

Hitting the ground running with Elixir - First impressions of Elixir as a JS developer

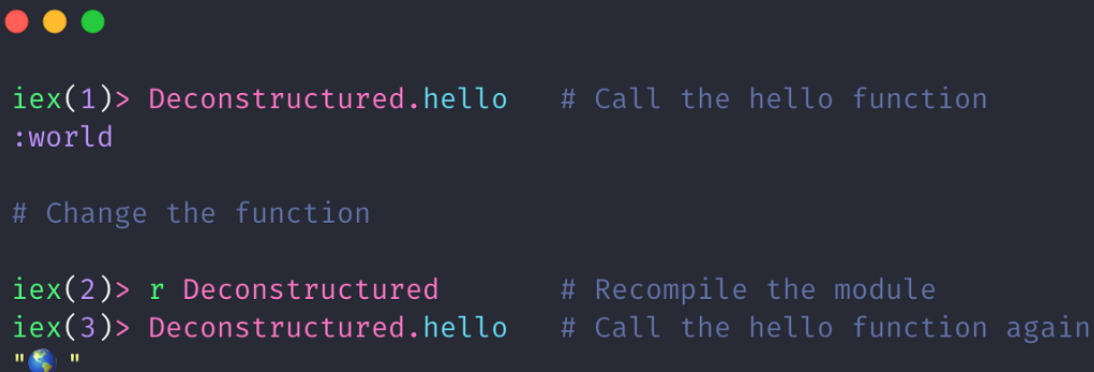
Written by Seth Corker on Benevolent Bytes

Why Elixir?

I stumbled upon Elixir when looking for something new to learn. Although I've used a few languages, I mostly write in JavaScript for work and by extension, many of my side projects too. I want to try something different and I'd heard some good things about Elixir. For the past week, I've been experimenting with Elixir and I've noticed some key things that help you hit the ground running. It reminds me of the productivity I saw with Ruby on Rails.

What is Elixir and how does it help productivity?

Elixir is a functional programming language which runs on the Erlang VM, it's touted as being "designed for building scalable and maintainable applications" and if you squint, it has a very Ruby-esque syntax on the surface. Elixir is easy to get started and what's included out of the box makes it one of the most accessible programming languages I've ever worked with. The tools included by default are great and make for a very good developer experience. So what are these tools and what sets them apart?



```
iex(1)> Deconstructed.hello    # Call the hello function
:world

# Change the function

iex(2)> r Deconstructed        # Recompile the module
iex(3)> Deconstructed.hello    # Call the hello function again
"🌍"
```

IEx REPL Elixir ships with IEx, much like IRB for Ruby but it has some more advanced features. IEx is indispensable when developing. Running your project with:

```
iex -S mix
```

Allows you to recompile modules without leaving iex:

```
iex(1)> r MyModule
```

A REPL is a nice way to try something out quickly without leaving the terminal.

Hex package manager As a JavaScript developer, I use npm or yarn on a daily basis. It's indispensable for a modern web developer and ubiquitous to modern web development. Hex is no different but it does have a few key advantages over the JavaScript or Ruby equivalents.

Hex comes out of the box. Although it might seem like a minor difference, learning how to get started with JavaScript doesn't always lead to learning how NPM works right away and neither does Ruby and RubyGems. Hex included in an Elixir installation makes takes an extra barrier away from getting started and that's a good thing.

ExDoc Documentation is a first-class citizen in Elixir. It's a pleasure to write and ExDoc generates a great looking static site. You can attach documentation for modules and functions like you would in most languages but there are some really cool additions. All documentation can include markdown which will render in the resulting web page and is non-intrusive.

Documentation ExDoc generates taken from the official Elixir documentation <https://hexdocs.pm/elixir/Kernel.html>

One of my favourite features is Doctests, if you include the title "Examples" in a test along with some sample code and the example will run during testing.

Testing with ExUnit Like everything else in Elixir, testing is easy and turned on by default. When you setup a project with the build tool Mix, a directory is created called **test** where all of your tests can sit suffixed with `"_test"`. Testing by default encourages good coding practice and is very easy with a functional language without side effects. It is much simpler to get started testing coming from a JavaScript perspective because all of the setup is done.

Where do I go from here?

If you want to give Elixir a try, I hope this article has convinced you that the barrier to entry is really low. It's got everything you need to get started and hit the ground running. You can toy around in it using the REPL (IEx) or learn how to get creating quickly with the helpful Elixir guides. Overall the documentation is really helpful and consistent, because of the focus on documentation from the beginning, most modules you find for Elixir should have enough to get started along with some example code to boot.

Website for Elixir

If you're curious about Elixir's syntax, I always recommend Learn X in Y minutes. It's a helpful cheat-sheet:

Learn elixir in Y Minutes

Thanks for taking the time to see why I think Elixir is a cool ecosystem to get started in and be productive from the outset. Let me know what your experience is with Elixir and if you're just dipping your toe in the water too, how easy is it to get started?

Background photo by Martin Sanchez on Unsplash