#!/usr/bin/env hy

(import os.path)

(import hy.compiler)

(import hy.core)

;; absolute path for Hy core

(setv \*core-path\* (os.path.dirname hy.core.--file--))

(defn collect-macros [collected-names opened-file]

(while True

(try

(let [data (read opened-file)]

(if (and (in (first data)

'(defmacro defmacro/g! defn))

(not (.startswith (second data) "\_")))

(.add collected-names (second data))))

(except [e EOFError] (break)))))

(defmacro core-file [filename]

`(open (os.path.join \*core-path\* ~filename)))

(defmacro contrib-file [filename]

`(open (os.path.join \*core-path\* ".." "contrib" ~filename)))

(defn collect-core-names []

(doto (set)

(.update hy.core.language.\*exports\*)

(.update hy.core.shadow.\*exports\*)

(collect-macros (core-file "macros.hy"))

(collect-macros (core-file "bootstrap.hy"))))