Diving In

Datatypes. Set aside your first Python program for just a minute, and let's talk about datatypes. In Python, every value has a datatype, but you don't need to declare the datatype of variables. How does that work? Based on each variable's original assignment, Python figures out what type it is and keeps tracks of that internally.

Python has many native datatypes. Here are the important ones:

1. **Booleans** are either `True` or `False`.
2. **Numbers** can be integers (1 and 2), floats (1.1 and 1.2), fractions (1/2 and 2/3), or even complex numbers.
3. **Strings** are sequences of Unicode characters, e.g. an `HTML` document.
4. **Bytes** and **byte arrays**, e.g. a `JPEG` image file.
5. **Lists** are ordered sequences of values.
6. **Tuples** are ordered, immutable sequences of values.
7. **Sets** are unordered bags of values.
8. **Dictionaries** are unordered bags of key-value pairs.

Of course, there are more types than these. Everything is an object in Python, so there are types like *module*, *function*, *class*, *method*, *file*, and even *compiled code*. You've already seen some of these: modules have names, functions have `docstrings`, &c. You'll learn about classes in Classes & Iterators, and about files in Files.

Strings and bytes are important enough — and complicated enough — that they get their own chapter. Let's look at the others first.