

Scala vs Python: Comparison Table

Sl.No.	Evaluation Criteria	Scala	Python
1	Relative Performance	Fast	Slow
2	Relative background of the development team	If Java, then Scala	If C/C++, then Python
3	Runtime environment	<ul style="list-style-type: none"> • JVM and Java Infrastructure • Compiler • Can be run in an 'interpreted' mode 	<ul style="list-style-type: none"> • Platform OS + Python Libraries • Combination of compiler and interpreter (like Java) • Interpreter (PyPy or CPython or Jython) • CPython (Compiled to IR and Interpreted) • PyPy (Interpreted and Compiled to machine code)
4	Relative suitability	If large project, Scala + AKKA + Java Libs can be used as required, else Scala.	If project large, use C/C++ for performance, else Python for simplicity. Do not go to Cython. Use Scala.
5	Type system	Static: Code robustness.	Dynamic: Flexibility & speed.
6	OO Support	Yes: <ul style="list-style-type: none"> • Classes & Objects • Every value or method or class instance is an object • Singleton for every class • No; Traits can be used to implement inheritance of multiple data model. • Parametric polymorphism • Work around for other polymorphisms 	Yes: <ul style="list-style-type: none"> • Classes & Objects (like C++) • Yes, like C++, you can use multiple inheritances from more than one base class. • Add hot polymorphism or Operator overloading with RT binding • Uniquely dynamic
7	Functional	Strong Support	Weak Support

	Programming		
8	Procedural Coding	More syntax and type declarations required compared to Python	Very intuitive and readable
9	3rd Party Lib support	Strong	Stronger
10	Multi-threading Support	Yes	Yes