perplexity

in a table form

Perfect ✅
Here’s the information presented in a **clear comparative table**:

Types of Programming Languages

Category	Definition	Examples	Key Features
Interpreted Languages	Executed line by line by an interpreter without prior compilation.	Python, Ruby, JavaScript	Easy debugging, portable, slower than compiled languages.
Functional Languages	Based on mathematical functions, recursion, immutability, higher-order funcs.	Haskell, Lisp, Erlang	Declarative style, fewer side-effects, focuses on "what".
Compiled Languages	Source code compiled into machine/bytecode before execution.	C, C++, Rust	Fast execution, optimized performance.
Procedural Languages	Organized around step-by-step procedures or functions.	C, Pascal, Fortran	Structured, modular, sequential execution.
Scripting Languages	Interpreted; mainly for automation, glue code, and small tasks.	Python, Perl, Ruby, Bash	Quick prototyping, task automation, lightweight.
Markup Languages	Define structure, meaning, and presentation of data (not full programming).	HTML, XML, JSON	Data representation, widely used in web/data exchange.
Concurrency-Oriented	Designed for handling simultaneous tasks/processes efficiently.	Go (Golang), Erlang	Concurrency, lightweight threads, scalability.
Object-Oriented (OOP)	Organized around classes/objects with encapsulation, inheritance, polymorphism.	Java, C++, C#, Python	Reusability, modularity, abstraction.

▯ Would you like me to also add a **Venn diagram-style overview** showing how some languages (like Python, Erlang, C++) fall into multiple categories at once?