

Preface

i. What is the purpose of this book?

There are many students, other than those specifically interested in computer-related areas such as computer science or computer engineering, who nevertheless need to know how to solve computational problems on computers. There are basically two approaches to meeting the needs of such students. One is to rely on software applications such as spreadsheets, using built-in functions as needed, without relying explicitly on any of the constructs, such as branching and looping, that are common to programming languages.

A second approach is to teach such students a traditional programming language, previously Fortran or Pascal, and more recently C, C++, or Java. These languages play important roles in certain kinds of work, such as computer science (C++, Java) or scientific computing (C and Fortran), but having to learn one of them may be viewed, possibly with good reason, as irrelevant by many students.

From a student's point of view, there is no painless solution to this dilemma, but in this book I assume that learning to solve computational problems in an HTML/JavaScript environment will at least appear to be a more relevant solution. Both **HTML**¹ and **JavaScript** are essential for Web development and some working knowledge of them is a useful and marketable skill. So, in addition to learning basic programming concepts, students can also learn something that may be more immediately valuable than older text-based languages.

In many ways, the HTML/JavaScript environment is more difficult to learn than a traditional text-based programming language such as C. C is a mature (some might prefer "obsolete"), fairly small language with an unambiguous set of syntax rules and a primitive text-based input/output interface. You can view the limitations of C as either a blessing or a curse, depending on your needs. A major advantage of C is that programs written in **ANSI Standard C** should work equally well on any computer that has a C compiler, making the language inherently **platform-independent**.

¹ See Glossary for definitions of terms appearing in bold font.

JavaScript and HTML, on the other hand, are immature and very unstable languages (if we can agree informally to call HTML a “language”) that function within a constantly changing Web environment. There are dialects of HTML and JavaScript that will work only on particular computing platforms and with specific software. While it is true that there are extensions to languages such as C and other older languages that are platform-dependent, the platform dependence of HTML and JavaScript is a major implementation issue rather than an occasional minor inconvenience.

As one indication of the teaching and learning challenges these environments provide, just three popular paperback HTML and JavaScript reference books occupy nearly 6 inches of space (15 cm in deference to non-U.S. readers) on my office bookshelf! A great deal of the material in those books is devoted to explaining the often subtle differences among various versions of HTML and JavaScript.

Fortunately, it is possible to present some core subsets of both HTML and JavaScript that can be used to solve some of the same kinds of computational problems that would be appropriate for a more traditional language such as C or C++. My motivation for writing this book was to learn how to use HTML and JavaScript to write my own online applications, and I now use this environment for many tasks that I previously would have undertaken in C. Based on this experience, I have concluded that, despite the fact that JavaScript is definitely not intended as a “scientific computing” language, it is nonetheless reasonable to present some basic programming skills of interest to science and engineering students and practitioners in the context of an HTML/JavaScript environment. The examples and exercises presented in the book do not require extensive science, engineering, or mathematics background (only rarely, in a few of the exercises) is knowledge beyond basic algebra needed), so I believe this book could serve as a beginning programming text even for high school students.

ii. Learning by Example

It is well known that people learn new skills in different ways. Personally, I learn best by having a specific goal and then studying examples that seem related to that goal. Once I understand those examples, I can incorporate them into my own work. I have used that learning model in this book, which contains many complete examples that can serve as starting points for your work.

This model works well in an online environment, too. The amount of online information about both HTML and JavaScript is so vast that it is

only a slight exaggeration to state that nobody writes original JavaScript code any more. If you have trouble “learning by example,” you will have trouble learning these languages, not just from this book, but in general because that is how most of the available information is presented.

It is an inescapable fact that a great deal of the source code behind Web pages involves nothing more (or less) than creative cutting, pasting, and tweaking of existing code. Aside from the issues of plagiarism and intellectual dishonesty that must be dealt with in an academic environment, there is also the practical matter of an effective learning strategy. You cannot learn to solve your own computational problems just by trying to paste together someone else’s work. (Believe me, I’ve tried!) Until you develop your own independent skills, you will be constantly frustrated because you will never find *exactly* what you need to copy and you will be unable to synthesize what you need from what is available.

So, while you should expect to find yourself constantly recycling your own code throughout a course based on this book, you need to be responsible for your own work. Make sure you really *learn* and don’t just *learn to copy*!

iii. Acknowledgments

I am once again indebted to my wife, Susan, for her patient editing of this manuscript. Considering that she also edited two of my previous computer programming manuscripts, I can conclude only that sufficient time has passed to dim her recollections of those experiences!

During the Fall of 2006, student comments in a class I taught for Drexel University’s School of Biomedical Engineering, Science and Health Systems, have provided valuable suggestions for improving the presentation and content of this manuscript.



<http://www.springer.com/978-1-84628-656-8>

An Introduction to HTML and JavaScript
for Scientists and Engineers

Brooks, D.R.

2007, XI, 200 p. 6 illus. in color. With online
files/update., Softcover

ISBN: 978-1-84628-656-8