MOL is the first language with native |> pipeline operators and automatic execution tracing — built for AI/RAG pipelines, cognitive computing, and data processing. Created by **Mounesh Kodi** for **IntraMind** at **CruxLabx**.

90+	8	33	68	2
Stdlib Functions	Domain Types	AST Nodes	Tests Passing	Transpile Targets

The Killer Feature: |> Auto-Tracing

```
-- Full RAG pipeline in ONE expression
let doc be Document("notes.txt", "MOL is...")
doc |> chunk(512) |> embed |> store("index")
-- Auto output (zero config):
  0. input  — Document("notes.txt")
  1. chunk  0.1ms → List<5 Chunks>
  2. embed  0.2ms → List<5 Embeddings>
  3. store  0.0ms → VectorStore
```

Why MOL?

Problem	Python/JS	MOL
Debugging	`print()`	|> auto-traces
Data flow	No pipes	|> left-to-right
AI types	Generic dicts	Native types
RAG setup	50+ lines	One expression
Safety	None	`guard` + `access`

Language Syntax

```
let name be "IntraMind"
let score : Number be 42
define greet(person)
  return "Hello, " + person
end
pipeline clean(data)
  return data |> trim |> lower
end
guard confidence > 0.8 : "Too low"
let doubled be map([1,2,3], double)
```

Domain Types

Type	Cat.	Constructor
Thought	Core	`Thought("idea", 0.9)`
Memory	Core	`Memory("key", value)`
Node	Core	`Node("label", 0.5)`
Stream	Core	`Stream("feed")`
Document	RAG	`Document("file", "text")`
Chunk	RAG	`Chunk("text", 0, "src")`
Embedding	RAG	`Embedding("text", "model")`
VectorStore	RAG	*via* `store()`

Standard Library (90+ Functions)

Category	Functions
General	len, type_of, to_text, to_number, range, abs, round, sqrt, max, min, sum
Functional	map, filter, reduce, flatten, unique, zip, enumerate, find, take, drop, group_by, every, some
Math	floor, ceil, log, sin, cos, tan, pi, e, pow, clamp, lerp
Statistics	mean, median, stdev, variance, percentile
Collections	sort, sort_by, sort_desc, binary_search, reverse, push, pop, keys, values, contains, join, slice
Strings	split, upper, lower, trim, replace, starts_with, ends_with, pad_left, pad_right, format
Crypto	hash, uuid, base64_encode, base64_decode
Random	random, random_int, shuffle, sample, choice
Maps	merge, pick, omit
Types	is_null, is_number, is_text, is_list, is_map
RAG	chunk, embed, store, retrieve, cosine_sim, think, recall, classify, summarize
Debug	display, tap, inspect, to_json, from_json

CLI & Tooling

```
mol run file.mol
mol run file.mol --no-trace
mol parse file.mol                          show AST
mol transpile file.mol -t python
mol transpile file.mol -t js
mol repl                                    interactive
mol version
```

VS Code Extension included — syntax highlighting, 20+ snippets, code folding for all block structures.

Version History & Roadmap

Ver.	Date	Status	Highlights
0.1.0	02-08	Done	Grammar, AST, interpreter, 4 types, CLI
0.2.0	02-09	Done	|> auto-trace, guard, RAG types
0.3.0	02-10	Done	42 new algorithms, 90+ stdlib
0.4.0	—	Next	Sovereign AI, agent blocks
0.5.0	—	Plan	Async pipelines, HTTP server
1.0.0	—	Vision	Package manager, cloud deploy