Needs Assessment

What is a Need?

Before sharing the top ten needs, we're briefly describing what a need is to help set the context for the following Findings section.

The need statements were informed from the fourteen pilot interviews conducted at the beginning of this project. The statements are written from the point of view of a web developer. The outline we used to create the need statements was:

I am a __________________ (persona) trying to __________________ (verb) but __________________ (barrier) because __________________ (cause), which makes me feel __________________ (emotional reaction).

Putting this into action, it could read as follows:

I am a tourist trying to travel to another country but am struggling to understand the Visa process because it's complex and poorly communicated, which makes me feel frustrated.

We drew upon common practices in design thinking as well as product-development processes for inspiration when deciding to use need statements in the survey. Because they are written from the point of view of developers, we felt it would be an intuitive way to read, interpret, and rank to get to the top ten.

The need statements for this project were centered around the emotional reaction of frustration. If a web developer experiences frustration in regards to web development, there may be an underlying opportunity for browser vendors to help solve that frustration.

Ranking Methodology

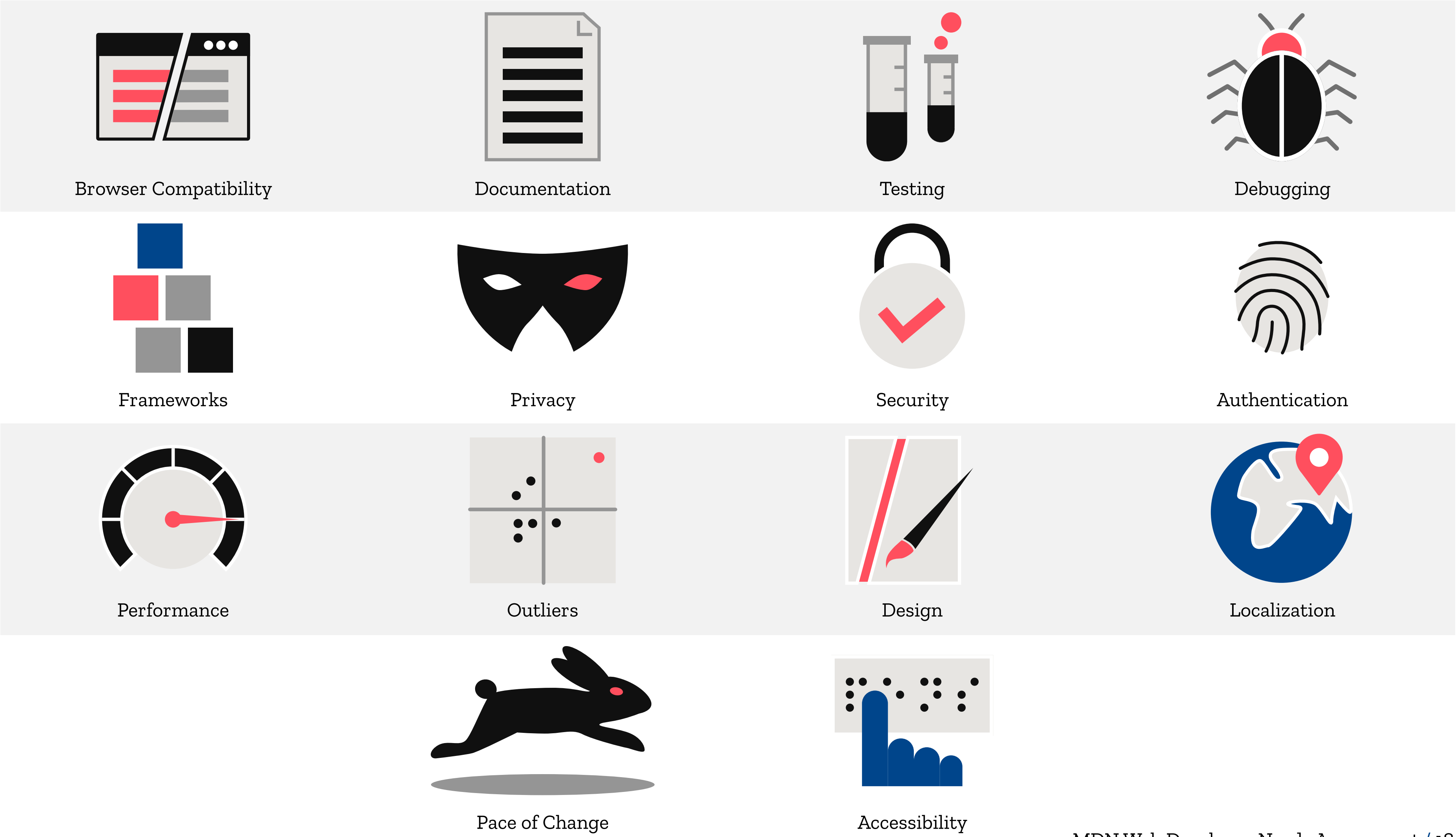
Using the MaxDiff methodology, we asked survey respondents to rank a total of 28 need statements. Respondents saw twelve sets comprised of four need statements. For each set they were instructed to pick the one need that causes them the least frustration and the one need that causes them the most frustration. A single need statement could appear more than once within the twelve sets.

It is important to note that just because a need may not rank as the least frustrating within a set, that does not mean it causes the least frustration. It could imply that the respondent does not have experience with the subject matter or does not prioritize that subject within their work.

For example, "Making sites accessible," ranked 24th out of 28, and that's pretty low. That doesn't mean making sites accessible is easy, without room for improvement. What we learned during the pilot interviews is that developers and designers are not always given latitude to spend the necessary time on accessibility. Therefore, because they cannot spend the time on it, accessibility does not create frustration. If in the future, developers and designers spend more time on accessibility, their perception of the frustration may change, and so would the ranking.

Themes

The 28 needs cover 14 different themes. As was mentioned earlier, four of the top ten needs relate to browser compatibility, making it the highest ranked theme. Rounding out the top five themes are Documentation, Testing, Debugging, and Frameworks.



Overall Needs Ranking

One is the most frustrating and 28 is the least frustrating.

1. Having to support specific browsers (e.g., IE11).
2. Outdated or inaccurate documentation for frameworks and libraries.
3. Avoiding or removing a feature that doesn't work across browsers.
4. Testing across browsers.
5. Making a design look/work the same across browsers.
6. Discovering bugs not caught during testing.
7. Supporting multiple frameworks in the same code base.
8. Keeping up with a large number of new and existing tools or frameworks.
9. Managing user data to comply with laws and regulations.
10. Understanding and implementing security measures.
11. Integrating with third parties for authentication.
12. Pinpointing existing performance issues.
13. Running end-to-end tests.
14. Lack of device APIs allowing for access to hardware.
15. Outdated documentation for HTML, CSS and JavaScript.
16. Determining the root cause of a bug.
17. Capability of the web to support a specified layout.
18. Knowing what browsers support a specific technology.
19. Achieving visual precision on stylized elements (e.g., buttons).
20. Running front-end tests.
21. Implementing localization.
22. Keeping up with changes to the web platform.
23. Implementing performance optimizations.
24. Making sites accessible.
25. Getting users to grant permissions to Web APIs (e.g., geo-location).
26. Deciding what to learn next to keep my skill set relevant.
27. Finding a community of peers.
28. Fixing a bug once it's been identified.

Segmented View of the Needs Ranking

The Overall Needs Ranking shows how the general population, the 28,474 completed survey responses, ranked the 28 needs.

The data is further analyzed by different segments to provide a more nuanced view. When seeing the differences, keep in mind that deviation from the general population is not good or bad; it's just a visual representation of the actual data.

The segments included in this report are:

Country - The seven shown are there because they are the countries that have the most traffic on MDN, the primary recruiting vehicle used for the survey.

Gender - When looking at the results, remember the percentage of respondents who selected "Man" when asked about the gender they identify with (87.1%) and the rate of respondents who selected "Woman" (8.2%). More respondents chose "Man," which helps explain why when filtering the needs list by gender, men's ranking of the needs doesn't differ from the general population and women do. Again, differing from the general population is not good or bad; it's just a visual representation of the data.

Satisfaction with the Web - In the survey, we asked respondents to rate their overall satisfaction with the Web as a platform and set of tools (more on this starting on page 27) using a Likert scale from very satisfied to very dissatisfied. This segmentation includes those who were very satisfied and satisfied compared to those who are dissatisfied and very dissatisfied.

Years of Experience - We provided data for those who have two years or less experience writing code for the Web and those with ten or more years of experience.

Type of Front End Developer - Survey respondents were able to pick which type of developer best describes them, and there were two options for Front End Developers. Those who primarily write code using JavaScript and those who primarily use HTML and CSS. We provided the difference in need ranking for these two.

Needs By Country

General Population		Brazil	China	France	India	Japan	Russia	U.S.
Having to support specific browsers (e.g., IE11).	1	1	1	1	1	1	1	1
Outdated or inaccurate documentation for frameworks and libraries.	2	3	2	2	2	3	5	2
Avoiding or removing a feature that doesn't work across browsers.	3	2	4	5	4	4	3	3
Testing across browsers.	4	4	3	3	6	2	2	4
Making a design look/work the same across browsers.	5	5	6	4	3	5	4	7
Discovering bugs not caught during testing.	6	6	9	6	8	8	7	5
Supporting multiple frameworks in the same code base.	7	7	7	7	7	6	6	8
Keeping up with a large number of new and existing tools or frameworks.	8	8	10	12	5	16	9	6
Managing user data to comply with laws and regulations.	9	9	23	8	13	10	10	13
Understanding and implementing security measures.	10	14	16	17	12	12	14	9
Integrating with third parties for authentication.	11	17	13	11	15	11	12	11
Pinpointing existing performance issues.	12	15	11	14	14	18	13	10
Running end-to-end tests.	13	11	12	10	11	9	25	12
Lack of device APIs allowing for access to hardware.	14	10	5	13	9	17	11	23
Outdated documentation for HTML, CSS and JavaScript.	15	16	8	9	18	7	19	19
Determining the root cause of a bug.	16	13	21	18	19	13	18	14
Capability of the web to support a specified layout.	17	19	24	15	16	14	17	18
Knowing what browsers support a specific technology.	18	12	17	19	10	20	8	25
Achieving visual precision on stylized elements (e.g., buttons).	19	22	15	20	25	19	15	21
Running front-end tests.	20	20	20	16	24	15	23	16
Implementing localization.	21	23	22	23	21	22	26	15
Keeping up with changes to the web platform.	22	25	19	25	20	21	16	20
Implementing performance optimizations.	23	26	14	24	17	23	20	22
Making sites accessible.	24	24	28	21	26	25	22	17
Getting users to grant permissions to Web APIs (e.g., geo-location).	25	21	18	22	23	24	21	26
Deciding what to learn next to keep my skill set relevant.	26	18	25	26	22	26	24	24
Finding a community of peers.	27	27	27	27	27	27	27	27
Fixing a bug once it's been identified.	28	28	26	28	28	28	28	28

Needs By Gender

General Population		*Female*	*Male*
Having to support specific browsers (e.g., IE11).	1	1	1
Outdated or inaccurate documentation for frameworks and libraries.	2	2	2
Avoiding or removing a feature that doesn't work across browsers.	3	3	3
Testing across browsers.	4	6	4
Making a design look/work the same across browsers.	5	8	5
Discovering bugs not caught during testing.	6	4	6
Supporting multiple frameworks in the same code base.	7	5	7
Keeping up with a large number of new and existing tools or frameworks.	8	7	8
Managing user data to comply with laws and regulations.	9	14	9
Understanding and implementing security measures.	10	9	10
Integrating with third parties for authentication.	11	11	11
Pinpointing existing performance issues.	12	10	12
Running end-to-end tests.	13	16	13
Lack of device APIs allowing for access to hardware.	14	12	14
Outdated documentation for HTML, CSS and JavaScript.	15	15	15
Determining the root cause of a bug.	16	17	16
Capability of the web to support a specified layout.	17	20	17
Knowing what browsers support a specific technology.	18	19	18
Achieving visual precision on stylized elements (e.g., buttons).	19	25	19
Running front-end tests.	20	23	20
Implementing localization.	21	22	21
Keeping up with changes to the web platform.	22	18	22
Implementing performance optimizations.	23	13	23
Making sites accessible.	24	26	24
Getting users to grant permissions to Web APIs (e.g., geo-location).	25	21	25
Deciding what to learn next to keep my skill set relevant.	26	24	26
Finding a community of peers.	27	27	27
Fixing a bug once it's been identified.	28	28	28

Needs By Satisfaction with the Web

General Population		*Satisfied*	*Dissatisfied*
Having to support specific browsers (e.g., IE11).	1	1	1
Outdated or inaccurate documentation for frameworks and libraries.	2	2	2
Avoiding or removing a feature that doesn't work across browsers.	3	3	6
Testing across browsers.	4	4	4
Making a design look/work the same across browsers.	5	5	3
Discovering bugs not caught during testing.	6	6	8
Supporting multiple frameworks in the same code base.	7	7	7
Keeping up with a large number of new and existing tools or frameworks.	8	8	5
Managing user data to comply with laws and regulations.	9	9	16
Understanding and implementing security measures.	10	10	13
Integrating with third parties for authentication.	11	12	10
Pinpointing existing performance issues.	12	11	15
Running end-to-end tests.	13	13	14
Lack of device APIs allowing for access to hardware.	14	14	19
Outdated documentation for HTML, CSS and JavaScript.	15	15	17
Determining the root cause of a bug.	16	16	20
Capability of the web to support a specified layout.	17	18	9
Knowing what browsers support a specific technology.	18	17	18
Achieving visual precision on stylized elements (e.g., buttons).	19	21	11
Running front-end tests.	20	19	21
Implementing localization.	21	20	22
Keeping up with changes to the web platform.	22	23	12
Implementing performance optimizations.	23	22	25
Making sites accessible.	24	25	23
Getting users to grant permissions to Web APIs (e.g., geo-location).	25	24	26
Deciding what to learn next to keep my skill set relevant.	26	26	24
Finding a community of peers.	27	27	27
Fixing a bug once it's been identified.	28	28	28

Needs By Years of Experience

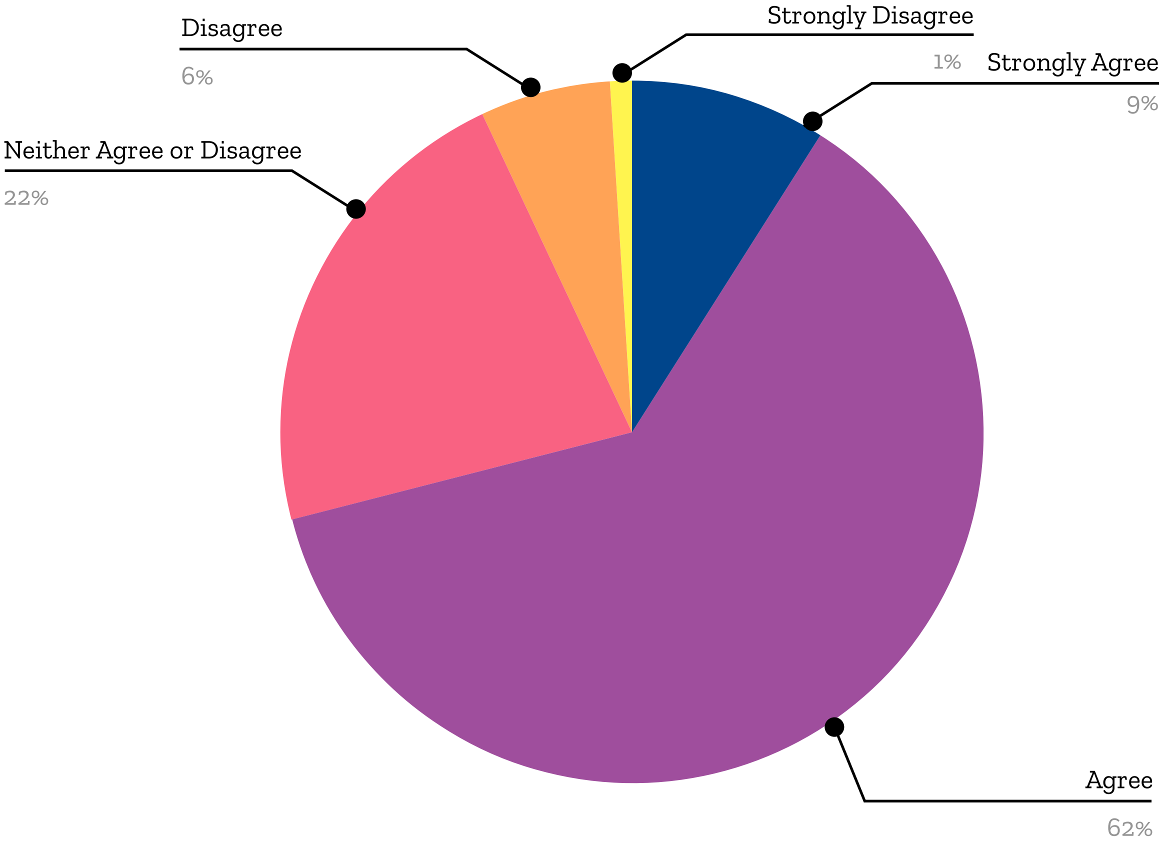
General Population		*2 or less*	*10+*
Having to support specific browsers (e.g., IE11).	1	1	1
Outdated or inaccurate documentation for frameworks and libraries.	2	2	2
Avoiding or removing a feature that doesn't work across browsers.	3	3	4
Testing across browsers.	4	5	3
Making a design look/work the same across browsers.	5	4	8
Discovering bugs not caught during testing.	6	6	6
Supporting multiple frameworks in the same code base.	7	7	7
Keeping up with a large number of new and existing tools or frameworks.	8	8	5
Managing user data to comply with laws and regulations.	9	11	9
Understanding and implementing security measures.	10	9	13
Integrating with third parties for authentication.	11	15	10
Pinpointing existing performance issues.	12	12	12
Running end-to-end tests.	13	17	11
Lack of device APIs allowing for access to hardware.	14	10	18
Outdated documentation for HTML, CSS and JavaScript.	15	14	17
Determining the root cause of a bug.	16	13	15
Capability of the web to support a specified layout.	17	18	16
Knowing what browsers support a specific technology.	18	16	23
Achieving visual precision on stylized elements (e.g., buttons).	19	19	20
Running front-end tests.	20	24	14
Implementing localization.	21	22	22
Keeping up with changes to the web platform.	22	21	19
Implementing performance optimizations.	23	20	24
Making sites accessible.	24	26	21
Getting users to grant permissions to Web APIs (e.g., geo-location).	25	23	25
Deciding what to learn next to keep my skill set relevant.	26	25	26
Finding a community of peers.	27	27	27
Fixing a bug once it's been identified.	28	28	28

By Front End Developer

General Population		JavaScript	CSS/HTML
Having to support specific browsers (e.g., IE11).	1	1	1
Outdated or inaccurate documentation for frameworks and libraries.	2	2	5
Avoiding or removing a feature that doesn't work across browsers.	3	4	3
Testing across browsers.	4	3	10
Making a design look/work the same across browsers.	5	5	13
Discovering bugs not caught during testing.	6	7	4
Supporting multiple frameworks in the same code base.	7	6	6
Keeping up with a large number of new and existing tools or frameworks.	8	8	2
Managing user data to comply with laws and regulations.	9	11	11
Understanding and implementing security measures.	10	9	7
Integrating with third parties for authentication.	11	12	9
Pinpointing existing performance issues.	12	10	12
Running end-to-end tests.	13	13	15
Lack of device APIs allowing for access to hardware.	14	14	14
Outdated documentation for HTML, CSS and JavaScript.	15	15	19
Determining the root cause of a bug.	16	17	8
Capability of the web to support a specified layout.	17	18	23
Knowing what browsers support a specific technology.	18	16	22
Achieving visual precision on stylized elements (e.g., buttons).	19	23	26
Running front-end tests.	20	22	24
Implementing localization.	21	20	21
Keeping up with changes to the web platform.	22	21	18
Implementing performance optimizations.	23	19	20
Making sites accessible.	24	25	25
Getting users to grant permissions to Web APIs (e.g., geo-location).	25	24	17
Deciding what to learn next to keep my skill set relevant.	26	26	16
Finding a community of peers.	27	27	27
Fixing a bug once it's been identified.	28	28	28

How Developers Felt About the Needs List

Because the Developer Needs Assessment is intended to be reproduced annually, we asked survey respondents whether the list of 28 needs was a fair representation of the needs they experience as a web developer. While most respondents agreed the list was representative, 21.6% neither agreed nor disagreed which means there is room for improvement in the needs list.



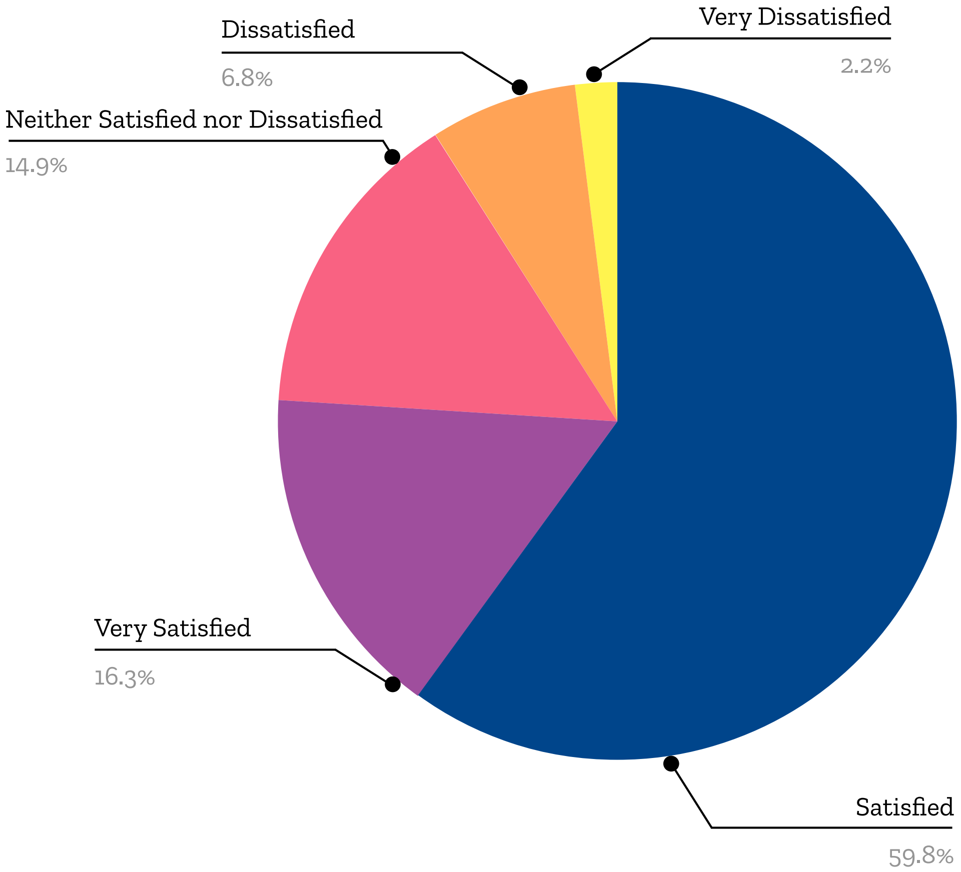
Overall Satisfaction With the Web

Overall Satisfaction With the Web

Through this project, we wanted a way to measure how satisfied web developers and designers are with the Web as a platform. We wanted this to be a repeatable question to measure changes in satisfaction over time.

We asked survey respondents, "How would you rate your overall satisfaction with the Web, as a platform and set of tools, to enable you to build what you need or want?"

We learned that a majority, 76.1%, of respondents are either very satisfied or satisfied with the Web, whereas 9% are either very dissatisfied or dissatisfied.


Dissatisfied
Very Dissatisfied
Neither Satisfied nor Dissatisfied
Very Satisfied
Satisfied