

A programmer's diary

Learning to code browser extensions

Marcin Borkowski

A programmer's diary

Learning to code browser extensions

Marcin Borkowski

This book is for sale at <http://leanpub.com/programmers-diary>

This version was published on 2023-06-26



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2023 Marcin Borkowski

Contents

1 Introduction	1
2 The sources	3
3 Getting started	5
3.1 What are extensions?	5
3.2 Your first extension	5
3.3 Your second extension	6
3.4 Anatomy of an extension	8
3.5 Example extensions	9
3.6 What next?	9
4 Concepts	11
4.1 Using the JavaScript APIs	11
4.2 Content scripts	11
4.3 Background scripts	13
4.4 Match patterns in extension manifests	14
4.5 Working with files	14
4.6 Internationalization	15
4.7 Content Security Policy	15
4.8 Native messaging	16
4.9 Differences between API implementations	18
4.10 Chrome incompatibilities	18
5 User interface	20
5.1 User interface	20

CONTENTS

5.2 Toolbar button	20
5.3 Address bar button	20
5.4 Sidebars	20
5.5 Context menu items	20
5.6 Options page	21
5.7 Extension pages	21
5.8 Notifications	21
5.9 Address bar suggestions	21
5.10 devtools panels	21
6 Howtos	23
6.1 Intercept HTTP requests	23
6.2 Modify a web page	23
6.3 Safely inserting external content into a page	23
6.4 Sharing objects with page scripts	23
6.5 Add a button to the toolbar	23
6.6 Implement a settings page	24
6.7 Working with the Tabs API	24
6.8 Work with the Bookmarks API	24
6.9 Work with the Cookies API	24
6.10 Work with contextual identities	24
6.11 Interact with the clipboard	25
6.12 Extending the developer tools	25
6.13 Building a cross-browser extension	25
7 Firefox differentiators	26
8 JavaScript APIs	27
8.1 Browser support for JavaScript APIs	27
8.2 action	27
8.3 alarms	27
8.4 bookmarks	27
8.5 browserAction	27
8.6 browserSettings	28
8.7 browsingData	28
8.8 captivePortal	28

CONTENTS

8.9 clipboard	28
8.10 commands	28
8.11 contentScripts	29
8.12 contextualIdentities	29
8.13 cookies	29
8.14 declarativeNetRequest	29
8.15 devtools	29
8.16 dns	30
8.17 dom	30
8.18 downloads	30
8.19 events	30
8.20 extension	30
8.21 extensionTypes	31
8.22 find	31
8.23 history	31
8.24 i18n	31
8.25 identity	31
8.26 idle	32
8.27 management	32
8.28 menus	32
8.29 notifications	32
8.30 omnibox	32
8.31 pageAction	33
8.32 permissions	33
8.33 pkcs11	33
8.34 privacy	33
8.35 proxy	33
8.36 runtime	34
8.37 scripting	34
8.38 search	34
8.39 sessions	34
8.40 sidebarAction	34
8.41 storage	35
8.42 tabs	35
8.43 theme	35

8.44 topSites	35
8.45 types	35
8.46 userScripts	36
8.47 webNavigation	36
8.48 webRequest	36
8.49 windows	36
9 Afterword	37

1 Introduction

Hello, dear reader! What you're reading right now is an experiment. Some time ago I decided to learn to write web browser extensions, with the aim of writing a book teaching how to write them, and document the learning process along the way. My assumption is that some insights into the thought processes of someone learning something new might be interesting to people. It may turn out that I was completely wrong—we'll see!

(A month later.) I am slowly making my way through the documentation I found on the web and writing down my impressions, and I decided that I need to update the introduction. See, I noticed that some patterns have emerged. Especially one pattern. It turned out that I am grumpier than I thought, and that this booklet became a bit too rantish. Sorry for that. I would like to make it crystal clear that my goal is not to pick on the authors of the docs I'm studying. Writing good documentation is *hard*, and there are plenty of reasons much of the documentation found on the web is not of the best quality or up to date. Many people just lack the resources—they don't have the time to write docs. A lot of software changes very rapidly and keeping the docs up to date is a constant uphill struggle. Many organizations lack the will to pour money (and time) to developing good docs. Most programmers do not have the abilities and experience to write well (not to mention hardly anyone *likes* writing manuals!), and especially with volunteer-driven FOSS projects there are no technical writers to aid engineers. In fact, now that more than a month has passed since I started this booklet, I realize that this is *far* from the worst documentation I've seen (and no, I won't show examples of worse docs, that would be cruel; -)). On the other hand, I don't want to edit too much what I've written so far—I'd prefer this booklet to stay a genuine record of my learning process. So, dear reader, please forgive me my

negativity and treat it more as a colleague sharing his frustration with you and less as a jerk making fun of people writing docs for Mozilla.

My general experience in life says that very good things can be results of bad things we do. (This is definitely not to justify them, of course!) Now that I noticed all that negativity in me while exploring the world of browser extensions, I can be more conscious about my emotions related to software in general, and take steps to make them more positive. As every programmer knows, making some piece of code *work* is a huge dopamine booster, so what I'm going to do from now on is to read slower and tinker with the actual code more. I sincerely hope this will make me less frustrated! And if that works, not only will the book be more pleasant to read (or so I hope), but some of you might want to learn from my errors and make your life better, using a similar approach. (See? Whatever I do, it seems I can't stop being a teacher...)

Anyway, back to reading. Where were we? Oh yes. The Pit of Despair. (Ok, I couldn't resist a little inside joke. I'm at the section about the settings page, if you really want to know.)

2 The sources

So, first things first. Where do I learn how to write browser extensions? [MDN](#)¹ seems a good place to start, no? There is a section devoted to web extensions in the “References”, but I don’t need a reference—I need a guide. Clicking “Guides” doesn’t seem to help—there are HTML guides, CSS guides, JS guides, accessibility guides and more, but no browser extension guide. Well. I click on the “web extensions” item in “References” menu (is it a menu? a tab? something else? no idea...), and the page I land on tells me about “extensive range of [tutorials and guides](#)”². Well. I go there and I see... something. Looks a bit like, well, a guide to writing extensions. But that’s not the whole story! A few paragraphs below there is this *another* item, called [Extension Workshop](#)³, which also looks like a guide for writing extensions! Does Mozilla have two competing products here? Are they just books written by different people? It’s a pity they didn’t tell me what’s the difference between those. Also, I already miss a *real* book—how am I supposed to know how long it is going to take me to study these guides? If I had a page count, I could make plans about how many pages I want to read daily or something... (And yes, I know that it often doesn’t make sense. For fiction literature, assuming a reading speed of 100 pages per hour may be perfectly reasonable. And back when I was studying math, we once had a full lecture—90 minutes weekly for one semester—based on about 20 pages of Rudin’s book on real and complex analysis...)

Ok, I’m getting grumpy. Let’s stop it. I decide to check out the “extensive range” first and leave the “extension workshop” for later.

¹<https://developer.mozilla.org/>

²https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/What_are_WebExtensions

³<https://extensionworkshop.com/>

Now that I've made a decision, I feel more excited than grumpy.
Good!

3 Getting started

So I think that copying the chapter structure from the guide I found may be a good idea. Hopefully the copyright police won't come to me with shotguns.

3.1 What are extensions?

Just an introduction page telling me what the possibilities for web extensions are. Nothing too exciting. One important piece of information—the API used to write extensions for Firefox is “to a large extent” compatible with the “extension API supported by Google Chrome and Opera”. Uh-oh. “To a large extent compatible” is a nice way of saying “incompatible”, right? I expected this, of course, but it's still a bit disappointing. I really hope the differences are minor enough that I won't have to bother too much, but this means I will need to test on Chromium, too.

3.2 Your first extension

So the first extension they are going to show me how to write adds a red border to every page coming from `mozilla.org`. Sounds good. As we all know, `border:1px solid red` is the `console.log` of CSS, so this is basically a “hello world” extension; -).

And they dare to tell me that “understanding how to use the command line terminal is a handy skill”! Kids these days... The only reason I don't *live* in command line is that I live in Emacs! I probably used command line before the folks who wrote this were *born*!

Ok, ok. Let's move on. They tell me to create a `manifest.json` using a "suitable text editor". Well, I know only one text editor which is "suitable". The `manifest.json` for this extension contains something called `content_scripts` (I assume this is a mandatory property in that JSON). And then they offer me two links, one to learn about `content_scripts` in general, and one to learn about matches, one if its subproperties.

Now don't get me wrong, we all love browsing the internet and Wikipedia-surfing. But this is a *textbook*. I'm ok with a textbook saying "we will cover this topic in more detail later, for now just type this and don't think too much"—but I really hope that I'm not *supposed* to click every link. What I hope for is for someone to have thought out the structure and have made it linear so that I can learn things in a resonable order. Links like that make me go FOMO. Let's not follow them and see what happens (although I admit that links like that are very distracting for me).

Reading further, I learn a bit about icons and installing temporary extensions. I also see *more* annoying links. There is one good thing about them, though: some of them lead to the "Extension Workshop" mentioned earlier, so I at least get an idea about what's that—it must be a more complete guide. Well, I'll check that up later anyway.

So, I did as they told me and played around with this extension, changing the color and style of the line, changing the background color, and changing the font family. (The last thing didn't work. I'm not a CSS expert by any means, but I suspected something was overriding my font selection. Inspecting the page confirmed my suspicions. After playing a bit, I was able to change the font, using `document.getElementsByTagName` and iterating over every element on the page instead of just modifying the style of the body element.)

Something is working. I like it!

3.3 Your second extension

Ok, so their suggestion of the second extension is a “tool” which lets you change the web page you’re viewing into a picture of an animal. How useful. (Though from what they tell me in the intro, it *will* showcase some *really* useful things, so let’s not get too grumpy.) Let’s ride with it!

One thing I notice almost immediately is that *lots* of information I’ve already read in the previous section (the one about my first extension) is repeated here verbatim. I guess this is because people will be skipping sections? Or forgetting what they read previously? Oh, I so miss a real book!

So, they now tell me to create a popup which will be shown when I click the toolbar button. The popup needs three files—an HTML file, a CSS file and a JS file. Sounds good. Turns out that the JS file is *huge*—a hundred lines just to show pictures of animals! (Well, 83 when I excluded blank lines and comments.) Let’s read it.

And done. Not to say I understood everything—the API still looks like a mystery to me. What is `browser.tabs`? Does `document` in the `listenForClicks` function refer to the popup only? What exactly are `sendMessage`, `insertCSS` and `removeCSS`? Questions, questions, questions. Hopefully they’ll get answered soon.

This makes me wonder if people who know little Emacs Lisp have a similar experience reading [my Elisp book](#)¹. I hope not, but it’s not easy to say. Note to myself: try to read it again one day, paying attention to stuff in the code which is not immediately explained.

Ah, and now that I scrolled down, I get to see more code. Of course. This time it seems to give more answers than questions (good!), but one question remains: why we hide everything in the code of the popup (`choose_beast.js`), but show the animal picture in the “content script” (`beastify.js`)? The guide doesn’t tell me now, so

¹<https://leanpub.com/hacking-your-way-emacs>

I doubt it will tell me that in the future. Anyway, maybe I'll find out myself. I'll probably just try to fix that architectural mistake myself when I learn more. If I succeed, it probably means it was a mistake indeed. If I fail, I'll probably understand the reason.

By the way, don't get me wrong: it's not me being grumpy again. I don't think they did a poor job by not telling me. I mean, I wish they *did* tell me, but it's so very hard to be aware that you *need* to tell the reader things like this... You need to anticipate their questions, and different readers may have different questions like that.

Anyway, they tell me next that I can use a command-line utility called `web-ext` to install a temporary extension. Cool! Let's try it out. I was wondering how it's going to know which Firefox to use (I often have more than one instance running—one for docs, email, company chat etc. and another one for developing). It turns out that it just runs a fresh instance of Firefox, which is very nice. And there's more—it watches the extension files and reloads it whenever I make any change to them. That is *really* cool! And the next section promises me to say something about “the anatomy of the extension”, so hopefully some explanations are coming.

3.4 Anatomy of an extension

Ok, so the beginning raises more questions (again). What are these categories exactly? Some of them look clear, some of them don't. And I won't even try to analyze the diagram. (Back when I was studying, I learned that some people just like to include nonsensical diagrams in their books. I never understood that. If the diagram isn't either making something clear or adding some information not present in the main text, why bother having it? There is this urge to break the “wall of text” with a diagram—but I've read hundreds of novels which don't have pictures, and I know it's perfectly fine not to have pictures!)

Coming back, there is again some mention of Manifest v2 and v3. This is something I don't get. I would expect v3 to be a strict superset of v2 in terms of features, and here I read that there are behaviors available in v2 but unavailable in v3. Are that bad behaviors and someone decided to forbid them? If so, why is v2 still around? Are they deprecated and v3 has something better I should use instead? I really hope to find out at some point. I did a cursory search, and it seems Manifest v3 is something newer, but not necessarily better, and that v2 is deprecated in Google Chrome... This means that the first example using Manifest v2 is probably not optimal, even if v2 stays in Firefox.

Anyway, it seems that the categories I mentioned in the beginning are more or less explained here. This is good—looks I have some overview of what an extension can contain and do.

3.5 Example extensions

This is just a huge list of example extensions, mentioning what things they illustrate. Wow. I really can't decide if its great or horrible. On the one hand, this allows you to choose which one to study if you want to learn a particular API. On the other hand, I'm convinced that a textbook should be more opinionated and that its authors should present you a series of carefully crafted examples, explaining new things along the way. This list is certainly a treasure trove for learners, but it kind of looks lazy that they just dumped this list on me with little commentary... Well, we'll see what happens next. (Incidentally, this is the title of the next section!)

3.6 What next?

Oh, so they just summarized briefly the journey so far and told me what I am going to find in this manual later on. They also mentioned the “Extension Workshop” I found earlier. Looks like my decision to start with this and not the workshop was good—it seems that the guide here is meant for beginners and the workshop is more advanced. I will get to it eventually. Still, the fact that I have completely no idea how much of this guide is behind me and how much is ahead of me makes me a bit uneasy. I do understand that the beginning sections are often easier and that the speed of studying is variable—after all, I dedicated a good portion of my life to mathematics, where you sometimes need 15 minutes and sometimes an hour or two for a page!—but a progress bar with a percentage, or an epub file, would be nice. On the other hand, maybe there are very few people who read this guide in its entirety. Good thing is, I don’t have a fixed deadline—I just want to get to the end, learning along the way. I assume about 20-30 hours will be enough—we’ll see.

4 Concepts

From the list of sections here I gather that this is one of the most important chapters. Let's dive in!

4.1 Using the JavaScript APIs

And this is just cheating! Plain cheating! Clicking on this section apparently brings me to *another chapter*! And clicking on *that* chapter just shows or hides its section list! So it seems that clicking the title of this section is the *only* way to get to this “ghost section”. No matter how useful this is, this is outrageous. I really don't want this diary to be one big rant, but this-is-the-most-horrible-user-experience-I-have-seen-in-a-very-long-time! I'm disappointed in you, Mozilla. I really am.

Anyway, this page seems really worth reading. I found the part on browser API differences especially important—it looks like the differences between Firefox and Chrome are really major, though they promise a way to write an extension working smoothly in both. (Apparently, Firefox uses promises and Chrome uses callbacks. Come on, it's 2023, who uses callbacks when promises and `async/await` are available?)

Also, they have a list of APIs. They really like long lists, these people.

4.2 Content scripts

So they remind me here that content scripts (they are the part of the extension which is allowed to interact with the actual page you are

viewing) are blocked on several Mozilla-owned domains. I accept this as a fact of life, but I'd really like to know *why* they do this. Sigh.

And the rest of this page is a mix of some useful information and stuff I cannot understand at this point.

Also, the example of connection-based messaging has one serious drawback. I'm a bit afraid this may become a common theme—any example should have the full scaffolding of a web extension, which is not convenient to put in a book. On the other hand, just a snippet of code is not something the user may copy to the devtools console and run to see what happens. (Admittedly, this is an advantage of Emacs and Emacs Lisp. The first useful example in my Elisp book was less than 10 lines of code, and it *just worked* right away, doing a useful thing. Here you need `manifest.json` and at least one other file. The remedy is simple—you put a short snippet in the book/guide and a full, working, tested example in a Git repo with examples. Good thing that they thought about providing so many full examples! Let's check if they have one about connection-based messaging... Oh. No, it looks they didn't.)

Also, the more I read this, the less it looks like a *tutorial*. It definitely started as one, but now it's leaning heavily on the side of a *reference*. I am now wondering if the “extension workshop” is better in that respect. (I am also wondering about the feelings of people reading my Elisp book. I learned about the [Diátaxis framework](https://diataxis.fr/)¹ after I already wrote it, but I am pretty confident that it is basically one big tutorial with a lot of interspersed explanation. I am not sure if this is a good idea—personally, I think it is a good idea for a book. I suppose most people want to complete a *tutorial* as fast as possible to learn what it teaches in a minimum time. With a *book*, the expectations are that it may take more time—and that it will provide deeper and more broad understanding than a short tutorial. At least, these seem to be my implicit expectations when reading a book.)

¹<https://diataxis.fr/>

Anyway, on to the section on background scripts. Let's move with this—now that I am somewhat disappointed with this document, I'd like to try to dive into the “extension workshop” sooner than later! I'm going to give this one a chance, though.

4.3 Background scripts

I hate to be grumpy again, but I really dislike this section. It is even more reference-y than the previous one (ok, I'm getting accustomed to this). Worse is the fact that there is *no* explanation why registering listeners in an asynchronous way won't work. And the two examples—of the (allegedly incorrect) asynchronous and the (correct) synchronous registering listeners—use completely different APIs (`browser.bookmarks.onCreated` vs. `browser.tabs.onUpdated`). Why is that? No idea. And it gets *yet* worse. In one subsection they *recommend* `browser.tabs.onUpdated`, and literally in the next one they say that you *should* use `browser.webNavigation.onCompleted` instead (because “the tabs API does not support filters”). Yet, when I click on the link to `tabs.onUpdated`, I can read that “You can filter this event”.

What. Is. Even. Going. On. Here?

My suspicion is that some time ago `tabs.onUpdated` *didn't* support filtering, then it acquired that feature, but the paragraph contradicting that was somehow missed and now it only serves to confuse the readers. Admittedly, with technology such as web extensions, which I heard is rather fluid and changes pretty often, this may be *very* difficult to avoid. Still, it leaves a very bad taste in my mouth.

Also, in the section about unloading background scripts they advise using the storage API to prevent data loss in the case of a crash “without receiving `runtime.onSuspend`”. Yet, the docs for that event say that “since the page is unloading, any asynchronous operations

started while handling this event are not guaranteed to complete”. Guess what? The storage API is asynchronous. I think that is what they mean when they say a bit later that “persisting data should be preferred rather than relying on `runtime.onSuspend`. It *doesn’t allow for as much cleanup as may be needed* [...]” (emphasis mine).

I think I might need to start writing my questions down and ask them on StackOverflow one day. What is the point of documentation if you need to go to some forums for clarifications anyway? Sigh (again).

Also, this reminds me of my favorite mode of reading books about technology. Read the book first (possibly skimming some parts), accepting the fact that you don’t understand everything. Then, try the technology out—write some code, play around etc. Then, read the book *again*, this time having much deeper understanding resulting from experience. This method is obviously rather slow, but the second reading can be very fulfilling. (Of course, it should *not* be necessary for very well-written books. I think you can understand everything in SICP² on the first reading (even if your understanding is shallow in the beginning)—at least, if you scratch your head and don’t know what is going on, it is because you didn’t pay enough attention, not because the authors were sloppy...

4.4 Match patterns in extension manifests

This is a pretty short section. Reference-style, but at least clear. I have nothing to say about it (besides the fact that it has no place in a beginner’s guide).

²<https://xuanji.appspot.com/isicp/>

4.5 Working with files

Yay, another reference section! From now on I'm going to stop complaining about those. Looks like the idea of a guide (implied by the "Getting started" chapter title and suggested by the "Concepts" chapter title) is long forgotten now.

So I read the first section, and guess what? It's much more like a guide, with a link to a proper reference at the end. What a rollercoaster!

Also, I skipped the section about storing files locally in IndexedDB. This looks extremely specialized; if I ever need this, I'll know where to look.

4.6 Internationalization

Ah. I18n. Such a wonderful topic—full of pitfalls for unsuspecting programmers. It turns out that i18n support for extensions looks actually pretty good, although you can always find cases where things get complicated and not very well supported. Dealing with grammatical gender is one of those—when addressing the user directly, we might want to use correct forms (which are different for men and women in many languages, like Romance or Slavic ones). Good luck with that! Even a seemingly simple thing as a plural form can be a trap. For example, in Polish you use a different form of many nouns for:

- one of them,
- 2, 3, 4 of them,
- 5—21 of them,
- and then the cycle repeats modulo 10 (more or less).

But i18n is a very complex topic anyway, and hardly any software can deal with all the corner cases.

4.7 Content Security Policy

Yet another reference-like section. Nothing interesting to see here, move on.

4.8 Native messaging

This is actually an interesting topic, at least for me. One of the extensions I actually use, [Org Capture](https://github.com/sprig/org-capture-extension)³, communicates with Emacs, and I can suppose that it uses the mechanism described here. Let's read on!

Well, first of all, it looks like it was written by a different person than the rest of the guide. This is a general problem with crowd-sourced texts. On sites like Wikipedia it is not that visible, because every page is a separate entity—and an encyclopedic style is pretty established anyway, so there is not a lot of stylistic variance. But if you try to write a book with more than one author, making it consistent is not very easy. (Trust me, I coauthored two books—one with two authors and one with three authors.) If you want to do it correctly, it requires either *very* good communication between the authors and an explicit, shared vision of the book they are writing, *or* a detailed style guide, and in practice usually both. (Do you use “you” or “one”? “We” or “I”? British or American spelling, in case of English? Formal or informal style? What about contractions? There are *lots* of little decisions like that when writing, and you'd better make them consciously and be consistent. When you're writing on your own, you may get away with dealing with these issues subconsciously, although it's risky even then. When there is more than one person writing, not only do you need to make these choices deliberately, but you also have to communicate them clearly and all coauthors must explicitly agree on them.)

³<https://github.com/sprig/org-capture-extension>

Anyway, in this chapter we get back to being vague. Sigh, again. For example, they say: “Create a JSON file called the «host manifest» or «app manifest». Install the JSON file in a defined location”. Where is the “defined location”? They *do* explain that on a linked page called “Native manifests”, but they could have said something along the lines of: “Install the JSON file in a defined location. This location is OS-specific and is explained on a page dedicated to «Native manifests»”. And the funniest thing here is, that page’s URL suggests it’s part of this whole document, but it appears that it is *not* in the table of contents on the left! How is that even possible? It starts to feel like a paragraph game of sorts, with a hidden room, instead like a proper book. I know, I know. I’m probably an old man yelling at cloud. But for all those hypertext aficionados out there, loving the non-linear structure of the Web, here is a piece of bad news: *time* is linear, so if you want to *teach* someone something via a written text, you should really think carefully about the order of presentation. And when you already have a well-defined order of presentation, why not make a proper table of contents, containing *every single chapter and section*?

As a side note, I must admit to one guilty pleasure here. In one of the books I had some say in, I pushed for inclusion of a “secret chapter”, not mentioned in the table of contents. So technically, I broke my own rule. Still, I feel justified—the chapter contains no important information related to the book topic, it’s basically one big Easter egg for devoted readers who don’t skip parts of the book, and you don’t lose *anything* but a bit of giggling if you don’t find it.

Coming back, it is sad (but not surprising) that there are a lot of differences between handling native app messaging in Firefox and Chrome, and also between MacOS/GNU+Linux and Windows. All of this suggests to me that actually using this feature is not something I would like to deal with when distributing an extension to regular users, who may have various OSs, browsers and amounts of experience. On the other hand, if I decide to write an extension

for myself, to make various things easier *for me*, that seems to be a non-issue.

Well, and one more thing. I decided to satisfy my curiosity and peeked into the source code of the Org Capture extension I mentioned above. And it turns out that no, it doesn't use the "native messaging" feature at all. But that's a side note.

4.9 Differences between API implementations

"Among the various browsers supporting the extensions API, Firefox is the most compliant with the emerging standard, and is, therefore, your best place to start when developing browser extensions", says this page. Well, it's MDN, so it's not surprising that they advertise their browser, but I'm still happy that as a Firefox user I'm kind of privileged here. On the other hand, Chrome-based browsers have a much larger market share, so if I want to do anything larger than writing an extension for myself, I'll have to deal with that anyway...

Also, it is hard to believe that "Firefox is the only major browser that has implemented promises for the extensions API. All other browsers use callbacks". Come on, Chrome! It's 2023, nobody wants to use callbacks anymore when promises (and `async/await`) are available!

4.10 Chrome incompatibilities

I hate to be The Grumpy Guy™ again, but it looks that bad structuring of this document strikes again. Right after the section about differences between implementations we get a section specifically about incompatibilities between Firefox and Chrome. I know

Chrome is the one with the most market share, but where are the sections about the others?

Also, reading this I have just learned that Chrome's callbacks don't even support the traditional JavaScript "error-first" approach—you need to refer to `chrome.runtime.lastError` to check for errors. What is that? Looks like the seventies might call and ask for their error handling back...

And the rest of this section is just a big reference list. And from skimming this list I can infer that the differences may cause some ugliness in the code, and maybe even take considerable experience to account for properly.

5 User interface

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

5.1 User interface

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

5.2 Toolbar button

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

5.3 Address bar button

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

5.4 Sidebars

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

5.5 Context menu items

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

5.6 Options page

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

5.7 Extension pages

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

5.8 Notifications

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

5.9 Address bar suggestions

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

5.10 devtools panels

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

6 Howtos

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

6.1 Intercept HTTP requests

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

6.2 Modify a web page

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

6.3 Safely inserting external content into a page

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

6.4 Sharing objects with page scripts

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

6.5 Add a button to the toolbar

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

6.6 Implement a settings page

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

6.7 Working with the Tabs API

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

6.8 Work with the Bookmarks API

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

6.9 Work with the Cookies API

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

6.10 Work with contextual identities

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

6.11 Interact with the clipboard

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

6.12 Extending the developer tools

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

6.13 Building a cross-browser extension

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

7 Firefox differentiators

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8 JavaScript APIs

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.1 Browser support for JavaScript APIs

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.2 action

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.3 alarms

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.4 bookmarks

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.5 browserAction

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.6 browserSettings

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.7 browsingData

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.8 captivePortal

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.9 clipboard

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.10 commands

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.11 contentScripts

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.12 contextualIdentities

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.13 cookies

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.14 declarativeNetRequest

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.15 devtools

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.16 dns

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.17 dom

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.18 downloads

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.19 events

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.20 extension

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.21 extensionTypes

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.22 find

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.23 history

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.24 i18n

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.25 identity

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.26 idle

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.27 management

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.28 menus

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.29 notifications

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.30 omnibox

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.31 pageAction

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.32 permissions

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.33 pkcs11

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.34 privacy

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.35 proxy

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.36 runtime

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.37 scripting

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.38 search

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.39 sessions

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.40 sidebarAction

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.41 storage

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.42 tabs

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.43 theme

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.44 topSites

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.45 types

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.46 userScripts

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.47 webNavigation

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.48 webRequest

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

8.49 windows

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>

9 Afterword

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/programmers-diary>