

# Think DSP

Digital Signal Processing in Python

Version 1.1.4



# Think DSP

Digital Signal Processing in Python

Version 1.1.4

Allen B. Downey

Green Tea Press

Needham, Massachusetts

Copyright © 2014 Allen B. Downey.

Green Tea Press  
9 Washburn Ave  
Needham MA 02492

Permission is granted to copy, distribute, and/or modify this document under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License, which is available at <http://creativecommons.org/licenses/by-nc-sa/4.0/>.

The L<sup>A</sup>T<sub>E</sub>X source for this book is available from <http://think-dsp.com>.

# Preface

Signal processing is one of my favorite topics. It is useful in many areas of science and engineering, and if you understand the fundamental ideas, it provides insight into many things we see in the world, and especially the things we hear.

But unless you studied electrical or mechanical engineering, you probably haven't had a chance to learn about signal processing. The problem is that most books (and the classes that use them) present the material bottom-up, starting with mathematical abstractions like phasors. And they tend to be theoretical, with few applications and little apparent relevance.

The premise of this book is that if you know how to program, you can use that skill to learn other things, and have fun doing it.

With a programming-based approach, I can present the most important ideas right away. By the end of the first chapter, you can analyze sound recordings and other signals, and generate new sounds. Each chapter introduces a new technique and an application you can apply to real signals. At each step you learn how to use a technique first, and then how it works.

This approach is more practical and, I hope you'll agree, more fun.

## 0.1 Who is this book for?

The examples and supporting code for this book are in Python. You should know core Python and you should be familiar with object-oriented features, at least using objects if not defining your own.

If you are not already familiar with Python, you might want to start with my other book, *Think Python*, which is an introduction to Python for people who have never programmed, or Mark Lutz's *Learning Python*, which might be better for people with programming experience.

I use NumPy and SciPy extensively. If you are familiar with them already, that's great, but I will also explain the functions and data structures I use.

I assume that the reader knows basic mathematics, including complex numbers. You don't need much calculus; if you understand the concepts of integration and differentiation, that will do. I use some linear algebra, but I will explain it as we go along.

## 0.2 Using the code

The code and sound samples used in this book are available from <https://github.com/AllenDowney/ThinkDSP>. Git is a version control system that allows you to keep track of the files that make up a project. A collection of files under Git's control is called a "repository". GitHub is a hosting service that provides storage for Git repositories and a convenient web interface.

The GitHub homepage for my repository provides several ways to work with the code:

- You can create a copy of my repository on GitHub by pressing the Fork button. If you don't already have a GitHub account, you'll need to create one. After forking, you'll have your own repository on GitHub that you can use to keep track of code you write while working on this book. Then you can clone the repo, which means that you copy the files to your computer.
- Or you could clone my repository. You don't need a GitHub account to do this, but you won't be able to write your changes back to GitHub.
- If you don't want to use Git at all, you can download the files in a Zip file using the button in the lower-right corner of the GitHub page.

All of the code is written to work in both Python 2 and Python 3 with no translation.

I developed this book using Anaconda from Continuum Analytics, which is a free Python distribution that includes all the packages you'll need to run the code (and lots more). I found Anaconda easy to install. By default it does a user-level installation, not system-level, so you don't need administrative privileges. And it supports both Python 2 and Python 3. You can download Anaconda from <https://www.anaconda.com/distribution/>.

If you don't want to use Anaconda, you will need the following packages:

- NumPy for basic numerical computation, <http://www.numpy.org/>;
- SciPy for scientific computation, <http://www.scipy.org/>;
- matplotlib for visualization, <http://matplotlib.org/>.

Although these are commonly used packages, they are not included with all Python installations, and they can be hard to install in some environments. If you have trouble installing them, I recommend using Anaconda or one of the other Python distributions that include these packages.

Most exercises use Python scripts, but some also use Jupyter notebooks. If you have not used Jupyter before, you can read about it at <http://jupyter.org>.

There are three ways you can work with the Jupyter notebooks:

**Run Jupyter on your computer** If you installed Anaconda, you probably got Jupyter by default. To check, start the server from the command line, like this:

```
$ jupyter notebook
```

If it's not installed, you can install it in Anaconda like this:

```
$ conda install jupyter
```

When you start the server, it should launch your default web browser or create a new tab in an open browser window.

**Run Jupyter on Binder** Binder is a service that runs Jupyter in a virtual machine. If you follow this link, <http://mybinder.org/repo/AllenDowney/ThinkDSP>, you should get a Jupyter home page with the notebooks for this book and the supporting data and scripts.

You can run the scripts and modify them to run your own code, but the virtual machine you run in is temporary. Any changes you make will disappear, along with the virtual machine, if you leave it idle for more than about an hour.

**View notebooks on nbviewer** When we refer to notebooks later in the book, we will provide links to nbviewer, which provides a static view of the code and results. You can use these links to read the notebooks and listen to the examples, but you won't be able to modify or run the code, or use the interactive widgets.

Good luck, and have fun!

## Contributor List

If you have a suggestion or correction, please send email to [downey@allendowney.com](mailto:downey@allendowney.com). If I make a change based on your feedback, I will add you to the contributor list (unless you ask to be omitted).

If you include at least part of the sentence the error appears in, that makes it easy for me to search. Page and section numbers are fine, too, but not as easy to work with. Thanks!

- Before I started writing, my thoughts about this book benefited from conversations with Boulos Harb at Google and Aurelio Ramos, formerly at Harmonix Music Systems.
- During the Fall 2013 semester, Nathan Lintz and Ian Daniher worked with me on an independent study project and helped me with the first draft of this book.
- On Reddit's DSP forum, the anonymous user RamjetSoundwave helped me fix a problem with my implementation of Brownian Noise. And andodli found a typo.
- In Spring 2015 I had the pleasure of teaching this material along with Prof. Oscar Mur-Miranda and Prof. Siddhartan Govindasamy. Both made many suggestions and corrections.
- Silas Gyger corrected an arithmetic error.
- Giuseppe Masetti sent a number of very helpful suggestions.
- Eric Peters sent many helpful suggestions.

Special thanks to Freesound, which is the source of many of the sound samples I use in this book, and to the Freesound users who uploaded those sounds. I include some of their wave files in the GitHub repository for this book, using the original file names, so it should be easy to find their sources.

Unfortunately, most Freesound users don't make their real names available, so I can only thank them using their user names. Samples used in this book were contributed by Freesound users: iluppai, wcf10, thirsk, docquesting, kleeB, landup, zippi1, themusicalnomad, bcjordan, rockwehrmann, marcgascon7, jveliz. Thank you all!