Basic Data Types. We begin by introducing some of the basic data types in Python. The first two types are the two different ways Python can represent numbers.

floating point number	think of this as a number with a decimal point i.e. 3.5
integer	an integer i.e. -78
string	a collection of characters enclosed in quotes i.e. "football"
boolean	True or False
list	an ordered collection of data types enclosed in square brackets i.e. [1,"cheese", 30, -2]

Definition. *A **variable** is a name given to a particular memory location in the computer. A variable can store any of the data types introduced above. We access the contents of a variable by referencing its name.*

Basic Commands. What follows is a collection of basic commands used extensively in Python.

(i) `x = y` - variable `x` is assigned the value of variable `y`.
(ii) `x == y` - tests whether variable `x` equals variable `y`, and outputs a Boolean value.
(iii) `# comment` - any text on a line begun with hash is ignored by Python, such annotations are called comments, and are added for the benefit of people reading the code, to explain what a section of code is doing.
(iv) `int("42")` - converts a string to an integer.
(v) `str(42)` - converts an integer to a string.
(vi) `print(x)` - prints the variable `x`.
(vii) `print("These are not the droids you are looking for.")` - prints a string.
(viii) `print("Red", x, "standing by.")` - prints a combination of types.
(ix) `answer = input("Have you accelerated to attack speed? ")` - asks the user for a string.
(x) `answer = int(input("How many Star Destroyers were there? "))` - asks the user for an integer.
(xi) `range(n)` - produces a sequence of numbers from `0` to `n-1`, typically used when iterating over a `for` loop.
(xii) `range(m,n,k)` - produces a sequence of numbers from `m` (inclusive) to `n` (not inclusive), with steps of size `k`.
(xiii) `list(range(n))` - produces an actual list of numbers from `0` to `n-1`.

Mathematical Operators. In Python we represent real numbers as either integers or as floating point numbers. Python can perform all of the typical arithmetic operations on these numbers as detailed below.

`+`	add
`-`	subtract
`*`	multiply
`/`	divide (normally)
`**`	exponentiate
`%`	modulo (gives the remainder)
`//`	divide (does not give the remainder)
`abs()`	absolute value

Note that integers are allowed to be as large as you wish (provided you have enough memory), but floating point numbers can only contain a limited number of digits (and so cannot be arbitrarily large, or arbitrarily small). Consider the two computations: `10**350` and `10.0**350`.

Comparative Operators. The following operators take two numbers and return a boolean value based on whether the resulting statement is True or False.

`==`	equal to
`!=`	not equal to
`>`	greater than
`<`	less than
`>=`	greater than or equal to
`<=`	less than or equal to

Boolean Operators. We also have the following boolean operators: `and`, `or`, `not`.