* Parsing
* Validation
* Code Generation
* Integration with Libraries
* Integration with web technologies

JRuby, Closure are dynamically typed languages that make good DSLs. They are easier to read with less noise and without the nitty gritty type annotations. These don’t use compile checking which is very useful when making DSLs, it can be used for code generation which can only be invoked at runtime through metamodels or reflection which is helpful when these languages being dynamically typed which is also referred to as Duck Typing. JRuby and Closure use Duck Typing.

Both languages have metaprogramming facilities whereby you are able to introduce objects, functions and attributes at runtime. Dynamically typed languages can be intercepted during t

These languages can be used for internal or external DSLs.

All these languages run on the JVM platform

Scala is often compared with [Groovy](https://en.wikipedia.org/wiki/Groovy_(programming_language)) and [Clojure](https://en.wikipedia.org/wiki/Clojure" \o "Clojure), two other programming languages also using the JVM. Substantial differences between these languages are found in the type system, in the extent to which each language supports object-oriented and functional programming, and in the similarity of their syntax to the syntax of Java.

Scala is [statically typed](https://en.wikipedia.org/wiki/Statically_typed), while both Groovy and Clojure are [dynamically typed](https://en.wikipedia.org/wiki/Dynamically_typed). This makes the type system more complex and difficult to understand but allows almost all type errors to be caught at compile-time and can result in significantly faster execution. By contrast, dynamic typing requires more testing to ensure program correctness and is generally slower in order to allow greater programming flexibility and simplicity. Regarding speed differences, current versions of Groovy and Clojure allow for optional type annotations to help programs avoid the overhead of dynamic typing in cases where types are practically static. This overhead is further reduced when using recent versions of the JVM, which has been enhanced with an *invoke dynamic* instruction for methods that are defined with dynamically typed arguments. These advances reduce the speed gap between static and dynamic typing, although a statically typed language, like Scala, is still the preferred choice when execution efficiency is very important.

Regarding programming paradigms, Scala inherits the object-oriented model of Java and extends it in various ways. Groovy, while also strongly object-oriented, is more focused in reducing verbosity. In Clojure, object-oriented programming is deemphasised with functional programming being the main strength of the language. Scala also has many functional programming facilities, including features found in advanced functional languages like [Haskell](https://en.wikipedia.org/wiki/Haskell_(programming_language)), and tries to be agnostic between the two paradigms, letting the developer choose between the two paradigms or, more frequently, some combination thereof.

Regarding syntax similarity with Java, Scala inherits much of Java's syntax, as is the case with Groovy. Clojure on the other hand follows the [Lisp](https://en.wikipedia.org/wiki/Lisp_(programming_language)) syntax, which is different in both appearance and philosophy. However, learning Scala is also considered difficult because of its many advanced features. This is not the case with Groovy, despite its also being a feature-rich language, mainly because it was designed to be mainly a scripting language.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

Scala (Find out more about scala)

Scala is a multi-paradigm general purpose programming language. Scala is an object-oriented programming language that uses many functional programming features. Scala was designed to bridge Java’s shortcomings. It uses the JVM platform and everything gets compiled to bytecode. It is statically types compared to Ruby which is dynamically typed.

Scala can be used to make internal and external dsls

Difference with Ruby duck-typing

Duck typing in Scala is statically checked