JENS OLIVER **MEIERT**

UPGRADE YOUR HTML HTML HTML HTML

FRONTEND DOGMA

Upgrade Your HTML

10 Examples to Improve Your Markup

Jens Oliver Meiert

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

This version was published on 2024-11-01

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.

© 2019 Jens Oliver Meiert

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

Intro

HTML is the backbone of the Web. We can build a website without JavaScript, even without CSS, but without HTML—*any* HTML—, no, we can't.

HTML is so important, yet so easy. Everyone writes HTML, and everyone says they can write HTML. They don't just mean they're able to write HTML, but that they write *good* HTML, where "good" means "high quality."

That would be great news, because writing good HTML is more important than writing good CSS or good JavaScript. This is at least the view of the first paradigm of web development. This paradigm sees a benefit in separating concerns, and assumes, as the end goal, a large code base of many HTML templates and files with few style sheets and scripts.

The problem is not that there's also a new paradigm, in which components may include HTML and CSS and JavaScript (or ECMAScript, if we want to adjust terminology). It's that *not* everyone writes good HTML. There are many reasons for that.

The most important one is that HTML isn't actually that easy, because it really is complex. Want an example?

HTML 5.2, which is the last HTML recommendation by the W3C (HTML is maintained by the WHATWG), contains 111 elements alone. *111.* How many do you know? How many does the average web developer know? What does it mean for their markup if they don't know all the elements?

Want another example? When people talk about HTML elements like `html` or `a` or `p`, then most of the time they do mean *elements*. But what they then *talk about* is "tags." Google, as of September, 2019, finds 25 million occurrences of "html tags" alone. HTML elements, what people mean, but don't call by name? 2.4 million hits, a tenth. This begs the question how well developers understand HTML if they can't tell the difference between elements and tags.

Another one? In a few years, HTML will celebrate its 30th birthday. Very cool! We will certainly have maxed out all options to reduce HTML payload to improve performance, wouldn't we? Well, no. One of the major options at our disposal to reduce HTML payload is not to write HTML that can be left out without a document turning invalid (please validate, by the way). Unfortunately, almost no one uses that option. Web developers have their concerns and habits, yes. The point is that the method is so under-utilized, we cannot speak of HTML mastery here, either.

We could go on, adding data and anecdotes to how HTML is both important and yet not well-understood, nor well-used. The route I want to take is to select 10 examples, from the wild, to make improvements to the respective HTML code. That code has been anonymized, for this booklet is not about pointing fingers. It's to look at HTML and note what else we can do, how we can improve it, how we can: upgrade it.

Welcome to this first and light and playful edition of *Upgrade Your HTML.*

—Jens Oliver Meiert (short: Jens)

PS.
The booklet really is about HTML. Not SGML, not XHTML, not XML, just HTML.

Acknowledgments

With special thanks to Kevin Khaw, Merci Niebres, Gabriele Kretzschmar, and Kirsty MacRae for reviewing this booklet, as well as Jukka K. Korpela for providing technical feedback.

1. Respect the `title` Attribute

```
<a mode="dark" type="expanded" name="politics" href="/politics" data-analytics="head\
er_expanded-nav" title="visit the US Politics section" class="nav-linksstyles__Link-\
sc-1tike8v-0 nav-linksstyles__SectionLink-sc-1tike8v-4 bwVECJ">US Politics</a>
```

You've read the short intro, and the first thing you're about to witness are equally short chapters. The way this works is that each chapter has a theme, announced in the title, for which I'll provide code from an undisclosed live website that I'll make remarks and recommendations about.

This snippet, taken from a U.S. news site, is interesting in many ways. There are overt and covert lessons to be learned. An example of an overt lesson? It would probably be smart to drop `name=politics`, or to use an `id`. A covert lesson? `mode=dark` and `type=expanded` smell and need examination. On the sparse information we have we can't tell. Another overt one? No matter the naming school, the class names are poor—after processing in the tool chain they end up verbose yet say way too little.

The point, however, is this: Don't use `titles` that parrot what the link text says. Don't use `titles` when they *almost* say what the link text says. In fact, use the `title` attribute sparingly: Use it only when context and contents do not provide sufficient cues to understand an element, and only once you balanced the need for `title` maintenance with the needs of your audience.

`titles` can be useful for users (UX +) but, relatively speaking, they are costly to add and maintain (DX –). Note "relatively speaking": It's easy to key in a few characters, but only looking at the few characters one `title` would use neglects that more such attributes might be added, and that all of them have to be maintained. Be mindful. Respect the `title` attribute.

Invisible content, in my view, is a disaster. Meta descriptions, `alt` attributes, `title` information… even with the support of the most advanced content management systems they're a pain to keep track of and maintain. The only thing that has kept this at bay is that the respective resources are often rather stable, that is, the image described "My first cup of coffee in 2002" is likely only to experience an update once it's being kicked out altogether. Be careful around invisible content unless it hugely benefits users, as with `alt`.

What's the bare minimum for the sample code above?

```
<a href=/politics>US Politics</a>
```

Much code that I'm showing in between, and all minimal code presented at the end of each chapter has been pruned of optional markup, like optional tags and quotes. Whether and how you adopt this practice is entirely up to you and your needs. It's my strong suggestion to omit such optional code, at least for production code.

2. Stop Escaping &

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

3. Think Through Your Markup

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

4. Don’t Double Boolean Attributes

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

5. If It Can Be Done Using an HTTP Header, Use an HTTP Header

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

6. Avoid `data` Images

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

7. Don’t Use the Classic “Clearfix”

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

8. Be Clear About Attribute Functions

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

9. Question Your Frontend Code

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

10. Add Meaning, Prune Meaninglessness

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

Outro

“Upgrade Your HTML”—or was this all “Downgrade Your HTML”? You’ve read ten short chapters and have encountered as many samples of HTML that were discussed and minimized.

Is this just correlation, or was there a mistake that all these upgrades resulted in less code? Is less code an upgrade all by itself?

I’m biased—I specialize in using as little code as possible and I love the idea of “minimal web development.” Yet I believe there’s a strong correlation. Sometimes, more markup leads to improvements and upgrades—for example, making sure an image gets its `alt` attribute. Oftentimes, it’s *less* code that improves matters, for less code reflects focus, improves performance, understanding, and maintainability.

As with many rules, there are exceptions, yet I don’t find it surprising that what this little book shows is that improvements often come with removing rather than adding code.

The aspect I like to close with, however, is a different one: There are always things to improve. (Even if Jesus or another amazing person joined us right here, right now, someone could find something objectionable about that.) Yet, as that statement, as the examples, as I tried to show, that’s not a negative thing, for if there’s *always* something to improve, it doesn’t mean that what was there before was necessarily “bad.” It’s also *constructive* to make suggestions for improvements, especially when there’s reason to believe that these do, indeed, solve what’s been pointed out as problems.

Making constructive suggestions around actual uses of HTML has been the idea behind this booklet. It’s likewise the idea for a series of such writings, similarly brief—for there are always things to improve, for there’s always HTML to, you name it: upgrade.

Thank you for following along, and the best wishes to you, and your HTML.

PS.
If you like to learn about other HTML blunders, have a look at Manuel Matuzović’s “HTMHell.” HTMHell shares the same spirit as *Upgrade Your HTML*, and Manuel does a great job explaining issues with HTML, and how to avoid or fix them.

Feedback

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

About the Author

Jens Oliver Meiert is a web developer and author (ex-Google, W3C, O'Reilly). He specializes in the management and quality assurance of complex websites, the development and deployment of web frameworks, as well as code efficiency and maintainability. 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*

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
Reviewers: Kevin Khaw, Merci Niebres

While this book has been produced with great care, the author, publisher, and contributors assume no liability for the up-to-dateness, correctness, and completeness of the information provided. Liability claims based on the use or non-use of this information are excluded, unless author, publisher, or contributors can be proven to have acted with intent or gross negligence. Use of the information in this book is on your own responsibility. When using code or content subject to open-source licenses or the rights of others, it is on you to ensure compliance with the respective licenses and rights.

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

Follow Frontend Dogma on Mastodon (or other networks).

[1.5.16]