



## Next steps

- [“indicators” vs “strategies”](#)
- [How scripts are executed](#)
- [Time series](#)
- [Publishing scripts](#)
- [Getting around the Pine Script™ documentation](#)
- [Where to go from here?](#)

After your [first steps](#) and your [first indicator](#), let us explore a bit more of the Pine Script™ landscape by sharing some pointers to guide you in your journey to learn Pine Script™.

## “indicators” vs “strategies”

Pine Script™ [strategies](#) are used to backtest on historical data and forward test on open markets. In addition to indicator calculations, they contain `strategy.*()` calls to send trade orders to Pine Script™’s broker emulator, which can then simulate their execution. Strategies display backtest results in the “Strategy Tester” tab at the bottom of the chart, next to the “Pine Script™ Editor” tab.

Pine Script™ indicators also contain calculations, but cannot be used in backtesting. Because they do not require the broker emulator, they use less resources and will run faster. It is thus advantageous to use indicators whenever you can.

Both indicators and strategies can run in either overlay mode (over the chart’s bars) or pane mode (in a separate section below or above the chart). Both can also plot information in their respective space, and both can generate [alert events](#).

## How scripts are executed

A Pine script is **not** like programs in many programming languages that execute once and then stop. In the Pine Script™ *runtime* environment, a script runs in the equivalent of an invisible loop where it is executed once on each bar of whatever chart you are on, from left to right. Chart bars that have already closed when the script executes on them are called *historical bars*. When execution reaches the chart’s last bar and the market is open, it is on the *realtime bar*. The script then executes once every time a price or volume change is detected, and one last time for that realtime bar when it closes. That realtime bar then becomes an *elapsed realtime bar*. Note that when the script executes in realtime, it does not recalculate on all the chart’s historical bars on every price/volume update. It has already calculated once on those bars, so it does not need to recalculate them on every chart tick. See the [Execution model](#) page for more information.

When a script executes on a historical bar, the [close](#) built-in variable holds the value of that bar’s close. When a script executes on the realtime bar, [close](#) returns the **current** price of the symbol until the bar closes.

Contrary to indicators, Pine Script™ strategies normally execute only once on realtime bars, when they close. They can also be configured to execute on each price/volume update if that is what you need. See the page on [Strategies](#) for more information, and to understand how strategies calculate differently than indicators.

## Time series

The main data structure used in Pine Script™ is called a [time series](#). Time series contain one value for each bar the script executes on, so they continuously expand as the script executes on more bars. Past values of the time series can be referenced using Pine Script™'s history-referencing operator: `[]`. `close[1]`, for example, refers to the value of `close` on the bar preceding the one where the script is executing.

While this indexing mechanism may remind many programmers of arrays, a time series is different and thinking in terms of arrays will be detrimental to understanding this key Pine Script™ concept. A good comprehension of both the [execution model](#) and [time series](#) is essential in understanding how Pine scripts work. If you have never worked with data organized in time series before, you will need practice to put them to work for you. Once you familiarize yourself with these key concepts, you will discover that by combining the use of time series with our built-in functions specifically designed to handle them efficiently, much can be accomplished in very few lines of Pine Script™ code.

## Publishing scripts

TradingView is home to a large community of Pine Script™ programmers and millions of traders from all around the world. Once you become proficient enough in Pine Script™, you can choose to share your scripts with other traders. Before doing so, please take the time to learn Pine Script™ well-enough to supply traders with an original and reliable tool. All publicly published scripts are analyzed by our team of moderators and must comply with our [Script Publishing Rules](#), which require them to be original and well-documented.

If want to use Pine scripts for your own use, simply write them in the Pine Script™ Editor and add them to your chart from there; you don't have to publish them to use them. If you want to share your scripts with just a few friends, you can publish them privately and send your friends the browser's link to your private publication. See the page on [Publishing](#) for more information.

## Getting around the Pine Script™ documentation

While reading code from published scripts is no doubt useful, spending time in our documentation will be necessary to attain any degree of proficiency in Pine Script™. Our two main sources of documentation on Pine Script™ are:

- This [Pine Script™ v5 User Manual](#)
- Our [Pine Script™ v5 Reference Manual](#)

The [Pine Script™ v5 User Manual](#) is in HTML format and in English only.

The [Pine Script™ v5 Reference Manual](#) documents what each variable, function or Pine Script™ keyword does. It is an essential tool for all Pine Script™ programmers; your life will be miserable if you try to write scripts of any reasonable complexity without consulting it. It exists in two formats: the HTML format we just linked to, and the popup version, which can be accessed from the Pine Script™ Editor, by either `ctrl` + `clicking` on a keyword, or by using the Editor's "More/Pine Script™ reference (pop-up)" menu. The Reference Manual is translated in other languages.

There are five different versions of Pine Script™. Ensure the documentation you use corresponds to the Pine Script™ version you are coding with.

## Where to go from here?

This [Pine Script™ v5 User Manual](#) contains numerous examples of code used to illustrate the concepts we discuss. By going through it, you will be able to both learn the foundations of Pine Script™ and study the example scripts. Reading

about key concepts and trying them out right away with real code is a productive way to learn any programming language. As you hopefully have already done in the [First indicator](#) page, copy this documentation's examples in the Editor and play with them. Explore! You won't break anything.

This is how the [Pine Script™ v5 User Manual](#) you are reading is organized:

- The [Language](#) section explains the main components of the Pine Script™ language and how scripts execute.
- The [Concepts](#) section is more task-oriented. It explains how to do things in Pine Script™.
- The [Writing](#) section explores tools and tricks that will help you write and publish scripts.
- The [FAQ](#) section answers common questions from Pine Script™ programmers.
- The [Error messages](#) page documents causes and fixes for the most common runtime and compiler errors.
- The [Release Notes](#) page is where you can follow the frequent updates to the Pine Script™.
- The [Migration guides](#) section explains how to port between different versions of Pine Script™.
- The [Where can I get more information](#) page lists other useful Pine Script™-related content, including where to ask questions when you are stuck on code.

We wish you a successful journey with Pine Script™... and trading!



---

[First indicator](#)

[Language](#)

© Copyright 2023, TradingView.

