

# Pencil Code

A Programming Primer

David Bau

Visit <http://pencilcode.net/> to run your programs.

*"Creativity takes courage."*  
- Henri Matisse

Second Edition. Copyright © 2013 David Bau.

Pencil Code is an open-source system that unites the CoffeeScript language by Jeremy Ashkenas in 2009, and Iced await/defer extensions created by Maxwell Krohn in 2012, with the jQuery-turtle plugin developed by the author in 2011, using the jQuery library invented by John Resig in 2006. This work is inspired by the beloved LOGO language created by Seymour Papert and Wally Feurzeig in 1967.

Special thanks to the students in Lincoln Massachusetts, Beaver Country Day School, and Dorchester McCormack School who vetted this material.

Post questions, ideas, and bug reports to <http://pencilcode.net/group>

Fancy Sun illustration contributed by Margaret Z.

Random Tree illustration contributed by Mike Koss.

Cover image by Phil Clements. Back cover image by Vinod Velayudhan.

This book is typeset in Łukasz Dziedzic's 2010 open font Lato and Paul D. Hunt's 2012 Adobe Source Code Pro.

## No Thresholds and No Limits

The aim of this book is to teach you to write programs as you would use a pencil: as an outlet for creativity and as a tool for understanding.

These pages follow a fifty-year tradition of using programming as a liberating educational tool, with no thresholds for beginners, and no limits for experts. Seymour Papert's LOGO is the inspiration. Start with a few lines of code, and progress to writing programs to explore art, mathematics, language, algorithms, simulation, and thought.

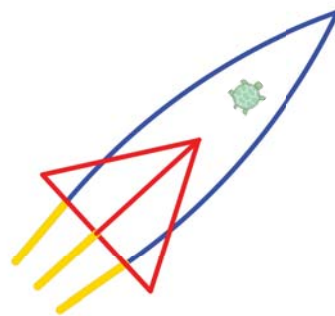
The language is CoffeeScript. Although CoffeeScript is a production programming language used by pros, it was chosen here because it has an elegance and simplicity well-suited for beginners. While the first examples make the language look trivial, CoffeeScript has a good notation for all the important ideas: algebraic expressions, lists, loops, functions, objects, and concurrency. As you learn the language, remember that the goal should be not mastery of the syntax, but mastery of the underlying concepts.

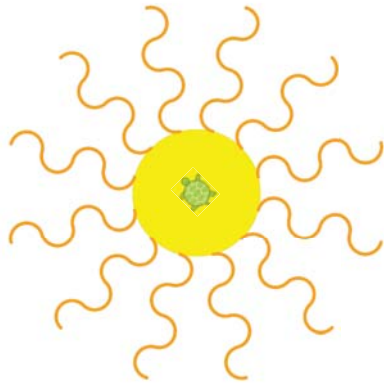
Edit and run your programs on [pencilcode.net](http://pencilcode.net). The site is an experiment in community learning: everything posted is public. Write programs that would be interesting to others. Accounts are free.

As you experiment by building your own ideas, you will find that at first your programs will behave in ways that you do not intend. Details matter, and persistence pays off. If you are patient in adjusting and perfecting your work, you will be rewarded with insight.

Read, think, play, and create something beautiful.

David Bau, 2013





# Contents

Part 1

## Primer: 100 Little Projects

1. Lines
2. Points
3. Loops
4. Nesting
5. Functions
6. Parameters
7. Time
8. Output
9. Input
10. Numbers
11. Computation
12. Objects
13. Arrays
14. Recursion
15. Randomness
16. Sets
17. Text
18. Motion
19. Concurrency
20. Styles
21. Selectors
22. Events
23. Slicing
24. Sorting
25. Search
26. Intelligence

# Contents

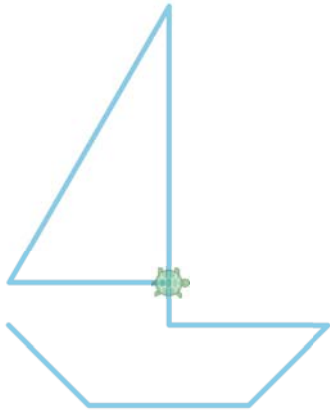
Part 2

## Appendix: One Project in Detail

1. Running Pencil Code
2. Keeping a Secret
3. Computers are Fine Calculators
4. Strings and Numbers
5. Creating Graphics
6. Making our First Program
7. Hurry Up and Wait
8. Using "for" to Repeat
9. Using "if" to Choose
10. Input with "read"
11. Using "while" to Repeat
12. Improving our Game
13. Making it Look Like Hangman
14. Picking a Random Secret
15. Loading a List from the Internet
16. The Whole Hangman Program
17. Making it Yours

Next Steps

Reference



## Primer

### 100 Little Projects

In the following pages, the basic concepts of modern computer programming are shown in a series of brief examples.

None of the examples come with explanations. Just try them.

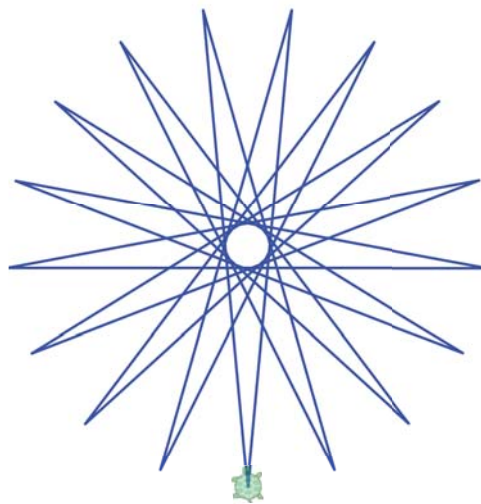
Once you get a program working, stop and think about how it works. Make your own customized version.

Can you draw a violet with a stem? Can you make a snowflake that really looks like a snowflake? And can you do these things in an elegant way?

For a tutorial introduction to CoffeeScript, turn to the Appendix which begins after example set 26. Also be sure to try the "help" command in the test panel.

Enough advice.

Let's play.



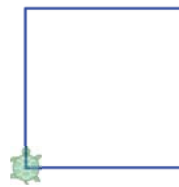


# 1. Lines

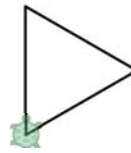
**First**



**Square**



**Triangle**



**House**



**Turtle**  
pen green

