Jennifer resources



The interpreter, the tools, the decks, and where each one lives.

Based on **Jennifer 0.26.0**, November 2026, by @mplx.

The map

What	Where
The language	jennifer-lang.dev
Documentation	docs.jennifer-lang.dev
Source and releases	github.com/jennifer-language
Decks	registry.jennifer-lang.dev
The deck manager	jvc.jennifer-lang.dev
Community decks	github.com/deckyard
Talks	academy.jennifer-lang.dev

The interpreter

One static binary: no runtime, no system libraries, copy it and run it.

Runs on Linux (amd64, arm64), OpenWrt routers (`.ipk`) and Android in Termux; macOS and Windows builds are best-effort.

Two builds from the same source:

- `jennifer`: the full build, multi-core `spawn`, network, processes, SQL
- `jennifer-tiny`: the TinyGo build, small and embeddable

43 built-in libraries and 80 shipped modules, over 2,300 functions.

```
sudo apt install ./jennifer_0.26.0_amd64.deb   # AUR: jennifer-bin
docker run --rm -it ghcr.io/jennifer-language/jennifer repl
```

jvc: the package & app manager

```
jvc add "@deckyard/epub"           # require a deck
jvc install                        # vendor exactly what camcorder.lock
pins
jvc app install https://github.com/jennifer-language/grimoire
jvc publish                        # tag, check, and announce your own
deck

import "@deckyard/epub/" as epub;
```

The public package registry

registry.jennifer-lang.dev: 65+ decks, searchable, each with its versions, documentation and repository.

Scope	Who	Examples
@jennifer	the project	telegram, influxdb, oauthext, mikrotik
@deckyard	the curated community	epub, geo, sheets, proxmox, openai
your own	anyone with a repository	claim a scope with jvc claim

Deckyard: community-based, but curated

A deck joins `@deckyard` only after review, and only while it keeps up:

- a named maintainer who uses what the deck wraps
- tests that run in CI without accounts or network
- documentation checked against the code
- strict input validation, typed errors, a cap and a deadline on every read
- the shared CI layer, so one fix reaches every deck

github.com/deckyard: API clients, file formats, science, home infrastructure.

Apps written in Jennifer

App	What it does
grimoire	Markdown to a documentation site and a PDF; builds every deck's book
cantrip	Markdown to slides: an HTML slideshow and a PDF (this deck)
jvc	the deck manager
registry	the deck registry server and its operator tool
ocsaml	Cisco AnyConnect VPN with SAML / SSO login

Grimoire plugins: grimoire-plugin on GitHub.

Demos

Demo	What it shows
demo-jsnake	a multiplayer snake game for the Linux console, over UDP and TCP, with CPU opponents
demo-micrograd	a scalar reverse-mode autodiff engine, built up in five steps

Start here

```
jennifer repl                                        # try the language
jvc app install https://github.com/jennifer-language/grimoire
jvc add "@deckyard/geo" && jvc install               # your first deck
```

- Learn: docs.jennifer-lang.dev
- Learn from talks: academy.jennifer-lang.dev
- Find a deck: registry.jennifer-lang.dev
- Share one: `jvc publish`