1.函数定义

func myfunc1(x,y) x+y\*sin(x\*y)

func myfunc2() 2+PI

func myfunc3(x, y) x+y

func myfunc4(\*) //这个需要支持自定义宏才可以（就是语句块，和一些顺序、分支、循环等结构）

func myfunc5(x,y,\*) //这个需要支持自定义语句才可以，但是可以限定最少的参数的类型

2.普通变量

def x,y,z=5,6

def i,j,k=x+5,y+3

3.计算

x+y

x+sin(x+y)

1+2

如何解析他们呢？

1.函数：

1.1特征

关键字1：函数的唯一标识 myfunc：这个主要是再解析表达式的时需要

关键字2：函数的签名 myfunc|3x表示一个需要至少三个的函数，这个用来进行函数的唯一性检查、以及动态抉择函数匹配;myfunc|4：只能是四个参数

1.2自定义函数的匹配的参数：需要再构造的时候根据参数的个数选择匹配的重载函数

1.3内置函数的参数匹配：这个再内部进行计算，但是也需要根据参数的种类选择合适的方法计算。

2.变量的管理：

2.1我觉得需要编写一个管理变量的类。目前可以借助栈这种结构模拟计算机的函数调用，因为我这个是C语言式的，主要的元素是函数。所以可以通过模拟栈：

意淫一下交互式的界面

>>func my(x,y) x+y-2

>>Okay, function defined successfully.

>>def x,y,z=1,2

>>Okay, variables defined successfully.

>>func(x,y)

>>result=1

>>func(2,z)

>>Error(s) occurred: uninitialized variable(s): z.

>>i+j+k

>>Please specify the parameters for i+j+k

>>i=25/

>>j=24/

>>k=0/

>>result=49

>>func(uu,ww)

>>Please specify the parameters for func(uu,ww)

>>uu=1

>>ww=myfunc(2,5)

>>result=4

***\*图中省略了函数定义的内容***

2.2.作用域的内容：

2.2.1：作用域相当于一个容器，就目前而言，只有main的作用域和他的子作用域（函数定义的空间）

如果再子函数的作用域中找不到变量的定义，那么就需要想行查找（由于目前只做了main的作用域，所以可以直接通过检查const是否存在即可，main保存的变量都是consts。）在函数的作用域中（这个过程在构建表达式式树中实例，如果以后支持自定义宏的话，就可以和main的作用域保持一致了），检查顺序：参数->main的consts

对于main的作用域，包含了x,y,z三个consts

做哟

**作用域定义的相对的“内存地址“？**

3.普通表达式的计算，需要提前进行参数检查，确保所有的参数能被初始化（初始化Present类型），放入临时函数定义的map中(方便解析的时候检查参数，有可能适当调整构建树的方式，以提高性能，也不需要产生很多的标识信息（利用作用域定义的相对的“内存地址“？）)