

Technical Writing for Beginners – An A-Z Guide to Tech Blogging Basics

 freecodecamp.org/news/technical-writing-for-beginners

Technical writing might also be for you if you love learning by teaching others, contributing to open source projects and teaching others how to do so, too, or basically enjoy explaining complex concepts in simple ways through your writing.

Let's dive into the fundamentals and learn about what you should know and consider when getting started with technical writing.

Table of Contents

In this article, we'll be looking at:

What is Technical Writing?

Technical writing is the art of providing detail-oriented instruction to help users understand a specific skill or product.

And a technical writer is someone who writes these instructions, otherwise known as technical documentation or tutorials. This could include user manuals, online support articles, or internal docs for coders/API developers.

A technical writer communicates in a way that presents technical information so that the reader can use that information for an intended purpose.

Benefits of Technical Writing

Technical writers are lifelong learners. Since the job involves communicating complex concepts in simple and straightforward terms, you must be well-versed in the field you're writing about. Or be willing to learn about it.

This is great, because with each new technical document you research and write, you will become an expert on that subject.

Technical writing also gives you a better sense of user empathy. It helps you pay more attention to what the readers or users of a product feel rather than what you think.

You can also make money as a technical writer by contributing to organizations. Here are [some organizations that pay you to write for them](#), like [Smashing Magazine](#), [Auth0](#), [Twilio](#), and [Stack Overflow](#).

In addition to all this, you can contribute to Open Source communities and participate in paid open source programs like [Google Season of Docs](#) and [Outreachy](#).

You can also take up technical writing as a full time profession – lots of companies need someone with those skills.

Necessary Skills to Have as a Technical Writer

Understand the use of proper English

Before you consider writing, it is necessary to have a good grasp of English, its tenses, spellings and basic grammar. Your readers don't want to read an article riddled with incorrect grammar and poor word choices.

Know how to explain things clearly and simply

Knowing how to implement a feature doesn't necessarily mean you can clearly communicate the process to others.

In order to be a good teacher, you have to be empathetic, with the ability to teach or describe terms in ways suitable for your intended audience.

| If you can't explain it to a six year old, you don't understand it yourself. Albert Einstein

Possess some writing skills

I believe that writers are made, not born. And you can only learn how to write by actually writing.

You might never know you have it in you to write until you put pen to paper. And there's only one way to know if you have some writing skills, and that's by writing.

So I encourage you to start writing today. You can choose to start with any of the platforms I listed in [this section](#) to stretch your writing muscles.

And of course, it is also a **huge benefit to have some experience in a technical field.**

The Technical Writing Process

Analyze and Understand who your Readers are

The biggest factor to consider when you're writing a technical article is your intended/expected audience. It should always be at the forefront of your mind.

A good technical writer writes based on the reader's context. **As an example**, let's say you're writing an article targeted at beginners. It is important not to assume that they already know certain concepts.

You can start out your article by outlining any necessary prerequisites. This will make sure that your readers have (or can acquire) the knowledge they need before diving right into your article.

You can also include links to useful resources so your readers can get the information they need with just a click.

In order to know for whom you are writing, you have to gather as much information as possible about who will use the document.

It is important to know if your audience has expertise in the field, if the topic is totally new to them, or if they fall somewhere in between.

Your readers will also have their own expectations and needs. You must determine what the reader is looking for when they begin to read the document and what they'll get out of it.

To understand your reader, ask yourself the following questions before you start writing:

- Who are my readers?
- What do they need?
- Where will they be reading?
- When will they be reading?
- Why will they be reading?
- How will they be reading?

These questions also help you think about your reader's experience while reading your writing, which we'll talk about more now.

Think About User Experience

User experience is just as important in a technical document as it is anywhere on the web.

Now that you know your audience and their needs, keep in mind how the document itself services their needs. It's so easy to ignore how the reader will actually use the document.

As you write, continuously step back and view the document as if you're the reader. Ask yourself: Is it accessible? How will your readers be using it? When will they be using it? Is it easy to navigate?

The goal is to write a document that is both useful to and useable by your readers.

Plan Your Document

Bearing in mind who your users are, you can then conceptualize and plan out your document.

This process includes a number of steps, which we'll go over now.

Conduct thorough research about the topic

While planning out your document, you have to research the topic you're writing about. There are tons of resources only a Google search away for you to consume and get deeper insights from.

Don't be tempted to lift off other people's works or articles and pass it off as your own, as this is plagiarism. Rather, use these resources as references and ideas for your work.

Google as much as possible, get facts and figures from research journals, books or news, and gather as much information as you can about your topic. Then you can start making an outline.

Make an outline

Outlining the content of your document before expanding on it helps you write in a more focused way. It also lets you organize your thoughts and achieving your goals for your writing.

An outline can also help you identify what you want your readers to get out of the document. And finally, it establishes a timeline for completing your writing.

Get relevant graphics/images

Having an outline is very helpful in identifying the various virtual aids (infographics, gifs, videos, tweets) you'll need to embed in different sections of your document.

And it'll make your writing process much easier if you keep these relevant graphics handy.

Write in the Correct Style

Finally, you can start to write! If you've completed all these steps, writing should become a lot easier. But you still need to make sure your writing style is suitable for a technical document.

The writing needs to be accessible, direct, and professional. Flowery or emotional text is not welcome in a technical document. To help you maintain this style, here are some key characteristics you should cultivate.

Use Active Voice

It's a good idea to use active voices in your articles, as it is easier to read and understand than the passive voice.

Active voice means that the **subject** of the sentence is the one actively performing the **action** of the verb. Passive voice means that a **subject** is the recipient of a verb's **action**.

Here's an example of **passive voice**: The documentation should be read six times a year by every web developer.

And here's an example of **active voice**: Every web developer should read this documentation 6 times a year.

Choose Your Words Carefully

Word choice is important. Make sure you use the best word for the context. Avoid overusing pronouns such as 'it' and 'this' as the reader may have difficulty identifying which nouns they refer to.

Also avoid slang and vulgar language – remember you're writing for a wider audience whose disposition and cultural inclinations could differ from yours.

Avoid Excessive Jargon

If you're an expert in your field, it can be easy to use jargon you're familiar with without realizing that it may be confusing to other readers.

You should also avoid using acronyms you haven't previously explained.

Here's an Example:

Less clear: **PWAs** are truly considered the future of multi-platform development. Their availability on both Android and iOS makes them the app of the future.

Improved: **Progressive Web Applications (PWAs)** are truly the future of multi-platform development. Their availability on both Android and iOS makes **PWAs** the app of the future.

Use Plain Language

Use fewer words and write in a way so that any reader can understand the text. Avoid big lengthy words. Always try to explain concepts and terms in the clearest way possible.

Visual Formatting

A wall of text is difficult to read. Even the clearest instructions can be lost in a document that has poor visual representation.

They say a picture is worth a thousand words. This rings true even in technical writing.

But not just any image is worthy of a technical document. Technical information can be difficult to convey in text alone. A well-placed image or diagram can clarify your explanation.

People also love visuals, so it helps to insert them at the right spots. Consider the images below:

First, here's a blog snippet without visuals:

Step 1

So, the first thing to do is to preview your article, by clicking on the preview button right on top of your draft page.

Step 2

Select the heading you'd love to skip to, right click and select inspect (if you're on chrome). This will launch you to the developer tools.

Here's a snippet of same blog, but with visuals:

Step 1

So, the first thing to do is to preview your article, by clicking on the preview button right on top of your draft page.



Step 2

Select the heading you'd love to skip to, right click and select inspect (if you're on chrome). This will launch you to the developer tools.

Adding images to your articles makes the content more relatable and easier to understand. In addition to images, you can also use gifs, emoji, embeds (social media, code) and code snippets where necessary.

Thoughtful formatting, templates, and images or diagrams will also make your text more helpful to your readers. You can check out the references below for a technical writing template from @Bolajiayodeji.

Do a Careful Review

Good writing of any type must be free from spelling and grammatical errors. These errors might seem obvious, but it's not always easy to spot them (especially in lengthy documents).

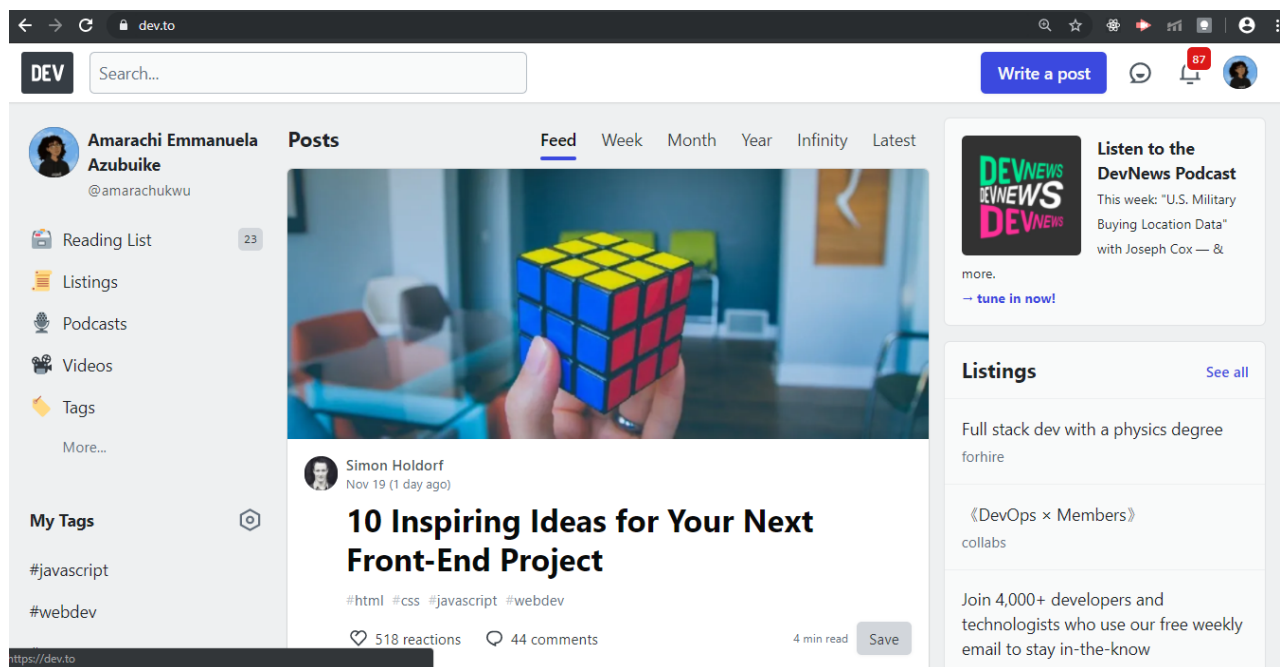
Always double-check your spelling (you know, dot your Is and cross your Ts) before hitting 'publish'.

There are a number of free tools like [Grammarly](#) and the [Hemingway app](#) that you can use to check for grammar and spelling errors. You can also share a draft of your article with someone to proofread before publishing.

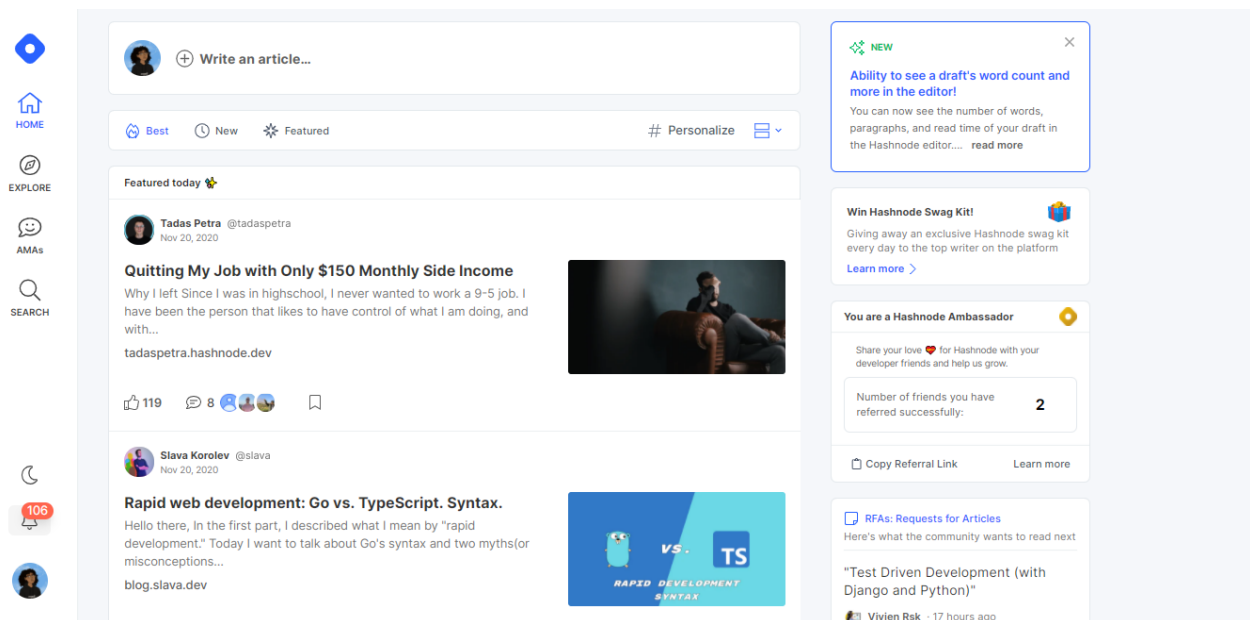
Where to Publish Your Articles

Now that you've decided to take up technical writing, here are some good platforms where you can start putting up technical content for free. They can also help you build an appealing portfolio for future employers to check out.

[**Dev.to**](#) is a community of thousands of techies where both writers and readers get to meaningfully engage and share ideas and resources.



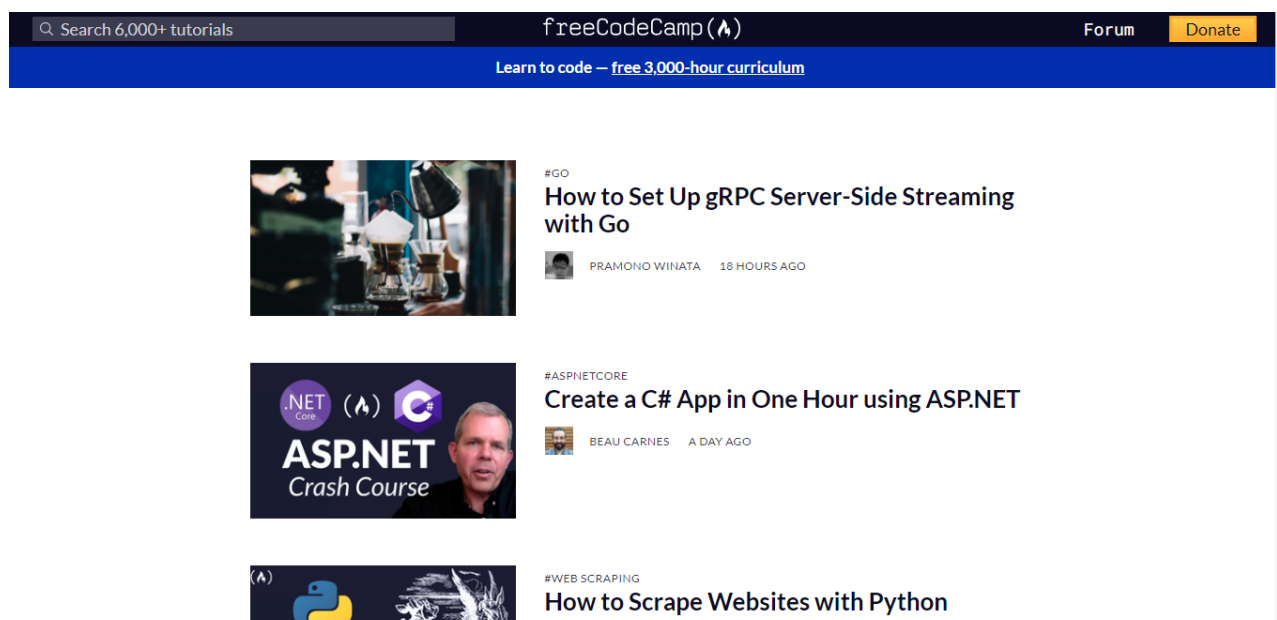
[**Hashnode**](#) is my go-to blogging platform with awesome perks such as custom domain mapping and an interactive community. Setting up a blog on this platform is also easy and fast.



freeCodeCamp has a very large community and audience reach and is a great place to publish your articles. However, you'll need to apply to write for their publication with some previous writing samples.

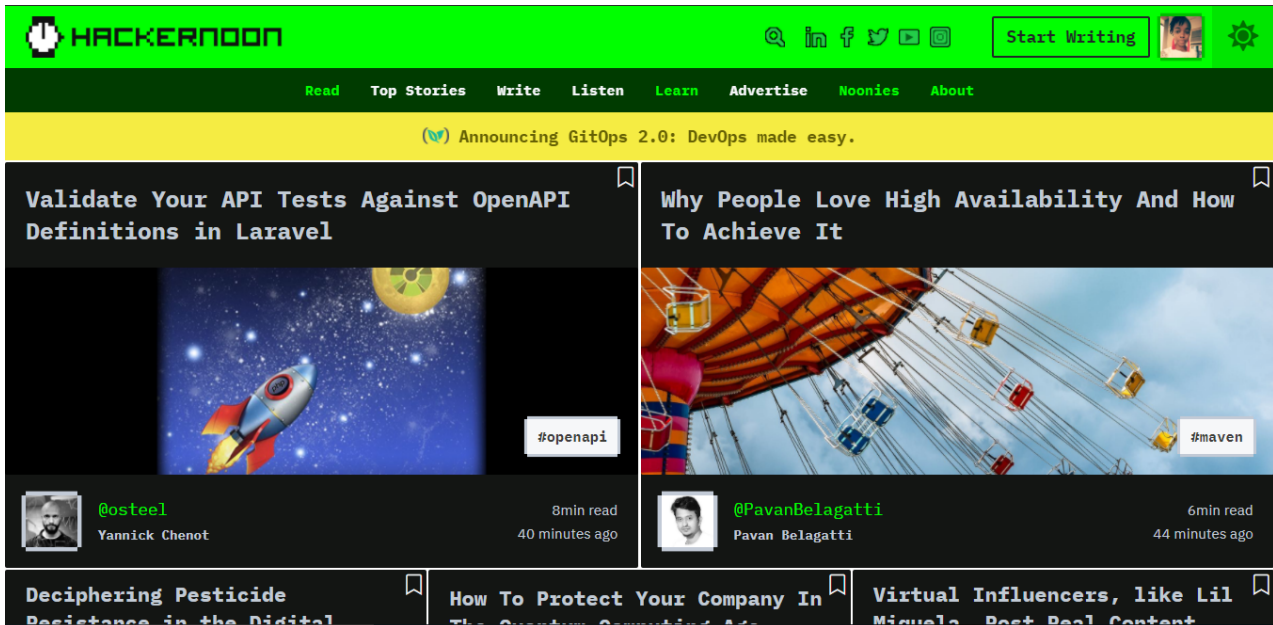
Your application could either be accepted or rejected, but don't be discouraged. You can always reapply later as you get better, and who knows? You could get accepted.

If you do write for them, they'll review and edit your articles before publishing, to make sure you publish the most polished article possible. They'll also share your articles on their social media platforms to help more people read them.



Hackernoon has over 7,000 writers and could be a great platform for you to start publishing your articles to the over 200,000 daily readers in the community.

Hacker Noon supports writers by proofreading their articles before publishing them on the platform, helping them avoid common mistakes.



Technical Writing Courses

Just like in every other field, there are various processes, rules, best practices, and so on in Technical Writing.

Taking a course on technical writing will help guide you through every thing you need to learn and can also give you a major confidence boost to kick start your writing journey.

Here are some technical writing courses you can check out:

- [Google Technical Writing Course](#) (Free)
- [Udemy Technical Writing Course](#) (Paid)
- [Hashnode Technical Writing Bootcamp](#) (Free)

Technical Writing Forums and Communities

Alone we can do so little, together, we can do so much ~ Helen Keller

Being part of a community or forum along with people who share same passion as you is beneficial. You can get feedback, corrections, tips and even learn some style tips from other writers in the community.

Here are some communities and forums for you to join:

- [Hashnode](#)
- [Dev.to](#)
- [Technical Writing World](#)
- [Technical Writer Forum](#)
- [Write the Docs Forum](#)

Some Amazing Technical Writers to follow

In my technical writing journey, I've come and followed some great technical writers whose writing journey, consistency, and style inspire me.

These are the writers whom I look up to and consider virtual mentors on technical writing. Sometimes, they drop technical writing tips that I find helpful and have learned a lot from.

Here are some of those writers (hyperlinked with their twitter handles):

- [Quincy Larson](#)
- [Eddiong Asikpo](#)
- [Catalin Pit](#)
- [Victoria Lo](#)
- [Bolaji Ayodeji](#)
- [Amruta Ranade](#)
- [Chris Bongers](#)
- [Colby Fayock](#)

Final words

You do not need a degree in technical writing to start putting out technical content. You can start writing on your personal blog and public GitHub repositories while building your portfolio and gaining practical experience.

Really – Just Start Writing.

Practice by creating new documents for existing programs or projects. There are a number of open source projects on GitHub that you can check out and add to their documentation.

Is there an app that you love to use, but its documentation is poorly written? Write your own and share it online for feedback. You can also quickly set up your blog on [hashnode](#) and start writing.

You learn to write by writing, and by reading and thinking about how writers have created their characters and invented their stories. If you are not a reader, don't even think about being a writer. - Jean M. Auel

Technical writers are always learning. By diving into new subject areas and receiving external feedback, a good writer never stops honing their craft.

Of course, good writers are also voracious readers. By reviewing highly-read or highly-used documents, your own writing will definitely improve.

Can't wait to see your technical articles!

References

[Introduction to Technical Writing](#)

How to structure a technical article

Understanding your audience, the why and how

Technical Writing template

I hope this was helpful. If so, follow me on Twitter and let me know!
