How to detect if JavaScript is disabled?

Asked 16 years, 3 months ago Modified 6 months ago

Viewed 527k times

712

There was a post this morning asking about how many people disable JavaScript. Then I began to wonder what techniques might be used to determine if the user has it disabled.

Does anyone know of some short/simple ways to detect if JavaScript is disabled? My intention is to give a warning that the site is not able to function properly without the browser having JS enabled.

Eventually I would want to redirect them to content that is able to work in the absence of JS, but I need this detection as a placeholder to start.

javascript html css

Share

Improve this question

Follow

edited Jan 5 at 7:02

community wiki
6 revs, 4 users 47%
MikeJ

What do you want to do with this information? It could change the answer you get - e.g. progressive enhancement should generally be favoured over trying to detect JavaScript being disabled and taking specific action based on that.
– Jonny Buchanan Sep 23, 2008 at 14:14

progressive enhancement is what I am looking for. I want ot be able to redirect them to alternate content that will function properly in the abscence of a JS enable or capable browser.
– MikeJ Sep 23, 2008 at 14:25

1 @expiredninja This post from Nicholas Zakas says about 2%, though it is over a year old. – sdleihssirhc Dec 12, 2011 at 20:47

The easiest way is to use noscript to show non javascript site, and use javascript to show any javascript depend elements by modifying style display. – Myforwik Feb 13, 2012 at 3:12

enable-javascript.com – Vitim.us Oct 9, 2013 at 19:36

41 Answers

Sorted by: Highest score (default)

398

I'd like to add my .02 here. It's not 100% bulletproof, but I think it's good enough.

The problem, for me, with the preferred example of putting up some sort of "this site doesn't work so well without Javascript" message is that you then need to make sure that your site works okay without Javascript.

And once you've started down that road, then you start realizing that the site should be bulletproof with JS turned off, and that's a whole big chunk of additional work.

So, what you really want is a "redirection" to a page that says "turn on JS, silly". But, of course, you can't reliably do meta redirections. So, here's the suggestion:

```
<noscript>
    <style type="text/css">
        .pagecontainer {display:none;}
    </style>
    <div class="noscriptmsg">
    You don't have javascript enabled.  Good luck with
    </div>
</noscript>
```

...where *all* of the content in your site is wrapped with a div of class "pagecontainer". The CSS inside the noscript tag will then hide all of your page content, and instead display whatever "no JS" message you want to show. This is actually what Gmail appears to do...and if it's good enough for Google, it's good enough for my little site.

Share Improve this answer Follow

edited Jan 25, 2012 at 17:47

community wiki
2 revs, 2 users 97%
hairbo

12 @steve ever try to validate google.com? validation just determines how close the document follows the spec sheet,

it in no way effects the usability of the page. – JKirchartz Feb 23, 2011 at 18:00

82 @JonasGeiregat What's wrong with forcing the user to require Javascript if your web app was built - and requires - Javascript to function? I hear this comment too often, yet it's only valid for the most simple of sites. Try Facebook without JS and see what happens. If your app was built FOR Javascript, then what's wrong with requiring what 99% of users have enabled anyway? – Lee Benson Mar 5, 2012 at 11:13

24 @Lee hits it on the nail, you can only do so much for the user before you draw the line on what you will support. We expect all users to have an internet enabled device, should I also implement a paper version of my site for those that refuse to access the internet? – Kolors Sep 4, 2014 at 13:59

19 @Kolors, I made that comment back in Mar 2012, and I'd argue that it's doubly-true today. Since then, we've had Meteor, Angular.Js, Famo.us, and a ton of amazing libraries that all require Javascript to even *start*. I do wonder what users who choose to disable JS are actually trying to achieve... clearly, they're not surfing the same web as the rest of us, and there's no reason a company should need to scale down their website for the tiniest sub-set of stubborn users... – Lee Benson Sep 4, 2014 at 18:59

7 Yeah, and I wondered why I couldn't upvote your answer, since I forgot to activate my JS :-)) I like and implement your solution! Thanks! – Garavani May 12, 2015 at 5:26

311

I assume you're trying to decide whether or not to deliver JavaScript-enhanced content. The best implementations degrade cleanly, so that the site will still operate without JavaScript. I also assume that you mean server-side

detection, rather than using the `<noscript>` element for an unexplained reason.

There is no good way to perform server-side JavaScript detection. As an alternative it is possible to set a cookie using JavaScript, and then test for that cookie using server-side scripting upon subsequent page views. However this would be unsuitable for deciding what content to deliver, as it would not distinguish visitors without the cookie from new visitors or from visitors who did not accept the JavaScript set cookie.

Share Improve this answer Follow

edited Sep 15, 2022 at 9:49

community wiki
7 revs, 4 users 50%
Marc Gear

135 <noscript> IS the most semanticly accurate way to specify non-javascript content - and rather then detecting if javascript is disabled, detect if it's enabled. So show the "you need javascript to properly use my site" message by default, but hide it with a javascript function immediately onLoad. – Matt Lohkamp Sep 23, 2008 at 23:58

62 @matt lohkamp: Or even hide the message by default, and put a `<style>` block inside the `<noscript>` to unhide (no reflow there if JS enabled). Surprisingly, this works in all modern browsers, and even in IE6
– Piskvor left the building Jul 16, 2010 at 14:31

7 @matt - I was doing this for a while, but the message stays up for a while on slow connections leaving users confused.

I'm here looking for a different solution. @Piskvor - the <noscript> tag in the header will not validate, and a <style> tag in the body will not validate. – Ben Nov 12, 2010 at 6:25

21 ...except why are you concerned with validation? Are you validating just to validate? It may not be "correct" but if it works and accomplishes it "with a little dirt under the hood" what's the issue? Don't worry, I won't judge ;-) – keif Nov 29, 2010 at 15:54

5 It stays valid if you just use the style attribute to the given elements within the <noscript> tags (like <noscript> blahblah </noscript> or sg. similar). BUT it also stays valid if you put the <style> BLOCK normally in the header defining the style of some .noscript (or similar) classed elements, and in the body, within the <noscript> tags, you just give the given elements the previously defined .noscript class, like this: <noscript> blahblah </noscript> – Sk8erPeter May 19, 2011 at 11:13

204

`noscript` blocks are executed when JavaScript is disabled, and are typically used to display alternative content to that you've generated in JavaScript, e.g.

```
<script type="javascript">
    ... construction of ajaxy-link,  setting of "js-en
</script>
<noscript>
    <a href="next_page.php?nojs=1">Next Page</a>
</noscript>
```

Users without js will get the `next_page` link - you can add parameters here so that you know on the next page whether they've come via a JS/non-JS link, or attempt to

set a cookie via JS, the absence of which implies JS is disabled. Both of these examples are fairly trivial and open to manipulation, but you get the idea.

If you want a purely statistical idea of how many of your users have javascript disabled, you could do something like:

```
<noscript>
    <img src="no_js.gif" alt="Javascript not enabled"
</noscript>
```

then check your access logs to see how many times this image has been hit. A slightly crude solution, but it'll give you a good idea percentage-wise for your user base.

The above approach (image tracking) won't work well for text-only browsers or those that don't support js at all, so if your userbase swings primarily towards that area, this mightn't be the best approach.

Share Improve this answer Follow

edited Feb 10, 2014 at 19:16

community wiki
5 revs, 2 users 97%
ConroyP

9 This isn't very effective. For example, it won't count anybody with text-only browsers, which normally don't have JavaScript support. At the very least, you should disable caching for that image. – Jim Sep 23, 2008 at 14:18

3 Would browsere which don't support JS at all not simply ignore the noscript tag and show the image? – LKM Oct 13, 2008 at 9:23

1 @LKM: Depends on how they're written, most likely they would, so you'd make it a 1x1px dot. That option is mainly for tracking usage patterns server-side, so would still be ok, as it is triggered by a user without javascript capability. – ConroyP Oct 13, 2008 at 13:30

Note that there is currently a bug with IE8 and the noscript tag if you style it... see positioniseverything.net/explorer.html – alex Jun 1, 2009 at 0:11

2 also you can serve that image as a server-side script, setting the mime-type, and use that to log some information about the user or drop a cookie... px.sklar.com/code.html/id=256 – JKirchartz Mar 4, 2011 at 15:29

52

This is what worked for me: it redirects a visitor if javascript is disabled

```
<noscript><meta http-equiv="refresh" content="0; url=w
</noscript>
```

> ## Permitted content
>
> When scripting is disabled and when it is a descendant of the `<head>` element: in any order, zero or more `<link>` elements, zero or more `<style>` elements, and zero or more `<meta>` elements. When scripting is disabled and when it isn't a descendant of the `<head>` element: any

> transparent content, but no `<noscript>` element must be among its descendants. Otherwise: flow content or phrasing content.

[Tecnichal Summary]

Share Improve this answer Follow

edited May 25 at 10:58

community wiki
3 revs, 3 users 53%
gurkan

6 This is invalid. noscript elements cannot contain meta elements. I would not trust this to work reliably in all browsers (including future browsers which you cannot currently test in), as they may perform error recovery in different ways. – Quentin Oct 27, 2009 at 10:10

16 This may be invalid, but look at Facebook's code, it uses the same solution in the <head> part - which doesn't mean it's a very good solution, because Facebook's code is far from a valid code, BUT it can mean that it works on many users' browser, which can be an important aspect. But it's true that it's not really suggested. This is their code: <noscript> <meta http-equiv=refresh content="0; URL=/about/messages/?_fb_noscript=1" /> </noscript> – Sk8erPeter May 19, 2011 at 11:26

IE9 (at least) has a setting "allow META REFRESH". I don't seem many turning it off, but presumably those that don't like javascript do, but if you want 100%, it's not this. – jenson-button-event Apr 29, 2013 at 7:56

It is valid in HTML5. – Naszta Aug 24, 2013 at 21:50

1 This destroys the link when trying to share on Linkedin, we found the hard way... this is not a good idea. Since linkedin follows the meta refresh, disregarding Og tags, etc.
– SomeDudeSomewhere Feb 3, 2015 at 20:28

49

I'd suggest you go the other way around by writing unobtrusive JavaScript.

Make the features of your project work for users with JavaScript disabled, and when you're done, implement your JavaScript UI-enhancements.

https://en.wikipedia.org/wiki/Unobtrusive_JavaScript

Share Improve this answer Follow

edited May 16, 2015 at 17:56

community wiki
2 revs, 2 users 67%
roosteronacid

157 Life's too short. It's 2012- Enable javascript or go home
– Yarin Jun 18, 2012 at 18:11

28 "Life's too short. It's 2012 - Enable javascript or go home!" is perfect little message to display in the event of no JS. logged in just to tell you that @Yarin – user900360 Oct 31, 2012 at 9:14

5 I also just logged in to say that is AWESOME! @Yarin. I'm making a small site that won't be generally public (consider like an intranet but for friends getting together to work on a project). I'm making it an SPA and not bothering to put in a

fallback. Unobtrusive JavaScript would nearly double the workload I'm putting into the SPA concept. With modern javascript libraries such as Knockout, Jquery, among others, we just have to accept that not every site can be easily made "unobtrusive." Javascript is the future of the web. – lassombra Jun 28, 2013 at 17:26

5 @Yarin I build PHP / JS apps for a living I never ever put in fallbacks... if it doesn't work for someone they contact me and I tell them to stop living in the past and enable JS. – Someone May 25, 2016 at 13:10

3 @n611x007 - Unethical things hide in all code, in every device, whether server or client side. Its not codes fault, its humans fault. I find a campaign against JS ironic because you still use a 3rd party ISP to connect to the internet, still use a computer with 3rd party hardware/firmware, still use insecure cell networks accessible by 3rd party injects, still own a cell phone that uses a 3rd party unpatched OS, still use a flat screen TV that uses a 3rd party wide open update service, etc etc. You are always a victim of MITM-ish channels regardless of whether you turn JS off in your browser. – dhaupin Jun 2, 2016 at 15:41

44

If your use case is that you have a form (e.g., a login form) and your server-side script needs to know if the user has JavaScript enabled, you can do something like this:

```
<form onsubmit="this.js_enabled.value=1;return true;">
    <input type="hidden" name="js_enabled" value="0">
    <input type="submit" value="go">
</form>
```

This will change the value of js_enabled to 1 before submitting the form. If your server-side script gets a 0, no JS. If it gets a 1, JS!

Share Improve this answer Follow

answered Sep 24, 2008 at 7:06

community wiki
Andrew Hedges

40

`<noscript>` isn't even necessary, and not to mention **not supported in XHTML**.

Working Example:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset/
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html>
<head>
    <title>My website</title>
    <style>
      #site {
          display: none;
      }
    </style>
    <script src="http://code.jquery.com/jquery-latest.
    <script>
      $(document).ready(function() {
          $("#noJS").hide();
          $("#site").show();
      });
    </script>
</head>
<body>
    <div id="noJS">Please enable JavaScript...</div>
    <div id="site">JavaScript dependent content here..
```

```
</body>
</html>
```

In this example, if JavaScript is enabled, then you see the site. If not, then you see the "Please enable JavaScript" message. **The best way to test if JavaScript is enabled, is to simply try and use JavaScript!** If it works, it's enabled, if not, then it's not...

Share Improve this answer Follow

edited Jun 27, 2017 at 20:18

community wiki
4 revs, 3 users 70%
de Raad

3 Ha, XHTML. Those were the days... Things have changed a lot: developer.mozilla.org/en-US/docs/Web/HTML/Element/noscript – Stephan Weinhold May 24, 2018 at 13:04

1 @StephanWeinhold, JSF uses XHTML. – Arash May 30, 2020 at 4:43

36

Use a .no-js class on the body and create non javascript styles based on .no-js parent class. If javascript is disabled you will get all the non javascript styles, if there is JS support the .no-js class will be replaced giving you all the styles as usual.

```
document.body.className = document.body.className.rep
```

trick used in HTML5 boilerplate http://html5boilerplate.com/ through modernizr but you can use one line of javascript to replace the classes

noscript tags are okay but why have extra stuff in your html when it can be done with css

Share Improve this answer Follow

edited Mar 2, 2013 at 13:29

community wiki
3 revs
Grundizer

5 Cannot believe it took this long before someone mentioned the `.nojs` class! This is one of the most seamless approaches and puts `noscript` to shame. – Moses Apr 25, 2012 at 16:57

16

just a bit tough but (hairbo gave me the idea)

CSS:

```
.pagecontainer {
  display: none;
}
```

JS:

```
function load() {
  document.getElementById('noscriptmsg').style.display
```

```
  document.getElementById('load').style.display = "blo
  /* rest of js*/
}
```

HTML:

```
<body onload="load();">

  <div class="pagecontainer" id="load">
    Page loading....
  </div>
  <div id="noscriptmsg">
    You don't have javascript enabled. Good luck with
  </div>

</body>
```

would work in any case right? even if the noscript tag is unsupported (only some css required) any one knows a non css solution?

Share Improve this answer

Follow

edited Feb 3, 2017 at 14:54

community wiki
3 revs, 3 users 72%
borrel

13

You can use a simple JS snippet to set the value of a hidden field. When posted back you know if JS was enabled or not.

Or you can try to open a popup window that you close rapidly (but that might be visible).

Also you have the NOSCRIPT tag that you can use to show text for browsers with JS disabled.

Share Improve this answer Follow

answered Sep 23, 2008 at 14:08

community wiki
rslite

This is not a good answer. The first solution requires a page reload and a form submission. The second solution opens a popup (ack!) and closes it "rapidly" - from the client side? -1 – Ben Nov 12, 2010 at 6:29

Sometimes you need to know server side if a site has JS enabled or not, so I don't see how you can do it other than posting something back. Agree that a popup is ugly nad I'm not really sure about this but it should be possible to open a popup outside the visible area of the screen so the user is not disturbed by this. No elegant but it works and there's no page reload. – rslite Dec 10, 2010 at 22:40

Spambots also posts to a hidden field. – Janis Veinbergs Mar 21, 2011 at 10:52

13

You'll want to take a look at the noscript tag.

```
<script type="text/javascript">
...some javascript script to insert data...
</script>
<noscript>
   <p>Access the <a href="http://someplace.com/data">d
</noscript>
```

Share Improve this answer Follow

edited Jun 2, 2013 at 13:49

community wiki
2 revs, 2 users 95%
anon

13

Because I always want to give the browser something worthwhile to look at I often use this trick:

First, any portion of a page that needs JavaScript to run properly (including passive HTML elements that get modified through getElementById calls etc.) are designed to be usable as-is with the assumption that there ISN'T javaScript available. (designed as if it wasn't there)

Any elements that would require JavaScript, I place inside a tag something like:

```
<span name="jsOnly" style="display: none;"></span>
```

Then at the beginning of my document, I use `.onload` or `document.ready` within a loop of `getElementsByName('jsOnly')` to set the `.style.display = "";` turning the JS dependent elements back on. That way, non-JS browsers don't ever have to see the JS dependent portions of the site, and if they have it, it appears immediately when it's ready.

Once you are used to this method, it's fairly easy to hybridize your code to handle both situations, although I am only now experimenting with the `noscript` tag and expect it will have some additional advantages.

Share Improve this answer Follow

edited Dec 21, 2016 at 22:31

community wiki
3 revs, 2 users 81%
Brian

12

The noscript tag works well, but will require each additional page request to continue serving useless JS files, since essentially noscript is a client side check.

You could set a cookie with JS, but as someone else pointed out, this could fail. Ideally, you'd like to be able to detect JS client side, and without using cookies, set a session server side for that user that indicates is JS is enabled.

A possibility is to dynamically add a 1x1 image using JavaScript where the src attribute is actually a server side script. All this script does is saves to the current user session that JS is enabled ($_SESSION['js_enabled']). You can then output a 1x1 blank image back to the browser. The script won't run for users who have JS disabled, and hence the $_SESSION['js_enabled'] won't be set. Then for further pages served to this user, you can decide whether to include all of your external JS files, but you'll always want to include the check, since some of your users might be using the NoScript Firefox add-on or have JS disabled temporarily for some other reason.

You'll probably want to include this check somewhere close to the end of your page so that the additional HTTP request doesn't slow down the rendering of your page.

Share Improve this answer Follow

answered Sep 27, 2008 at 6:53

community wiki
David Wees

12

Add this to the HEAD tag of each page.

```
<noscript>
        <meta http-equiv="refresh" runat="server" id="
content="0;logon.aspx" />
</noscript>
```

So you have:

```
<head>
    <noscript>
        <meta http-equiv="refresh" runat="server" id="
content="0;logon.aspx" />
    </noscript>
</head>
```

With thanks to Jay.

Share Improve this answer

Follow

answered Apr 1, 2011 at 9:42

community wiki
the_seeker_who

11

A common solution is to the meta tag in conjunction with noscript to refresh the page and notify the server when JavaScript is disabled, like this:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <noscript>
            <meta http-equiv="refresh" content="0; /?j
        </noscript>
        <meta charset="UTF-8"/>
        <title></title>
    </head>
</html>
```

In the above example when JavaScript is disabled the browser will redirect to the home page of the web site in 0

seconds. In addition it will also send the parameter javascript=false to the server.

A server side script such as node.js or PHP can then parse the parameter and come to know that JavaScript is disabled. It can then send a special non-JavaScript version of the web site to the client.

Share Improve this answer Follow

answered Feb 22, 2013 at 15:49

community wiki
Umesh Patil

1 But, if javascript is disabled, then this will loop forever?
– Jakob Sternberg Mar 9, 2021 at 2:49

10

This is the "cleanest" solution id use:

```
<noscript>
    <style>
        body *{ /*hides all elements inside the body*/
            display: none;
        }
        h1{ /* even if this h1 is inside head tags it
we have to display it again after all body elements ar
            display: block;
        }
    </style>
    <h1>JavaScript is not enabled, please check your b
</noscript>
```

Share Improve this answer

Follow

answered Apr 2, 2015 at 14:04

community wiki
Alpha2k

9

If javascript is disabled your client-side code won't run anyway, so I assume you mean you want that info available server-side. In that case, *noscript* is less helpful. Instead, I'd have a hidden input and use javascript to fill in a value. After your next request or postback, if the value is there you know javascript is turned on.

Be careful of things like noscript, where the first request may show javascript disabled, but future requests turn it on.

Share Improve this answer

Follow

edited Jun 1, 2009 at 0:07

community wiki
2 revs
Joel Coehoorn

5

You might, for instance, use something like document.location = 'java_page.html' to redirect the browser to a new, script-laden page. Failure to redirect implies that JavaScript is unavailable, in which case you can either resort to CGI ro utines or insert appropriate

code between the tags. (NOTE: NOSCRIPT is only available in Netscape Navigator 3.0 and up.)

credit
http://www.intranetjournal.com/faqs/jsfaq/how12.html

Share Improve this answer Follow

answered Sep 23, 2008 at 14:16

community wiki
Brad8118

4

A technique I've used in the past is to use JavaScript to write a session cookie that simply acts as a flag to say that JavaScript is enabled. Then the server-side code looks for this cookie and if it's not found takes action as appropriate. Of course this technique does rely on cookies being enabled!

Share Improve this answer Follow

answered Sep 23, 2008 at 14:10

community wiki
John Topley

4

I think you could insert an image tag into a noscript tag and look at the stats how many times your site and how often this image has been loaded.

Share Improve this answer

Follow

answered Sep 23, 2008 at 14:14

community wiki
MrVolley

4

People have already posted examples that are good options for detection, but based on your requirement of "give warning that the site is not able to function properly without the browser having JS enabled". You basically add an element that appears somehow on the page, for example the 'pop-ups' on Stack Overflow when you earn a badge, with an appropriate message, then remove this with some Javascript that runs as soon as the page is loaded (and I mean the DOM, not the whole page).

Share Improve this answer

Follow

answered Sep 23, 2008 at 21:54

community wiki
roryf

4

code inside `<noscript>` tags will be executed when there is no js enabled in browser. we can use `noscript` tags to display msg to turn on JS as below.

```
<noscript>
    <h1 style="text-align: center;">
       To view this page properly, please
       enable JavaScript and reload the page
```

```
    </h1>
</noscript>
```

while keeping our website content inside body as hidden. as below

```
<body>
<div id="main_body" style="display: none;">
    website content.
</div>
</body>
```

now if JS is turned on you can just make the content inside your main_body visible as below

```
<script type="text/javascript">
    document.getElementById("main_body").style.display
</script>
```

Share Improve this answer Follow

edited Feb 1, 2023 at 18:24

community wiki
2 revs, 2 users 78%
Pavan Kumar

3

Why don't you just put a hijacked onClick() event handler that will fire only when JS is enabled, and use this to append a parameter (js=true) to the clicked/selected URL (you could also detect a drop down list and change the value- of add a hidden form field). So now when the

server sees this parameter (js=true) it knows that JS is enabled and then do your fancy logic server-side.
The down side to this is that the first time a users comes to your site, bookmark, URL, search engine generated URL- you will need to detect that this is a new user so don't look for the NVP appended into the URL, and the server would have to wait for the next click to determine the user is JS enabled/disabled. Also, another downside is that the URL will end up on the browser URL and if this user then bookmarks this URL it will have the js=true NVP, even if the user does not have JS enabled, though on the next click the server would be wise to knowing whether the user still had JS enabled or not. Sigh.. this is fun...

Share Improve this answer Follow

answered Aug 13, 2009 at 21:16

community wiki
FirstSOURCE

3

To force users to enable JavaScripts, I set 'href' attribute of each link to the same document, which notifies user to enable JavaScripts or download Firefox (if they don't know how to enable JavaScripts). I stored actual link url to the 'name' attribute of links and defined a global onclick event that reads 'name' attribute and redirects the page there.

This works well for my user-base, though a bit fascist ;).

Share Improve this answer Follow

answered Jul 26, 2010 at 14:12

community wiki
Jan Sahin

and a bit annoying if the user does have JS enabled and clicks on the link before your progressive javascript kicks in... – Simon_Weaver Jul 26, 2010 at 22:33

1 Sure, but no issue reported yet. – Jan Sahin Jul 27, 2010 at 8:17

3

You don't detect whether the user has javascript disabled (server side or client). Instead, you assume that javascript is disabled and build your webpage with javascript disabled. This obviates the need for `noscript`, which you should avoid using anyway because it doesn't work quite right and is unnecessary.

For example, just build your site to say `<div id="nojs">This website doesn't work without JS</div>`

Then, your script will simply do `document.getElementById('nojs').style.display = 'none';` and go about its normal JS business.

Share Improve this answer Follow

answered Aug 31, 2011 at 15:03

community wiki
Explosion Pills

this is a great seamless approach <noscript> tags are much harder to maintain on different pages with this approach you can maintain your noscript styling through your css file.
– zadubz Apr 30, 2013 at 10:49

3

Check for cookies using a pure server side solution i have introduced here then check for javascript by dropping a cookie using Jquery.Cookie and then check for cookie this way u check for both cookies and javascript

Share Improve this answer Follow

edited May 23, 2017 at 12:26

community wiki
2 revs
pouya

3

In some cases, doing it backwards could be sufficient. Add a class using javascript:

```
// Jquery
$('body').addClass('js-enabled');

/* CSS */
.menu-mobile {display:none;}
body.js-enabled .menu-mobile {display:block;}
```

This could create maintenance issues on anything complex, but it's a simple fix for some things. Rather than trying to detect when it's not loaded, just style according to when it is loaded.

Share Improve this answer Follow

answered Jan 13, 2013 at 9:52

community wiki
Jen

3

I would like to add my solution to get reliable statistics on how many real users visit my site with javascript disabled over the total users. The check is done one time only per session with these benefits:

- Users visiting 100 pages or just 1 are counted 1 each. This allows to focus on single users, not pages.
- Does not break page flow, structure or semantic in anyway
- Could logs user agent. This allow to exclude bots from statistics, such as google bot and bing bot which usually have JS disabled! Could also log IP, time etc...
- Just one check per session (minimal overload)

My code uses PHP, mysql and jquery with ajax but could be adapted to other languanges:

Create a table in your DB like this one:

```
CREATE TABLE IF NOT EXISTS `log_JS` (
  `logJS_id` int(11) NOT NULL AUTO_INCREMENT,
  `data_ins` timestamp NOT NULL DEFAULT CURRENT_TIMEST
  `session_id` varchar(50) NOT NULL,
  `JS_ON` tinyint(1) NOT NULL DEFAULT '0',
  `agent` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`logJS_id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8;
```

Add this to every page after using session_start() or equivalent (jquery required):

```
<?  if (!isset($_SESSION["JSTest"]))
    {
        mysql_query("INSERT INTO log_JS (session_id, a
mysql_real_escape_string(session_id()) . "', '" .
mysql_real_escape_string($_SERVER['HTTP_USER_AGENT']).
        $_SESSION["JSTest"] = 1; // One time per sessi
        ?>
        <script type="text/javascript">
            $(document).ready(function() { $.get('JSOK
        </script>
        <?
    }
?>
```

Create the page JSOK.php like this:

```
<?
include_once("[DB connection file].php");
mysql_query("UPDATE log_JS SET JS_ON = 1 WHERE session
mysql_real_escape_string(session_id()) . "'");
```

Share Improve this answer edited Aug 23, 2013 at 8:37

Follow

community wiki
2 revs
Unbroken

3

I've figured out another approach using css and javascript itself.
This is just to start tinkering with classes and ids.

The CSS snippet:
1. Create a css ID rule, and name it #jsDis.
2. Use the "content" property to generate a text after the BODY element. (You can style this as you wish).
3 Create a 2nd css ID rule and name it #jsEn, and stylize it. (for the sake of simplicity, I gave to my #jsEn rule a different background color.

```
<style>
#jsDis:after {
    content:"Javascript is Disable. Please turn it ON!
    font:bold 11px Verdana;
    color:#FF0000;
}

#jsEn {
    background-color:#dedede;
}

#jsEn:after {
    content:"Javascript is Enable. Well Done!";
    font:bold 11px Verdana;
    color:#333333;
}
</style>
```

The JavaScript snippet:
1. Create a function.
2. Grab the BODY ID with getElementById and assign it to a variable.
3. Using the JS function 'setAttribute', change the value of the ID attribute of the BODY element.

```
<script>
function jsOn() {
    var chgID = document.getElementById('jsDis');
    chgID.setAttribute('id', 'jsEn');
}
</script>
```

The HTML part.
1. Name the BODY element attribute with the ID of #jsDis.
2. Add the onLoad event with the function name. (jsOn()).

```
<body id="jsDis" onLoad="jsOn()">
```

Because of the BODY tag has been given the ID of #jsDis:
- If Javascript is enable, it will change by himself the attribute of the BODY tag.
- If Javascript is disable, it will show the css 'content:' rule text.

You can play around with a #wrapper container, or with any DIV that use JS.

Hope this helps to get the idea.

Share Improve this answer Follow

answered Sep 12, 2013 at 3:17

community wiki
Chris Pesoa

2

Detect it in what? JavaScript? That would be impossible. If you just want it for logging purposes, you could use some sort of tracking scheme, where each page has JavaScript that will make a request for a special resource (probably a very small `gif` or similar). That way you can just take the difference between unique page requests and requests for your tracking file.

Share Improve this answer Follow

answered Sep 23, 2008 at 14:09

community wiki
Hank Gay

---

1 2 Next