



3. OBJECT-ORIENTED PROGRAMMING

In *object-oriented programming* we break a large and potentially complex program into a set of interacting elements, or *objects*. The idea originates from modeling (in software) real-world entities such as electrons, people, buildings, or solar systems and readily extends to modeling abstract entities such as bits, numbers, colors, images, or programs.

As discussed in Section 1.2, a data type is a set of values and a set of operations defined on those values. In Python, the values and operations for many data types such as `int` and `float` are predefined. In object-oriented programming, we compose code to define new data types.

This ability to define new data types and to manipulate objects holding data-type values is also known as *data abstraction*, and leads us to a style of modular programming that naturally extends the function abstraction style that was the basis for Chapter 2. A data type allows us to isolate data as well as functions.

- [3.1 Data Types](#) describes how to use existing data types, for text processing image processing.
- [3.2 Creating Data Types](#) describes how to create user-defined data types using Python's class mechanism.
- [3.3 Designing Data Types](#) considers important techniques for designing data types, emphasizing APIs, encapsulation, immutability, and design-by-contract.
- [3.4 Case Study: N-Body Simulation](#) presents a case study that simulates the motion of n particles, subject to Newton's laws of gravity.

Python Programs in this Chapter

Below is a list of Python programs and data files used in this chapter.

REF	PROGRAM	DESCRIPTION	DATA
3.1.1	potentialgene.py	potential gene identification	–
3.1.2	chargeclient.py	charged particle client	–
3.1.3	alberssquares.py	Albers squares	–
3.1.4	luminance.py	luminance library	–
3.1.5	grayscale.py	converting color to grayscale	mandrill.jpg mandrill.png darwin.jpg darwin.png
3.1.6	scale.py	image scaling	mandrill.jpg mandrill.png darwin.jpg darwin.png
3.1.7	fade.py	fade effect	mandrill.jpg mandrill.png darwin.jpg darwin.png
3.1.8	potential.py	visualizing electric potential	charges.txt
3.1.9	cat.py	concatenating files	in1.txt in2.txt
3.1.10	stockquote.py	screen scraping for stock quotes	–
3.1.11	split.py	splitting a file	djia.csv
3.2.1	charge.py	charged-particle data type	–
3.2.2	stopwatch.py	stopwatch data type	–
3.2.3	histogram.py	histogram data type	–
3.2.4	turtle.py	turtle graphics data type	–
3.2.5	koch.py	Koch curve	–
3.2.6	spiral.py	spira mirabilis	–
3.2.7	drunk.py	drunken turtle	–
3.2.8	drunks.py	drunken turtles	–
3.2.9	complex.py	complex number data type	–
3.2.10	mandelbrot.py	Mandelbrot set	–
3.2.11	stockaccount.py	stock account data type	turing.txt
3.3.1	complexpolar.py	complex numbers (revisited)	–
3.3.2	counter.py	counter data type	–
3.3.3	vector.py	spatial vector data type	–
3.3.4	sketch.py	sketch data type	genome20.txt
3.3.5	comparedocuments.py	similarity detection	documents.txt constitution.txt tomsawyer.txt huckfinn.txt prejudice.txt djia.csv amazon.html actg.txt
3.4.1	body.py	gravitational body data type	–
3.4.2	universe.py	n-body simulation	2body.txt 3body.txt 4body.txt 2bodytiny.txt