

# Elixir (programming language)

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

Elixir is used by companies such as E-MetroTel, Pinterest<sup>[4]</sup> and Moz.<sup>[5]</sup> Elixir is also used for web development, by companies such as Bleacher Report, Discord, and Inverse (<https://www.inverse.com/>),<sup>[6]</sup> and for building embedded systems.<sup>[7][8]</sup> The community organizes yearly events in United States<sup>[9][10][11]</sup>, Europe<sup>[12]</sup> and Japan<sup>[13]</sup> as well as minor local events and conferences.<sup>[14][15]</sup>

## Contents


- History
- Features
- Examples
- References
- External links

## History

José Valim is the creator of the Elixir programming language, an R&D project of Plataformatec (<http://plataformatec.com.br>). His goals were to enable higher extensibility and productivity in the Erlang VM while keeping compatibility with Erlang's ecosystem.<sup>[16]</sup> <sup>[17]</sup>

## Features

- A language that compiles to bytecode for the Erlang Virtual Machine (BEAM)<sup>[18]</sup>
- Everything is an expression<sup>[18]</sup>
- Erlang functions can be called from Elixir without run

Elixir	
	
<b>Paradigm</b>	multi-paradigm: functional, concurrent, distributed, process-oriented
<b>First appeared</b>	2011
<b>Stable release</b>	1.6.0 / 18 January 2018 <sup>[1]</sup>
<b>Typing discipline</b>	dynamic, strong
<b>Platform</b>	Erlang
<b>License</b>	Apache License 2.0 <sup>[2]</sup>
<b>Filename extensions</b>	.ex, .exs
<b>Website</b>	elixir-lang.org (https://elixir-lang.org)
<b>Influenced by</b>	
Erlang, Ruby, Clojure	
<b>Influenced</b>	
LFE	

time impact, due to compilation to Erlang bytecode, and vice versa

- Meta programming allowing direct manipulation of AST<sup>[18]</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>[18]</sup>
- Shared nothing concurrent programming via message passing (Actor model)<sup>[19]</sup>
- Emphasis on recursion and higher-order functions instead of side-effect-based looping
- Lightweight concurrency utilizing Erlang's mechanisms.<sup>[18]</sup>
- Lazy and async collections with streams
- Pattern matching<sup>[18]</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

```
iex> [1, a] = [1, 2]  
iex> a  
2  
  
iex> {:ok, [hello: a]} = {:ok, [hello: "world"]}  
iex> a  
"world"
```

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  
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
```

## References

---

1. <https://elixir-lang.org/blog/2018/01/17/elixir-v1-6-0-released/>. Retrieved 17 January 2018. Missing or empty |title= (help)
2. "elixir/LICENSE at master · elixir-lang/elixir · GitHub" (<https://github.com/elixir-lang/elixir/blob/master/LICENSE>). *GitHub*.
3. "Elixir" (<https://elixir-lang.org>). *José Valim*. Retrieved 2013-02-17.
4. "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.
5. "Unlocking New Features in Moz Pro with a Database-Free Architecture" (<https://moz.com/devblog/moz-analytics-db-free/>). Retrieved 2016-08-01.
6. "What big projects use Elixir?" (<https://www.quora.com/What-big-projects-use-Elixir>). Retrieved 2016-08-01.
7. "Elixir in production interview: Garth Hitchens" (<http://blog.plataformatec.com.br/2015/06/elixir-in-production-interview-garth-hitches/>). Retrieved 2016-08-01.
8. "Nerves - Craft and deploy bulletproof embedded software in Elixir" (<http://nerves-project.org/>). Retrieved 2016-08-01.
9. "ElixirConf 2014" (<http://elixirconf.com/2014/>). Retrieved 2016-08-01.
10. "ElixirConf 2015" (<http://elixirconf.com/2014/>). Retrieved 2016-08-01.
11. "ElixirConf" (<http://elixirconf.com/>). Retrieved 2016-08-01.
12. "ElixirConf" (<http://elixirconf.eu/>). Retrieved 2016-08-01.
13. "ElixirConf.jp" (<http://www.elixirconf.jp/>). Retrieved 2017-07-09.
14. "Elixir LDN" (<http://www.elixir.london/2016>). Retrieved 2016-08-01.
15. "EMPEX - Empire State Elixir Conference" (<http://empex.co/>). Retrieved 2016-08-01.
16. *Elixir - A modern approach to programming for the Erlang VM* (<http://vimeo.com/53221562>). Retrieved 2013-02-17.
17. *José Valim - ElixirConf EU 2017 Keynote* (<https://www.youtube.com/watch?v=IZvpKhA6t8A>). Retrieved 2017-07-14.
18. "Elixir" (<https://elixir-lang.org/>). Retrieved 2014-09-07.

19. 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.

## External links

---

- [Elixir language website \(https://elixir-lang.org\)](https://elixir-lang.org)
  - [Code on GitHub \(https://github.com/elixir-lang/elixir/\)](https://github.com/elixir-lang/elixir/)
  - [RED9 \(https://red9.ir/what-is-red9?\)](https://red9.ir/what-is-red9?) / Service Delivery Platform developed by Elixir
  - [Elixir - A modern approach to programming for the Erlang VM \(http://vimeo.com/53221562\)](http://vimeo.com/53221562) (video presentation)
  - Dave Thomas: "Programming Elixir 1.2: Functional → Concurrent → Pragmatic → Fun" (book) (<https://pragprog.com/book/elixir12/programming-elixir-1-2>)
  - Simon St. Laurent, J. David Eisenberg: "Introducing Elixir" (book) (<http://shop.oreilly.com/product/0636920030584.do>)
  - Chris McCord: "Metaprogramming Elixir " (book) (<http://shop.oreilly.com/product/9781680500417.do>)
- 

Retrieved from "[https://en.wikipedia.org/w/index.php?title=Elixir\\_\(programming\\_language\)&oldid=821434085](https://en.wikipedia.org/w/index.php?title=Elixir_(programming_language)&oldid=821434085)"

---

**This page was last edited on 20 January 2018, at 12:57.**

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.