# 0.1 — Introduction to these tutorials

👤 **ALEX**    🕔 **JULY 19, 2022**

### Welcome!

Welcome to the Learn C++ tutorials! Above all else, these tutorials aim to make learning C++ easy.

Unlike many other sites and books, these tutorials don't assume you have any prior programming experience. We'll teach you everything you need to know as you progress, with lots of examples along the way.

Whether you're interested in learning C++ as a hobby or for professional development, you're in the right place!

A testimonial from reader Syam from Trinidad:

> It's crazy how perfect this website explains C++ concepts and teaches programming. I read the book "Programming: Principles and Practice Using C++" [by Bjarne Stroustrup, the creator of C++] and it was rather confusing…. This website makes it clear the first time, and they explain everything you need to know. Somehow they know what I don't know, and what I need to know. It's almost like my future self went back in the past to teach me what mistakes I will make.

## Lesson structure

The lessons in this introductory chapter are aimed at giving you some context around what C++ is, how it came about, how programs work, and what software you need to install to create your own programs. You'll even write your own first program.

Further chapters will explore different parts of the C++ language. In the first chapter (chapter 1), you'll get a broad but shallow overview of many fundamental C++ concepts, so we can start writing some simple programs. Further chapters will explore those concepts in depth, or introduce new concepts.

Each chapter has a theme, with all of the sections underneath it being generally related to that theme. There is no suggested amount of time that you should spend with each lesson or chapter; progress through the material at a pace that is comfortable for you.

## Goals

Before we get started, let's cover a few important goals for these tutorials:

- Cover programming topics as well as C++. Traditional textbooks do a pretty good job of teaching the basics of a given programming language, but they often do not cover relevant programming topics that are incidental to the language. For example, books will omit sections on programming style, common pitfalls, debugging, good/bad programming practices, and testing. Consequently, by the time you finish the book, you may understand how to program in a language, but you might also have picked up bad habits that will come back to bite you later! One of the goals of these tutorials is to make sure that all of these incidental topics are covered along the way, in the sections where it naturally makes sense to discuss them. When you finish, you will not only know how to program in C++, you will know how NOT to program in C++, which is arguably as important.

- Provide a lot of examples. Most people learn as much or more from following the examples as they do from reading the text. These tutorials will endeavor to provide plenty of clear, concise examples to show the application of concepts you are learning. We will also avoid (as much as possible) the twin evils: the ellipses (also known as …), where a necessary or non-obvious part of an example is omitted in the interest of space, and the unexplained new concept, where a new concept that is integral to the example is introduced without any mention of what it is or how it works. Both of these tend to lead to getting stuck.

- Provide practice programs. The end of many lessons and sections will contain some exercises that you can attempt to answer on your own, along with solutions. You can compare your solution against ours to see what we did differently, or, if you get stuck, how we solved the problem. Then you can go back and refocus on the areas you need more work on.

- Most importantly: have fun. Programming can be a lot of fun, and if you're not generally having fun, you're not in the right mindset to be programming. Tired or unhappy programmers make mistakes, and debugging code tends to take much longer than writing it correctly in the first place! Often you can save yourself some time by going to bed, getting a good night's sleep, and coming back to a problem in the morning.

## Getting the most out of these tutorials

As you go through these tutorials, we recommend a number of practices to maximize your learning experience:

- Type in the examples by hand and compile them yourself. Do not copy and paste them! This will help you learn where you commonly make errors, as well as becoming familiar with compiler warnings and errors. Don't just transcribe the programs mindlessly -- think about what each of the lines you are typing in does, and how it contributes to the overall program. If you encounter anything that you don't understand, that's something to investigate further.

- As you make mistakes or find bugs in your programs, fix them. Try to solve your own problems before asking others for help. Learning how to find and fix errors is a key skill to successful programming. Don't neglect learning how to use a debugger (we'll explain how in a future chapter) -- it's a key tool in figuring out where your programs are going wrong.

- Experiment with the examples. Change numbers and text to see what happens. Modify the programs to do additional things (e.g. if a program adds two numbers, make it add three numbers). Try to find different ways to break the programs (if a program asks you to enter a number, try entering a letter instead and see what happens). You'll learn more by modifying the examples than by simply following them.

- Plan to spend some time with the quizzes. If you're new to programming, you may find these challenging (and that's normal, as your brain acclimates to the programming mindset). Don't be discouraged if you don't get the right answer the first time. You may need to try several different approaches before you find a path to success. It's okay to look at the answer if you're really stuck. Just make sure you understand how the provided answer works before proceeding.

- Write your own short programs using the concepts you have learned. This will reinforce your learning and improve your retention.

## ()Common site-related questions 🔗 (#FAQ)

### Q: How do I sign up for the site? How do I get a login?

All parts of this site are accessible anonymously -- therefore, no user account or signup is needed!

### Q: Is there a PDF version of this site available for offline viewing?

Unfortunately, there is not. The site is able to stay free for everyone because we're ad-sponsored -- that model simply doesn't work in PDF format. You are welcome to convert pages from this website into PDF (or any other) format for your own private use, so long as you do not distribute them.

### Q: What should I do if I get stuck on a concept?

If you don't understand something or feel stuck:

- Read through the comments. Other readers may have encountered similar challenges.
- Scan through the next lesson or two in the series -- your question may be answered there.
- Use a search engine to see if your question (or error message) has been addressed elsewhere.
- Ask your question on a site that is designed for programming Q&A, like Stack Overflow.

If all else fails, skip the material you don't understand, and come back to it later. You may find that something that was hard to understand is easier with the additional knowledge and context provided by other articles.

### Q: What do I do if I forget what something means?

Use the Site index (/learn-c-site-index/). Look up any topics you want to know more about there, and you'll find links to the lessons where that topic is discussed.

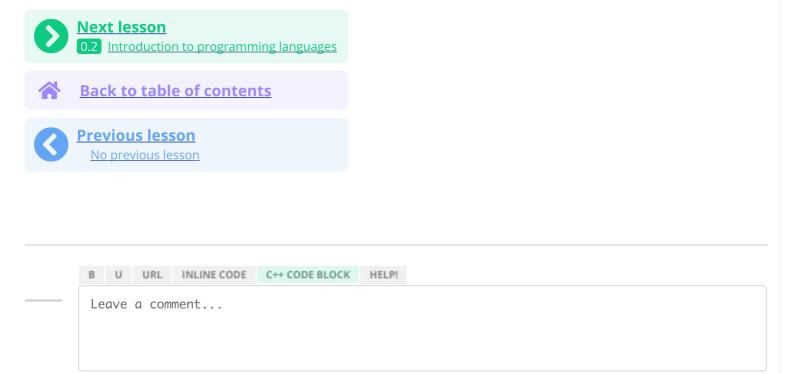### Q: Are there any similar sites for other languages?

Nope. That would require being able to clone myself a few times.

### Q: Can you do a dark mode for this site?

Not easily, but you can! See darkreader.org (https://darkreader.org/).

Finally, one small nag: This site is free because it is ad-supported. If you find yourself enjoying the lessons, please consider disabling your ad blocker.

Alright, let's get on with it!

| B | U | URL | INLINE CODE | C++ CODE BLOCK | HELP! |

Leave a comment...

Name*

Email* ⓘ

Find a mistake?  Leave a comment! ⓘ

Avatars from https://gravatar.com/ are connected to your provided email address.

Notify me about replies: 🔔  **POST COMMENT**

**1.1K COMMENTS**

Newest ▾