

```

Welcome to Berkeley Logo version 6.2.4
? print 2 + 3
5
? print 2+3
5
? print sum 2 3
5
? print (sum 2 3 4 5)
14
? print sum 2 3 4 5
5
You don't say what to do with 4
? 2+3
You don't say what to do with 5
? print "yesterday
yesterday
? print "julia"
julia"
? print revolution
I don't know how to revolution
? print [blue jay way]
blue jay way
? show [eight days a week]
[eight days a week]
? show first [golden slumbers]
golden
? print first bf [she loves you]
loves
? pr first first bf [yellow submarine]
s
? to second :stuff
> output first bf :stuff
> end
second defined
? second "something
You don't say what to do with o
? print second "piggies
i
? pr second [another girl]
girl
? pr first second [carry that weight]
t
? pr second second [i dig a pony]
i
? to pr2nd :thing
> print first bf :thing
> end
pr2nd defined
? pr2nd [the 1 after 909]
1
? print first pr2nd [hey jude]
jude
pr2nd didn't output to first
? repeat 5 [print [this boy]]
this boy
this boy
this boy
this boy

```

```

this boy
? if 3 = 1+1 [print [the fool on the hill]]
? print ifelse 2=1+1 ~
~ [second [your mother should know]] ~
~ [first "help]
mother
? print ifelse 3=1+2 ~
~ [strawberry fields forever] ~
~ [penny lane]
I don't know how to fields
? print ifelse 4=1+2 ~
~ ["flying] ~
~ [[all you need is love]]
all you need is love
? to greet :person
> say [how are you,]
> end
greet defined
? to say :saying
> print sentence :saying :person
> end
say defined
? greet "ringo
how are you, ringo
? show map "first [paperback writer]
[p w]
? show map [word first ? last ?] ~
~ [lucy in the sky with diamonds]
[ly in te sy wh ds]
? to who :sent
> foreach [pete roger john keith] "describe
> end
who defined
? to describe :person
> print se :person :sent
> end
describe defined
? who [sells out]
pete sells out
roger sells out
john sells out
keith sells out
? print :bass
bass has no value
? make "bass "paul
? print :bass
paul
? print bass
paul
? to bass
bass is already defined
? output [johnny cymbal]
Can only use output inside a procedure
? end
I don't know how to end
? print bass
paul
? print :bass

```

```

paul
? print "bass
bass
? to countdown :num
> if :num=0 [print "blastoff stop]
> print :num
> countdown :num-1
> end
countdown defined
? countdown 5
5
4
3
2
1
blastoff
? to downup :word
> print :word
> if empty? bl :word [stop]
> downup bl :word
> print :word
> end
downup defined
? downup "rain
rain
rai
ra
r
ra
rai
rain
? ;;;; The following stuff will work
? ;;;; only on an X workstation:
? cs
? repeat 4 [forward 100 rt 90]
? cs
? repeat 10 [repeat 5 [fd 150 rt 144] rt 36]
? cs repeat 36 [repeat 4 [fd 100 rt 90]
~ setpc remainder pencolor+1 8
~ rt 10]
? to tree :size
> if :size > fd :size/2
> lt 30 tree :size*3/4 rt 30
> fd :size/3
> rt 45 tree :size*2/3 lt 45
> fd :size/6
> bk :size
> end
tree defined
? cs pu bk 100 pd ht tree 100
?

```