Scheme 101

Jackson

January 31, 2022

Contents

1 Basic functions

- (+ 1 2 3)
- =6
- Functions are a list where the first elements in each list "()" is a function
- So for the first example "+" is the function we are calling, and "1", "2", "3" are our arguments
- Some examples of basic operators are "+, -, /, *, etc"

1.1 Using multiple functions inline

- Since everything is Scheme is a list we can call functions inside of functions, and these inner functions evaluate to a return value that is then passed to the next function out
- (sqrt (+ (* 5 5) (* 12 12)))
- 13
- Let's break this example down
  - There are two ways to think about this single line:

    - Inside-out, and Outside-in
  - Let's start with Inside-out as this will help us view the function left-to-right
    1. (sqrt ...)
       - sqrt takes an argument and returns its square root
    2. (+ ...)
       - + is the addition opperator and it will add all arguments together
    3. (* ...)
       - * is the multiplication opperator that will multiply all arguments together and return the output
  - So our line of code will return the square root of the result of the addition of the product of (5*5) and (12*12)
    - Seems complicated when you look at it that way, so lets take a look from the inside out
  - From the inside-out we can start by finding the inner-most argument, which in out case is the second multiplication opperator
    1. (* 12 12)
       - = 144
    2. Next we can perform next operation outwards, (* 5 5)
       - = 25
    3. Now that we have performed both of the operations inside the inntermost function we can perform that function, (+ 25 144)
       - = 169
    4. So then we move outward and we can see we are at the outermost operation being sqrt, sqrt take exactly one argument being our number 169, (sqrt 169)
       - = 13

```
(sqrt (+ (* 5 5) (* 5 5)))
```

1.1.1 Side Note on Creating a List

- Since Scheme uses lists for both function and data we need a way to tell it which is which. For this we prefix our list with a singlr quote '()

- Notice that we only use a beginning quote, and there is no need for an ending quote mark.

```
(* 4 5)
= 20

'(* 4 5)
=(* 4 5)
```