How can you check for a #hash in a URL using JavaScript?

Asked 10 years, 8 months ago Active 1 month ago Viewed 581k times

735

148

I have some jQuery/JavaScript code that I want to run only when there is a hash (`#`) anchor link in a URL. How can you check for this character using JavaScript? I need a simple catch-all test that would detect URLs like these:

- `example.com/page.html#anchor`
- `example.com/page.html#anotheranchor`

Basically something along the lines of:

```
if (thereIsAHashInTheUrl) {
    do this;
} else {
    do this;
}
```

If anyone could point me in the right direction, that would be much appreciated.

javascript jquery anchor fragment-identifier

edited Jun 24 at 3:39
a stone arachnid
1,176 1 8 23

asked Nov 18 '08 at 11:35
Philip Morton
57.8k 36 81 93

18 Answers

1318

Simple:

```
if(window.location.hash) {
  // Fragment exists
} else {
    // Fragment doesn't exist
}
```

edited Aug 30 '12 at 1:52

answered Nov 18 '08 at 11:37

Gareth
95.8k 29 138 149

39 Additional: the .location object is only available on the current window's URL, you can't do this for an arbitrary URL (e.g. one stored in a string variable) – Gareth Nov 18 '08 at 11:39

63 Also, location properties like .hash and .query are also available on <a> elements – Gareth Jul 30 '10 at 13:52

18 `.search` is available on an `<a>`, not `.query`. Sample jQuery: `$("<a/>").attr({ "href": "http://www.somewhere.com/a/b/c.html?qs=1#fragmenttest" })[0]`. `.hash => "#fragmenttest"` and `.search` = `?qs=1`. From there, hit up the querystring extraction question to get something other than a string. – patridge Jul 29 '11 at 16:48

2 @Gareth Yes, you're right. Just realized that the link I posted was for the `hashchange` event and not `window.location.hash`. Although the latter is not supported by IE < 8. – hitautodestruct May 19 '13 at 6:37

4 `(new URL("http://www.yahoo.com/#hi")).hash` – Muhammad Umer Oct 7 '15 at 16:39

300

```
if(window.location.hash) {
    var hash = window.location.hash.substring(1); //Puts hash in variable, and removes the # character
    alert (hash);
    // hash found
} else {
    // No hash found
}
```

edited Jun 1 '16 at 13:07

Martijn
12.6k 3 22 56

answered Jul 13 '11 at 16:46

Mark Notton
3,587 2 15 22

51

Put the following:

```
<script type="text/javascript">
    if (location.href.indexOf("#") != -1) {
        // Your code in here accessing the string like this
        // location.href.substr(location.href.indexOf("#"))
    }
</script>
```

edited Sep 6 '12 at 15:35
saluce
10.4k 2 43 61

answered Nov 18 '08 at 11:41
José Leal
4,531 6 30 52

1 Thanks, window.location.hash only has a value if there is a value following the #. for example. //www.example.com# returns '' when checking the hash. – Jessycormier Feb 16 '17 at 12:42

29

If the URI is not the document's location this snippet will do what you want.

```
var url = 'example.com/page.html#anchor',
    hash = url.split('#')[1];

if (hash) {
    alert(hash)
} else {
    // do something else
}
```

answered Mar 8 '12 at 10:42
Marc Diethelm
979 1 10 15

+1 for Javascript not knowing what out of bounds is :) – Dunc Aug 19 '14 at 20:04

2 @Dunc: Well JS arrays are basically Objects. Accessing them by index is like accessing an Object property like so: `obj['0']` . In JS this is true: `arr[0] === arr['0']` Therefore if the index/key doesn't exist the returned value is undefined instead of out of bounds. jsfiddle.net/web5me/Mw376 – Marc Diethelm Oct 30 '14 at 11:48

Have you tried this?

18

```
if (url.indexOf("#") != -1)
{
}
```

(Where `url` is the URL you want to check, obviously.)

answered Nov 18 '08 at 11:37
Jon Skeet
1124k 713 8168
8568

15

```
$('#myanchor').click(function(){
    window.location.hash = "myanchor"; //set hash
    return false; //disables browser anchor jump behavior
});
$(window).bind('hashchange', function () { //detect hash change
    var hash = window.location.hash.slice(1); //hash to string (= "myanchor")
    //do sth here, hell yeah!
});
```

This will solve the problem ;)

answered Aug 31 '11 at 13:44
Oral ÜNAL
185 1 13

Liked this answer coz of mixing JS+JQ, cross-browser and simple solution, already implemented this approach. – Arthur Kushman Aug 7 '16 at 16:27

12

```
window.location.hash
```

will return the hash identifier

answered Apr 27 '13 at 18:31
user2327502

141 1 4

Short and clean, is perfect – Jam Dec 6 '17 at 10:32

10

...or there's a jquery selector:

```
$('a[href^="#"]')
```

answered Apr 19 '12 at 13:18

BaronVonKaneHoffen
1,063 1 12 18

6

Here's what you can do to periodically check for a change of hash, and then call a function to process the hash value.

```
var hash = false;
checkHash();

function checkHash(){
    if(window.location.hash != hash) {
        hash = window.location.hash;
        processHash(hash);
    } t=setTimeout("checkHash()",400);
}

function processHash(hash){
    alert(hash);
}
```

answered Jul 5 '10 at 22:23

Emmanuel
3,202 5 32 58

10 thats only nessesary in ie 6 + 7. Al other browsers have included the onhashchange event – Tokimon Mar 25 '11 at 12:44

@Tokimon - great! I didn't know that. But I guess we've still got to support those old versions of IE – Emmanuel Mar 26 '11 at 8:57

1 Yeah sadly... Even IE 8 wont go away soon as IE 9 is not supported on XP :(– Tokimon Apr 5 '11 at 22:52

1 modernizr.com implements the hashchange event on older browsers (along with a bunch of other modern features) – guigouz Nov 27 '11 at 15:41

4 This is not a recommended code at all: it should at least be: `setTimeout(checkHash, 400)` . Plus, modern browsers have the `hashchange` event so you can do a `window.addEventListener('hashchange', function(){ … })` . Finally, leaking the global `hash` variable is another non-recommended practice, even for an exemple. – Oncle Tom Dec 10 '13 at 23:04

5

Most people are aware of the URL properties in document.location. That's great if you're only interested in the current page. But the question was about being able to parse anchors on a page not the page itself.

What most people seem to miss is that those same URL properties are also available to anchor elements:

```
// To process anchors on click
jQuery('a').click(function () {
   if (this.hash) {
      // Clicked anchor has a hash
   } else {
      // Clicked anchor does not have a hash
   }
});

// To process anchors without waiting for an event
jQuery('a').each(function () {
   if (this.hash) {
      // Current anchor has a hash
   } else {
      // Current anchor does not have a hash
   }
});
```

answered Sep 5 '13 at 17:32

Nicholas Davison

51 1 1

4

```
function getHash() {
  if (window.location.hash) {
    var hash = window.location.hash.substring(1);
    if (hash.length === 0) {
      return false;
    } else {
      return hash;
    }
  } else {
    return false;
  }
}
```

edited May 23 '14 at 9:27
jokklan
3,145 12 34

answered Jun 7 '13 at 23:58
user2465270
41 1

3 Should that not be "hash.length" as opposed to "hash.length()"? :) – Nathan Pitman Dec 18 '13 at 22:20

2 Please don't add code-only answers. – Daniel W. Nov 14 '14 at 12:51

3

Partridge and Gareths comments above are great. They deserve a separate answer. Apparently, hash and search properties are available on any html Link object:

```
<a id="test" href="foo.html?bar#quz">test</a>
<script type="text/javascript">
   alert(document.getElementById('test').search); //bar
   alert(document.getElementById('test').hash); //quz
</script>
```

Or

```
<a href="bar.html?foo" onclick="alert(this.search)">SAY FOO</a>
```

Should you need this on a regular string variable and happen to have jQuery around, this should work:

```
var mylink = "foo.html?bar#quz";

if ($('<a href="'+mylink+'">').get(0).search=='bar')) {
    // do stuff
}
```

(but its maybe a bit overdone ..)

edited Dec 8 '12 at 4:08
McGarnagle
88.4k 24 193 238

answered Feb 20 '12 at 22:42
commonpike
5,807 1 44 50

3

```
var requestedHash = ((window.location.hash.substring(1).split("#",1))+"?").split("?",1);
```

edited Jul 6 '13 at 10:57
Yotam Omer
13.5k 10 49 60

answered Jul 6 '13 at 10:37
Graham
211 2 3

3

Throwing this in here as a method for abstracting location properties from arbitrary URI-like strings. Although `window.location instanceof Location` is true, any attempt to invoke `Location` will tell you that it's an illegal constructor. You can still get to things like `hash`, `query`, `protocol` etc by setting your string as the `href` property of a DOM anchor element, which will then share all the address properties with `window.location`.

Simplest way of doing this is:

```
var a = document.createElement('a');
a.href = string;

string.hash;
```

For convenience, I wrote a little library that utilises this to replace the native `Location` constructor with one that will take strings and produce `window.location`-like objects: [Location.js]()

answered Sep 10 '13 at 13:58
Barney
13.8k 4 42 66

Usually clicks go first than location changes, so after a click is a good idea to setTimeOut to get updated window.location.hash

1

```
$(".nav").click(function(){
    setTimeout(function(){
        updatedHash = location.hash
    },100);
});
```

or you can listen location with:

```
window.onhashchange = function(evt){
   updatedHash = "#" + evt.newURL.split("#")[1]
};
```

I wrote a jQuery plugin that does something like what you want to do.

It's a simple anchor router.

edited Jun 21 '12 at 14:18
HackedByChinese
38.4k 8 122 137

answered Jun 20 '12 at 18:18
Rolando
121 1 3

0

Here is a simple function that returns `true` or `false` (has / doesn't have a hashtag):

```
var urlToCheck = 'http://www.domain.com/#hashtag';

function hasHashtag(url) {
    return (url.indexOf("#") != -1) ? true : false;
}

// Condition
if(hasHashtag(urlToCheck)) {
    // Do something if has
}
else {
    // Do something if doesn't
}
```

Returns `true` in this case.

Based on @jon-skeet's comment.

answered Nov 29 '15 at 11:22

dnns
152 2 9

0

This is a simple way to test this for the current page URL:

```
function checkHash(){
    return (location.hash ? true : false);
}
```

answered Jul 18 '18 at 9:31

aabiro
547 1 8 18

-2

sometimes you get the full query string such as "#anchorlink?firstname=mark"

this is my script to get the hash value:

```
var hashId = window.location.hash;
hashId = hashId.match(/#[^?&\/]*/g);

returns -> #anchorlink
```

answered Oct 27 '14 at 8:45

markg
91 1 6

5 Not possible as hash is appended after the query string and is never sent to server. The only time hash would appear before query string is when you modified the url by hand. – user1537415 Dec 6 '14 at 18:52

1 yep, but sometimes people send urls on this format. this is just so you can get only the hash value and neglect the others. :) – markg Jul 27 '15 at 8:43