

# RIFTLang

A programming language for governance-first computation. Code how you live - with care, with autonomy, with clarity.

## ?? What Is RIFT?

RIFT is not just a language. It's a **governance ecosystem**.

Born from failure - medical hardware that couldn't be trusted, developers who weren't listened to, and systems that forgot to care - RIFTLang is a new computing philosophy designed to ensure that what runs **is safe**, what's written **is intentional**, and what's stored **is yours**.

RIFT is defined by one idea: > “*We compile what has been governed.*”

## ?? Core Concepts

- `.rift` files define policies that **govern compilation**.
- Components can be written in any language, but **linked through Gossi drivers**.
- RIFT follows the **RIFTer's Way**: a manifesto of intentional design, respectful concurrency, and thread-safe logic.

## ?? Key Principles

- **Interference**: Allocate bits before you compute. No ambiguity.
- **Substitution**: Encode without mutation. Wrap meaning with intent.
- **Entanglement**: Synchronize buffers like shared memory across dimensions.
- **Superposition**: Mount entropy to memory. Register it for reuse.

Every file is a breath. Every token, a truth.

## ?? Components

- `LibRIFT` - Pattern matching and type wrapping
- `NLINK` - Policy-aware linker
- `GossiLang` - Gossip-powered polyglot binding
- `Rift.exe` / `LRift.so` - The compiler, the enforcer

## ?? Import(disk)

Import disk - not data, but **meaning**.

Resuming a session in RIFT restores **context**, not just memory. It's a declarative restoration of intention. You don't just load a file - you restore a breath you left behind.

## **?? Documentation**

- The RIFTer's Way Manifesto
- Base Shift Allocator Spec
- Quantum Interference System

## **?? Status**

??? Early development. Structure in place. Meaning under construction.

## **?? License**

MIT - because the code should be free, but never reckless.

---