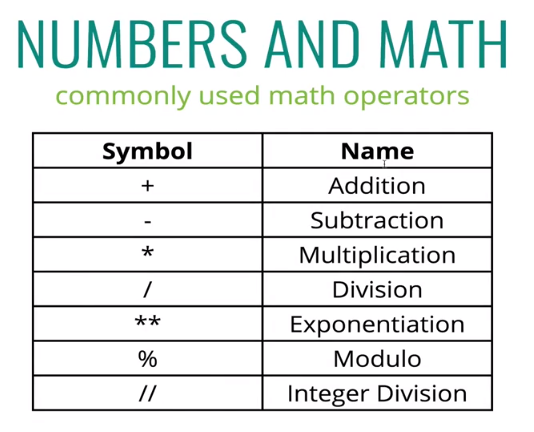
Section 6: Numbers, Operators and Comments

* How does the language represent data? One of those ways is numeric
  + Such as user age, shoes in stock, price for shoes, etc
* Python represents a number as an int (integer) or a float (floating point)
  + It’s all about saving space in memory as 1’s and 0’s
    - A decimal number may take up a lot more space than a whole number. That is because there is a finite number of whole numbers between any two given numbers, whereas there is an infinite number of decimal numbers between any two given numbers.
  + Integers are whole numbers, positive or negative
  + Floating points are also positive or negative, but have decimals after them (even if the digits after the decimal points are all zeroes
* You can use the type() function to discern the type of number you are working with
* In a mathematical operation in Python, if the operation includes a float, then the solution will automatically default to a float
* Common python mathematical operators
  + 
  + In Python, dividing any two numbers, regardless of whether they are a float or integer, the result in a float
    - Division ALWAYS returns a float in Python
    - Some languages don’t do this
  + Order of operations is standard: PEMDAS
* Comments – a weird one-off topic
  + Comments are a way for you to write little notes in your code without actually having it run or interpreted by Python
  + The single line comment is made using the hash or octothorp (#)
  + Comments should generally be used whenever or wherever it is useful to you
    - Oftentimes just looking at code is confusing or not self-explanatory
* Exponentiation (\*\*) raises a number to a given power
* Modulo ( % ) gives you the **remainder** after dividing the second number into the first number as many times as possible
  + 10 % 3 = 1
    - 3 goes into 10 nine times, with 1 a remainder of 1
  + Actually useful for figuring out if a number is even or odd ( number % 2 ) when you’re working with numbers that aren’t hard-coded in
* Integer division is done with ( // ). What makes it different from regular division is that it will return an integer, or the number of times the second number will go into the first number with no partials or decimals
  + The integer division “floors” the operation by rounding the answer down to the nearest integer. It does not ever round up