

@UdoG I am curious, how do you know that men? can you please explain. – [Juanma Menendez](#) Jan 7 at 3:07

@JuanmaMenendez: *expressions* in functions parameters are always evaluated before calling the function itself. Sorry, I have no document at hand that makes that clear but try yourself: `function() { alert("foo"); }` is a valid *expression*. `void(function() { alert("foo"); })` returns `undefined` and does **not** show the alert, whereas `myVoid(function() { alert("foo"); })` does (in your version, not the one of [Gopal Yadav](#)). – [Udo G](#) Jan 7 at 8:08 ✎

The `void` operator evaluates the given expression and then returns undefined. It avoids refreshing the page.

14

edited May 2 '14 at 14:31



katalin_2003

488 1 11 23

answered Jul 24 '13 at 13:47



Abhay Singh

594 1 12 27

Usage of `javascript:void(0)` means that the author of the HTML is misusing the anchor element in place of the button element.

12

Anchor tags are often abused with the onclick event to create pseudo-buttons by setting href to "#" or "javascript:void(0)" to prevent the page from refreshing. These values cause unexpected behavior when copying/dragging links, opening links in a new tabs/windows, bookmarking, and when JavaScript is still downloading, errors out, or is disabled. This also conveys incorrect semantics to assistive technologies (e.g., screen readers). In these cases, it is recommended to use a `<button>` instead. In general you should only use an anchor for navigation using a proper URL.

Source: [MDN's <a> Page](#).

edited Oct 22 '17 at 12:45



Styx

5,992 6 28 39

answered Oct 4 '17 at 22:43



Ron Royston

7,298 3 32 53

2 +1 for bringing up semantic html on an old question... Links go places, buttons do things - if we don't want it looking like a button, we should just clear the styling. – [kevlarr](#) Oct 11 '17 at 17:14

The big exception to this would be image maps, which may need to execute JavaScript; and since it is not a button, it is a link that has polygonal bounds, this is the "only" way. – [user4914655](#) Apr 28 '18 at 6:16

A link must have an HREF target to be specified to enable it to be a usable display object.

6

Most browsers will not parse advanced JavaScript in an

`<A HREF=""`

tag such as:

```
<A href="JavaScript:var elem = document.getElementById('foo');"
```

because the HREF tag in most browsers does not allow whitespace, or will convert whitespace to %20, the HEX equivalent of a SPACE, which renders your JavaScript absolutely useless to the interpreter.

So if you want to use an A HREF tag to execute inline JavaScript, you must specify a valid value for HREF FIRST that isn't too complex (doesn't contain whitespace), and then provide the JavaScript in an event attribute tag like OnClick, OnMouseOver, OnMouseOut, etc.

The typical answer is to do something like this:

```
<A HREF="#" onclick="var elem = document.getElementById('foo');">Get the object</a>
```

This works fine but it makes the page jump to the top because of the pound sign/hash tag tells it to do so.

Just providing a pound sign/hash tag in an A HREF tag actually specifies the root anchor, which is always, by default the top of the page, you can specify a different location by using specifying the NAME attribute inside an A HREF tag.

```
<A NAME='middleofpage'></A>
```

You can then change your A HREF tag to jump to 'middleofpage' and execute the JavaScript in the OnClick event, once that happens like so:

```
<A HREF="#middleofpage" onclick="var elem = document.getElementById('foo');">Get the object</a>
```

There will be MANY times where you do not want that link jumping around, so you can do two things:

```
<A HREF="#thislinkname" name='thislinkname' onclick="var elem = document.getElementById('foo');">Get the object</a>
```

Now it will go nowhere when clicked, but it could cause the page to re-center itself from it's current viewport. This is not pretty. What is the best way to provide in-line javascript, using an A HREF, but without having to do any of the above? JavaScript:void(0);

```
<A HREF="JavaScript:void(0);" onclick="var elem = document.getElementById('foo');">Get the object</a>
```

This tells the browser to go NOWHERE, but instead execute the valid JavaScript:void(0); function first in the HREF tag because it contains no whitespace, and will not be parsed as a URL. It will instead be run by the compiler. VOID is a keyword which, when supplied with a parameter of 0 returns UNDEFINED, which does not use any more resources to handle a return value that would occur without specifying the 0 (it is more memory-management/performance friendly).

The next thing that happens is the OnClick gets executed. The page does not move, nothing happens display-wise.

answered Apr 28 '18 at 6:12
user4914655



5

Web Developers use `javascript:void(0)` because it is the easiest way to prevent default behavior of a tag. `void(*anything*)` returns `undefined` and it is a falsy value. and returning a falsy value is like `return false` in `onclick` event of a tag that prevent its default behavior.



So I think `javascript:void(0)` is the simplest way to prevent default behavior of a tag.

answered Aug 21 '18 at 5:13



[Mohamad Shiralizadeh](#)

4,906 4 42 70



0



JavaScript: URLs aside; this is where void can be useful to write shorter code.

```
var error1 = false,
    error2 = false,
    error3 = false;

function error1() {
    error1 = true;
}

function myFunction() {
    // You can easily return and call a function at once, if you don't care about
    // myFunction's return value
    if (!someCondition1)
        return error1();

    // What if you want to set a value first?
    if (!someCondition2) {
        error2 = true;
        return
    }

    // Shortest way to return and set a value at once
    if (!someCondition3)
        return void(error3 = true);

    // More code goes here
}
```



It would be nice if the downvoters could at least tell why they downvoted. I know it's not on topic of the question, but I'm thinking about all the people that come here from the search results. – [Maciej Krawczyk](#) May 2 '18 at 9:47

Another example where is used `javascript:void(0)` . I'm not sure whether it is the correct way, but does the job

-2

```
$(document).ready(function() {
  jQuery(".show-hide-detail").hide();
  jQuery(".show-hide-detail:first").show();
  jQuery(".show-hide-btn a").click(function() {
    var thid_data = jQuery(this).attr('data-id');
    jQuery(".show-hide-btn a").removeClass('active');
    jQuery(this).addClass('active');
    if (!jQuery("#" + thid_data).is(":visible")) {
      jQuery(".show-hide-detail").hide();
      jQuery("#" + thid_data).show();
    }
  });
});

<section>
  <div class="features">
    <div class="container">
      <h1>Room Dimensions</h1>
      <p class="description">
        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tempor arcu non
        ligula convallis, vel tincidunt ipsum posuere.sollicitudin. Duis iaculis, arcu ut
        hendrerit pharetra, elit augue pulvinar magna
      </p>
      <div class="dimension-btn show-hide-btn">
        <a class="active" data-id="LivingRoom">Living Room</a>
        <a href="javascript:void(0)" data-id="DiningRoom">Dining Room</a>
        <a href="javascript:void(0)" data-id="Kitchen">Kitchen</a>
        <a href="javascript:void(0)" data-id="MasterBedroom">Master Bedroom</a>
        <a href="javascript:void(0)" data-id="Bedroom2">Bedroom 2</a>
        <a href="javascript:void(0)" data-id="Bedroom3">Bedroom 3</a>
        <a href="javascript:void(0)" data-id="Bathroom">Bathroom</a>
        <a href="javascript:void(0)" data-id="Gym">Gym</a>
      </div>
    </div>
  </div>
</section>
```

```
    <a href="javascript:void(0)" data-id="SwimmingPool">Swimming Pool</a>
  </div>
  <div class="row">
    <div class="LivingRoom Dimension-detail show-hide-detail" id="LivingRoom"
style="display: block;"></div>
    <div class="LivingRoom Dimension-detail show-hide-detail" id="DiningRoom"
style="display: none;"></div>
    <div class="LivingRoom Dimension-detail show-hide-detail" id="Kitchen"
style="display: none;"></div>
    <div class="LivingRoom Dimension-detail show-hide-detail" id="MasterBedroom"
style="display: none;"></div>
    <div class="LivingRoom Dimension-detail show-hide-detail" id="Bedroom2"
style="display: none;"></div>
    <div class="LivingRoom Dimension-detail show-hide-detail" id="Bedroom3"
style="display: none;"></div>
    <div class="LivingRoom Dimension-detail show-hide-detail" id="Bathroom"
style="display: none;"></div>
    <div class="LivingRoom Dimension-detail show-hide-detail" id="SwimmingPool"
style="display: none;"></div>
  </div>
</div>

</section>
```

[Run code snippet](#)[Expand snippet](#)

answered Dec 30 '16 at 13:00



Mile Mijatovic

1,257 12 27

1 JQuery isnt JavaScript. – user4914655 Apr 28 '18 at 6:13

protected by [Robert Harvey](#) ♦ Sep 26 '12 at 21:01

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 [reputation](#) on this site (the [association bonus does not count](#)).

Would you like to answer one of these [unanswered questions](#) instead?

