Lisp Interpreter in Python 3.4

Concepts of Modern Programming Languages

Maria Florusß

August 18, 2014

Overall Structure

Lisp Objects

TODO

Functionality

Builtin Syntax

define

lambda

if

set!

let

begin

quote

and

or

Builtin Functions

Arithmetic

add

Description:	Adds an arbitrary amount of numbers and returns the accumulated value as `SchemeNumber`. If only one argument is given, the arguments value is returned as `SchemeNumber`. If no argument is given the return value is 0.
Symbol:	`+`
Arguments:	0+ `SchemeNumber`s

Example of usage:

```
> (+ 1 2)
3
> (+ 2 3 4)
9
> (+)
0
> (+ 42)
42
```

subtract

Description:	Subtracts an arbitrary amount of numbers from the first number and returns the accumulated value as `SchemeNumber`. If only one argument is given, the arguments value is negated and returned as `SchemeNumber`. If no argument is given an ArgumentCountException is risen.
Symbol:	-
Arguments:	1+ `SchemeNumber`s

Example of usage:

```
> (- 0.5 2)
-1.5
> (- 10 3 4)
3
> (-)
ArgumentCountException: 'function - expects at least 1 argument.'
> (- 42)
-42
```

multiply

Description:	Multiplies an arbitrary amount of numbers and returns the resulting value as `SchemeNumber`. If only one argument is given, the arguments value is returned as `SchemeNumber`. If no argument is given the return value is 1.
Symbol:	`*`
Arguments:	0+ `SchemeNumber`s

Example of usage:

```
> (* 3.5 4)
14.0
> (* 2 3 4)
24
>(*)
1
> (* 42)
42
```

divide

Description:	Divides the first argument by the second, the result by the third and so on. If only one argument is given, the result is 1 devided by the argument. If no argument is given an ArgumentCountException is risen.
Symbol:	`/`
Arguments:	1+ `SchemeNumber`s

Example of usage:

```
> (- 0.5 2)
-1.5
> (/ 12 3 2)
2.0
> (/)
ArgumentCountException: 'function - expects at least 1 argument.'
> (/ 3)
```

```
8 0.3333333333333333
```

arithmetic equals

modulo