$(document).ready(function(){}); vs script at the bottom of page

Asked 13 years, 7 months ago Modified 9 years, 2 months ago Viewed 44k times

93

what is the difference / advantage / disadvantage of writing script at the bottom of the page and writing the script in

```
$(document).ready(function(){});
```

javascript jquery

Share Improve this question Follow

asked May 17, 2011 at 5:36
Sourav
17.5k ● 35 ● 105 ● 161

learningjquery.com/2006/09/introducing-document-ready – Saleh May 17, 2011 at 5:45

2 Answers

Sorted by: Highest score (default)

144 ✓

Very little in and of itself, either way the DOM will be ready for you to operate on (I was nervous about that until I read this from Google). If you use the end of page trick, your code may get called the slightest, slightest bit sooner, but nothing that will matter. But more importantly, this choice relates to where you link your JavaScript into the page.

If you include your `script` tag in the `head` and rely on `ready`, the browser encounters your `script` tag before it displays anything to the user. In the normal course of events, the browser comes to a screeching halt and goes and downloads your script, fires up the JavaScript interpreter, and hands the script to it, then waits while the interpreter processes the script (and then jQuery watches in various ways for the DOM to be ready). (I say "in the normal course of things" because some browsers support the `async` or `defer` attributes on `script` tags.)

If you include your `script` tag at the end of the `body` element, the browser doesn't do all of that until your page is largely already displayed to the user. This improves perceived load time for your page.

So to get the best perceived load time, put your script at the bottom of the page. (This is also the guideline from the Yahoo folks.) And if you're going to do that, then there's no need to use `ready`, though of course you could if you liked.

There's a price for that, though: You need to be sure that the things the user can see are ready to be interacted with. By moving the download time to after the page is largely displayed, you increase the possibility of the user starting to interact with the page before your script is loaded. That's one of the counter-arguments to putting the `script` tag at the end. Frequently it's not an issue, but you have to look at your page to see whether it is and, if so, how you want to deal with it. (You can put a small *inline* `script` element in the `head` that sets up a document-wide event handler to cope with this. That way, you get the improved load time but **if** they try to do something too early, you can either tell them that or, better, queue the thing they wanted to do and do it when your full script is ready.)

Share

Improve this answer

Follow

edited May 28, 2015 at 6:56

answered May 17, 2011 at 5:47

T.J. Crowder

1.1m ● 199 ● 2k ● 1.9k

17 +1. Another way to deal with interaction before the js is loaded is to make the page work without javascript to begin with. Make sure all links work etc, although they will of course trigger a page reload. Then hijack the links and other stuff with js, and add your ajax or other bells and whistles :) – Adrian Schmidt May 17, 2011 at 5:54

1 How would you handle queuing an action? Is there a design pattern or some "common solution" that I can look into regarding this? – HC_ Aug 8, 2016 at 19:24

@HC_: What do you mean by "queuing an action"? – T.J. Crowder Aug 8, 2016 at 20:53

@T.J.Crowder From the last sentence of your answer: "or, better, queue the thing they wanted to do and do it when your full script is ready." I was just wondering, if there are "optimal" ways of doing this, since I'm sure I could figure out "some" way of doing this, but I'm sure it would be... frowned upon if anyone else ever looked at whatever code is involved. – HC_ Aug 8, 2016 at 20:59

2 @HC_: I've never felt the need to do it, and don't know of any particular standard for it. And five years later, it seems to me it'd be better to just go with "Sorry, loading, one sec.." instead. (Although if things are *that* slow to load, you have a bigger problem.) If I see something I think is clickable, and click it, and nothing happens, heaven only knows what I'll click next. If you queue them up and then play them back when you're ready, it could be...less than ideal. – T.J. Crowder Aug 8, 2016 at 21:06

3

Your page will load slower by scattering `$(document).ready()` scripts throughout the DOM (instead of all at the end) because it requires jQuery to be **synchronously** loaded first.

`$ = jQuery`. So you can't use `$` in your scripts without first loading jQuery. This approach forces you to load jQuery near the beginning of the page, which **will halt your page load** until jQuery is fully downloaded.

You cannot `async` load jQuery either because in many cases, your `$(document).ready()` script(s) will try to execute before jQuery is fully async loaded and cause an error, because again, `$` isn't defined yet.

That being said, there is a way to fool the system. Essentially setting `$` equal to a function that pushes `$(document).ready()` functions into a queue/array. Then at the bottom of the page, load jQuery then iterate through the queue and execute each `$(document).ready()` one at a time. This allows you to defer jQuery to the bottom of the page but still use `$` above it in the DOM. I personally haven't tested how well this works, but the theory is sound. The idea has been around for a while but I've very, very rarely seen it implemented. But it seems like a great idea:

http://samsaffron.com/archive/2012/02/17/stop-paying-your-jquery-tax

Share Improve this answer Follow

answered Oct 22, 2015 at 21:59

Jake Wilson

91k ● 96 ● 259 ● 371