Contributing to Picca

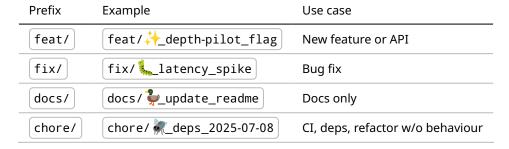
Thank you for taking the time to contribute!

We follow a **lightweight**, **automation-first** workflow so that Codex-1, human developers, and CI bots all speak the same language.

0. Quick Start (TL;DR)

Tip for Codex-1 Use the make dev task to spin up a local stack identical to Cloud Run.

1. Branch Naming



Emoji is optional but encouraged. Keep slug short & kebab-case.

2. Commit Message Format

We use Conventional Commits. Example:

```
feat(api): add squat-jump symmetry metric
```

This feeds automated changelogs & semantic versioning.

3. Pre-commit Hooks

Installed via pre-commit:

Hook	Purpose
gofumpt	Go formatter
black / isort	Python style & import
eslint	Next.js lint
markdownlint	Docs style
detect-secrets	Prevent key leakage
commitlint	Enforce commit style

Run automatically; or manually with:

```
$ pre-commit run --all-files
```

4. Tests & Lint

```
$ make lint # Go + Python + TS
$ make test # go test ./... && pytest
```

CI (Cloud Build) blocks merge if either target fails.

5. Pull Request Checklist

-

At least 1 core maintainer review required. Draft-PRs welcome.

6. Issue & RFC Process

- **Bug / Feature** → .github/ISSUE_TEMPLATE/*
- Large design → docs/RFC-<topic>.md (Google-style RFC)
- Tag with size/{XS,S,M,L} for triage.

7. Code of Conduct

We adopt the Contributor Covenant v2.1. See CODE_OF_CONDUCT.md

8. License Grant

By submitting code, you agree it is licensed under the **Apache 2.0 License**, and you have the right to do so.

9. Local Dev Cheatsheet

- # Rebuild AI model locally
 \$ make ai-build
- # Seed sample run
- \$ scripts/dev_sample.sh

Happy hacking!

Last updated 2025-07-08