UX Writing – A Quick Start Guide

By Mitch Krpata

mkrpata@gmail.com
www.writemitchwrite.com

Contents

Introduction

UX writing is the art of writing for digital channels, most often for the web. Effective UX writing helps to achieve business objectives by anticipating a user's needs and speaking to them in a concise and relevant way. It helps customers accomplish their goals on your site and leaves them feeling satisfied, even delighted, with the experience.

The purpose of this short book is to provide a handy overview of best practices in UX writing. As the name implies, it is not an in-depth, expert-level guide. (UX writing is, itself, only a portion of the content strategist's purview.) Entire books could be – and probably have been – written about each section that follows.

My hope is that *UX Writing – A Quick Start Guide* helps newer people in the field to get a handle on the fundamentals. It may also be useful for product owners and designers who are, tragically, bereft of the services of a full-time content strategist. While much of what I've written here pertains specifically to writing for the web, the principles are applicable to other digital media as well.

For the most part, the same rules of good writing in general apply to UX writing in particular. Favor plain language, brevity, and the active voice. Use inclusive language. Sound human.

Voice and tone should fit with a company's brand guidelines, but remember that there are places to turn the volume up or down depending on the user's goals. For example, a checkout flow is not a place to get cute or make jokes, but a purchase confirmation page can be more fun.

Great UX writing values function over form. It's an integral part of the business relationship with a current or prospective customer. When done well, most of the time users won't even notice it. When done poorly, everyone will notice – including corporate leadership.

Above all, remember that your goal in UX writing is to help users accomplish *their* goals, and every decision you make should be in service of that.

Using the right tool for the job

When writing for the web, how you format content is just as important as the words you choose. Keep in mind general web design principles and be sure that you're using the right modality for your content. Understanding common affordances and best practices is critical. You do not need to be a visual designer or an engineer, but you should understand how visual design and engineering can support or interfere with a user's goals.

Hyperlinks

When to use them: Hyperlinks should be used for navigation, defined as when the user will load a new page (more common) or jump to a different spot on the current page (less common).

Hyperlink text should be specific enough about where the user will land that it makes sense even without reading the context. Generic phrases like "learn more" and "click here" should not be used as hypertext.

> **Bad:** Click here to read more about Labrador Retrievers.
> **Good:** Discover the common traits of Labrador Retrievers.

Because hyperlinks are used within copy, they can be reasonably long. Several words are fine. But for readability's sake, try not to extend them past one sentence. Users like to scan for hyperlinked phrases.

Links should look like links. They should be styled with an underline and a font color that contrasts with the rest of the text. A different color for visited links remains extremely helpful.

Buttons

When to use them: Buttons should be used for any user-initiated action that changes the state of the page in some way, such on the front end, the back end, or both. An example would be adding an item to the user's cart.

As with links, it's best to be very specific about what clicking the button will do.

> **Bad (but sometimes unavoidable):** Submit
> **Good:** Add to cart

In general, it's good to keep button copy short, because very long buttons can look off style. Try to balance brevity with clarity.

Buttons should look like buttons and they should have a bounding box. There's nothing wrong with a little shading or depth, as long as the copy remains easily readable.

Calls to action (CTAs)

When to use them: Ideally there should not be more than one per page. The CTA should reflect the purpose of the page.

People tend to think of anything clickable or interactive as a CTA, but that's not really true. Nav elements are not CTAs, for instance. A CTA is something that offers value to the user in exchange for clicking. Strong CTAs are clear and specific, and they promise a return for the user's input.

For instance, imagine that we have asked for a user's contact information to sign them up for a fitness newsletter. We want to be specific about what clicking the CTA will do, and if possible we want to promise the value to the user.

> **Bad:** Submit
> **Better:** Sign up
> **Good:** Get free weekly fitness tips in your inbox

A good way to determine the CTA is to imagine that you are the user and complete this sentence: "I want to _____."

How you fill in the blank is your CTA.

Lists

When to use them: Whenever you are writing out three or more of the same kinds of things, a list tends to be preferable to a sentence. It is far more scannable.

Use an unordered (bulleted) list when the order doesn't matter.

> **OK:**
> Most owners of Labrador Retrievers describe their dogs as friendly, playful, good with kids, obedient, and eager to please.

> **Better:**
> Most owners of Labrador Retrievers describe their dogs as:
> - Friendly
> - Playful
> - Good with kids
> - Obedient
> - Eager to please

(However, even if the order doesn't matter you may still find it advisable to alphabetize or otherwise arrange the items in a logical way.)

Use an ordered (numbered) list when the order does matter; for instance, when the user must follow step-by-step instructions.

Bad:

1. Mail the test sample to our lab
2. Gather a test sample with the included swab
3. Open the test kit

Good:

1. Open the test kit
2. Gather a test sample with the included swab
3. Mail the test sample to our lab

When list items are short or incomplete sentences, you can do without terminal punctuation. But do use it when they are complete or multiple sentences. And every item in the list should be consistent in this regard. If it's unclear whether you should include or omit punctuation, then consider rewriting one or more of the items.

It is best to avoid multiple sentences within a single list item.

Both ordered and unordered lists should be styled correctly with HTML. Some web designers like to make fancy number graphics to use with ordered lists, but that destroys their utility. As with just about everything, use the built-in functionality that web browsers give you for free.

Forms

When to use them: Any time you need to collect information from the user in order for them to accomplish their goals.

When not to use them: Any time you don't need to collect information from the user in order for them to accomplish their goals – even if you want that information really, really bad.

The first rule of form design is to ask for no more than you truly need. There may be more information you would like to have from the user, certainly, but this is a case where you need to prioritize the user's desires over your own. Every field on your form should have a purpose.

Ask questions like:

- Have we earned the user's trust to give us this information yet?
- Are we offering the user enough value in exchange for this information?
- Do we need this information now, or is there a chance to ask for it later?
- Will asking for this much information lower conversion?

As an example, say you want users to sign up for a free weekly newsletter about fitness with the aim of converting them to paying customers for a wearable fitness tracker. You'd love to know things like their current diet and exercise habits, their health goals, maybe some demographic info, any number of things. But what do you *need*? Just an email address.

Eventually, you'll need to ask for more information in order to sell products, let users create accounts, and so on, and you'll do that when the time is right. Always be ruthless in asking for only the minimum that you need.

If it's axiomatic that users come to a website not to look at it but to use it, then it's doubly true for a form. No user desires or values a sexy-looking form. They are not delighted by floating labels or little animations. Forms should just work.

In practice, this means:

> **Good labelling.** Labels are best positioned atop their fields and left-aligned. They should be static, meaning that they are visible to the user whether or not the user has typed anything or even focused on the field. Also, field labels must be specific and unambiguous.

> **Vertical orientation.** Forms are easiest to use when they can be scanned from top to bottom. There are some exceptions when related values may be more helpfully positioned in line. First and last name could go side by side, for example. (Although it's even better if you can choose not to require that kind of formatting!)

> **Logical grouping.** You may be asking for different types of information in a form. Presuming it's all stuff that you absolutely need to know, then it's helpful to the user if similar questions are visibly grouped. A common example of this is shipping information appearing separately from billing information. (You may consider paginating a long form that asks for many types of information.)

> **Optional values clearly marked.** Two schools of thought here. One believes in marking required fields, while the other believes in marking optional fields. I'm part of the latter, and here's why: An ideally designed form wouldn't ask for *any* optional information. This is a fight you will never win, but it's important to keep in mind. The default assumption should be that everything in a form is required unless marked optional.

> **Helper text when needed.** Using helper text can be a crutch. For instance, if your back end will only accept a phone number that is formatted as (123) 456-7890, then a lot of people will want you to write helper text saying so. And this is certainly preferable to throwing an error, but not nearly as preferable as just accepting the phone number – no matter how it's formatted. With many forms, however, helper text comes in handy when you're asking for something unusual. If asking for an employee ID number, for instance, you could indicate where to find it or how many characters it is.

Congruent field sizes. You have a pretty good idea of what to expect for most values that you're asking for. The container for each value should correspond to the character count you expect. This gives users a heuristic for understanding what you're asking for beyond the label. For instance, a ZIP code is five digits long. Therefore the field for the user's ZIP code should not be as long as the one asking for their full name.

Auto-complete functionality that correctly auto-completes. Yes, this is an engineering matter more than a content design matter, but it's one in which you may find yourself needing to be a pain in the butt.

Helpful error messaging, and a thoughtful design that prevents common errors altogether. I'll talk more about good error handling later on.

All controls are not created equal. Know when to use radio buttons, when to use checkboxes, and when to use a dropdown. There will be cases where the answer is not obvious, though not as many as you'd think.

A well-designed form is a subtle work of art that few will appreciate. Be one of those few.

Tables

The good old HTML table seems to be a lost art these days, but it's still an excellent way to convey certain information – especially when users want to compare quantities of similar items.

Tables are far better than images for displaying data for reasons too numerous to mention, yet you still see such images everywhere. Because tables are coded in HTML, they'll work across all browsers, devices, and assistive technology with no information loss.

They also help to reduce repetitive copy. For instance, when a company is selling a product that comes in different counts, you might see a paragraph that reads like this:

1 kit - $99
3-5 kits - $89/kit (you save $10 per kit)
6+ kits - $79/kit (you save $20 per kit)

A table simplifies this information and makes it far easier to compare.

Quantity	Price/kit	Your savings/kit
1	$99	--
3-5	$89	$10
6+	$79	$20

The benefit is obvious. Use tables when the content calls for it.

Error handling

Errors will happen. But before you jump straight into writing an error message for a particular scenario, take a step back. Could the page be designed in such a way that this error could be prevented altogether?

Imagine that your form will only accept a properly formatted phone number and you are asked to write an error message that will display when users input the wrong format. Before delivering the copy and moving on with your day, try some other solutions, like:

- Providing an example of the accepted format as helper text (better than nothing)
- Auto-formatting the number as the user types it (helpful)
- Accepting 10 digits in literally any format (genius!)

The last suggestion is by far the best one. Your system can easily parse a phone number from any array of spaces, dashes, and parentheses, so long as there are 10 digits. This might mean slightly more work for a web developer, but it will make your users' lives far easier, and that's work worth doing.

Again, errors in general are unavoidable. Most specific errors are not predestined.

User errors

The bad thing about user error is that users will still get mad at you when it happens. The good thing is that user error is easily resolved. Your aim is to identify the issue and steer the user toward a resolution with a minimum of fuss.

Do:

- Write a message that is short and to the point
- Indicate exactly how to resolve the problem
- Use plain language

Don't:

- Apologize for the user's error
- Use cutesy language like "Oops!"
- Use technical terms

As an example, if you're asking for somebody's birth year and they enter the year 2053 (a valid value!), all you need to say is "Your birth year cannot be in the future. Please enter a year from the past."

Depending on your brand voice, there might be room for something like "Unless you're a time traveler..."

System errors

System errors can be tricky because they're unpredictable and the way to resolve them isn't always apparent. Sometimes it's a temporary glitch and simply reloading the page will do. Sometimes there's an unexpected outage and there's nothing to do but wait.

Unfortunately, these types of errors are prone to happen at the worst times, such as when posting data from a form. Nothing gets a user's goat quite like spending time and effort on something that gets vaporized due to a server error. And if it gets the user's goat, it should get your goat too.

These types of error messages are hard to generalize, but it's also hard to come up with a tailored message to every possible situation. A few guidelines:

Do:
- Explain the nature of the error as clearly as you can
- Give the user something to go on, such as a customer service contact
- Apologize if a cost has been incurred, such as lost data
- If you can, assure the user of bad outcomes that have not happened

Don't:
- Spit out some weird error code with no context
- Be cute or funny
- Dead-end the user on a page with no relevant links

In a situation where a user tries to make a purchase and the transaction fails, you might say something like: "Your purchase was not completed due to a system error. Your card has not been charged. If you need help, contact customer support at..."

Form errors

Errors are very common in forms. Since users already don't want to be completing a form, and since multiple errors are possible, make it easy for users to identify and resolve them. (It is preferable, but not always possible, to prevent errors with smart design and in-line validation.)

When the user submits a form with errors, first reload the page with all of the user's inputs retained. Show the user the top of the page (the beginning of the form) and not the bottom (the end of the form).

At the top of the form, display a general error message with a list of jump links to each specific problematic field. The text of each jump link should match the text of its corresponding error message, and each error message should be displayed in line with its field.

It's not always possible to write awesome in-line error messages but avoid simply repeating the field label if you can.

An error message of this type might look like this:

Fix 3 errors before continuing
Your email address must include a valid domain (e.g., "@gmail.com")
Your date of birth is required
Select a shipping method

When all else fails

Sometimes it can't be helped. You need a catch-all error message that is vague enough to cover any unforeseen circumstances but specific enough that users don't feel abandoned. In this case, you'll want to use a combination of the points above.

- Be as specific as you can about the nature of the problem
- Provide some information about when and how it will be resolved
- Direct the user to customer service or a help page

"We're sorry, but we're experiencing an unexpected system error. Our team will have things up and running as soon as possible. If you need help sooner, contact customer support at..."

Accessibility

Building an accessible website is the right thing to do for users. An accessible website is one on which all users have a comparable experience regardless of their modes of input and output. It makes allowances for assistive devices, sensory deficits, and cognitive disabilities. To focus on what makes a site accessible, remember the acronym **POUR**. Accessible websites are:

- Perceivable
- Operable
- Understandable
- Robust

The minimum goal for most companies is to meet AA-level compliance as defined by WCAG 2.2 accessibility standards.

Alt tags and figure captions

When to use them: Most images need an alt tag. This is how screen reader users will understand what the image represents.

A good alt tag describes the image as specifically as possible. It should not editorialize, upsell, or stuff keywords. The aim is to provide a comparable experience for non-sighted users.

> **Bad:** Doggone Good Brand DNA Tests for Dogs are the best DNA Tests for Dogs and you should buy one.
> **Good:** A dog owner swabs the cheek of a mixed breed dog to collect a saliva sample for DNA testing.

Some images can be said to be purely decorative, in which case they do not require an alt tag and should be set to invisible for screen readers. For the most part, though, if the image is

important enough to include on the page, then it's important enough for all users to "see." Otherwise, why is it there?

A figure caption is not as important as an alt tag. Most images don't need one. But it can be very helpful for users to understand or interpret the image. Unlike alt text, which is invisible unless users are looking for it, all users will see a figure caption as it will usually be displayed directly under the image. Use a figure caption when an image might need further explanation in order to be of value to the user.

Bad: A world map showing a red dot in Central Asia with arrows pointing from it to Europe, the Middle East, and Europe, and a thinner arrow pointing to Australia.

Good: Dogs in the D haplogroup are thought to have originated in Central Asia before spreading mainly to Europe, the Middle East, and Africa – although some made their way to Australia!

(In the examples above, note that the "bad" caption would actually be a fine alt tag. The distinction is that one describes the image while the other interprets it.)

Screen reader-specific guidance

The good news is that if you've done your job well to this point, then you're at least 90 percent of the way toward providing a good experience for screen reader users. Clear, readable copy written in plain language and using proper semantic markup will ensure that all users can accomplish their goals on your site. You do, however, need to keep a few things in mind to optimize the experience for the screen reader user.

Screen reader software struggles to interpret certain abbreviations and uses of punctuation that sighted users take for granted. The problem is that each of the different tools on the market may give a different output depending on how these things are written. That's contrary to our goal, which is a consistent experience for all users. This is not an exhaustive list, but some common issues and their solutions are:

Hashtags. When writing hashtags, capitalize the first letter of each word in your hashtag for accessibility. This can help screen readers decipher the hashtag as multiple words and also benefits anyone who may not be able to easily identify the patterns and relationships between words.

Day and date abbreviations. It's best to write out the full names of days and months if character limits allow. If you must use abbreviations, then consider including screen reader-specific copy that reads the words in their entirety.

Slash marks. For some reason, screen readers can screw up when two words are connected by a slash mark. In digital, include a space before and after the punctuation, e.g. "It's not an either / or situation." Yes, even if Grammarly underlines it in red.

Screen reader-only text. There is some common visual shorthand on web pages that sighted users may take for granted, such as using the letter "X" as a clickable spot to close a modal. For the non-sighted user, we need to make sure we include screen reader-specific copy with this element that says something like "Close modal" or "Cancel."

Writing for readers with dyslexia and attention-deficit disorders

To make life easier for readers with dyslexia:

- Use short sentences
- Left-justify text in paragraphs
- Avoid writing in all caps or all lowercase

To make life easier for readers with an attention-deficit disorder:

- Ensure sufficient padding and white space
- Create a consistent nav between pages and sections within a page
- Do not use text or images that move, scroll, or automatically refresh

For all users, make blocks of text more readable by using line breaks, section headers, and lists.

Miscellaneous

Although it's helpful to think in terms of general concepts and best practices, no UX writing challenge is exactly like another. Your professional judgment is often the difference between good copy and great copy – and sometimes it's all that stands between your users and a crummy experience.

Building consensus with stakeholders is always the preferred way of doing things. Occasionally, though, you will need to dig in. In this section I'll cover some topics that have proven surprisingly contentious in my career.

How not to sound like a robot

For reasons of character count, UX writers are often tempted to omit parts of speech that seem superfluous – especially connective words like articles and possessive adjectives ("a," "the," "your," etc.). Resist the temptation. Phrases without these small but important words invariably sound like the utterances of a 1950s sci-fi robot.

Which of these sounds more like a human wrote it?

    A. Enter email address
    B. Enter an email address

That's right, it's B. These three extra characters more than earn their place. If character count remains an issue, consider different wording altogether. (Simply saying "Email address" would work fine in a form.)

Also keep your guard up for using jargon when a plain word will do. Don't ask people to "input," "utilize," and so on. Try to use the same words the user would: "enter" or "use," for instance.

Which of these sounds most like a human wrote it?

    A. Input valid email address
    B. Enter an email address
    C. `Destroy all humans beep bop boop`

Yep, B again.

The best way to tell if your copy sounds human or robotic is to read it out loud. Go ahead. No one will judge you.

Left-justified text

Avoid center-justified text where possible. It's harder to read, especially for users with dyslexia. It is okay to center a single line of text, such as a section header, and it may be okay to center-justify two lines of text, but try to balance the character count between them. Anything more than that should be left-justified.

In general, when paragraphs of text are the main content of the page, they should be left-justified for readability.

When you find yourself discussing whether center- or left-justified text "looks" better, remember that in all situations its purpose is not to be looked at but to be understood.

Casing

Sentence case is the easiest to read. Therefore, use it for everything except proper nouns. That means using sentence case for nav elements, page headers, CTAs (both links and buttons), and almost everything else.

Title case should only be used for proper nouns, which may include things like the company name, specific products, or certain trademarks.

Use lowercase only for URLs. (E.g., "Visit www.writemitchwrite.com to read my latest blog posts!")

All caps should be used sparingly. I said SPARINGLY. It doesn't look half as cool as your agency thinks it does.

Inclusive language

Put people first. Address them the way they ask to be addressed. Yes, it means using a person's preferred pronouns. It means capitalizing "Black" but not "white." It means not arrogating to yourself the power to dictate other people's identities.

When writing about persons or groups, do not define them by a single characteristic. Sometimes this means using more characters. For example:

> **Bad:** Deaf people
> **Good:** People with partial or complete deafness

Extra character count is a small price to pay for treating people respectfully.

Be wary of using idiomatic language that implies a certain way of doing something. For example, you may tell customers to "walk right in," but not all of your prospective customers may be ambulatory. This doesn't mean you should never say something like "walk-in hours" if

you believe that's the most commonly understood way to tell people that they don't need an appointment. Just be intentional about your word choice.

Watch out too for assuming cognitive or physical ability. It's common to see descriptions like "quick" or "easy," but you simply can't be sure that your users will find something to be quick or easy. A better option is to be specific about the content in a way that will apply to every user.

> **Bad:** Take a short survey about your viewing habits!
> **Good:** Answer 5 questions about your viewing habits!

Further reading

These are all good books about writing and designing for the web. Some are available for free.

[Form Design Patterns: A practical guide to designing and coding simple and inclusive forms for the web](#) by Adam Silver

Forms are unsexy, sure. They're also critical. You have a choice: you can get caught up in surface-level flashiness or you can become an expert in what really matters to users.

[Microcopy: The Complete Guide](#) by Kinneret Yifrah

Practical, specific advice for handling common – and not-so-common – microcopy placements.

[Resilient Web Design](#) by Jeremy Keith

Web browsers are more powerful than we give them credit for. Understanding the fundamentals of HTML and semantic markup will improve your content's form and function.

[Technically Wrong: Sexist Apps, Biased Algorithms, and Other Threats of Toxic Tech](#) by Sara Wachter-Boettcher

Don't want your company to become today's main character on Twitter? Read this book.

[Writing Is Designing: Words and the User Experience](#) by Michael J. Metts & Andy Welfle

Your job is about much more than putting words into boxes. Understand and embrace that good design begins with you and your ability to define what users want and how your company will deliver it to them.

When you're stuck, revisit the [Inclusive Design Principles](#) to remember what matters most.

About me

My name is Mitch Krpata. I'm a content strategist who's built better digital experiences for customers in retail, health care, gaming, and professional channels. I believe that putting people first and focusing relentlessly on solving their problems is the best way for a business to grow and thrive.

Outside of work, I love reading, gaming, and watching weird movies. I lift weights in case an emergency one day arises in which someone needs 275 pounds picked up exactly once. My DuoLingo streak is over a year long (Russian, which was less problematic when I started). And I should probably say something about my wonderful family, which includes my wife, two children, and a dog.

I love hearing from kind people. There are, frankly, too many communication methods available to you. Some of them are:

       Email: mkrpata@gmail.com

       LinkedIn: https://www.linkedin.com/in/mitch-krpata-20aa37/

       On the web: https://www.writemitchwrite.com/

Or by signal flare. No promises, though.