Python as a Calculator (for Scalars)

Command	Meaning	Example
Arithmetic:		
`x + y, x - y, x * y, x / y`	$x+y$, $x-y$, xy, x/y	`7 / 3`
`( )`	grouping	`2 * 3 + 4, 2 * (3 + 4)`
`x ** y`	x^y (exponent)	`8 ** (1/3)`
`x // y`	integer division	`7 // 3`
`x % y`	modulo (remainder)	`7 % 3`
Calculator functions:		
`import numpy as np`	access `numpy` module code	`# call "import numpy" first`
`np.exp()`	exponential	`np.exp(1)`
`np.log(x)`	natural logarithm	`np.log(np.e ** 2)`
`np.cos(), np.sin()`	trigonometry	`np.sin(np.pi / 2)`
`np.sqrt()`	square root	`np.sqrt(9)`
Other easy functions:		
`np.fabs(x)`	absolute value	`np.fabs(-3)`
`np.floor(x)`	greatest integer $\leq$ x	`np.floor(-1.5)`
`np.ceil(x)`	ceiling: smallest integer $\geq$ x	`np.ceil(-1.5)`
`np.round(number, decimals=0)`	round to #decimal places	`np.round(4/3, 2)`
("=" indicates default)		`np.round(4/3)`
Miscellaneous:		
`help(name)`	check documentation	`help(np.round)`
`variable = value`	assign variable to value	`x = 3`
`variable_name`	`print(variable_name)`	`x`
`type(object)`	find object type	`type(2)`, `type(np.pi)`, `type('a')`
`#`	comment rest of line	`n = 3 # number of points`

A variable name begins with a letter; contains letters, digits and underscores; and is not in `help('keywords')`. By convention, use lowercase words separated by underscores for multiword variable names. e.g.

```
rate = 0.07
doubling_time = np.log(2) / np.log(1 + rate)
doubling_time
# in print(f'...'), {} gives value of enclosed variable
print(f'The rate is {rate} and doubling_time={doubling_time}.')
```

To learn more, see: https://numpy.org/doc/stable/reference/routines.math.html.