



Herbie automatically improves the error of floating point expressions. Visit our website for tutorials, documentation, and an online demo. Herbie has semi-regular releases once a year, maintains backwards compatibility, and uses standardized formats.

Installing

For full details on installing Herbie, please see the documentation.

Installing from Source

Installing from source requires Racket 8.0 or later, Rust 1.60.0 or later, and supports Windows, macOS, and Linux for various architectures.

Install Racket from [here](#). We recommend the official Racket installer over Snap. If your configuration depends on Racket being installed via Snap, you will need to ensure that Herbie and all packages are located in your home directory or another allow-listed directory. Install Rust from [here](#). In this directory, build Herbie with:

```
1 make install
```

This will install a `herbie` binary somewhere under `~/`. `racket` (Linux) although this path is dependent on OS and Racket version (check installation messages for a possible location). Depending on your configuration, you may need to add the binary to your path manually. You can also run `src/herbie.rkt` directly.

Installing from the Racket package index

Use this method for installing Herbie if Rust is not on your system. Installing via the Racket package index requires Racket 8.0 or later and supports Windows, macOS, and Linux on x86-64 architectures.

Install Racket from [here](#). Install Herbie with:

```
1 raco pkg install --auto herbie
```

This will install a `herbie` binary somewhere under `~/`. `racket` (Linux) although this path is dependent on OS and Racket version (check installation messages for a possible location). You can also run `src/herbie.rkt` directly.

Please note that this method of installation **will fail** for *Apple M1* systems and other *ARM* architectures. We hope to support this in the near future.

Running Herbie

For full details on running Herbie, please see the tutorial.

Herbie's input format is the Scheme-like FPCore; for example $(1 + x) - x$ is written

```
1 (FPCore (x) (- (+ 1 x) x))
```

You can see more examples in `bench/`. To use Herbie, run `herbie shell` and enter an FPCore expression:

```
1 $ herbie shell
2 Herbie 1.3 with seed 1866513483
3 Find help on https://herbie.uwplse.org/, exit with Ctrl-D
4 herbie> (FPCore (x) (- (+ 1 x) x))
5 (FPCore (x) ... 1)
```

In this case Herbie's improved, more-accurate expression is the constant 1.

Besides `shell`, Herbie also has a `web` interface, and can be run in batch mode on files with the `improve` and `report` commands. Consult the documentation. for more.

Helping Out

Herbie is developed as a joint project of the Universities of Washington and Utah. We use Github and Trello to organize development goals.

Running Tests

Herbie has unit tests for basic functionality, though coverage is far from complete. You can run the test suite by downloading the source code, changing to the source code directory, and running:

```
1 make install
2 raco test src/
```

Herbie also contains a large benchmark suite drawn from open source projects, examples from users, and numerical analysis textbooks. This suite is found in [bench/](#). The full test can be run with

```
1 herbie report bench/ report/
```

You may see warnings; these are expected. The output is HTML files in [report/](#). This full test can take a few hours to run. We often test Herbie on basic but representative examples with:

```
1 herbie report bench/hamming/ graphs/
```

This takes approximately 10 minutes.

Historic and nightly test results are collected on nightly.