them. Finally, two chapters are devoted to Node.js, another environment to program JavaScript in.

Throughout the book, there are five *project chapters*, which describe larger example programs to give you a taste of real programming. In order of appearance, we will work through building an artificial life simulation, aprogramming language, a platform game, a paint program, and a dynamic website.

The language part of the book starts with four chapters to introduce the basic structure of the JavaScript language. They introduce control structures (such as the `while` word you saw in this introduction), functions (writing your own operations), and data structures. After these, you will be able to write simple programs. Next, Chapters 5 and 6 introduce techniques to use functions and objects to write more *abstract* code and thus keep complexity under control.

After a first project chapter, the first part of the book continues with chapters on error handling and fixing, on regular expressions (an important tool for working with text data), and on modularity—another weapon against complexity. The second project chapter concludes the first part of the book.

The second part, Chapters 12 to 19, describes the tools that browser JavaScript has access to. You'll learn to display things on the screen (Chapters 13 and 16), respond to user input (Chapters 14 and 18), and communicate over the network (Chapter 17). There are again two project chapters in this part.

After that, Chapter 20 describes Node.js, and Chapter 21 builds a simple web system using that tool.

Typographic conventions