

[Getting Started](#)[Scala 3](#) ▾[Learn](#) ▾

# GETTING STARTED

The instructions below cover both Scala 2 and Scala 3.

## Resources For Newcomers



### Are You Coming From Java?

What you should know to get to speed with Scala after your initial setup.



### Scala in the Browser

To start experimenting with Scala right away, use "Scastie" in your browser.

## Install Scala on your computer

Installing Scala means installing various command-line tools such as the Scala compiler and build tools. We recommend using the Scala installer tool "Coursier" that automatically installs all the requirements, but you can still manually install each tool.

### Using the Scala Installer (recommended way)

The Scala installer is a tool named [Coursier](#), whose main command is named `cs`. It ensures that a JVM and standard Scala tools are installed on your system. Install it on your system



with the following instructions.

macOS **Linux** Windows Other

Run the following command in your terminal, following the on-screen instructions.

On the x86-64 architecture:

```
curl -fL https://github.com/coursier/coursier/releases/latest/download/cs-
```

Otherwise, on the ARM64 architecture:

```
curl -fL https://github.com/VirtusLab/coursier-m1/releases/latest/download
```

**i** You may need to restart your terminal, log out, or reboot in order for the changes to take effect.

Testing your setup



Along with managing JVMs, `cs setup` also installs useful command-line tools:

Commands	Description
<code>scalac</code>	the Scala compiler
<code>scala</code>	the Scala REPL and script runner
<code>scala-cli</code>	<a href="#">Scala CLI</a> , interactive toolkit for Scala
<code>sbt</code> , <code>sbtn</code>	The <a href="#">sbt</a> build tool
<code>amm</code>	<a href="#">Ammonite</a> is an enhanced REPL
<code>scalafmt</code>	<a href="#">Scalafmt</a> is the Scala code formatter

For more information about `cs`, read [coursier-cli documentation](#).

`cs setup` installs the Scala 3 compiler and runner by default (the `scalac` and `scala` commands, respectively). Whether you intend to use Scala 2 or 3, this is



*usually not an issue because most projects use a build tool that will use the correct version of Scala irrespective of the one installed “globally”. Nevertheless, you can always launch a specific version of Scala using*

```
$ cs launch scala:2.13.10  
$ cs launch scalac:2.13.10
```

*If you prefer Scala 2 to be run by default, you can force that version to be installed with:*

```
$ cs install scala:2.13.10 scalac:2.13.10
```

## ...or manually

You only need two tools to compile, run, test, and package a Scala project: Java 8 or 11, and sbt. To install them manually:

1. if you don't have Java 8 or 11 installed, download Java from [Oracle Java 8](#), [Oracle Java 11](#), or [AdoptOpenJDK 8/11](#). Refer to [JDK Compatibility](#) for Scala/Java compatibility detail.
2. Install [sbt](#)

## Create a “Hello World” project with sbt

Once you have installed sbt, you are ready to create a Scala project, which is explained in the following sections.

To create a project, you can either use the command line or an IDE. If you are familiar with the command line, we recommend that approach.

### Using the command line

sbt is a build tool for Scala. sbt compiles, runs, and tests your Scala code. (It can also publish libraries and do many other tasks.)

To create a new Scala project with sbt:

1. `cd` to an empty folder.
2. Run the command `sbt new scala/scala3.g8` to create a Scala 3 project, or `sbt new scala/hello-world.g8` to create a Scala 2 project. This pulls a project template



from GitHub. It will also create a `target` folder, which you can ignore.

3. When prompted, name the application `hello-world`. This will create a project called “hello-world”.
4. Let’s take a look at what just got generated:

```
- hello-world
  - project (sbt uses this for its own files)
    - build.properties
  - build.sbt (sbt's build definition file)
  - src
    - main
      - scala (all of your Scala code goes here)
        - Main.scala (Entry point of program) <-- this is all we need
```

More documentation about sbt can be found in the [Scala Book](#) (see [here](#) for the Scala 2 version) and in the official sbt [documentation](#)

## With an IDE

You can skip the rest of this page and go directly to [Building a Scala Project with IntelliJ and sbt](#)

## Open hello-world project

Let’s use an IDE to open the project. The most popular ones are [IntelliJ](#) and [VSCode](#). They both offer rich IDE features, but you can still use [many other editors](#).

### Using IntelliJ

1. Download and install [IntelliJ Community Edition](#)
2. Install the Scala plugin by following [the instructions on how to install IntelliJ plugins](#)
3. Open the `build.sbt` file then choose *Open as a project*

### Using VSCode with metals

1. Download [VSCode](#)
2. Install the Metals extension from [the Marketplace](#)
3. Next, open the directory containing a `build.sbt` file (this should be the directory `hello-world` if you followed the previous instructions). When prompted to do so,



select *Import build*.

## Play with the source code

View these two files in your IDE:

- `build.sbt`
- `src/main/scala/Main.scala`

When you run your project in the next step, the configuration in `build.sbt` will be used to run the code in `src/main/scala/Main.scala`.

## Run Hello World

If you're comfortable using your IDE, you can run the code in `Main.scala` from your IDE.

Otherwise, you can run the application from a terminal with these steps:

1. `cd` into `hello-world`.
2. Run `sbt`. This opens up the sbt console.
3. Type `~run`. The `~` is optional and causes sbt to re-run on every file save, allowing for a fast edit/run/debug cycle. sbt will also generate a `target` directory which you can ignore.

When you're finished experimenting with this project, press `[Enter]` to interrupt the `run` command. Then type `exit` or press `[Ctrl+D]` to exit sbt and return to your command line prompt.

## Next Steps

Once you've finished the above tutorials, consider checking out:

- [The Scala Book](#) (see the Scala 2 version [here](#)), which provides a set of short lessons introducing Scala's main features.
- [The Tour of Scala](#) for bite-sized introductions to Scala's features.
- [Learning Resources](#), which includes online interactive tutorials and courses.
- [Our list of some popular Scala books](#).



- [The migration guide](#) helps you to migrate your existing Scala 2 code base to Scala 3.

## Getting Help

There are a multitude of mailing lists and real-time chat rooms in case you want to quickly

Need Help?



and for where to reach out for help.

### DOCUMENTATION

[Getting Started](#)

[API](#)

[Overviews/Guides](#)

[Language Specification](#)

### DOWNLOAD

[Current Version](#)

[All versions](#)

### COMMUNITY

[Community](#)

[Mailing Lists](#)

[Chat Rooms & More](#)

[Libraries and Tools](#)

[The Scala Center](#)

### CONTRIBUTE

[How to help](#)

[Report an Issue](#)

### SCALA

[Blog](#)

[Code of Conduct](#)

[License](#)

### SOCIAL

[GitHub](#)

[Twitter](#)

