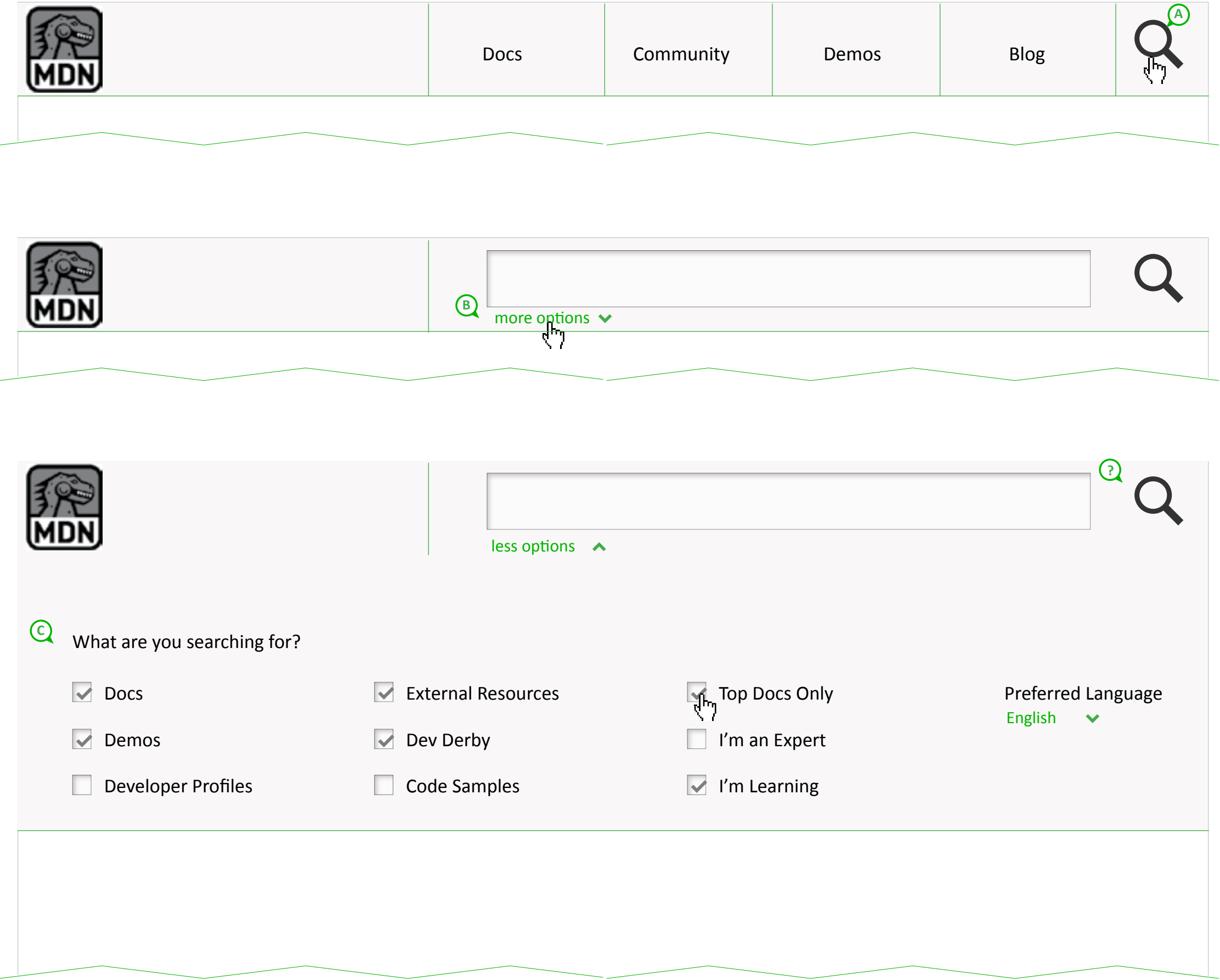


3

Phase 1: Updated Search

(Layout & Interaction)

Ph 1: Search Interaction



A. Trigger Seach Mode

User Triggers search mode by selecting search icon. (hover for desktop?)

B. Display Filtering Options

Selecting “more options” opens panel below menu with search filter options.

C. Filtering Options

Exact Filters TBD. Deendent on elastic search and docs organization.

?. Outstanding Issues for Prototype

1. When prototyping, need to test what the best search trigger is. For mobile, we need to have an explicit action button for this. Will search icon work for this, or will a “search” button be better?


2. What returns the user to the navigation? Does the same trigger that opened the search, close it? When user hovers outside of nav bar, does it return? If so, what do we do with mobile?

Search

vs.



Ph 1: Search Results



less options ^

What are you searching for?

☒ Docs

☒ External Resources

☒ Top Docs Only

Preferred Language

English v

☒ Demos

☒ Dev Derby

☐ I'm an Expert

☐ Developer Profiles

☐ Code Samples

☒ I'm Learning

A

RESULTS:

X

B

Docs

HTML10

CSS* Code Sample6

JavaScript2

HTML10

CSS* Code Sample6

JavaScript2

C

Demos

HTML56

JavaScript* Code Sample2

JavaScript technologies overview

Data structures

HTML10

CSS* Code Sample6

JavaScript2

Dev Derby

HTML5: Drag & DropSee the Winners

The Mobile WebCurrent Challenge, enter now!

Web Workers

HTML 5: Drag & Drop

External Resources

W3c Schools: Siti re con eatendis

Code Academy: Siti re con eatendis

W3c Schools: Siti re con eatendis

A. Search Results

Search results populate below search bar (and filters, if applicable). User can dismiss results and return to previous page, if needed, by selecting close (X) icon in right corner of results pane.

B. Topics & Details

Exact content is TBD, dependent on capabilities of new elastic search improvements. Included here are:

1. Category name (& visual, illustration, etc)
2. Document (or content) Type listed below category
3. Indicate if there is a code sample, or some extra content within result that makes it more valuable or appealing. (ie: & code sample)
4. Number of results related to each type.

C. Next Interaction

The next interaction and how we can get the user to their results asap still in discussion. We can either expand the selected choice and display results right here, or have a dedicated page as done at A List Apart (<http://alistapart.com/topics>). Getting the user to results with less clicks, is preferred.

D Possible Icon interaction

For prototyping: While viewing current search results, if user alters search term in field, either results refresh immediately, or we offer a visual to let them know the new search hasn't been performed yet. ie: switching search icon to refresh icon. ie:



Keep in mind...

- once user opens a search result, they should have an easy way to return to these results once again.
- consider common key commands best used to navigate search results
- consider how to display(number) results to correspond with key commands.

Ph 1: “stumble-upon” / horizontal navigation results bar

Docs

Community

Demos

Blog

A

◀ Next: Drag & Drop demo ▶

✓

✕

B

JavaScript Reference

HISTORY

EDIT

About this Reference

Some programming experience with a language such as C or Visual Basic is useful, but not required.

The JavaScript language is intended to be used within some larger environment, be it a browser, server-side scripts, or similar. For the most part, this reference attempts to be environment-agnostic and does not target a web browser environment. For demonstration purposes, this reference uses a function, `println`, which is not part of JavaScript and can be mapped to environment-specific functionality to display given values. For example, in a web browser `println` might have been defined as follows:

```
1 function println(string) {
2   window.alert(string);
3 }
```

Mapping to equivalent functionality in other environments is left as an exercise for the reader.

Formatting conventions

This reference includes descriptive syntax sections to demonstrate appropriate or common usage of the subject of documentation. Within these sections, all text literals to be reproduced verbatim are non-italicized, with the exception of ellipses. Words in italics represent user-defined names or statements. Any portions enclosed in square brackets (`[]`) are optional. A comma-delimited sequence that includes an ellipsis (`...`) indicates that the sequence is a list and all items in the sequence except the first are optional (e.g. only `param1` is required in "`param1, param2, ..., paramX`").

JavaScript history

Recent versions of Mozilla-based browsers support newer versions of JavaScript. The following table lists the JavaScript version supported

TABLE OF CONTENTS

About this Reference

Formatting conventions

JavaScript history

JavaScript/Browser support history

Where to find JavaScript information

Global Objects

General-purpose constructors

Typed array constructors

Error constructors

Non-constructor functions

Other

Functions and function scope

Statements

Operators and other keywords

Comments

E4X (extension)Deprecated

Appendix A – Reserved Words

Appendix B – Deprecated Features

TAGSFILES

A. Search navigator

The user can arrive at a docs page in the following ways:

- via Google Search
- via MDN Search
- via shared link kon Twitter or any other non-Mozilla site
- via link from another Mozilla related site. (mozilla.org, DevHub, etc)

When a user arrives at a doc via the above paths, the docs navigator is either populated with MDN links from search results or related docs.

(ie: when coming from DevHub, we can populate with a curated list, from Google Seearch we would serve up related links, from MDN we would serve search results.)

This promotes horizontal navigation within MDN, but also supports our goal in having a less jarring experience for the user as they find what they are looking for because they will be less likely to hit the back button.

B. Navigator Display

We can test a few layouts for this, but keep in mind keycommands and hierarchy of search results.

Functionality includes back and forward, ◀ ▶ as well as expanding the entire menu of results. ✓

- 1 JavaScript technologies overview

2 Data structures

3 JavaScript technologies overview

4 Data structures

5 Next: Drag & Drop demo

6 JavaScript technologies overview

7 Data structures

8 JavaScript technologies overview

9 Data structures

10 JavaScript technologies overview

11 Data structures

12 JavaScript technologies overview

Ph 1: Key Commands

TBD: select key commands for navigating search. Here are commands from Duck Duck Go and GitHub to help us use common commands found in web apps.

DDG <http://help.duckduckgo.com/customer/portal/articles/300862>
GITHUB: <https://github.com/scrod/nv/wiki/List-of-All-Keyboards-Shortcuts>

* also look at vim, emacs, and sublime text editor.

Keyboard Shortcuts

Last Updated: Jan 09, 2012 09:07PM EST



You can use keyboard shortcuts to navigate the results (as opposed to the mouse).

Open results:

- Enter or l or o — go to the highlighted result, or use it right away to go to first result.
- Ctrl/Cmd+Enter — open a result in the background
- d — domain search (if a result is highlighted)
- ' or v — open the highlighted result in a new window/tab. Since this uses JavaScript, you need to turn off pop-up blockers first.

Move around:

- ↓ or j — next result
- ↑ or k — prev result
- / or h — go to search box
- esc — get out of search box
- r — go to related topics (if any)
- s — go to misspelling link (if any)
- t — go to top
- m — go to main results
- 1 or l — open lbang dropdown

If you don't like the keyboard shortcuts, you can turn them off [in the settings](#).

Thank You

github

In the order of most to least useful:

⌘ L	Place insertion point in search/title area
Escape	Clear current search
Return	Begin editing note, creating it if necessary
Tab	Move insertion point between text areas
⌘ K	Select previous note in list
⌘ J	Select next note in list
⌘ D	De-select current note, restoring the search text
⌘ ⇧ V	Paste clipboard as a new note (Services Menu)
⌘ R	Rename the selected note
⌘ Delete	Delete the selected note(s)
⌘ ⇧ T	Tag the selected note
⌘ G	Find next occurrence of search text
⌘ ⇧ G	Find previous occurrence of search text
⌘ ↵	Open any URL under the insertion point
⌘ ←	At the beginning of a note-body, edit the title
⌘ →	At the end of a title, edit the note-body
⌘ ⇧ D	Bookmark the selected note
⌘ P	Print the selected note(s)
⌘ E	Export the selected note(s) to other formats

Text formatting shortcuts:

⌘ [(left square bracket) Outdent one or more lines
⌘]	(right square bracket) Indent one or more lines
⌘ Tab	Indent
⌘ T	Make text plain; remove styles
⌘ B	Embolden
⌘ I	Italicize
⌘ U	Underline

Last edited by scrod, 2 years ago

* Also, ‘s’ enters search/command mode. while ‘t’ enters navigation mode

= applicable to us

Document:	UX: Planning	Date:	24.03.2013	Author:	H. Habstritt
Project:	Mozilla: MDN Redesign	Sprint:	n/a	Email:	habber@mozilla.com