

The Pros and Cons of Outsourcing Enterprise Emails. [Click here to read this article.](#)

10 Tips for Writing Consistently Popular Programming Articles

By [Russell Jones](#) on January 27, 2009 12:10 PM | [Permalink](#) | [Comments \(0\)](#) | [TrackBacks \(0\)](#)

For authors, it's a sad fact that when it comes to popularity, not all articles are created equal; some prove to be more popular than others. The truth is that no matter how well you select topics, or how well you write, you won't be able to hit a home run every time. Nonetheless, if you're already writing programming articles (or you're thinking about writing them), the tips listed here are proven to help vault articles from the ho-hum heap to the top of the most-viewed lists.

1. *Choose a Pithy Topic.* Picking a pithy topic is tough, because many interesting topics aren't concise at all; they're long and complex. Sometimes you can create nicely self-contained individual articles that work in concert as a longer topic, but usually not. You may need to write a series or a book instead. Either pick something else or break the topic up into pithy chunks.

The next few tips can help you meet the requirement of tip #1:

2. *Augment Existing Documentation.* Some of the most consistently popular articles are essentially augmented documentation. There's nothing wrong with that, because technical documentation is often produced hurriedly, is incomplete, isn't written by developers, or lacks pertinent examples. In nearly every case, documentation (or the lack thereof) provides a rich vein of topics for authors to mine.
3. *Compare Things.* Another often-popular tack is to write a comparison between two or more popular items. These could be languages, language versions, APIs, databases, operating systems, frameworks, programming methodologies, patterns—in the rapidly-changing developer world, there are innumerable opportunities here. Pick two or more technologies that developers use, and write the article that best helps developers transition or choose them.
4. *Make a List.* You may think it's tired, but the "10 for/about " type articles tend to do well. (If this blog post does well, I'll consider it validation of this point; if it doesn't, maybe I'll reduce this to "Nine Tips...."). Editors and publishers love these articles because readers like them. I suspect readers like them because any article that contains a number of things increases the probability that at least one will contain needed or at least interesting material, but maybe people just like lists. At any rate...

After you have a solid topic, and you're ready to start writing, keep these points in mind:

5. *Ignore History.* Yes, I know you think everyone needs to understand the background history for your particular subject before they get to the meat, but the fact is, they rarely do. Your readers no more want to ponder your historical comments than you wanted your father to explain the history of math when you were learning algebra in high school. Here's an inside tip from reader behavior analysis: Most readers never get past the first page. So, if you don't answer their question or grab their interest immediately, it doesn't matter how on-target the rest of your article is; they won't see

Small

Online E
Innovati
and Rec

This service
protected at
operations.
deployed cl

Our backup
backup and
of critical se
efficient, se
without sign
\$112.50.

▶ Visit the



- it. Link to the history, get to the meat.
6. *Eschew "HelloWorld" Examples.* Just because you've seen a thousand "HelloWorld" code examples in articles and books that you've read doesn't mean they're good! They're not. No one likes them. They're all but completely useless for learning anything about programming. They're not entertaining either. It's perfectly possible to write clear and straightforward examples that both teach and aren't boring.
 7. *Illustrate Your Points.* Developers like code, sure, but you can save them time and effort by including illustrations and screenshots, too. That's because many of them may not run your fascinating sample code, but if they're reading your article, they're probably interested in the results. Show the input or output whenever it matters.
 8. *Show the Interesting Code.* Many technical authors seem to think that providing a brief explanation and following that up with reams of example code (or worse, just showing the code without the explanation) will spur their readers to study the code to gain enlightenment. I assure you that's not true. The best articles explain the topic, show only code fragments, and then explain or show (or both) what that code does, how it fits in with the surrounding code or overall topic, when you should use it, when you should not use it, and only then—and only if it's truly useful—do they refer you to longer chunks of code. Instead, put only the interesting code in your article, and provide the rest as a runnable, complete-project download.
 9. *Make the Complex Simple.* Avoid the urge to tell people how complicated your subject is. They know it's complicated, or they probably wouldn't be reading your article. Instead, think of ways to make your complicated subject seem simpler.

And perhaps the most important point of all is:

10. *Be Brief.* The most consistently popular technical articles give readers just what they need—and no more.

Finally, it's true that some articles are popular despite having few or none of the characteristics listed here—but that's not something you can control. Trying to write one of those articles is much like drawing to an inside straight; you'll lose most of the time. Concentrate on the basics, write a lot of articles, and some of them will be winners.

Happy writing.

Tags: [author](#), [programming](#), [technical article](#)

0 TrackBacks

Listed below are links to blogs that reference this entry: [10 Tips for Writing Consistently Popular Programming Articles](#).

TrackBack URL for this entry: <https://swarm.jupitermedia.com/mt-tb.cgi/6464>

Leave a comment

[Sign in](#) to comment on this entry, or [comment anonymously](#).



Sun

4
11
18
25

Small

Hos
Exc

- Free
- Email
- 24/7 S
- Requ

Offer beg
Eligibility
Exchang

Premier CL
APIfinder
Web Site A
Drivers & S
Menuing Sc
Product Ba
Shop DevX
Technology

About thi

This page c
on January

[JavaFX Tea](#)
previous en

Find recent
to find all c