

Peer Reviews in Technical Communications: An Example

In late Spring 2014, Penny Orwick and I undertook separate peer reviews on an example piece of technical writing. The following page shows the original text, with subsequent pages that contain my comments and suggestions to the writer. Penny's comments are published on the Steyer Associates blog.

This is part of a series of topics on technical communications, emphasizing writing for software documentation, that Penny and I write for Steyer Associates. The topic that launched this discussion: [Lions and Tigers and Peer Review—Oh No!](#)

See the archive of Steyer Associates blogs at <http://steyer.net/blog>.

[See the archive of my related articles](#), that include similar topics for fiction writers.

— Annie Pearson

May 2014

www.anniepearson.com

[Developer Central](#) > [Learning](#) > [MyCSS](#) > **ORIGINAL FIRST DRAFT**

Beginner's Guide to CSS

This page contains resources to help developers get up to speed with CSS. For additional resources see the [Learning Resources](#) page. For more resources see the [Resources](#) page.

Tutorials & Articles

- [Learning CSS Overview](#)
Which CSS API should I use--System.CSS, HTML to CSS, or MyCSS? What tools are available in MyTools? Also includes links to many articles.
- [Building CSS Solutions](#)
Extensive guidance for working with MyCSS, CSS and the database, schemas, stylesheets, and more.
- [CSS Tools](#)
Using MyTool's integrated CSS editor, schema explorer, and XCSST debugger.
- [MyCSSserv](#)
MyCSS Core Services (MyCSSserv) is the native API for building high-performance CSS-based solutions that adhere to CSS 1.0.
- [XCSST](#)
The Extensible CSS Transformation (XCSST) lets you transform the content of a source CSS file into another file that is different in format or structure.
- [CSS Namespaces](#)
All about namespaces in an CSS file.

=====
[Developer Central](#) > [Learning](#) > [MyCSS](#) > Learning CSS - Overview

Learning CSS - Overview

CSS is an almost universally supported way for describing the look and formatting of a document written in a markup language for display across applications and platforms. MyTools was a pioneer in CSS development and takes pride in the quality, performance, and standards conformance of our CSS technologies. We have a family of CSS technologies that allows users with differing requirements to do what they need, as simply and efficiently as possible.

Which CSS application programming interface (API) should you use? Here are our top-level guidelines:

- If you are writing managed code targeting the MyTools Framework in C#, Basic, J#, managed C++, or any other managed language, you should use [System.Css](#) and/or [HTML to CSS](#) in the MyTools Framework.
- If you are writing native code using C, C++, or a scripting languages you will probably want to use the [MyCSS](#) library:
 - MyCSS6 is the latest version that's included with MyApps 2009 and all versions of MyApps2010. New applications should be using MyCSS6.
 - MyCSS5 is an older library optimized for MyApps 2003 and 2007 and can only be used on machines that have a MyApps license.
 - MyCSS4 is nearing deprecation. MyCSS6 should be used for new applications
 - MyCSS3 is included in MyApps 2000 and higher.
 - [Comprehensive list of all MyCSSversions and variants](#)
- If you are writing native code and your application has tight performance or memory constraints, consider the [CSSLite API](#).

What CSS Tools Are Available? MyTools offers a core collection:

- [CSS Editor](#)
- [CSS Schema Explorer](#)
- [XCSST Debugger](#)
- [CSS Tools in MyTools overview](#)

[Developer Central](#) > [Learning](#) > [MyCSS](#) > **FIRST DRAFT PEER REVIEW BY AnnieP**

AP> All my peer review comments look like this. I didn't edit for detail, because this content is still too rough and incomplete for a detail edit at this point. Confusing terms & style issues are in yellow.

Beginner's Guide to CSS

AP> I highlighted my initial confusion — is this topic about our proprietary tools (MyCSS) or about the worldwide standard CSS technology? Because CSS is a well-established standard, I think we only want to teach our toolset and let external sources teach the standard. We don't have the budget to document what's already widely available on the web.

This page contains resources to help developers get up to speed with CSS. For additional resources see ~~the Learning Resources page.~~ For more resources see the [Resources](#) page.

AP> I struck out the repetition ... But neither [duplicate] cross-references tell me what kinds of Resources are at that target. Move the [Resources](#) link to the navigation list below and say what kind of resources are available. [And please check style guide for structure & punctuation of cross references.] Also, the 1st sentence says who the topic is for (which is the right way to start — let people know where they landed]. The next sentence should send non-novice developers to the appropriate master topic for them.

Tutorials & Articles

AP> None of these topics appears to be a tutorial (which would include exercises). These all appear to be overviews and descriptions of the product features and components.

Also — if there's a subheading in a topic, there has to be at least 2 subheads. Just strike this headline.

*AP> I'm not marking all the style items such as **&** above (instead of and) or **—** (instead of em dash). Please check the Punctuation topic in the style guide and fix these before passing to peer reviewers or the editor so other team members aren't wasting their time on basic mechanics.*

AP> Our style is parallel construction for headings in a set of topics and in bullet lists.

Please use all gerunds to describe the developers' tasks. Or, if you are merely describing tool capabilities, list them under on collective noun.

Because this is Beginner's Guide, I expect this to be an ordered list of topics, yet I can't tell from this list whether there is an order for learning. Explicitly state whether or not topics must be read in order.

In addition to headings not being parallel, the descriptions of the topics should also be parallel.

I suggest recasting each of the descriptors as "How to..."

You also switch between "I" and "you" as the target—our style is to use second person ("you") to refer to the reader, and plural third person when referring to the writer's voice ("we recommend...")

... And again, with each heading and its descriptor, clarify the difference between CSS (a standard technology) and MyCSS (our proprietary tools)

- [Learning CSS Overview](#)

Which CSS API should I use—System.CSS, HTML to CSS, or MyCSS? What tools are available in MyTools? ~~Also includes links to many articles.~~

AP> An overview presents the structure of the collection of topics. However, the content is more a "Getting Started" topic. Recast heading, and change to 2nd person.

Also, in the first topic, MyCSS is described as a tool set. Here, and below, it's called an API and a library, and other tools are presented as supporting MyCSS. Please follow the style guide definition for tool ("a

feature that aids in accomplishing a task,” and please clarify and differentiate here among tasks, tools, and APIs.

- [Building CSS Solutions](#)

Extensive guidance for working with MyCSS, CSS and the database, schemas, stylesheets, and more.

AP> Per style guide, “build” means to compile and link code. The developer’s tasks haven’t been made clear here, but I believe the solution artifacts are “created” and there is no build process. Please clarify and fix the headline as appropriate.

[I think this topic should be titled “Creating CSS Solutions with MyCSS” — correct?]

- [CSS Tools](#)

Using **MyTool's** integrated CSS editor, schema explorer, and XCSST debugger.

AP> Check style guide — don’t use a possessive with a proprietary/trademarked term.

I believe the topic name should be “MyTool Features for CSS Solutions” — there are many CSS tools in the world, and this topic will only describe what we deliver via the MyTool suite.

- [MyCSSserv](#)

MyCSS Core Services (MyCSSserv) is the native API for **building** high-performance CSS-based solutions that adhere to CSS 1.0.

AP> Ditto earlier note on “build” ...

What’s the difference between MyCSSserve as native API and the various MyCSS# versions described in the next topic?

Because this is a beginner’s guide, you need to provide information here that helps developers new to MyCSS build a mental map to understand “MyCSSserv” in relation to the API set you describe in the next topic, where you never mentioned MyCSSserv.

- [XCSST](#)

The Extensible CSS Transformation (XCSST) **lets** you transform the content of a source CSS file into another file that is different in format or structure.

AP> Why isn’t the XCSST debugger discussed in this topic instead of in the CSS Tools topic?

Also, features don’t “let” developers do things; developers use the feature to achieve their goal.

[For a gerund title, I think this should be “Transforming a Source CSS file with XCSST” — correct?]

- [CSS Namespaces](#)

All about namespaces in an CSS file.

AP> This is a reference, isn’t it—not a part of the MyCSS API-and-toolset?

I suggest you have a quick reference section, labeled as such, and the namespaces are just an entry in the quick reference.

=====

Learning CSS - Overview

AP> This isn't a learning topic—it's a truncated overview of product features. Either change the title to be an overview of MyCSS, or recast the topic so it's structured to match the developer's processes, and then "teaches" MyCSS in relation to the developer's tasks.

CSS is an almost universally supported way for describing the look and formatting of a document written in a markup language for display across applications and platforms. MyTools was a pioneer in CSS development and takes pride in the quality, performance, and standards conformance of our CSS technologies. We have a family of CSS technologies that allows users with differing requirements to do what they need, as simply and efficiently as possible.

AP> "universal" is an absolute ... it doesn't take "almost" as a modifier. What's the difference between "look," "formatting" and "display" in this definition?

Strike sentence #2 — it's marketing language and doesn't belong in a user guide.

Sentence #3 doesn't clarify what user tasks / develop needs will be described in the Learning guide.

Without first summarizing the developer's tasks, the choice of APIs in the next subsection has no context.

Also, technologies don't "allow" (or "enable") — Developers can use the MyCSS technologies to solve their problems.

Which CSS application programming interface (API) should you use? Here are our top-level guidelines:

AP> "top level" is meaningless, unless you have a more detailed set of guidelines to point to.

And, because you are using Q&A style to present the material, the "here are..." is assumed in the format and doesn't need to be stated explicitly.

HOWEVER: I do not recommend using Q&A format in a beginners guide. Turn the subheading into a gerund headline — and note that you do not need to spell out an acronym in a headline, per the style guide.

- If you are writing managed code targeting the MyTools Framework in C#, Basic, J#, managed C++, or any other managed language, you should use [System.Css](#) and/or [HTML to CSS](#) in the MyTools Framework.

AP> Beyond basic style issue of "and/or"—this text fails to give direction to the user. Why choose one over the other?

Each of these choices should have a 2nd-level bullet summarizing what it does and why that solution should be chosen.

Also note: "you should" in 1st bullet and "you will probably want to" in 2nd bullet.

Neither matches style guide nor provides developers new to MyCSS with definitive guidance.

Use the style guide format: either say "we recommend" or be definitive with just "use xyz."

[Deletion strikes a duplicate phrase that 1st appears in the wrong place in the sentence.]

- If you are writing native code using C, C++, or a scripting languages you will probably want to use the [MyCSS](#) library:

AP> Combine the "If you are writing native code..." bullet item from the end of this list here and clarify the choice between MyCSS library and the CSSlite API.

Then introduce the list of MyCSS versions as a separate topic. [Also fix typo in red]

- MyCSS6 is the latest version that's included with MyApps 2009 and all versions of MyApps2010. New applications should be using MyCSS6.

AP> Suggest "Use MyCSS6 for new applications" to avoid anthropomorphism. (Applications don't use libraries—developers do.)

- MyCSS5 is an older library optimized for MyApps 2003 and 2007 and can ~~only~~ be used **only** on **machines** that have a MyApps license.

AP> Check style guide of use of "machine" versus computer. Suggest "...can be used only where a MyApps license is present." But I think this bullet is missing information—this implies that any MyApps license will work, and it doesn't say exactly when the developer should use this library. I believe this statement should be "MyCSS5 is an older library that can be used only where the target computer has a licensed version of MyApps 2003 or MyApps 2007." But check with the program manager to clarify this message and related restrictions.

- MyCSS4 is nearing deprecation. MyCSS6 should be used for new applications

AP> "nearing deprecation"? Be specific, because this topic will remain in the online docset for a long time. State planned year of deprecation or link to its status.

- MyCSS3 is included in MyApps 2000 and higher.

AP> This item doesn't present answers for when this library should be used (or not used), as promised in the introduction. MyCSS4 is to be deprecated but MyCSS3 isn't? What does "and higher" actually mean?

- [Comprehensive list of all MyCSS versions and variants](#)

AP> Put this cross reference in the introductory bullet for the list of MyCSS versions, and then describe this list as a partial summary. ... Also, fix typo.

- If you are writing native code and your application has tight performance or memory constraints, consider the [CSSLite API](#).

AP> As I described above, this item needs to be combined with the initial bullet "If you are writing native code using C..." and provide more substance about choosing between the MyCSS library and the CSSLite API. You probably need to discuss the decision criteria with the program manager.

What CSS Tools Are Available? MyTools offers a core collection:

AP> Each of these should have a descriptor that summarizes in 1 sentence what the tool is used for and where it is used in the development process.

... Which is again a major deficit in this first topic: What are the developer's tasks, in what order are they performed, and how do MyCSS libraries and MyTools match the developer's problem?

Note also: the list in "Tutorials & Articles" shows CSS Tools as a separate topic.

There's no reason for this topic to be here, except as a cross reference, unless this subsection provides a framework that shows how each of these is used in the developer's process for creating a solution.

- [CSS Editor](#)
- [CSS Schema Explorer](#)
- [XCSST Debugger](#)
- [CSS Tools in MyTools overview](#)