write the prefix keybinding for
`cljr-add-keybindings-with-prefix` here

clj-refactor cheatsheet *Namespaces*

ai

add `:import`

- add an `:import` to `ns` and puts cursor there to type name
- hit *Tab* to move back to where you were

am

add a missing `:import`/`:require`

- uses *symbol at point*
- gives you a choice when it's ambiguous

ar

add `:require`

- multiple tab stops

mf

move form(s) to another namespace

- uses *toplevel form(s) at point*
- choose file
- adds `:require` `:refer` if needed

rd

remove debug function calls

- `println`, `pr`, `prn`

rr

remove all unused `:require` statements

ru

replace all `:use` with `:require` `:refer` `:all`

sn

sort `ns` form

sr

stop referring

- remove `:refer` *clause at point*
- replaces uses with alias/namespace prefix

write the prefix keybinding for
`cljr-add-keybindings-with-prefix` here

clj-refactor cheatsheet *Toplevel*

as
add method stubs
- uses *interface or protocol at point*

cp
cycle privacy
- uses *toplevel* `def`/`defn` *at point*

dk
destructure keys
- uses *argument at point*

ef
extract `defn`
- uses *form at point*
- finds and replaces other occurrences

fe
create `defn` from example
- uses *function call at point*

fu
find usages
- uses *symbol at point*

is
inline symbol
- uses *symbol at point*

pf
promote function
- `#()` at point => `fn`
- `fn` at point => `defn`

rs
rename symbol
- uses *symbol at point*

write the prefix keybinding for
`cljr-add-keybindings-with-prefix` here

clj-refactor cheatsheet *Expressions*

cc
cycle collection type
- uses *surrounding collection at point*

ci
cycle between `if` and `if-not`
- uses nearest *surrounding `if` at point*

el
extract `let`
- uses *surrounding `let` at point*
- moves `let` up one level

il
introduce `let`
- uses *form at point*
- makes it the value of a binding in a new `let`

ml
move to `let`
- uses *form at point*
- moves it to nearest surrounding `let` binding

rl
remove let
- uses *surrounding `let` at point*
- inlines bound values

tf
thread first
- uses *form at point*

th
thread one form
- uses *form at point*

tl
thread last
- uses *form at point*

ua
unwind threaded expression
- uses *form at point*

uw
unwind one form
- uses *form at point*

write the prefix keybinding for `cljr-add-keybindings-with-prefix` here

_____ clj-refactor cheatsheet *Project*

ap

add and hotload dependency

- autocompletes from Clojars
- autocompletes version string

hd

hotload dependency

- uses *dependency at point*

pc

project clean

- goes through all namespaces and `project.clj`
- remove unused `:require`
- sort `ns`
- sort project dependencies

rf

rename file or directory

- renames `ns`
- renames references

sp

sort project dependencies

up

update project dependencies