

In the 1980s, to solve the problem of phone switching with more and more subscribers coming in, Ericsson developed **Erlang** which solved the problem to a great extent. Erlang consisted of tools that offered scalability, reliability and concurrency. To solve a similar problem between the connections presented on the web, *Jose Valim* created Elixir.

It is inherently similar to Ruby and offers features and functionalities which are already available with Ruby and hence developers argue why they should turn to Elixir. Learn it and incorporate it in their software when they are already well versed with Ruby.

The answers to those questions lie in the further parts of this article.

The world wide web is growing wider as you are reading this article. A couple of years ago, Elixir emerged as the newest addition to the modern [web technologies](#) which aimed at solving the problem of concurrency. While there [exist languages](#) like Java, Ruby, etc. which have been on the scene for a long time now.

It is still gaining huge popularity and a lot of developers are putting it to good use. It offers excellent scalability, excellent reliability and offers the developers a chance to write clean code.

Although it has gained popularity in the world of professional developers, students are still aloof to the benefits of this technology and are not entirely aware of the possibilities it has to offer and the additions it can do to your piece of software.

Table of Contents

- History of Elixir
- Application Concurrency in Elixir
- Features of Elixir
- Benefits of using Elixir
- Phoenix: The Framework of all Frameworks
- Frequently Asked Questions
- Conclusion