

Mini_lisp 拓展功能

本 Mini_lisp 解释器主要按照 Mini_lisp 标准，同时对部分功能有修改，并添加了一些新的功能。关于标准文档，详见[标准文档](#)。

与 Mini_lisp 标准不同的地方：

append 内置过程

本 Mini_Lisp 解释器实现的 append 内置过程允许不接受参数，这将返回空表。

```
1 | (append) ; 返回 ()
```

允许接受若干个列表，将其中的元素按顺序拼接成一个新的列表。

```
1 | (append '(1 2 3) '(a b c) '(foo bar baz)) ; 返回 '(1 2 3 a b c foo bar baz)
```

允许最后一个元素不是列表，将返回对子。

```
1 | (append '(1 2 3) 4) ; 将返回 '(1 2 3 . 4)
```

关于文件模式

本 Mini_Lisp 解释器支持对文本进行解释，需要在后面接上文件路径，即可对文件进行解释。

```
1 | mini_lisp C:/path/to/the/test.scm
```

文件模式同样支持多行输入，同时允许空行和注释。

Lv7+ 介绍

1. 增加了多行输入和换行缩进

支持多行输入，如果解释器接受的输入尚未完成，将会支持在下一行继续输入，而不是报错。比如这样的输入

```
1 | >>> (+  
2 | ... 1 2)
```

同时还会自动识别缩进。

2. 增加了 `read` 内置过程

本 `Mini_Lisp` 解释器支持从标准输入读取输入，并将输入缓存，如果缓存仍有，将直接从缓存读取，否则要求用户重新输入。比如这样：

```
1 | (+ (read) 2) ; 将会等待用户输入
2 | ; 如果用户输入 1, 将直接返回3
3 | (read)
4 | ; 如果输入 1 2 3, 那将会返回1, 并将2 3读入缓存
5 | (+ (read) (read))
6 | ; 将直接从缓存读取输入2 和 3, 也就是将返回 5
```

3. 增加了有理数及其运算

本 `Mini_Lisp` 解释器支持读入、输出有理数以及进行有理数的简单运算，包括 加法、减法、乘法等。

```
1 | (define x 1/2) ; 将 x 定义为 1/2, 并返回空表
2 | (+ 1/2 3) ; 将返回 7/2
3 | (+ 1/2 2/3) ; 将返回 7/6
4 | (+ 1/2 3/2) ; 将返回 2
5 | (+ 1/3 0.4) ; 将返回 1.06667
```

4. 增加了 `when` 和 `unless` 特殊形式

本 `Mini_Lisp` 解释器增加了 `when` 和 `unless` 特殊形式。

1. `when` 特殊形式

使用方法为：

```
1 | (when <子句1> <子句2> ...)
```

如果子句1为 `true`，则依次执行后面的子句，否则不执行。

2. `unless` 特殊形式

使用方法为：

```
1 | (unless <子句1> <子句2> ...)
```

如果子句1为 `false`，则依次执行后面的子句，否则不执行。

5. 文件模式下的错误定位

文件模式下，如果文件中某一条语句出现错误，将会在报错中指出大致的范围，报错方式：

```
1 | Error in line 3 to line 5 : Unimplement
```

6. 类型转换函数

本 `Mini_Lisp` 解释器实现了部分的类型转换的内置过程，包括 `number->string` 等。使用时在后面加上需要转换的表达式。比如

```
1 | (number->string 1) ; 返回 "1"  
2 | (number->integer 2.3) ; 返回 2
```