Chapter 2

Variables and arithmetic

Code Structure

Like many other programming languages, it is very important to pay close attention to the **structure** of your code when writing.

Structure largely refers to "syntax" used, which includes specific characters, spaces, tabs to convey exactly the operations python needs to perform.

For exmaple:

```
# This is a function calculates 1 + 2 + ... + n
#   using the formula sum = n(n+1)/2
def add_numbers(n):
  """Return 1 + 2 + ... + n."""
  return n * (n+1)//2   # The // operator means integer division
```

Reserved Words

There are a certain set of words that python **reserves** for itself. When python sees these keywords it runs specific predefined functions.

For example:

- `import` tells python to bring in code that has been written in a different file
- `def` defines a function that can be used elsewhere
- `while` instructes python to repeat code over and over while a condition is true

Some Keywords in python:

```
import, from, def, return, None, while, break, continue, pass, True, False, elif, and, not
```

Variables

A variable in programming is really a "named container". Another way to think of it is a named memory location.

A variable can **vary** over time, because it's a variable.

```
x = 3
x = 5
x = 2*x - 1
```

With each line above, the value contained in the variable x changes. First it's set to `3`, then `5`, and finally computes `2 times 5 minus 1` and stores the result into x

Variables

Variables can have different **types**, depending on what it is currently holding:

- an **int** is a number without a decimal (integer) `5`
- a **float** is a number with a floating point decimal `3.14159`
- a **string** is a string of characters, or a single character `"hello"`
- a **boolean** is a value which is either true or false

There are many other types in python. Python chooses the type for you, but you can override it's choice if you need to.

Arithmetic

Standard operators

Python uses the standard arithmetic operators that we have in algebra: `+ - * /`

Unlike in the syntax we use in math, the multiplication sign `*` must be used when multiplying.

Additional operators

Python, and other languages, also include additional operators for arithmetic:

- `%` is the modulo, it calculates the remainder after division (eg `17 % 3 --> 2`)
- `//` is the integer division (eg `7//2 --> 3`)
- `**` is for exponents, (eg `2**3 --> 8`)