# Swindle

Version 9.4.0.5

November 9, 2026

`#lang swindle` package: `swindle`

Swindle extends Racket with many additional features. The main feature that started this project is a CLOS-like object system based on Tiny-CLOS from Xerox, but there is a lot more.

Some documentation is available at http://barzilay.org/Swindle/.

# 1 Features

The following is a high-level description of major features provided by Swindle. For every feature, the file that provides it is specified, if only a subset of the system is needed.

- Some basic syntax extensions, including lambda &-keywords, and improved `define` and `let` forms. (Available separately using `swindle/base`)
- Generic setters with `set!`, additional useful mutation forms: `pset!`, `shift!`, `rotate!`, and some simple ones like `inc!`, and `push!`. (Available separately using `swindle/setf`, where the names `setf!` and `psetf!` are used to avoid changing the Racket form)
- Easy macro-defining macros — simple `syntax-rules` macros with `defsubst`, and a generic `defmacro` utility, all with a local `let...` form, and extended to easily create symbol macros. (`swindle/misc`)
- A `collect` macro that provides very sophisticated list comprehensions and much more. (`swindle/misc`)
- An `echo` mechanism which is an alternative to using format strings, and contains many useful features including a list iteration construct, and is easy to extend. (`swindle/misc`)
- A `regexp-case` syntax which is similar to a `case` on strings with easy access to submatches. (`swindle/misc`)
- A CLOS-like object system – based on Tiny CLOS, but with many extensions that bring it much closer to CLOS, and heavily optimized. Some added features include singleton and struct classes, applicable stand-alone methods, method-combination, and some MOP extensions. (Available without syntax bindings in `swindle/tiny-clos`)
- Good integration with the Racket implementation: primitive values have corresponding Swindle classes, and struct types can also be used as type specializers. A Swindle class will be made when needed, and it will reflect the struct hierarchy. In addition, structs can be defined with a Swindle-line `defstruct` syntax which will also make it possible to create these structs with `make` using keyword arguments. (`swindle/tiny-clos` and `swindle/extra`)
- Many hairy macros that make the object system much more convenient (CLOS has also a lot of macro code). Some of the macros (especially `defclass`) can be customized. (`swindle/clos`)
- Useful generic functions, including `print-object` which is used to display all objects. (`swindle/extra`)
- A `match` mechanism with a generic-like interface. (`swindle/extra`)

- The fun `amb` toy. (`swindle/extra`)
- A language that can easily create HTML, where the result is human-editable. (`swindle/html`)
- Customizable syntax: easy to add customized languages to DrRacket. (`custom`)

# 2 Libraries

Files marked with “module” provide a module by the same name, files marked with "language module" modify the language and should be used as an initial import for other modules. Most files (and especially all language modules) are useful by themselves, even without using the whole Swindle environment.

- `swindle/base` (language module) — Basic syntax extensions, mainly Lisp-like lambda argument &-keywords.
- `swindle/setf` (module) — Generic setters similar to `setf` in Lisp, and a few more useful macros.
- `swindle/misc` (module) — Lots of useful functionality bits, including everything from frequently useful Racket legacy libraries (`mzlib/list`, `mzlib/etc`, and `mzlib/string`).
- `swindle/turbo` (language module) — A module that packages functionality from `swindle/base`, `swindle/setf` (overriding `set!` with `setf!`), and `swindle/misc`.
- `swindle/tiny-clos` (module) — The core object system, based on Tiny CLOS from Xerox, but heavily modified, optimized and extended.
- `swindle/clos` (module) — Convenient macro wrappers for `swindle/tiny-clos`.
- `swindle/extra` (module) — Extra functionality on top of `swindle/clos`.
- `swindle/swindle` (language module) — The main Swindle environment module: packages `swindle/tiny-clos`, `swindle/clos`, and `swindle/extra` on top of `swindle/turbo`, and some more general definitions.
- `swindle/info` (module) — Compilation definitions.
- `swindle/tool` (module) — Setup for Swindle in DrRacket: makes some languages available in DrRacket, including custom Swindle-based languages.
- `swindle/custom` (module) — A sample file that demonstrates how to create a Swindle-based customized language; see the source for instructions.
- `swindle/html` (module) — A language for creating HTML.