Code Samples Survey 2014 — Results —
About this survey
This survey is an attempt to understand what needs to be improved regarding the various code samples on MDN.

The survey was run from May 22th to June 16, 2014 and was filled by 950 participants.

Participants
Among the 950 participants, we get answers from beginners as well as advanced users, from occasional readers as well as MDN contributors.

Regarding web technologies, participants are:
- Beginners (19.4%)
- Average web developers (37.1%)
- Seasoned web developers (34.2%)
- A Jedi/Sith (9.3%)

The population is quite well balanced with average web developer representing 37.1% of the participants and seasoned web developer representing 34.2%.

Regarding MDN, participants are:
- Occasional readers (35.8%)
- Regular readers (54.5%)
- Contributors (7.5%)
- Obi Wan Kenobi/Darth Vador (2.2%)
About the existing code samples
We first asked their opinion to the participants about existing code samples available on MDN.

Generally speaking, all participants agreed that code samples are useful.

In general, do you find code samples useful?

- Yes (94.1%)
- No (5.9%)

Regarding interactive code, participants also think it's very useful.

In general, do you find editable/interactive code samples useful?

- Yes (79.1%)
- No (20.9%)

It worth noting that beginners and average developers (81.5%) are a little more interested in that feature than seasoned developers (77.2%).
When we ask about the quality of the code samples on MDN, we get some very positive review:

Do you think code samples on MDN are accurate?

It worth noting that the more a user is advanced, the more he finds relevant code samples on MDN, however, beginner are more willing to find code samples that match exactly their needs.

3rd party code samples providers
When ask for reference web sites that handle code samples well, we get the following:

Blog

- css-tricks.com (8)
- tympanus.net/codrops (2)
- bradfrostweb.com
- html5doctor.com
- lea.verou.me
- loopinfinito.com.br
- usabili.ru
- worrydream.com (/#!/LearnableProgramming)
Technologies and libraries

• api.jquery.com (7)
• php.net (7)
• angularjs.org (3)
• emberjs.com (3)
• getbootstrap.com (3)
• cppreference.com (2)
• docs.sencha.com (2)
• redis.io (2)
• api.rubyonrails.org
• backbonejs.org
• codex.wordpress.org
• coffeescript.org
• cplusplus.com
• cytoscape.org
• d3js.org
• docs.casperjs.org
• docs.python.org
• docs.sqlalchemy.org
• dojotoolkit.org
• glsl.heroku.com
• ionicframework.com/docs
• nodejs.org
• perlmonks.org
• play.golang.org
• qt-project.org/docs
• stripe.com/docs
• tryruby.org
• underscorejs.org
• unity3d.com
• vertex.io
• yuilibrary.com

Community web sites

• stackoverflow.com (8)
• stackexchange.com

Code playground

• jsfiddle.net (16)
• github.com (9)
• codepen.io (8)
• jsbin.com (8)
• gist.github.com (7)
• bl.ocks.org
• dabblet.com
• livecoder.net
• jsdo.it

Educational web sites

• w3schools.com (14)
• msdn.microsoft.com (8)
• codeacademy.com (6)
• developers.google.com (5)
• plnkr.co (3)
• khanacademy.org (2)
• code.org
• codeschool.com
• developer.apple.com
• html.net
• learnstreet.com
• proprofs.com
• thecodeplayer.com
• typing.io
• weberdev.com
• xahlee.info
**Expectation around code samples**

In order to see how to improve code samples, we asked participants to tell us what they expect from code samples:

What are you expecting from code samples?

It worth noting than Average and seasoned developers expect more code samples which highlight some difficult code usages when beginners are more eager to find fully functional code samples.

One of the most interesting result is that almost nobody seems really interested about being able to “export” or “share” code samples out of MDN.

Participants were invited to extend that list of expectation. Among those others expectation, we mostly found:

- Having up to date samples
- Having samples with some context explaining the why and the how.
- Having samples focused on a single code problem
- Having samples with explicit and verbose naming of variables and functions
- Having samples promoting best practices and being cross browser compatible.
Beginners’ top 3

- To be short and simple
- To have comments to explain the code
- To be fully functional

Average developers’ top 3

- To be short and simple
- To have comments to explain the code
- To highlight some difficult code usage

Seasoned developers’ top 3

- To be short and simple
- To highlight some difficult code usage
- To have comments to explain the code

Jedi/Sith’s top 3

- To be short and simple
- To highlight some difficult code usage
- To have comments to explain the code
We also provided a list of possible enhancements to help users of code samples on MDN, participants were asked to rate them from 1 (useless) to 5 (must have):

Which feature around code samples on MDN will help you the most?

1. Having a code that is always nicely formatted.
2. Being able to see and understand errors if I try to play with the code.
3. Being able to see the code side by side with the live result.
4. Being able to play with the code and see the live result.
5. Being able to only watch the live result of each sample.
6. Being able to download code samples.
7. Being able to share the code samples without the whole page.
8. Being able to fork code samples on CodePen.io (or other similar services).
9. Being able to fork code samples on GitHub (or other similar services).

It's interesting to see that the most wanted feature is to have code samples nicely formatted where social interaction (fork) are clearly unwanted. Having the possibility to fiddle with the code is definitely a highly wanted feature especially if it comes with a mechanism to explain code error.
Contribution to code samples
Because MDN is a wiki, we also wish to know more about contribution to code samples.

Do you ever contribute to code samples on MDN?

- Yes (13.7%)
- No (86.3%)

Surprisingly, even among regular MDN contributors, not that much had contributed to code samples.

As an MDN contributor, do you ever contribute to code samples on MDN?

- Yes (52.1%)
- No (47.9%)
**Why no contribution**

In order to understand why we have so few contributors on that area of MDN, we ask the participants what prevent them to contribute:

**What did prevent you to contribute to code samples**

- I wasn't aware it was possible.
- I don't know what samples are needed.
- I had no time.
- I don't have the required skills.
- I don't want to create an account.
- I found the editing system too complicated.
- I found the editing system too limited.

For almost half of the participants, it’s a lack of information that prevents them to contribute to code samples. Worst, for MDN contributors, 26.8% (more than a quarter!) told they weren’t aware it was possible, and 29.6% have no idea which code samples are needed. This clearly states that MDN do not communicate well on that area.

Among other reasons provide by participants, we mostly found:

- The fear of doing wrong
- The lack of willingness
- There is enough code samples on MDN

**Verbatim:**

“An 'Add sample...' page with a template and some best practices would be useful and would encourage me to contribute more.”
**Issues while contributing**

On the other side, we asked the code samples contributor if they found something annoying while contributing. Here are some of the most relevant verbatim we get:

**Community issues**

- “I did not get any encouragement / motivation in mdn to participate more”
- “When people replace irrefutably accurate code and polyfills with garbage hacks.”

**Issues with the MDN editing systems**

- “[...] interrupting workflow to figure out how to edit (which I do only very infrequently) breaks concentration on the programming problem [...]”
- “I’d do it more often if it was FASTER to edit things.”
- “I wasn’t able to modify just a little part of the page without entering in edit mode for the whole page. Per section editing like wikipedia would be great.”
- “It appears the editor is WYSIWYG editor, not really designed for writing code.”
- “The normal editor doesn’t do a very good job of supporting code editing, especially around indentation and syntax support.”

**Issues with code formatting**

- “Finding examples to adhere to MDN styling, syntax, variable nomenclature”

**Fuzzy or inaccurate process**

- “It took a while to figure out how to effectively contribute live code samples; and unless I’m doing it repeatedly, there’s a bit of a rediscovery process each time.”
- “Iteration over preview checking.”
- “If you guys could make an Google+ or Facebook Login Button available i’ll be the first to use it”
- “to have to hide AND mix my js code to make the sample functional, and the code a actually wanted to show”

**Kuma issues**

- “The Kumascript language for templates and some little bugs.”
- “The mess between wiki templates and WYSIWYG editing... which is quite a standard problem and only possible to overcome with dedicated software rather than standard wiki software.”
Encouraging contribution
In order to encourage contribution, we thought about some features that could be helpful to contributors. We asked the participant to rate them from 1 (pointless) to 5 (absolute requirement to contribute).

Which feature will encourage you to contribute code samples on MDN?

- Having the code automatically formatted to follow MDN's code convention.
- Having a mentor to review and provide feedback on code samples I'm working on.
- Being able to write code samples directly inside the MDN page editor.
- Being able to write code samples on MDN but in a dedicated editor.
- Being able to push code samples through GitHub or a similar service.
- Being able to link any third party library to use it in code samples.
- Being able to push code samples through codepen.io or a similar service.
- Being able to write some code hidden from the code displays to the user.

It is quite coherent with what we get on the expectation: the top wanted feature to encourage contribution is an automated system to format the code to make sure it follows MDN coding standard. Mentoring is also something highly rated which could require to rethink the contribution workflow on MDN.

Surprisingly using third party code workflow provider such as GitHub or Codepen does not seem really appealing to users. They do not reject it, but it will not be a game changer for MDN. Actually the real surprise comes from the fact that this lack of serious interest is shared among participants regardless of their skill (beginners or advanced). Interestingly actual MDN contributors rate such feature a little higher (3.3) than the others. This indicates clearly that the current MDN workflow needs to be adjusted but there is no need to out source it to a third party service.
Conclusion

This survey was quite enlightening and quite relevant regarding the number of participants. There mostly 3 points to remind about the results:

1. Code samples on MDN are considered reasonably good and can be easily improved.
2. If we want to encourage contribution we must focus our effort on communication and on automated code formatting.
3. Interactive code samples will be a good feature to set up if it comes with an error explanation system.