

Should I use window.onload or script block?

Asked 16 years, 3 months ago Modified 9 years, 7 months ago Viewed 8k times



14



I have a javascript function that manipulates the DOM when it is called (adds CSS classes, etc). This is invoked when the user changes some values in a form. When the document is first loading, I want to invoke this function to prepare the initial state (which is simpler in this case than setting up the DOM from the server side to the correct initial state).

Is it better to use window.onload to do this functionality or have a script block after the DOM elements I need to modify? For either case, why is it better?

For example:

```
function updateDOM(id) {  
    // updates the id element based on form state  
}
```

should I invoke it via:

```
window.onload = function() { updateDOM("myElement"); };
```

or:

```
<div id="myElement">...</div>  
<script language="javascript">  
    updateDOM("myElement");  
</script>
```

The former seems to be the standard way to do it, but the latter seems to be just as good, perhaps better since it will update the element as soon as the script is hit, and as long as it is placed after the element, I don't see a problem with it.

Any thoughts? Is one version really better than the other?

javascript dom

Share

Improve this question

Follow

edited May 8, 2015 at 17:55



canolucas

1,490 ● 1 ● 15 ● 32

asked Sep 3, 2008 at 0:32



Mike Stone

44.6k ● 30 ● 114 ● 140

9 Answers

Sorted by: Highest score (default)



The onload event is considered the proper way to do it, but if you don't mind using a javascript library, jQuery's `$(document).ready()` is even better.

17



```
$(document).ready(function(){  
    // manipulate the DOM all you want here  
});
```



The advantages are:

1. Call `$(document).ready()` as many times as you want to register additional code to run - you can only set `window.onload` once.
2. `$(document).ready()` actions happen as soon as the DOM is complete - `window.onload` has to wait for images and such.

I hope I'm not becoming The Guy Who Suggests jQuery On Every JavaScript Question, but it really is great.

[Share](#) [Improve this answer](#) [Follow](#)

answered Sep 3, 2008 at 1:19



[Neall](#)

27.1k ● 5 ● 50 ● 49



10



I've written lots of Javascript and `window.onload` is a **terrible** way to do it. It is brittle and waits until **every** asset of the page has loaded. So if one image takes forever or a resource doesn't timeout until 30 seconds, your code will not run before the user can see/manipulate the page.

Also, if another piece of Javascript decides to use `window.onload = function() {}`, your code will be blown away.

The proper way to run your code when the page is ready is wait for the element you need to change is ready/available. Many JS libraries have this as built-in functionality.

Check out:

- <http://docs.jquery.com/Events/ready#fn>
- <http://developer.yahoo.com/yui/event/#onavailable>

Share Improve this answer Follow

answered Sep 3, 2008 at 2:12



[Ryan Doherty](#)

38.7k ● 4 ● 57 ● 62



5



Definitely use `onload`. Keep your scripts separate from your page, or you'll go mad trying to disentangle them later.

Share Improve this answer Follow

answered Sep 3, 2008 at 0:37



[John Millikin](#)

201k ● 41 ● 215 ● 227



3



Some JavaScript frameworks, such as [mootools](#), give you access to a special event named "domready":

Contains the window Event 'domready', which will execute when the DOM has loaded. To ensure that DOM elements exist when the code attempting to access them is executed, they should be placed within the 'domready' event.

```
window.addEvent('domready', function() {  
    alert("The DOM is ready.");  
});
```



matt b

140k ● 66 ● 284 ● 350



2



window.onload on IE waits for the binary information to load also. It isn't a strict definition of "when the DOM is loaded". So there can be significant lag between when the page is perceived to be loaded and when your script gets fired. Because of this I would recommend looking into one of the plentiful JS frameworks (prototype/jQuery) to handle the heavy lifting for you.



Share Improve this answer Follow

answered Sep 3, 2008 at 1:25



TonyLa

734 ● 4 ● 6



1



@The Geek

I'm pretty sure that window.onload will be called again when a user hits the back button in IE, but doesn't get called again in Firefox. (Unless they changed it recently).



In Firefox, `onload` is called when the DOM has finished loading regardless of how you navigated to a page.



Share Improve this answer Follow

answered Sep 3, 2008 at 1:15



John Millikin

201k ● 41 ● 215 ● 227



1



While I agree with the others about using window.onload if possible for clean code, I'm pretty sure that window.onload will be called again when a user hits the back button in IE, but doesn't get called again in Firefox. (Unless they changed it recently).

Edit: I could have that backwards.



In some cases, it's necessary to use inline script when you want your script to be evaluated when the user hits the back button from another page, back to your page.



Any corrections or additions to this answer are welcome... I'm not a javascript expert.

Share

edited Sep 3, 2008 at 1:22

answered Sep 3, 2008 at 1:13

Improve this answer



[The How-To Geek](#)

1,105 ● 9 ● 13

Follow



My take is the former because you can only have 1 window.onload function, while inline script blocks you have an n number.

0

Share Improve this answer Follow



answered Sep 3, 2008 at 0:39



[David Basarab](#)

73.2k ● 43 ● 130 ● 157



onLoad because it is far easier to tell what code runs when the page loads up than having to read down through scads of html looking for script tags that might execute.

0

Share Improve this answer Follow



answered Sep 3, 2008 at 1:09



[DancesWithBamboo](#)

4,156 ● 1 ● 20 ● 20

