

SCHEME's **eval** function

scheme

Scheme saved on Sunday November 21, 1993 at 9:15:23 PM

Release 7.3.1

Microcode 11.146

Runtime 14.166

1]=>(define u 7)

;Value: u

1]=>u

;Value: 7

1]=>'u

;Value: u

1]=>(eval 'u (nearest-repl/environment))

;Value: 7

[The second argument to eval is an environment. The value of the expression "(nearest-repl/environment)" is exactly the current environment in which the read-eval-print loop is executing.]

1]=>x

;Unbound variable: x

;To continue, call RESTART ...

2 error>

;Quit!

```
1 ]=>'x;
```

```
Value: x
```

```
1 ]=>(eval 'x (nearest-repl/environment) )
```

```
;Unbound variable: x
```

```
;To continue, call RESTART...
```

```
2 error>
```

```
;Quit!
```

```
1 ]=>(eval ``x (nearest-repl/environment) )
```

```
Value: x
```

```
(define eval_experiment (lambda (z)
  (eval z (nearest-repl/environment)) ))
```

```
;Value: eval_experiment
```

```
1 ]=> (eval_experiment 'u)
```

```
;Value: 7
```

```
1 ]=>(eval_experiment u)
```

```
;Value: 7
```

```
1 ]=>(eval_experiment ``u)
```

```
;Value: u
```