

# Should you design websites that require JavaScript in this day & age?

Asked 16 years, 3 months ago

Modified 9 years ago

Viewed 8k times



It's fall of 2008, and I still hear developers say that you should not design a site that requires JavaScript.

15



I understand that you should develop sites that degrade gracefully when JS is not present/on. But at what point do you not include functionality that can only be powered by JS?



I guess the question comes down to demographics. Are there numbers out there of how many folks are browsing without JS?

javascript

Share

edited Sep 26, 2009 at 16:44

Improve this question

Follow

community wiki

2 revs, 2 users 100%

Baer

**27**

Just as long as you're aware of the accessibility limitations you might be introducing, ie for users of screen-reading software, etc.



It's one thing to exclude people because they choose to turn off JS or use a browser which doesn't support it, it's entirely another to exclude them because of a disability.



Share Improve this answer

answered [Sep 15, 2008 at 13:24](#)

Follow

community wiki

[warren\\_s](#)

---

While answered awhile ago, anyone coming across this might be interested in this page [a11yproject.com/posts/myth-screen-readers-dont-use-javascript](http://a11yproject.com/posts/myth-screen-readers-dont-use-javascript) which highlights, "A common misconception among web developers is that screen readers only read the non-JavaScript page. Due to this misconception, we sometimes assume it's unnecessary to make JavaScript apps and other functionality accessible. This is categorically false." – [Software Prophets](#) Jul 8, 2015 at 11:39

---

**8**

Two simple questions to help you decide...

1. Does using javascript provide some core functionality of your site?



2. Are you prepared to limit your potential users to those who have JS? (ie. Most people)



If you answer yes to both of those, go for it!



Websites are moving (have moved?) from static pages of information to interactive web applications. Without something like Javascript or Flash, making compelling user interactions is sometimes not possible.

Share Improve this answer

answered [Sep 15, 2008 at 13:23](#)

Follow

community wiki

[Marc Hughes](#)



4

Designing to degrade gracefully is the most that should be done. We are moving/have moved past the point of simple web "sites" to web "applications". The only option besides client side scripting to add round trips to the server.



I think (personal opinion) that the "don't use JavaScript" comes more from a lack of understanding of what JavaScript is/does than any actual market data that shows a significant number of people are browsing without it.



Share Improve this answer

answered Sep 15, 2008 at 13:24

Follow

community wiki

[user1921](#)



3



It's reasonable to design sites that use JavaScript but it is not safe to assume that all clients have support for Javascript and therefore it is important that you provide a satisfactory experience even when JavaScript is not available



Share Improve this answer

answered Sep 15, 2008 at 13:27



Follow

community wiki

[Crippledsmurf](#)



3



Search Engines don't support JavaScript. They're also blind and don't support CSS. So my suggestion to you is to make sure that the part of your product that needs to be indexable by search engines works without JavaScript and CSS. After that, it really depends on the needs of your users.



If you have a very limited subset of users, then you can actually query them. But to remember that 10% of the population has some form of impairment ranging from vision issues (low vision, color-blindness, etc.) or motor

functions (low hand dexterity). These problems tend to be more prominent in the elderly and the knowingly disabled

If your site will target the general audience of Internet users then please make it degrade gracefully, but if you can't do that, then make a no-JavaScript version (like G-mail has).

Share Improve this answer

answered [Sep 15, 2008 at 14:21](#)

Follow

community wiki

[Orion Adrian](#)

---



2



I think the days of "content sites only" are gone. What we see now is WWW emerging as the platform of web applications, and the latest developments in the browser front (speeding up JS in particular) ar indication of this. There can be no yes/no answer to your question - you should decide, where on content site<---->web application continuum your site is and how essential is the experience provided by JavaScript. In my opinion - yes it is acceptable to have web applications which require Javascript to function.

Share Improve this answer

answered [Sep 15, 2008 at 13:24](#)

Follow

community wiki  
[Rimantas](#)



2



Degrading gracefully is a must. At a minimum, you sure make use of the NOSCRIPT tag in order to inform potential *customers* first that your site requires javascript, and secondly why you require it.

If it's for flashy menus and presentations that I could honestly care less about then I probably won't bother coming back. If there's a real reason that you're requiring javascript (client-side validation on forms, or a real situation that requires AJAX for performance reasons) then say so and your visitors will respond accordingly.

I install extensions that limit both Javascript and Cookies. Websites that don't prominently state their requirements of both usually don't get a second visit unless there's a real need for it.

Share Improve this answer  
Follow

answered [Sep 15, 2008 at 13:31](#)

community wiki  
[Peter Bernier](#)



2



You should never design a public site to rely on ANY technology/platform. The user agent may not display colour (think screen readers), display graphics (again, think screen readers or text only browsers such as links), etc.



Design your site for the lowest common denominator and then progressively enhance it to add support for specific technologies.

To answer the question directly: No, you cannot assume your users have Javascript, so your site should work without it. Once it does, enhance it with Javascript.

Share Improve this answer  
Follow

answered [Sep 15, 2008 at 13:40](#)

community wiki  
[RichardAtHome](#)

---

1 "You should never design a public site to rely on ANY technology/platform" - LOL. Really? You mean relying on some specific technology, like, say HTML or HTTP would be bad? :P Ha ha. Give me a break, at some point you have to rely on something, and JavaScript is now at that point

– [Jason Bunting](#) Sep 15, 2008 at 20:49

---

1 And what of your visitors who use screen readers or other assistive technologies that don't support Javascript?. Don't they deserve to access your content too? – [RichardAtHome](#)

Jan 2, 2009 at 16:20

---



2



it's not about browser capability, it's about user control. People who install the noscript plugin for firefox so they don't have to put up with punch-the-monkey garbage ( the same problem that inspired stack overflow) will not allow your web site to do anything non-static until they trust you.



Share Improve this answer

answered [Sep 15, 2008 at 13:46](#)

Follow

community wiki  
[davidnicol](#)

---





2



In terms of client software consider users/customers who are using a browser that supports some but not all Javascript. For example, most mobile phone browsers support a bit of Javascript but nothing very complicated. The browsers on devices such as the Playstation 3 are similar.



Then there are browsers such as [Opera Mini](#), which support a lot of Javascript but are operating in an environment where the scripts are running on a server that then sends the results to a mobile device.

Share Improve this answer

answered Sep 15, 2008 at 13:50

Follow

community wiki  
[Kevin ORourke](#)



2



You should design websites with Javascript in mind--but not implemented. Consider, build it where every click, every action, performs a round trip to the server. That's the default functionality for older browsers, and those without JS turned on.



Then, after it's all built, and everything is working properly, add in JavaScript which hijacks the link, button and other events, and overlay their standard functionality with the Javascript functionality you're wanting.

Building the application like this means that it will ALWAYS work, which ultimately is what you're wanting.

Share Improve this answer

answered Sep 15, 2008 at 14:00

Follow

community wiki

Stephen Wrighton



2

The received wisdom answer is that you can use JavaScript (or any other technology) providing that it 'degrades gracefully'...



I have experience with disability organisations, so accessibility is important to me. But equally, I'm in the business of building attractive, usable websites, so javascript can be a powerful ally. It's a difficult call, but if you can build a rich, javascript-aided site, without completely alienating non-js visitors, then do so. If not, you will have to look at the context of the site and decide which way to jump.

Regardless, there are no rights and wrongs with this question. However, in some countries, there is a requirement to build 'public' sites to be accessible, so this may be yet another factor in your decision. [In the UK, it is the Disability Discrimination Act.. though to my knowledge, no company has been prosecuted for failure to comply]

**2**

JavaScript is great for extending the browser to do things like google maps. But it's a pointy instrument, so use it with care.



My bank web site uses JavaScript for basic navigation between pages. **Sigh**. As a result, it's not usable from my mobile device.



Make sure you're familiar with the [Rule of Least Power](#) when considering JavaScript:

When designing computer systems, one is often faced with a choice between using a more or less powerful language for publishing information, for expressing constraints, or for solving some problem. This finding explores tradeoffs relating the choice of language to reusability of information. The "Rule of Least Power" suggests choosing the least powerful language suitable for a given purpose.



5% according to these statistics:

[http://www.w3schools.com/browsers/browsers\\_stats.asp](http://www.w3schools.com/browsers/browsers_stats.asp)

1



Share Improve this answer

answered Sep 15, 2008 at 13:23

Follow



community wiki  
Antonio Cangiano



---

7 This number covers only the users of w3school.com I think, and might be a bit atypical - you can see the Firefox percentage is unusually high. – Kristian J. Sep 15, 2008 at 13:30

---



1

As you said, demographics. The web is expanding onto devices that doesn't have very much power, for instance cellphones. If your site is usable without javascript, Opera Mini will likely show your site without any problems.



Share Improve this answer

answered [Sep 15, 2008 at 13:22](#)



Follow



community wiki  
[RobbieGee](#)



1

I think Javascript implementations in most modern browsers have now reached a reasonable level of maturity and there are a bunch of Javascript UI frameworks which let you build very attractive Javascript based web applications using web-services and such (regardless of the back-end server platform).



An example is [ExtJS](#) - they have got a very extensive AJAX + UI widget framework which I recently used to build a full fledged internal web-app for a client with an ASP.NET backend (for webservice).

Share Improve this answer

answered [Sep 15, 2008 at 13:23](#)

Follow

community wiki  
[Krishna](#)



1



I think it comes down to what you're about to do. Are you writing a web APPLICATION? Then I think you're bound to use javascript and/or something like [GWT](#). Just have a look at all the social sites, and google applications like gmail. If you're writing a webpage with product descriptions and hardly any interactivity, then you can make the javascript optional.

Share Improve this answer

answered [Sep 15, 2008 at 13:26](#)

Follow

community wiki  
[Chris M.](#)



1



I agree with the majority of the stackoverflow respondents. JavaScript has matured and offers an "extra" level of functionality to a webpage, especially for forms. Those who turn off cookies and JS have likely been bitten while surfing in dangerous waters. For the corporate power users that pay my way either in B2B or retail sites, JS is a proven and trusty tool. Until something better comes along (and it will) I'm sticking with JS.

Share Improve this answer

answered [Sep 15, 2008 at 13:57](#)

Follow

community wiki  
[user7428](#)



1

There's add-on for Firefox called [NoScript](#) which have 27,501,701 downloads. If your site won't work without JavaScript most of those guys wouldn't want to use it.



Why would you install that add-on? Ever wanted to get rid of the popup on the site that covers the most of the useful text you want to read? Or disable flash animation? Or be sure that evil site won't steal your cookies?



Share Improve this answer

answered [Sep 15, 2008 at 14:11](#)

Follow

community wiki  
[vava](#)

---

NoScript users could always whitelist the site, they'd just have to be made aware that they need to do so.

– [Sam Hasler](#) [Sep 15, 2008 at 14:15](#)

---



1

Some corporate environments won't allow Javascript, by policy or by firewall. It closes the door to one avenue of virus infection.



Whether you think this is a good idea or not, realize that not everyone has full control over their browser and it might not be their choice.



Share Improve this answer

answered Sep 15, 2008 at 14:50

Follow

community wiki

[Mark Ransom](#)



1

There is a gradient between web sites and web applications. However, you should always be able to say "we are building a web site" or "we are building a web application".



Web sites should be readable down to plain HTML (no CSS, no images, no JavaScript).



Web applications, of course, could just say "Sorry, JavaScript is needed" (which also assumes CSS for layout). Application should still be able to work without images.

Share Improve this answer

answered Sep 15, 2008 at 15:10

Follow

community wiki

[squadette](#)



1

The accessibility issue is the only important technical issue, all other issues can be socially engineered. When one says that javascript reduced accessibility and another says that Web Applications can use javascript, can we





take these two together to imply that all blind people are unemployed? There has to be some kind of momentum in making javascript accessible. Maybe a Screenreader object on the javascript side which can detect the presence of a screenreader and then maybe send hints to the screenreader, Screenreaders which can hook onto the browser, and maybe it gets glued together with a screenreader toolbar.

Share Improve this answer

answered [Sep 15, 2008 at 15:23](#)

Follow

community wiki  
[Kinjal Dixit](#)



**1**

if you want your site viewable by the top 100 companies in the US. I would write without javascript.

Share Improve this answer

answered [Sep 15, 2008 at 18:52](#)

Follow



community wiki  
[mikedopp](#)



1



Independence from Javascript and graceful degradation are important to an application despite the actual demographics -- because such an application probably has better software design.

The "human user without Javascript" may be purely hypothetical (for example, if you're trying to make money with your product). But designing for that hypothetical user encourages modular software design which will pay off as you continue to develop your app.

Javascript provides functionality. HTML provides data (on the page itself, and via links that point to more data). As a general rule that reaches well beyond browser apps: A well-designed software product will separate data from functionality. All data should be available, and the functionality should be a separate layer that consumes the data.

If your Javascript is creating data at runtime, then it's time to get specific and figure out whether your webpage really is a piece of software (e.g. a mortgage calculator) or whether it's a document containing data (e.g. a list of mortgage interest rates). This should tell you whether it makes sense to rely on Javascript.

As a final note/example, demographics can be misleading. Relatively few humans browse your site without Javascript, but lots of **machines** (search bots, data miners, screen readers for the disabled, etc.) are browsing your site without Javascript. Again, the

distinction between data and functionality are important -- the bots are just making requests and looking for data in the responses. They don't need functionality. But if your user needs to invoke functions just to make your data accessible, the bots are getting no value from your site.

---

One side point about the screen readers and other accessibility considerations for the disabled. This is an important niche demographic: a mind that navigates data in a human way, but who can only get data from your site in the same way machines get it. By providing data cleanly and semantically on your page, you make it available to the largest possible set of accessibility tools.

*Note this doesn't exclude Javascript from consideration.* Our mortgage calculator example can still work: accept input from the user, invoke Javascript, and write the output back into the clean semantic data layer of the page. Screen readers can then read it! And if they can't, you're encouraging the development of better screen readers that can.

Share Improve this answer

answered [Sep 15, 2008 at 18:53](#)

Follow

community wiki  
[Travis Wilson](#)



If you expect your app to work for everyone, you'll need a backup for all your javascript functionality. If it's form

1

validation, you should also check the data on the server before saving it. So the answer is Yes, it's okay, but have a backup. Do not rely on it.



Share Improve this answer

answered [Sep 16, 2008 at 1:05](#)



Follow

community wiki

[Mike](#)



1

As many people are saying, it's important to consider your user base, but whoever your users are there's a strong possibility that some (stats say 10%) of them will have some sort of disabilities, and screen readers don't like javascript. If you're only adding simple things, a javascript menu or something, then just make it degrade (or don't do it). If the site depends on javascript to work properly, make two versions, one for javascript and one without.



I generally find that anything too javascript heavy is very difficult to make degrade well without just having javascript re-writing the page to a javascript version if the user can take it. Given this, it's well worth writing two pages from square one for complicated stuff.

I would say that there are very very very few web sites that should be running without some support for users without javascript. You'd need to have a very dynamic application that completely didn't make sense as static

pages, or you'd need to have an audience you could guarantee were OK with it (like on an office Intranet say).

Share Improve this answer

answered Feb 22, 2009 at 13:51

Follow

community wiki

Tim Perry



1



Well, it depends on your userbase. If you know that people will be using your site from mobile devices, it's good to have unobtrusive JavaScript. However, if you're trying to appeal to a tech-savvy crowd, don't bother with it.



However, if you're appealing to a crowd that may be using screen readers (blind people), I'd highly suggest using [WAI-ARIA](#) standards. Dojo's widget system has full support for this, and would be a great and easy way to do it.

Anyway, in most cases, you don't need unobtrusive JavaScript. Most people who have JavaScript disabled are either using a smartphone, using Lynx, or have NoScript installed. It's enabled by default in all the major browsers, so you shouldn't have to worry.

Lastly, it's good to at least have some unobtrusive JavaScript. `<noscript>` tags are your best friend. For example, one may want to replace a widget that draws rating stars with text. Example using dojo:

```
<div dojoType="dojox.Rating" stars="5" value="4"></div>  
<noscript>4/5</noscript>
```

Share Improve this answer

edited Dec 17, 2015 at 10:55

Follow

community wiki

2 revs, 2 users 89%

HFLW



-1



It's the 21st century. People not permitting JavaScript need to exit the last millennium, posthaste. It's a mature, widely used, and very useful technology that is one of the foundations of the recent expansion in useful web services.



Share Improve this answer

answered Sep 15, 2008 at 13:23



Follow

community wiki

ceejayoz



-1



You should be tying the functionality of your website to your audience. That being said, every modern browser (save for the mobile platform) includes javascript, and so unless your audience includes luddites with decade old computers, you can assume they *have* javascript.



The people you need to worry about, then, are those that specifically turn it off. This includes:

- Corporate networks with tough security (not common, but some financial and defense institutions)
- Paranoid web-heads

So, first, who is your audience? Are there other websites that are comparable to your target? Look at their site and success - do they degrade gracefully, and would you be satisfied with their level of success?

If you are targeting mobile applications, though, you can't guarantee javascript.

-Adam

Share Improve this answer

answered [Sep 15, 2008 at 13:27](#)

Follow

community wiki

[Adam Davis](#)



-1



I would say that you should look at your target audience. If you can reasonably expect that they will have js enabled, and making everything work without any js is too much of a pain, then by all means - go ahead and ignore the non-js crowd, if, on the other hand you have to create a site that will be used by a very large audience/or you are perhaps building a government web site, then you



must make sure that everything works, and it is easier in those cases to first build the site so that it works without any js, and add all the nice time-saving ajaxy bits later.

In general though, almost everyone has js enabled by default.

Though you should be aware that server-side validation of user posted data is a must in either case.

Share Improve this answer

answered [Sep 15, 2008 at 13:27](#)

Follow

community wiki

[Ramuns Usovs](#)

---