JENS OLIVER **MEIERT**

FRONTEND DOGMA

UPGRADE YOUR HTML HTML HTML HTML II

FOREWORD BY MANUEL **MATUZOVIĆ**

Upgrade Your HTML II

10 More Examples to Improve Your Markup

Jens Oliver Meiert

This book is available at http://leanpub.com/upgrade-your-html-2

This version was published on 2024-11-01

Tweet This Book!

Please help Jens Oliver Meiert by spreading the word about this book on Twitter!

The suggested hashtag for this book is #htmlupgrade.

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

#htmlupgrade

Contents

Foreword 1

Intro 2

Acknowledgments 3

1. Cut the Fat 4

2. Mark Titles of Works Correctly 6

3. Review and Improve Third-Party Code 7

4. Keep It Simple 8

5. Use Lists 9

6. Skip `type` 10

7. Work Your Way Up 11

8. Know the Past, Know the Future 12

9. Ignore AMP 13

10. Avoid Prefetching and Prerendering Everything 14

Outro 15

Feedback 16

About the Author 17

About *Upgrade Your HTML II* 19

Foreword

It's impressive how much effort we've put into optimizing the way we write and organize our CSS and JavaScript files, especially in the last 10 years. We went from writing large files to splitting code into components; we care about maintainability, performance, modularity, and scalability. While there are also downsides to these developments, it's great that we attempt to get the best out of frontend languages.

At the same time, it seems that we've forgotten to put similar effort into the quality of our markup. Documents are bloated, inaccessible, most pages contain invalid HTML, and we only use a fraction of the 110+ elements we have at our disposal. It's hard to pin down the exact reasons, but it's fair to say that the Web is in a bad shape and we should pay more attention to writing clean and semantic documents. We can do that by starting to question our code as well as code provided by frameworks and libraries.

In this little book, Jens Oliver Meiert shares 10 examples where he does exactly that; he takes code from real websites and tries to optimize them right down to the last detail. While his approach is radical in some cases, the message counts: analyze, scrutinize, optimize. Take Jens's advice and upgrade your HTML, so your users will benefit from clean and semantic HTML documents.

—Manuel Matuzović

Intro

HTML is one of the most important standards of the Web. It's the *backbone* of the Web, as the first edition of this book explained. HTML is important.

Therefore, HTML is one of those standards everyone puts on their CV. Everyone can write HTML. But, as the first edition of this book suggested, too, that's questionable. HTML is more complex than even experienced developers admit, and has not nearly been exhausted. The craft of HTML is lacking. Not surprisingly, we even *teach* HTML the wrong way (to be demonstrated elsewhere).

Clearly, there's always something to nitpick in people's HTML code. That's what this book does. That's what this whole book series does. But it tries to do so constructively; fairly; and playfully. You can always upgrade your HTML, and we're not here to point fingers, but to have a look at the HTML landscape around us and take away a thing or two. It's not about explaining the world—even if that rests on HTML. Let's go.

—Jens Oliver Meiert

Who is this Jens guy? There's a section at the end of this book for that (*About the Author*). Jens has spent much of his life working with HTML and CSS, including developing compact HTML/CSS frameworks, defining quality HTML/CSS coding guidelines, and maxing out HTML/CSS optimization options.

Acknowledgments

With great thanks to Manuel Matuzović for reviewing this booklet and contributing the foreword, Gabriele Kretzschmar and Kirsty MacRae for reviewing and editing, and Ingo Helmdach for sharing design feedback.

1. Cut the Fat

```
<a rel="nofollow noopener noreferrer" href="https://www.facebook.com/sharer/sharer.p\
hp?u=https%3A%2F%2Fexample.com%2Fexample%2F%3Futm_source%3Dfacebook%26utm_medium%3DS\
ocial%26utm_campaign%3DSocial-Pug" class="dpsp-network-btn dpsp-facebook dpsp-no-lab\
el dpsp-first" ><span class="dpsp-network-icon"></span><span class="dpsp-network-lab\
el-wrapper"></span></a>
```

The first edition of *Upgrade Your HTML* had two chapters on writing HTML in general (*Think Through Your Markup* and *Question Your Frontend Code*). This edition and future ones will stress the importance of monitoring your overall markup, too.

The sample code above uses too much code. To illustrate that, it's better not to look at the code itself but its purpose: Allow the user to share the respective page on Facebook.

When you hear that, the first response should not be three `rel` values, four classes, two child elements with one class each, but: one link. Something like this:

```
<a href=https://www.facebook.com/sharer/>Share on Facebook</a>
```

It behooves, then, to check whether this code is enough.

If it isn't, it's useful to check each piece of code whether it's necessary. (That's the school of Minimal Web Development.)

It's hard and sometimes impossible to reproduce and relate to *all* decisions made by other developers—especially when we don't know them and their projects. In this case, we can infer a few things, starting with that they will have liked to share a *particular* page. We'll assume the reference to that page should include a few parameters for their site analytics. Therefore, we'll keep the link they've already used:

```
<a href="https://www.facebook.com/sharer/sharer.php?u=https%3A%2F%2Fexample.com%2Fex\
ample%2F%3Futm_source%3Dfacebook%26utm_medium%3DSocial%26utm_campaign%3DSocial-Pug">\
Share on Facebook</a>
```

Given how specific the use of `rel` was, we can assume that their `rel` values were also intended and needed. However, as the link would not open in a new tab or window—which in most cases, it shouldn't—, `noopener` and `noreferrer` are not needed. Only `nofollow` remains. Reluctantly.

```
<a href="https://www.facebook.com/sharer/sharer.php?u=https%3A%2F%2Fexample.com%2Fex\
ample%2F%3Futm_source%3Dfacebook%26utm_medium%3DSocial%26utm_campaign%3DSocial-Pug" \
rel=nofollow>Share on Facebook</a>
```

The rest is speculative and gives reason for concern. One, two, maybe three of the classes may be necessary, but it's impossible to tell which ones. The `span` elements seem unnecessary, though that assumption is still speculative. What's entirely been missing is some cue for users as well as search engines what the element was about at all—text content, which "Share on Facebook" now covers.

For all we know, that last snippet is the minimum we need. Less code, more content.

Why Less Code?

Why all of this, why this focus on so little code? What's the purpose of this all, does that not make our work more difficult than it already is?

Here's my thinking! Less code:

- is less code to write
- is… or can be easier to understand
- is usually faster (at least the payload is smaller)
- is often easier to maintain
- is more beautiful

The first reason is obviously correct, because it's tautological.

Understandability and maintainability depend on a few factors, particularly whether one inadvertently drifts into writing magic code. I believe that's not an actual danger with HTML: Assuming the code is still valid, understandability and maintainability don't seem to suffer when, for example, leaving out optional code—or would you suggest that `<p>Hi` is harder to understand than `<p>Hi</p>`?

Less code (meaning a smaller payload, which leads to improved performance) seems to reflect solid reasoning. I'd argue that exceptions prove this rule, and that we enter the realm of micro-optimization when we reject them.

That less code is more beautiful—this is certainly individual, subjective, personal. Yet personally, subjectively, individually, I cherish and promote minimal HTML, like the smallest valid document:

```
<!DOCTYPE html>
<title>␣</title>
```

2. Mark Titles of Works Correctly

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/upgrade-your-html-2.

3. Review and Improve Third-Party Code

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/upgrade-your-html-2.

4. Keep It Simple

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/upgrade-your-html-2.

5. Use Lists

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/upgrade-your-html-2.

6. Skip `type`

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/upgrade-your-html-2.

7. Work Your Way Up

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/upgrade-your-html-2.

8. Know the Past, Know the Future

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/upgrade-your-html-2.

9. Ignore AMP

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/upgrade-your-html-2.

10. Avoid Prefetching and Prerendering Everything

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/upgrade-your-html-2.

Outro

What did we cover?

Cut the fat! If it looks like too much markup, it probably is.

Mark titles of works correctly. Don't `em`phasize what you can `cite`.

Review and improve third-party code. Some, much, or most third-party code is, unfortunately, of bad quality. Take a closer look and improve it yourself, if need be.

Keep it simple! (Qed.)

Use lists. When you list things, that is.

Skip `type`. For style sheets and (JavaScript) scripts, you usually don't need it.

Work your way up. Or down. Review elements, attributes, contents.

Know the past, know the future. That was probably not a great chapter title.

Ignore AMP. You don't want it.

Avoid prefetching and prerendering everything. Resource hints are scalpels, not sledgehammers.

*
**

Thank you, and see you in the next episode of *Upgrade Your HTML*!

Feedback

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/upgrade-your-html-2.

About the Author

Jens Oliver Meiert is a web developer and author who, after several years as a tech lead at Google, works as an engineering manager at Jimdo. He's an expert in web development, specializing in HTML and CSS optimization. Jens contributes to technical standards and regularly writes about his work and research, particularly on his website, meiert.com.

Other titles by Jens Oliver Meiert:

The Web Development Glossary 3K (2023)

> What is a BHO? CQRS? An EMD? What is Goanna? Hooking? Sharding? How about dynamic color, the phoenix server pattern, or the rules of ARIA? Covering more than 3,000 terms and concepts, and including explanations from Wikipedia and MDN Web Docs, *The Web Development Glossary 3K* provides an overview of web development unlike any other book or site.

Available at Apple Books, Kobo, Google Play Books, and Leanpub. (Try the glossary online at WebGlossary.info!)

The Little Book of Little Books (2021)

> *The Little Book of Little Books* consists of lovingly polished editions of *The Little Book of HTML/CSS Frameworks* (originally published in 2015), *The Little Book of HTML/CSS Coding Guidelines* (2015), and *The Little Book of Website Quality Control* (2016).

Available at Amazon, Apple Books, Kobo, Google Play Books, and Leanpub.

CSS Optimization Basics (2018)

> Are you unsure about your style sheets' quality, or whether you've maxed out your options? *CSS Optimization Basics* covers the necessary mindsets, discusses the main optimization methods, and presents useful resources to write higher-quality CSS.

Available at Amazon, Apple Books, Kobo, Google Play Books, and Leanpub.

On Web Development (2015)

> *On Web Development* bundles 134 articles and the last 11 years of technical writings by Jens Oliver Meiert (meiert.com). Freshly reordered and commented, the articles cover processes and maintenance, HTML and CSS, standards, as well as development and design in general; they include coding basics and principles, carefully scathing criticism, and tips and tricks and trivia.

Available at Amazon.

The Little Book of HTML/CSS Frameworks (2015)

> With the speed of web development today, it's little wonder that so many frameworks are available, since they come with a promise of saving development and design time. But using the wrong framework, or wrongly using the right framework, can be costly. This concise book shares higher-level ideas around web development frameworks that govern HTML and CSS code, whether you're looking at an external option or planning to build your own.

Available at O'Reilly.

About *Upgrade Your HTML II*

Written by Jens Oliver Meiert.

Published by Frontend Dogma, c/o Jens Oliver Meiert, Apartado de correos 3, 36070 Pontevedra, Spain.

Editors: Gabriele Kretzschmar, Kirsty MacRae
Reviewer: Manuel Matuzović

Contact +34-610859489 or info@frontenddogma.com for questions and more information.

Follow Frontend Dogma on Mastodon (or other networks).

[1.5.22]