

# Elixir (programming language)

**Elixir** is a functional, concurrent, general-purpose programming language that runs on the BEAM virtual machine used to implement the Erlang programming language.<sup>[3]</sup> Elixir builds on top of Erlang and shares the same abstractions for building distributed, fault-tolerant applications. Elixir also provides productive tooling and an extensible design. The latter is supported by compile-time metaprogramming with macros and polymorphism via protocols.<sup>[4]</sup>

Elixir is used by companies such as PagerDuty,<sup>[5]</sup> Discord,<sup>[6]</sup> E-MetroTel,<sup>[7]</sup> Pinterest,<sup>[8]</sup> Moz,<sup>[9]</sup> Bleacher Report,<sup>[10]</sup> The Outline,<sup>[11]</sup> Inverse,<sup>[12]</sup> Divvy,<sup>[13]</sup> FarmBot<sup>[14]</sup> and for building embedded systems.<sup>[15][16]</sup> The community organizes yearly events in the United States,<sup>[17]</sup> Europe<sup>[18]</sup> and Japan<sup>[19]</sup> as well as minor local events and conferences.<sup>[20][21]</sup>

## Contents

### History

### Versioning

### Features

### Examples

### Noteworthy Elixir projects

### See also

### References

### External links

## History

José Valim is the creator of the Elixir programming language, a research and development project of Plataformatec. His goals were to enable higher extensibility and productivity in the Erlang VM while keeping compatibility with Erlang's ecosystem.<sup>[22][23]</sup>

José Valim aimed to create a programming language for large-scale sites and apps. Being a Ruby developer, he used features of Ruby, Erlang, and Clojure to develop a high-concurrency and low-latency language. Elixir was designed to handle large data volumes. Its speed and capabilities spread Elixir in telecommunication, eCommerce, and finance industries.<sup>[24]</sup>

On July 12, 2018, Honeypot released a mini-documentary on Elixir.<sup>[25]</sup>

## Versioning

### Elixir



<b>Paradigm</b>	multi-paradigm: <u>functional</u> , <u>concurrent</u> , <u>distributed</u> , <u>process-oriented</u>
<b>First appeared</b>	2011
<b>Stable release</b>	1.11.1 / 16 October 2020 <sup>[1]</sup>
<b>Typing discipline</b>	<u>dynamic</u> , <u>strong</u> , <u>duck</u>
<b>Platform</b>	<u>Erlang</u>
<b>License</b>	<u>Apache License 2.0</u> <sup>[2]</sup>
<b>Filename extensions</b>	.ex, .exs
<b>Website</b>	<u>elixir-lang.org</u> ( <u>https://elixir-lang.org</u> )
<b>Influenced by</b>	<u>Clojure</u> , <u>Erlang</u> , <u>Ruby</u>
<b>Influenced</b>	<u>LFE</u>

Elixir mostly<sup>[26]</sup> follows semantic versioning and has only 1 major version with no plans for a second. Each of the minor versions supports a specific range of Erlang/OTP versions.<sup>[27]</sup>

## Features

---

- compiles to bytecode for the Erlang Virtual Machine (BEAM)<sup>[28]</sup>
- Everything is an expression<sup>[28]</sup>
- Erlang functions can be called from Elixir without run time impact, due to compilation to Erlang bytecode, and vice versa
- Meta programming allowing direct manipulation of abstract syntax tree (AST)<sup>[28]</sup>
- Polymorphism via a mechanism called protocols. Like in Clojure, protocols provide a dynamic dispatch mechanism. However, this is not to be confused with multiple dispatch as Elixir protocols dispatch on a single type.
- Support for documentation via Python-like docstrings in the Markdown formatting language<sup>[28]</sup>
- Shared nothing concurrent programming via message passing (Actor model)<sup>[29]</sup>
- Emphasis on recursion and higher-order functions instead of side-effect-based looping
- Lightweight concurrency utilizing Erlang's mechanisms<sup>[28]</sup>
- Railway oriented programming via the `with` construct
- Built-in tooling for managing dependencies, code compilation, running tests, formatting code, remote debugging and more
- Lazy and async collections with streams
- Pattern matching<sup>[28]</sup> to promote assertive code<sup>[30]</sup>
- Unicode support and UTF-8 strings

## Examples

---

The following examples can be run in an `iex` shell or saved in a file and run from the command line by typing `elixir <filename>`.

Classic Hello world example:

```
iex> IO.puts("Hello World!")
Hello World!
```

Comprehensions

```
iex> for n <- [1,2,3,4,5], rem(n, 2) == 1, do: n*n
[1, 9, 25]
```

Pattern Matching (destructuring)

```
iex> [1, a] = [1, 2]
iex> a
2

iex> {:ok, [hello: a]} = {:ok, [hello: "world"]}
iex> a
"world"
```

## Pattern Matching (multiple clauses)

```
iex> case File.read("path/to/file") do
iex>   {:ok, contents} -> IO.puts("found file: #{contents}")
iex>   {:error, reason} -> IO.puts("missing file: #{reason}")
iex> end
```

## Pipe Operator

```
iex> "1" |> String.to_integer() |> Kernel.*(2)
2
```

## Modules

```
defmodule Fun do
  def fib(0), do: 0
  def fib(1), do: 1
  def fib(n), do: fib(n-2) + fib(n-1)
end
```

## Sequentially spawning a thousand processes

```
for num <- 1..1000, do: spawn fn -> IO.puts("#{num * 2}") end
```

## Asynchronously performing a task

```
task = Task.async fn -> perform_complex_action() end
other_time_consuming_action()
Task.await task
```

## Noteworthy Elixir projects

---

- Mix is a build automation tool that provides tasks for creating, compiling, and testing Elixir projects, managing its dependencies, and more.<sup>[31]</sup>
- Phoenix is a web development framework written in Elixir which implements the server-side Model View Controller (MVC) pattern.<sup>[32]</sup>
- Nerves is a platform, framework, and tooling environment for building embedded systems and devices.<sup>[16][33]</sup>
- Ecto is the database wrapper and query generator for Elixir.<sup>[34]</sup>

## See also

---

- Concurrent computing
- Distributed computing

## References

---

1. "Releases - elixir-lang/elixir" (<https://github.com/elixir-lang/elixir/releases>). Retrieved 16 October 2020 – via GitHub.
2. "elixir/LICENSE at master · elixir-lang/elixir · GitHub" (<https://github.com/elixir-lang/elixir/blob/master/LICENSE>). *GitHub*.

3. "Most Popular Programming Languages of 2018 - Elite Infoworld Blog" (<https://www.eliteinfoworld.com/blog/popular-programming-languages-2018/>). 2018-03-30. Retrieved 2018-05-08.
4. "Elixir" (<https://elixir-lang.org>). José Valim. Retrieved 2013-02-17.
5. "Elixir at PagerDuty" (<https://www.pagerduty.com/blog/elixir-at-pagerduty/>). PagerDuty. 2018-06-14. Retrieved 2019-04-21.
6. Vishnevskiy, Stanislav (Jul 6, 2017). "How Discord Scaled Elixir to 5,000,000 Concurrent Users" (<https://blog.discordapp.com/scaling-elixir-f9b8e1e7c29b?gi=3afa589deb67>). Retrieved 2019-04-21.
7. "What's New in Release 6.0 | Documentation" (<https://www.emetrotel.com/tsd/content/whats-new-release-60>). *www.emetrotel.com*. Retrieved 2019-04-21.
8. "Introducing new open-source tools for the Elixir community" (<https://engineering.pinterest.com/blog/introducing-new-open-source-tools-elixir-community>). Retrieved 2016-08-01.
9. "Unlocking New Features in Moz Pro with a Database-Free Architecture" (<https://moz.com/devblog/moz-analytics-db-free/>). Retrieved 2016-08-01.
10. "Elixir" (<https://dev.bleacherreport.com/tagged/elixir>). *Bleacher Report Engineering*. Retrieved 2019-05-22.
11. Lucia, Dave (Sep 24, 2018). "Two years of Elixir at The Outline" (<https://blog.usejournal.com/two-years-of-elixir-at-the-outline-ad671a56c9ce?gi=9931fa1dcdbc>). Retrieved 2019-05-22.
12. "What big projects use Elixir?" (<https://www.quora.com/What-big-projects-use-Elixir>). Retrieved 2016-08-01.
13. "Why Divvy uses Elixir instead of more popular coding languages" (<https://medium.com/@divvyhq/why-divvy-uses-elixir-instead-of-more-popular-coding-languages-92c514dc47d0>). Retrieved 2019-04-30.
14. *The operating system and all related software that runs on FarmBot's Raspberry Pi.: FarmBot/farmbot\_os* ([https://github.com/FarmBot/farmbot\\_os](https://github.com/FarmBot/farmbot_os)), FarmBot, 2019-10-28, retrieved 2019-10-29
15. "Elixir in production interview: Garth Hitchens" (<http://blog.plataformatec.com.br/2015/06/elixir-in-production-interview-garth-hitches/>). Retrieved 2016-08-01.
16. "Nerves - Craft and deploy bulletproof embedded software in Elixir" (<http://nerves-project.org/>). Retrieved 2016-08-01.
17. "ElixirConf" (<http://elixirconf.com/>). Retrieved 2018-07-11.
18. "ElixirConf" (<http://elixirconf.eu/>). Retrieved 2018-07-11.
19. "Erlang & Elixir Fest" (<https://elixir-fest.jp/>). Retrieved 2019-02-18.
20. "Elixir LDN" (<http://www.elixir.london/>). Retrieved 2018-07-12.
21. "EMPEX - Empire State Elixir Conference" (<http://empex.co/>). Retrieved 2018-07-12.
22. *Elixir - A modern approach to programming for the Erlang VM* (<https://vimeo.com/53221562>). Retrieved 2013-02-17.
23. José Valim - ElixirConf EU 2017 Keynote (<https://www.youtube.com/watch?v=IZvpKhA6t8A>). Retrieved 2017-07-14.
24. "Behinde the code: The One Who Created Elixir" (<https://www.welcometothejungle.com/en/articles/btc-elixir-jose-valim/>). Retrieved 2019-11-25.
25. "Elixir: A Mini-Documentary" (<http://doc.honeypot.io/elixir-documentary-2018/>). Retrieved 2018-07-12.
26. "Imperative Assignments are breaking the application in 1.7 update · Issue #8076 · elixir-lang/elixir" (<https://github.com/elixir-lang/elixir/issues/8076>). *GitHub*. Retrieved 2020-02-10.
27. *Elixir is a dynamic, functional language designed for building scalable and maintainable applications: elixir-lang/elixir* (<https://github.com/elixir-lang/elixir>), Elixir, 2019-04-21, retrieved 2019-04-21
28. "Elixir" (<https://elixir-lang.org/>). Retrieved 2014-09-07.

29. Loder, Wolfgang (12 May 2015). *Erlang and Elixir for Imperative Programmers* (<https://leanpub.com/erlangandelixirforimperativeprogrammers>). "Chapter 16: Code Structuring Concepts", section title "Actor Model": Leanpub. Retrieved 7 July 2015.
30. "Writing assertive code with Elixir" (<http://blog.plataformatec.com.br/2014/09/writing-assertive-code-with-elixir/>). Retrieved 2018-07-05.
31. "Mix" (<https://hexdocs.pm/mix/Mix.html>). Retrieved 2019-04-18.
32. "Overview" (<https://hexdocs.pm/phoenix/overview.html>). Retrieved 2019-04-18.
33. "Getting Started" (<https://hexdocs.pm/nerves/getting-started.html>). Retrieved 2019-04-18.
34. "Getting Started" (<https://hexdocs.pm/ecto/getting-started.html#content>). Retrieved 2019-04-16.

## External links

---

- [Elixir language website \(https://elixir-lang.org\)](https://elixir-lang.org)
- 

Retrieved from "[https://en.wikipedia.org/w/index.php?title=Elixir\\_\(programming\\_language\)&oldid=983840263](https://en.wikipedia.org/w/index.php?title=Elixir_(programming_language)&oldid=983840263)"

---

This page was last edited on 16 October 2020, at 15:18 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.