

> (+ (let ((x 10)) x) 3)

13 [actually eval'd to 26, print the answer]

> (+ true 1)

Invalid argument [should not panic Rust] → IDEAS

> (define x 100)

> (+ x 3)

103

> (set! x 500)

When we see (set! x ...) in a repl entry, promote x to a "variable" that varies

Id(x)

if in env ⇒ ...

if in def-env ⇒ ...

1. catch-unwind panic from throw-error (snek-error)

2. print + return from snek-error-repl

snek-error-repl()  
jitted-fn()  
repl()

enum DefEnvElement

| Imm(i64)

| Var(&mut i64)

program of concern:

(loop

...  
(set! x (+ x true))  
...)

| Set(x, expr) ⇒

Vec<String> find-variables(expr: &Expr)

mov

, rax