Pre-Class material
Introduction (3 min)
Just enough grammar (optional) (10 min)
Words (10 min)
Active voice (15 min)
Clear sentences (10 min)
Short sentences (20 min)
Lists and tables (15 min)
Paragraphs (10 min)
Audience (10 min)
Documents (10 min)
Punctuation (optional) (5 min)
Markdown (variable)
Summary (1 min)

Punctuation (optional)

On this page

Commas
Exercise
Semicolons
Exercise
Em-Dashes
En-dashes and hyphens
Parentheses

Estimated Time: 5 minutes

This optional unit provides a quick refresher on punctuation marks.

Commas

Programming languages enforce clear rules about punctuation. In English, by contrast, the rules regarding commas are somewhat hazier. As a guideline, insert a comma wherever a reader would naturally pause somewhere within a sentence. For the musically inclined, if a period is a whole note rest, then a comma is perhaps a half-note or quarter-note rest. In other words, the pause for a comma is shorter than that for a period. For example, if you read the following sentence aloud, you probably rest briefly before the word *just*:

> C behaves as a mid-level language, just a couple of steps up in abstraction from assembly language.

Some situations *require* a comma. For example, use commas to separate items in an embedded list like the following:

> Our company uses C++, Python, Java, and JavaScript.

You might be wondering about a list's final comma, the one inserted between items N-1 and N. This comma—known as the **serial comma** or **Oxford comma**—is controversial. We recommend supplying that final comma simply because technical writing requires picking the least ambiguous solution. That said, we actually prefer circumventing the controversy by converting embedded lists into bulleted lists.

In sentences that express a condition, place a comma between the condition and the consequence. For example, both of the following sentences supply the comma in the correct place:

> If the program runs slowly, try the `--perf` flag.
>
> If the program runs slowly, then try the `--perf` flag.

You can also wedge a quick definition or digression between a pair of commas as in the following example:

> Python, an easy-to-use language, has gained significant momentum in recent years.

Finally, avoid using a comma to paste together two independent thoughts. For example, the comma in the following sentence is guilty of a punctuation felony called a **comma splice**:

Not recommended

> Samantha is a wonderful coder, she writes abundant tests.

Use a period rather than a comma to separate two independent thoughts. For example:

Recommended

> Samantha is a wonderful coder. She writes abundant tests.

Exercise

Add commas where appropriate to the following passage:

> Protocol Buffers sometimes known as protobufs are our team's main structured data format. Use Protocol Buffers to represent store and transfer structured data. Unlike XML Protocol Buffers are compiled. Consequently clients transmit Protocol Buffers efficiently which has led to rapid adoption.

Hint: Read the passage aloud and put a comma everywhere you hear a short pause.

Click the icon to see the answer.

Here is one possible solution:

> Protocol Buffers, sometimes known as protobufs, are our team's main structured data format. Use Protocol Buffers to represent, store, and transfer structured data. Unlike XML, Protocol Buffers are compiled. Consequently, clients transmit Protocol Buffers efficiently, which has led to rapid adoption.

Semicolons

A period separates distinct thoughts; a semicolon unites highly related thoughts. For example, notice how the semicolon in the following sentence unites the first and second thoughts:

Recommended

> Rerun Frambus after updating your configuration file; don't rerun Frambus after updating existing source code.

Before using a semicolon, ask yourself whether the sentence would still make sense if you flipped the thoughts to opposite sides of the semicolon. For example, reversing the earlier example still yields a *valid* sentence:

> Don't rerun Frambus after updating existing source code; rerun Frambus after updating your configuration file.

The thoughts preceding and following the semicolon must each be grammatically complete sentences. For example, the following semicolon is *incorrect* because the passage following the semicolon is a clause, not a complete sentence:

Not recommended

> Rerun Frambus after updating your configuration file; not after updating existing source code.

Recommended

> Rerun Frambus after updating your configuration file, not after updating existing source code.

You should almost always use commas, not semicolons, to separate items in an embedded list. For example, the following use of semicolons is *incorrect*:

Not recommended

> Style guides are bigger than the moon; more essential than oxygen; and completely inscrutable.

As mentioned earlier in this lesson, technical writing usually prefers bulleted lists to embedded lists. However, if you truly prefer an embedded list, use commas rather than semicolons to separate the items, as in the following example:

Recommended

> Style guides are bigger than the moon, more essential than oxygen, and completely inscrutable.

Many sentences place a transition word or phrase immediately after the semicolon. In this situation, place a comma after the transition. Note the comma after the transition in the following two examples:

> Frambus provides no official open source package for string manipulation; however, subsets of string manipulation packages are available from other open source projects.

> Even seemingly trivial code changes can cause bugs; therefore, write abundant unit tests.

Exercise

Which of the following periods or commas could you replace with a semicolon?

1. Python is a popular programming language. The C language was developed long before Python.
2. Model learning for a low value of X appears in the top illustration. Model learning for a high value of X appears in the bottom illustration.
3. I'm thankful for my large monitor, powerful CPU, and blazing bandwidth.

Click the icon to see the answer.

1. You may not convert the period in #1 to a semicolon because the two sentences are only vaguely related.
2. You may replace the period in #2 with a semicolon because the two sentences are so highly related.
3. You may not convert the commas in #3 to semicolons. Use commas to separate items in an embedded list.

Em-Dashes

Em-dashes are compelling punctuation marks, rich with punctuation possibilities. An em-dash represents a longer pause—a bigger break—than a comma. If a comma is a quarter note rest, then an em-dash is a half-note rest. For example:

> C++ is a rich language—one requiring extensive experience to master.

Writers sometimes use a pair of em-dashes to block off a digression, as in the following example:

> **Protocol Buffers**—often nicknamed **protobufs**—encode structured data in an efficient yet extensible format.

Could we have used commas instead of em-dashes in the preceding examples? Sure. Why did we choose an em-dash instead of a comma? Feel. Art. Experience.

En-dashes and hyphens

Consider the horizontal punctuation marks shown in the following table:

Name	Mark	Relative width
em-dash	—	widest (usually, the length of the letter `m`)
en-dash	–	medium (usually, the length of the letter `n`)
hyphen	-	narrowest

Some style guides recommend the **en-dash** for certain uses. The Google Style Guide, however, offers the following blunt advice about en-dashes:

> Don't use.

Hyphens are tricky. Within technical writing, hyphens connect words in certain compound terms, such as:

- self-attention
- N-gram

Confusingly, three-word compound terms often contain a hyphen between the first and second word but not between the second and third word. For example:

- decision-making system
- floating-point feature

When in doubt about hyphens, consult a dictionary, glossary, or style guide.

Note: If you consult *more than one* dictionary, glossary, or style guide about hyphens, you may encounter inconsistencies.

Parentheses

Use parentheses to hold minor points and digressions. Parentheses inform readers that the enclosed text isn't critical. Because the enclosed text isn't critical, some editors feel that text that deserves parentheses doesn't deserve to be in the document. As a compromise, keep parentheses to a minimum in technical writing.

The rules regarding periods and parentheses aren't always clear. Here are the standard rules:

- If a pair of parentheses holds an entire sentence, the period goes inside the closing parenthesis.
- If a pair of parentheses ends a sentence but does not hold the entire sentence, the period goes just outside the closing parenthesis.

For example:

> (Incidentally, Protocol Buffers make great birthday gifts.)
>
> Binary mode relies on the more compact native form (described later in this document).

Next unit: Markdown

Was this helpful?

Send feedback

Based on this session

Words

Educational resources about technical writing.

Updated Nov 8, 2021

Paragraphs

Educational resources about technical writing.

Updated Feb 20, 2021

Markdown (optional)

Educational resources about technical writing.

Updated Nov 4, 2021

Except as otherwise noted, the content of this page is licensed under the Creative Commons Attribution 4.0 License, and code samples are licensed under the Apache 2.0 License. For details, see the Google Developers Site Policies. Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2021-11-08 UTC.