Arithmetic

Use IDLE or python shell

Currently IDLE shell doesn’t support JAWS very well, while the python shell does a better job.

**Python shell:**

Python 3.9.4 (tags/v3.9.4:1f2e308, Apr 6 2021, 13:40:21) [MSC v.1928 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> 2+2

4

>>> 3\*\*2

9

>>>>>> 10\*\*0

1

>>> 10\*\*1

10

>>> 10\*\*2

100

>>>

>>> 10\*\*3

1000

>>> 10\*\*6

1000000

>>> 10\*\*9

1000000000

>>> 10\*\*12

1000000000000

>>>

>>> str(10\*\*12)

'1000000000000'

>>> len(str(10\*\*12))

13

>>>

>>> len(str(10\*\*100))

101

>>>

>>> 2\*\*8

256

>>> 2\*\*16

65536

>>> 2\*\*32

4294967296

>>> 2\*\*64

18446744073709551616

>>> 0b11111111

255

>>> log10(2\*\*128)

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'log10' is not defined

>>>

>>> import math

>>> math.log10(2\*\*128)

38.53183944498959

>>>

>>> math.sqrt(625)

25.0

>>> help math

File "<stdin>", line 1

help math

^

SyntaxError: invalid syntax

>>> help("math")

Help on built-in module math:

NAME

math

DESCRIPTION

This module provides access to the mathematical functions

defined by the C standard.

FUNCTIONS

acos(x, /)

Return the arc cosine (measured in radians) of x.

… more than you want to show …

log(...)

log(x, [base=math.e])

Return the logarithm of x to the given base.

If the base not specified, returns the natural logarithm (base e) of x.

log10(x, /)

Return the base 10 logarithm of x.

… still more than you want to show …

>>> from math import \* # bring in all the parts

>>> log10(2\*\*128)

38.53183944498959

>>>

Buzz Lightyear…

>>> 60\*60\*24\*365 # How many seconds in a year

31536000

>>> sec\_in\_year=60\*60\*24\*365

>>> sec\_in\_year

31536000

>>> mile\_per\_sec=186000 # Approximate speed of light

>>> mile\_per\_sec/25000 # Times around the earth

7.44

>>> mile\_per\_sec\*sec\_in\_year # Miles in a lightyear

5865696000000

>>> mile\_per\_sec\*sec\_in\_year/10\*\*6 # Million

5865696.0

>>> mile\_per\_sec\*sec\_in\_year/10\*\*9 # Billion

5865.696

>>> mile\_per\_sec\*sec\_in\_year/10\*\*12 # Trillion

5.865696

>>>