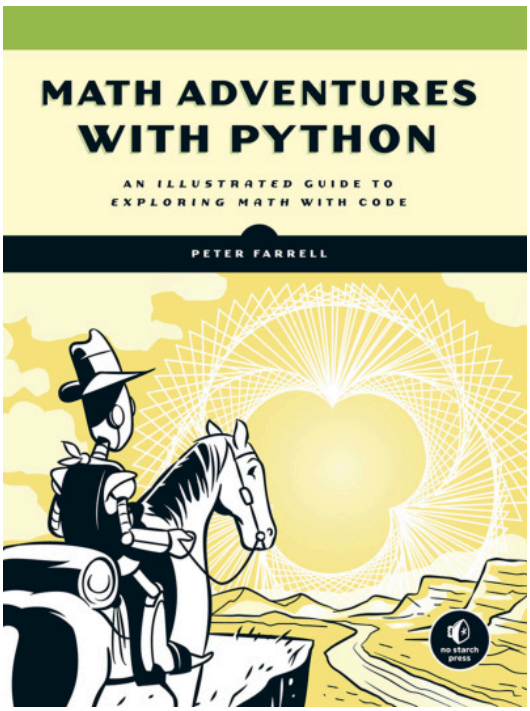


ebookbig.com
de to access the d st

making cool graphics is nearly impossible without knowing the math behind geometry, for example. As you'll see in this book, knowing a little of the math behind triangles, like the centroid, will make it easy to create our artworks. A student who knows math and can create cool designs is more likely to delve into a little geometry and put up with a few square roots or a trig function or two. A student who doesn't see any outcome, and is only doing homework from a textbook, probably doesn't have much motivation to learn geometry.

In my eight years of experience as a math teacher and three years of experience as a computer science teacher, I've met many more math learners who prefer the visual approach to the academic one. In the process of creating something interesting, you come to understand that math is not just following steps to solve an equation. You see that exploring math with programming allows for many ways to solve interesting problems, with many unforeseen mistakes and opportunities for improvements along the way.

This is the difference between school math and real math.

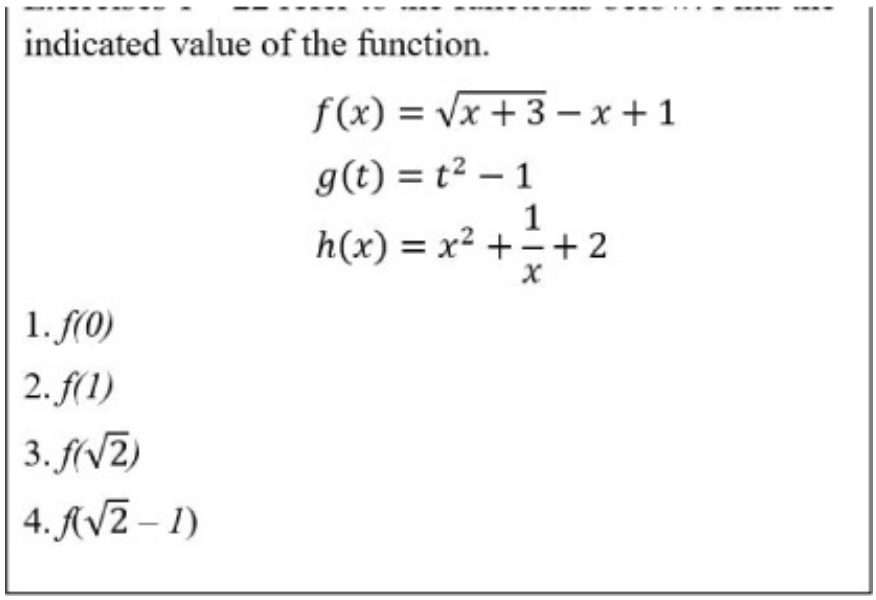
THE PROBLEM WITH SCHOOL MATH

What do I mean by "school math" exactly? In the US in the 1860s, school math was preparation for a job as a clerk, adding columns of numbers by hand. Today, jobs are different, and the preparation for these jobs needs to change, too.

People learn best by doing. This hasn't been a daily practice in schools, though, which tend to favor passive learning. "Doing" in English and history classes might mean students write papers or give presentations, and science students perform experiments, but what do math students do? It used to be that all you could actively "do" in math class was solve equations, factor polynomials, and graph functions. But now that computers can do most of those calculations for us, these practices are no longer sufficient.

Simply learning how to automate solving, factoring, and graphing is not the final goal. Once a student has learned to automate a process, they can go further and deeper into a topic than was ever possible before.

Figure 2 shows a typical math problem you'd find in a textbook, asking students to define a function, "f(x)," and evaluate it for a ton of values.


indicated value of the function.

$$f(x) = \sqrt{x+3} - x + 1$$

$$g(t) = t^2 - 1$$

$$h(x) = x^2 + \frac{1}{x} + 2$$

1. $f(0)$
2. $f(1)$
3. $f(\sqrt{2})$
4. $f(\sqrt{2} - 1)$

Figure 2: A traditional approach to teaching functions

This same format goes on for 18 more questions! This kind of exercise is a trivial problem for a programming language like Python. We could simply define the function `f(x)` and then plug in the values by iterating over a list, like this:

```
import math

def f(x):
    return math.sqrt(x + 3) - x + 1

#list of values to plug in
for x in [0,1,math.sqrt(2),math.sqrt(2)-1]:
    print("f({:.3f}) = {:.3f}".format(x,f(x)))
```

The last line just makes the output pretty while rounding all the solutions to three decimal places, as shown here:

```
f(0.000) = 2.732
f(1.000) = 2.000
f(1.414) = 1.687
f(0.414) = 2.434
```

In programming languages like Python, JavaScript, Java, and so on, functions are a vitally important tool for transforming numbers and other objects—even other functions! Using Python, you can give a descriptive name to a function, so it's easier to

```
def calculateArea(width,height):
```

A math textbook published in the 21st century, decades after Benoit Mandelbrot first generated his famous fractal on a computer when working for IBM, shows a picture of the Mandelbrot set and gushes over the discovery. The textbook describes the Mandelbrot set, which is shown in Figure 3, as "a fascinating mathematical object derived from the complex numbers. Its beautiful boundary illustrates chaotic behavior.

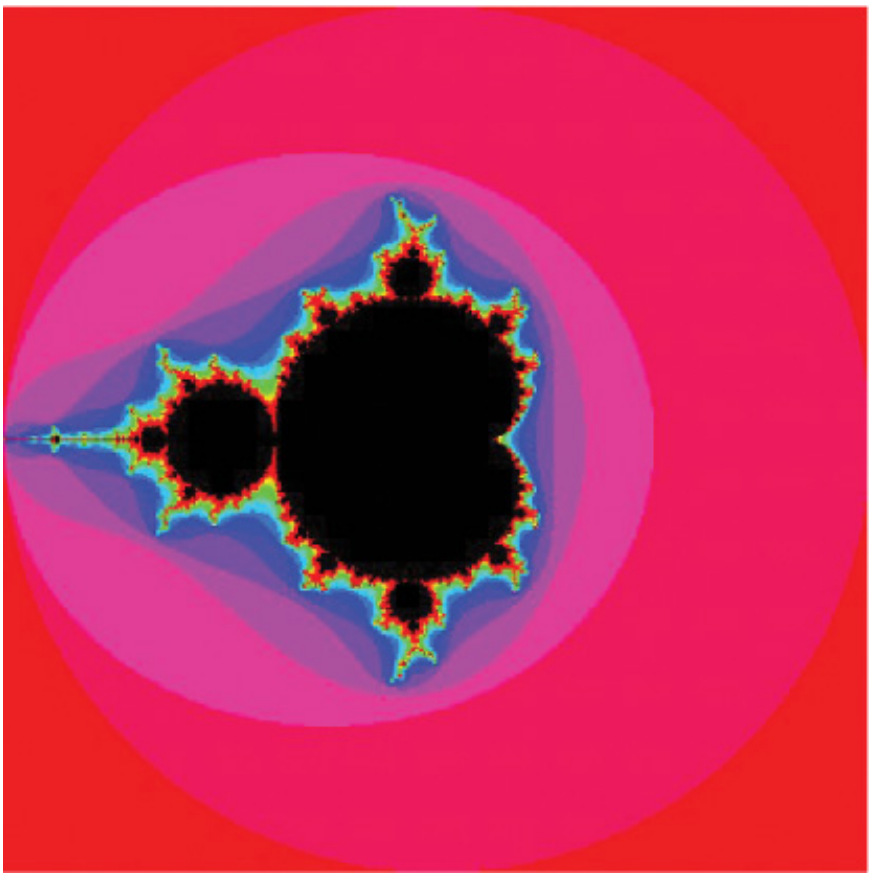


Figure 3: The Mandelbrot set

The textbook then takes the reader through a painstaking "exploration" to show how to transform a point in the complex plane. But the student is only shown how to do this on a calculator, which means only two points can be transformed (iterated seven times) in a reasonable amount of time. Two points.

In this book, you'll learn how to do this in Python, and you'll make the program transform hundreds of thousands of points automatically and even *create* the Mandelbrot set you see above!

ABOUT THIS BOOK

being challenging. You'll make graphs to show all the possible outputs of a function. You'll make dynamic, interactive works of art. You'll even make an ecosystem with sheep that move around, eat grass, and multiply, and you'll create virtual organisms that try to find the shortest route through a bunch of cities while you watch!

You'll do this using Python and Processing in order to supercharge what you can do in math class. This book is not about skipping the math; it's about using the newest, coolest tools out there to get creative and learn real computer skills while discovering the connections between math, art, science, and technology. Processing will provide the graphics, shapes, motion, and colors, while Python does the calculating and follows your instructions behind the scenes.

For each of the projects in this book, you'll build the code up from scratch, starting from a blank file, and checking your progress at every step. Through making mistakes and debugging your own programs, you'll get a much deeper understanding of what each block of code does.

WHO SHOULD USE THIS BOOK

This book is for anyone who's learning math or who wants to use the most modern tool available to approach math topics like trigonometry and algebra. If you're learning Python, you can use this book to apply your growing programming skills to nontrivial projects like cellular automata, genetic algorithms, and computational art.

Teachers can use the projects in this book to challenge their students or to make math more approachable and relevant. What better way to teach matrices than to save a bunch of points to a matrix and use them to draw a 3D figure? When you know Python, you can do this and much more.

WHAT'S IN THIS BOOK?

This book begins with three chapters that cover basic Python concepts you'll build on to explore more complicated math. The next nine chapters explore math concepts and problems that you can visualize and solve using Python and Processing. You can try the exercises peppered throughout the book to apply what you learned and challenge yourself.

Chapter 1: Drawing Polygons with Turtles teaches basic programming concepts like loops, variables, and functions using Python's built-in `turtle` module.

programming concepts like lists and Booleans.

Chapter 3: Guessing and Checking with Conditionals applies your growing Python skills to problems like factoring numbers and making an interactive number-guessing game.

Chapter 4: Transforming and Storing Numbers with Algebra ramps up from solving simple equations to solving cubic equations numerically and by graphing.

Chapter 5: Transforming Shapes with Geometry shows you how to create shapes and then multiply, rotate, and spread them all over the screen.

Chapter 6: Creating Oscillations with Trigonometry goes beyond right triangles and lets you create oscillating shapes and waves.

Chapter 7: Complex Numbers teaches you how to use complex numbers to move points around the screen, creating designs like the Mandelbrot set.

Chapter 8: Using Matrices for Computer Graphics and Systems of Equations takes you into the third dimension, where you'll translate and rotate 3D shapes and solve huge systems of equations with one program.

Chapter 9: Building Objects with Classes covers how to create one object, or as many as your computer can handle, with roaming sheep and delicious grass locked in a battle for survival.

Chapter 10: Creating Fractals Using Recursion shows how recursion can be used as a whole new way to measure distances and create wildly unexpected designs.

Chapter 11: Cellular Automata teaches you how to generate and program cellular automata to behave according to rules you make.

Chapter 12: Solving Problems Using Genetic Algorithms shows you how to harness the theory of natural selection to solve problems we couldn't solve in a million years otherwise!

DOWNLOADING AND INSTALLING PYTHON

The easiest way to get started is to use the Python 3 software distribution, which is available for free at *https://www.python.org/*. Python has become one of the most

crunch numbers in various fields, from astronomy to zoology. The latest version released to date is Python 3.7. Go to *https://www.python.org/downloads/* and choose the latest version of Python 3, as shown in Figure 4.



Figure 4: The official website of the Python Software Foundation



Figure 5: Click the downloaded file to start the install

You can choose the version for your operating system. The site detected that I was usin Windows. Click the file when the download is complete, as shown in Figure 5.

Follow the directions, and always choose the default options. It might take a few minutes to install. After that, search your system for "IDLE." That's the Python IDE, or *integrated development environment*, which is what you'll need to write Python code. Why "IDLE"? The Python programming language was named after the Monty Python comedy troupe, and one of the members is Eric Idle.

STARTING IDLE

Find IDLE on your system and open it.

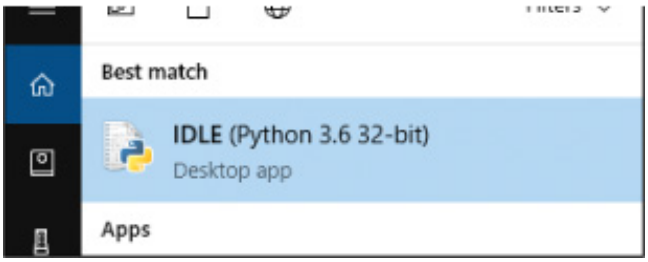


Figure 6: Opening IDLE on Windows

A screen called a "shell" will appear. You can use this for the interactive coding environment, but you'll want to save your code. Click **File▸New File** or press ALT-N, and a file will appear (see Figure 7).

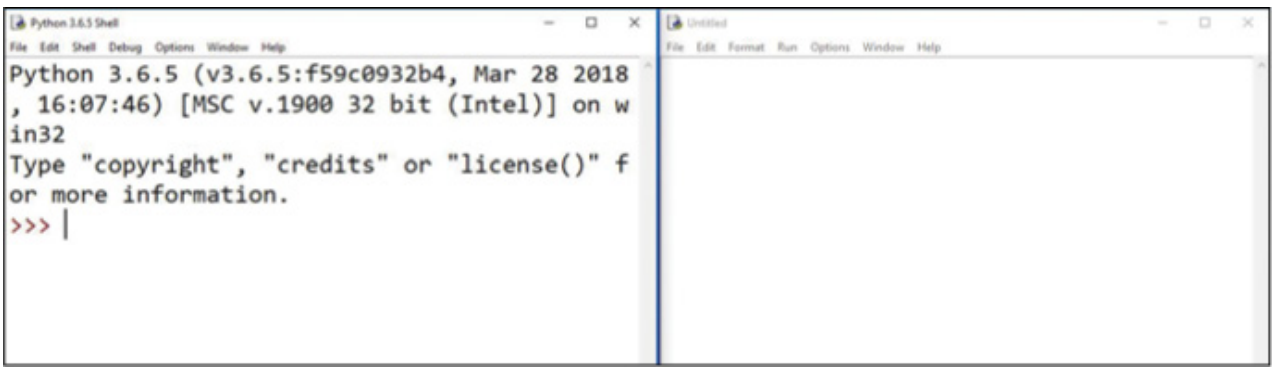


Figure 7: Python's interactive shell (left) and a new module (file) window, ready for code!

This is where you'll write your Python code. We will also use Processing, so let's go over how to download and install Processing next.

INSTALLING PROCESSING

There's a lot you can do with Python, and we'll use IDLE a lot. But when we want to do some heavy-duty graphics, we're going to use Processing. Processing is a professional-level graphics library used by coders and artists to make dynamic, interactive artwork and graphics.

Go to *https://processing.org/download/* and choose your operating system, as shown in Figure 8.

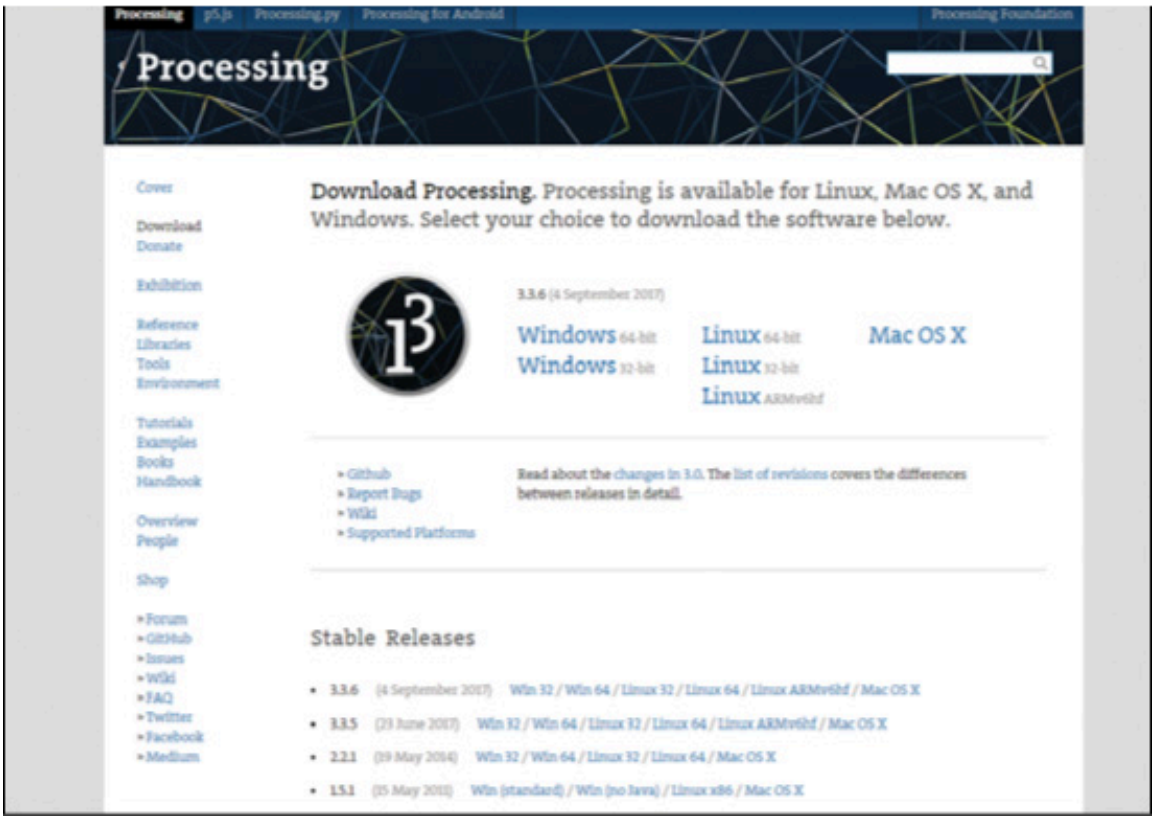


Figure 8: The Processing website

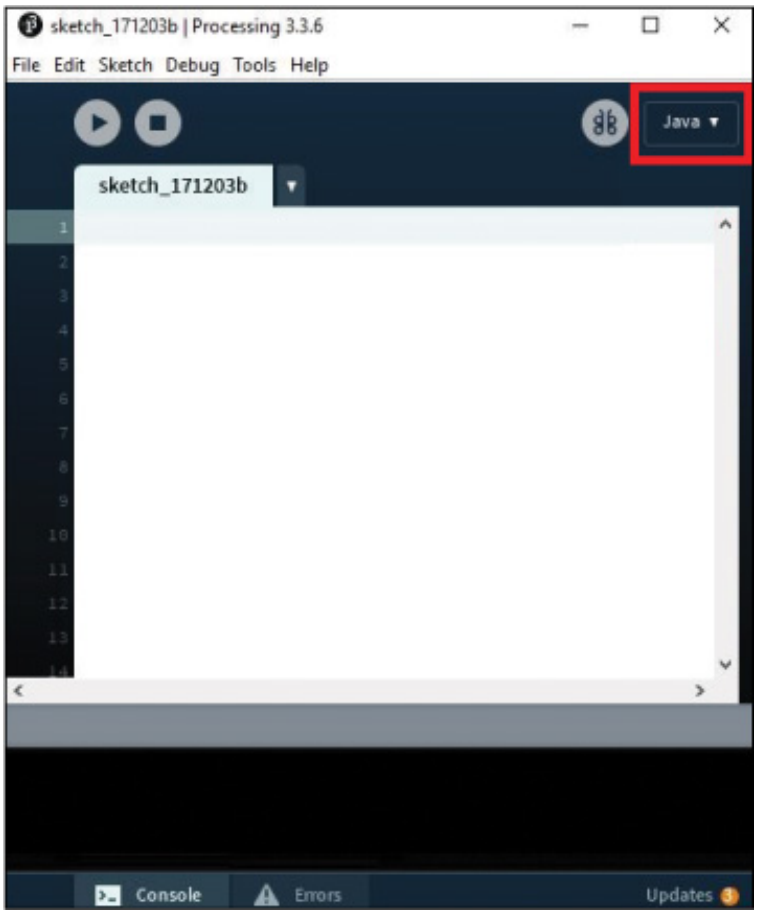


Figure 9: Where to find other Processing modes, like the Python mode we'll be using

Download the installer for your operating system by clicking it and following the

Java to open the drop-down menu, as shown in Figure 9, and then click **Add Mode**.

Select **Python Mode▸Install**. It should take a minute or two, but after this you'll be able to code in Python with Processing.

Now that you've set up Python and Processing, you're ready to start exploring math!