



Learn by doing: less theory, more results

# Scratch 2.0

*Second Edition*

Create digital stories, games, art, and animations through  
six unique projects

*Beginner's Guide*

Michael Badger

[PACKT]  
PUBLISHING

---

# **Scratch 2.0 Beginner's Guide**

## ***Second Edition***

Create digital stories, games, art, and animations through  
six unique projects

**Michael Badger**



BIRMINGHAM - MUMBAI

---

# **Scratch 2.0 Beginner's Guide**

## ***Second Edition***

Copyright © 2014 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First Published: July 2009

Second Edition: April 2014

Production Reference: 1080414

Published by Packt Publishing Ltd.  
Livery Place  
35 Livery Street  
Birmingham B3 2PB, UK.

ISBN 978-1-78216-072-4

[www.packtpub.com](http://www.packtpub.com)

Cover Image by Ross Manges ([ross@rossmanges.net](mailto:ross@rossmanges.net))

---

# Credits

**Author**

Michael Badger

**Project Coordinator**

Binny K. Babu

**Reviewers**

Samyak Bhuta

Manuel Menezes de Sequeira

Franklin Webber

**Proofreaders**

Simran Bhogal

Maria Gould

Ameesha Green

Paul Hindle

**Acquisition Editor**

Joanne Fitzpatrick

**Indexer**

Mehreen Deshmukh

**Content Development Editor**

Dayan Hyames

**Production Coordinator**

Nitesh Thakur

**Technical Editors**

Shubhangi Dhamgaye

Shweta Pant

Mrunmayee Patil

Aman Preet Singh

**Cover Work**

Nitesh Thakur

**Copy Editors**

Sarang Chari

Brandt D'Mello

Mradula Hegde

---

# About the Author

**Michael Badger** is a writer and technical communicator who has worked in a range of technical roles, including support, automated software testing, and project management. He has authored several books for Packt Publishing, including *Scratch 1.4 Beginner's Guide*. He also authors a regular Scratch column for Raspberry Pi Geek Magazine, which focuses on Scratch 1.4.

---

I'd like to thank the team at Packt Publishing for putting up with me and helping me make this revision the best it could be. My loving wife Christie and son Cameron also deserve credit for allowing me the flexibility to complete this book.

---

---

# About the Reviewers

**Samyak Bhuta** is fascinated by art and technology and is always excited when they both meet. He is a software architect by profession with over a decade of experience. He started programming in his childhood with GWBasic and quickly moved over to QBasic. Professionally, he has worked on Java, JavaScript, Python, and PHP. He enjoys coding user interfaces as well as working on backend programming. Samyak believes in the open source philosophy and has been active in his local community. He loves to eat dal bati, an Indian dish, and has dreams to become a flautist.

---

I would like to thank Packt Publishing for keeping patience when I couldn't submit my reviews on time.

---

**Manuel Menezes de Sequeira** has been teaching programming since 1995. He started teaching programming using C, then moved to C++, and later to Java. Nowadays, in his lectures, he usually starts programming with Scratch and Snap!, and then moves on to text-based languages such as Java. Manuel teaches at the Universidade Europeia | Laureate International Universities in Lisbon, Portugal, where he also champions in CoderDojo LX, the Lisbon-based CoderDojo, where children can learn to program for free while having fun. He lives in Lisbon, Portugal, and has been involved for a few years in the translation of Scratch, SNAP!, and other projects to Portuguese.

**Franklin Webber** is a software professional whose professional experience comes from a testing background where he sought to automate himself out of a job. A college teaching assistant once told Frank that he was a great software developer and a terrible computer scientist, and that the software he wrote cared more for the user experience than the size of its Big O Notation. As a software developer, he became the resident generalist who was always willing to step up to learn new technologies. He now spends most of his time teaching software design to students, both young and old.

---

# www.PacktPub.com

## Support files, eBooks, discount offers, and more

You might want to visit [www.PacktPub.com](http://www.PacktPub.com) for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at [www.PacktPub.com](http://www.PacktPub.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [service@packtpub.com](mailto:service@packtpub.com) for more details.

At [www.PacktPub.com](http://www.PacktPub.com), you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

## Why Subscribe?

- ◆ Fully searchable across every book published by Packt
- ◆ Copy and paste, print and bookmark content
- ◆ On demand and accessible via web browser

## Free Access for Packt account holders

If you have an account with Packt at [www.PacktPub.com](http://www.PacktPub.com), you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

---

# Table of Contents

<b>Preface</b>	<b>1</b>
<b>Chapter 1: Welcome to Scratch 2.0</b>	<b>7</b>
<b>About Scratch</b>	<b>8</b>
Encouraging everyone to think programmatically	8
Sample Scratch uses	9
Computational thinking	9
<b>Finding a project for you</b>	<b>10</b>
Making animations	10
Telling stories	11
Building games	12
Programming games of chance	12
Creating art projects	13
Sensing the real world	13
Programming concepts	13
<b>Using Scratch 2.0</b>	<b>15</b>
Looking inside a Scratch project	17
The stage area	18
The sprites pane	18
The scripts area	18
The built-in image editor	20
Using Scratch 2.0 offline	21
Encountering Scratch 1.4	22
Tinkering encouraged	22
<b>Summary</b>	<b>23</b>



<b>Chapter 2: A Quick Start Guide to Scratch</b>	<b>25</b>
<b>Joining the Scratch community</b>	<b>25</b>
<b>Time for action – creating an account on the Scratch website</b>	<b>26</b>
<b>Time for action – understanding the key features of your account</b>	<b>28</b>
Abiding by the terms of use	30
Creating projects under Creative Commons licenses	30
Finding free media online	30
<b>Taking our first steps in Scratch</b>	<b>31</b>
<b>Time for action – moving the cat across the stage</b>	<b>31</b>
Using events to trigger an action	33
<b>Time for action – animating a walking motion with the cat</b>	<b>33</b>
Understanding the basics of a Scratch Project	35
Saving early, often, and automatically	36
<b>Time for action – saving our work</b>	<b>36</b>
Undoing a deletion	37
<b>Introducing forever loops</b>	<b>37</b>
<b>Time for action – setting the cat in motion, forever</b>	<b>37</b>
Controlling a sprite with loops	39
<b>Time for action – flipping the cat right-side up</b>	<b>39</b>
Clicking on a block runs the command	41
<b>Adding sprites to the project</b>	<b>41</b>
<b>Time for action – adding a second sprite and script</b>	<b>42</b>
<b>Reviewing a video-sensing project</b>	<b>44</b>
<b>Time for action – reviewing pop the balloon - video starter</b>	<b>44</b>
Sensing video	46
<b>Summary</b>	<b>47</b>
<b>Chapter 3: Creating an Animated Birthday Card</b>	<b>49</b>
<b>Introducing the paint editor</b>	<b>50</b>
<b>Time for action – painting a happy birthday sprite</b>	<b>50</b>
Changing the size of a bitmap image	52
Choosing bitmap or vector images	53
<b>Time for action – drawing a vector image</b>	<b>53</b>
Changing the size of the vector image	54
Reviewing the image editing tools	55
Erasing in the vector mode	56
Filling the stage with color	57
<b>Time for action – using the fill with color tool to paint the stage</b>	<b>57</b>
<b>Adding gradients</b>	<b>59</b>
<b>Time for action – applying a gradient</b>	<b>59</b>
<b>Time for action – adding more sprites to address the card</b>	<b>61</b>
<b>Initializing a sprite's starting values</b>	<b>62</b>

<b>Time for action – hiding all sprites when the flag is clicked</b>	<b>62</b>
<b>Time for action – displaying happy birthday</b>	<b>64</b>
<b>Specifying memorable names and comments</b>	<b>66</b>
<b>Time for action – renaming sprites</b>	<b>66</b>
<b>Inserting comments into our code</b>	<b>67</b>
<b>Time for action – adding comments to a script</b>	<b>67</b>
<b>Transforming sprites with graphical effects</b>	<b>68</b>
<b>Time for action – transforming sprites</b>	<b>68</b>
Graphical transformations	71
Comparing the repeat and forever blocks	72
<b>Time for action – turning m in a circle</b>	<b>73</b>
<b>Time for action – making a sprite fade in with the ghost effect</b>	<b>74</b>
Two ways to control timing	76
<b>Summary</b>	<b>78</b>
<b>Chapter 4: Creating a Scratch Story Book</b>	<b>79</b>
<b>Designing the outline of a barnyard joke book</b>	<b>79</b>
<b>Time for action – designing a clickable table of contents</b>	<b>80</b>
<b>Time for action – adding pages to the book</b>	<b>81</b>
<b>Time for action – adding a sprite to the Backpack</b>	<b>83</b>
Using the Backpack to store sprites and scripts	84
<b>Building a joke with say blocks and sounds</b>	<b>85</b>
<b>Time for action – making a horse talk with the say block</b>	<b>85</b>
<b>Time for action – synchronizing and animating the horse</b>	<b>86</b>
<b>Time for action – importing a horse sound</b>	<b>87</b>
Playing supported sound formats	89
<b>Positioning a sprite by its coordinates</b>	<b>89</b>
<b>Time for action – moving the dog based on x and y coordinates</b>	<b>89</b>
Locating sprites with x and y coordinates	91
Creating a new costume	93
<b>Time for action – duplicating, flipping, and switching a sprite's costume</b>	<b>93</b>
Comparing costumes to sprites	94
<b>Composing custom sound effects</b>	<b>95</b>
<b>Time for action – creating drum sound effects</b>	<b>96</b>
Creating sound effects and music	96
<b>Time for action – integrating the dog's joke sequence</b>	<b>98</b>
<b>Navigating the story and coordinating scenes</b>	<b>100</b>
<b>Time for action – hiding the table of contents</b>	<b>100</b>
<b>Time for action – displaying the dog scene</b>	<b>101</b>
Coordinating scenes by backdrop name	102
<b>Time for action – navigating back to the table of contents</b>	<b>103</b>
<b>Summary</b>	<b>105</b>

<b>Chapter 5: Creating a Multimedia Slideshow</b>	<b>107</b>
<b>Importing photos as backdrops</b>	<b>107</b>
<b>Time for action – importing photos from files</b>	<b>108</b>
<b>Working with images</b>	<b>111</b>
Resizing images	112
Using caution while resizing images	112
<b>Adding slideshow controls to display images</b>	<b>114</b>
<b>Time for action – flipping through the photos</b>	<b>114</b>
Related backdrop blocks	116
<b>Playing and recording sounds</b>	<b>118</b>
<b>Time for action – adding a sound from Scratch's library</b>	<b>118</b>
<b>Time for action – recording sounds in the sound editor</b>	<b>119</b>
Understanding sound related blocks	122
<b>Editing sounds</b>	<b>123</b>
<b>Time for action – editing a recorded sound</b>	<b>123</b>
<b>Time for action – appending a sound</b>	<b>124</b>
<b>Time for action – adding sound effects to recordings</b>	<b>125</b>
Reviewing available sound effects	126
<b>Using x and y coordinates to find the position of the mouse's pointer</b>	<b>126</b>
<b>Time for action – using mouse location to hide arrows</b>	<b>127</b>
<b>Time for action – providing user instructions</b>	<b>130</b>
<b>Displaying a project in presentation mode</b>	<b>131</b>
<b>Time for action – presenting a fullscreen slideshow</b>	<b>132</b>
<b>Summary</b>	<b>134</b>
<b>Chapter 6: Making an Arcade Game – Breakout (Part I)</b>	<b>135</b>
<b>Learning about the Breakout game</b>	<b>135</b>
Discovering Pong	136
<b>Time for action – importing and playing the Pong starter project</b>	<b>136</b>
Remixing a legacy Scratch project	138
Moving a sprite with the mouse or arrows	138
Using reporter blocks to set values	138
Customizing the gameplay of the Pong project	139
<b>Time for action – adding the left and right arrow controls</b>	<b>139</b>
Evaluating the y position of the ball to end the game	140
<b>Time for action – determining if the ball is below the paddle</b>	<b>140</b>
<b>Time for action – adjusting the center of a sprite costume</b>	<b>142</b>
<b>Cloning to create identical sprites</b>	<b>144</b>
<b>Time for action – drawing bricks</b>	<b>144</b>
<b>Time for action – cloning bricks</b>	<b>145</b>
Dealing with the cloned sprite	147

<b>Time for action – breaking bricks when I start as a clone</b>	<b>147</b>
Cloning explained	149
Rapid fire shooting with cloning	149
Cloning related blocks	150
<b>Ricocheting with the point in direction block</b>	<b>151</b>
<b>Time for action – changing a sprite's direction</b>	<b>151</b>
Figuring out the direction	153
<b>Time for action – setting the starting position and the direction</b>	<b>153</b>
<b>Time for action – ricocheting off bricks</b>	<b>154</b>
Conditional statements	156
Conditional statements in real life	157
<b>Defining a variable to keep score</b>	<b>157</b>
<b>Time for action – adding a score variable</b>	<b>158</b>
Setting variables For all sprites	159
Setting variables For this sprite only	160
<b>Summary</b>	<b>161</b>
<b>Chapter 7: Programming a Challenging Gameplay – Breakout (Part II)</b>	<b>163</b>
<b>Implementing lives</b>	<b>164</b>
<b>Time for action – adding a variable to track lives</b>	<b>164</b>
<b>Time for action – checking for game over</b>	<b>165</b>
Evaluating multiple programming solutions	167
<b>Adding more bricks to the level with a custom block</b>	<b>169</b>
<b>Time for action – creating a second brick</b>	<b>169</b>
<b>Time for action – drawing rows of bricks with custom blocks</b>	<b>170</b>
Introducing procedures by way of custom blocks	173
Setting custom block inputs	173
<b>Time for action – coordinating the ball play</b>	<b>176</b>
<b>Increasing ball speed and difficulty</b>	<b>177</b>
<b>Time for action – increasing ball speed</b>	<b>177</b>
Using Boolean evaluations	179
Keeping score based on a clone's costume	180
<b>Time for action – decreasing the paddle size based on the clones' costume</b>	<b>180</b>
Considering alternative solutions	183
<b>Time for action – detecting when we clear the level</b>	<b>184</b>
<b>Keeping the score using cloud variables</b>	<b>186</b>
<b>Time for action – keeping a global scoreboard</b>	<b>186</b>
Understanding cloud variables in Scratch 2.0	188
Viewing the cloud data log	188
<b>Summary</b>	<b>190</b>

<b>Chapter 8: Chatting with a Fortune Teller</b>	<b>191</b>
<b>Creating, importing, and exporting lists</b>	<b>192</b>
<b>Time for action – creating lists to store multiple values</b>	<b>192</b>
Working with an item in a list	194
Importing a list	195
<b>Time for action – importing fortunes to a list</b>	<b>195</b>
Exporting a list from Scratch	197
<b>Prompting the player for a question</b>	<b>197</b>
<b>Time for action – asking a question</b>	<b>197</b>
Using stored questions	200
<b>Time for action – validating the seeker's question</b>	<b>200</b>
Deleting the list values	201
Selecting a random fortune	202
<b>Time for action – selecting a random fortune</b>	<b>202</b>
<b>Time for action – counting our fortunes with mod</b>	<b>203</b>
Using magic numbers	206
Creating a custom say fortune block	206
<b>Time for action – creating a custom say fortune block</b>	<b>207</b>
Using the if () then else block	208
Manipulating the text	209
<b>Time for action – ensuring grammatically correct questions</b>	<b>209</b>
Testing your project	210
<b>Creating a keyword scanner</b>	<b>211</b>
<b>Time for action – scanning a text string to build a list of words</b>	<b>211</b>
<b>Summary</b>	<b>215</b>
<b>Chapter 9: Turning Geometric Patterns into Art Using the Pen Tool</b>	<b>217</b>
<b>Drawing basic shapes</b>	<b>218</b>
<b>Time for action – drawing our first square</b>	<b>218</b>
<b>Time for action – building on the square</b>	<b>220</b>
Drawing user-defined shapes	221
<b>Time for action – enabling the user to create custom shapes</b>	<b>222</b>
<b>Time for action – turning triangles into pinwheels</b>	<b>224</b>
Defining procedures for home and shapes	226
<b>Time for action – creating a custom shapes procedure</b>	<b>226</b>
Plotting the coordinates of shapes	228
<b>Time for action – plotting x,y coordinates to draw a square</b>	<b>229</b>
<b>Understanding and using color</b>	<b>230</b>
<b>Time for action – coloring our shapes</b>	<b>230</b>
Understanding color shades	232
Working with the set pen color to () block	232

<b>Time for action – finding a color picker workaround</b>	<b>232</b>
Finding a color to use by its number	233
<b>Time for action – creating a color palette</b>	<b>234</b>
Adding color slider inputs to the shapes project	237
<b>Time for action – limiting color values with a slider</b>	<b>237</b>
<b>Creating asymmetrical patterns</b>	<b>239</b>
<b>Time for action – creating an explosion</b>	<b>240</b>
<b>Turning straight lines into string art</b>	<b>242</b>
<b>Time for action – animating a radar screen</b>	<b>242</b>
<b>Time for action – breaking out of the circle</b>	<b>244</b>
<b>Summary</b>	<b>246</b>
<b>Appendix A: Connecting a PicoBoard to Scratch 1.4</b>	<b>247</b>
<b>Using Scratch 1.4, the PicoBoard, and Raspberry Pi</b>	<b>248</b>
Finding Scratch 1.4	248
<b>Purchasing the PicoBoard</b>	<b>248</b>
<b>Time for action – enabling and testing the PicoBoard support in Scratch 1.4</b>	<b>249</b>
Adding the PicoBoard support to Scratch 2.0	250
<b>Animating webcam images by detecting sound</b>	<b>251</b>
<b>Time for action – creating a talking head</b>	<b>251</b>
Sharing Scratch 1.4 projects online	254
Sensing the environment with the PicoBoard	254
<b>Measuring resistance</b>	<b>254</b>
<b>Time for action – recording the resistance of a thermistor over time</b>	<b>255</b>
Completing a circuit	257
<b>Time for action – charting our measurements</b>	<b>258</b>
Interpreting the graph	261
<b>Time for action – revising the graph</b>	<b>262</b>
<b>Summary</b>	<b>264</b>
<b>Appendix B: Pop Quiz Answers</b>	<b>265</b>
<b>Index</b>	<b>269</b>



---

# Preface

This book demystifies Scratch programming through a variety of projects. The book assumes that you have no programming experience when you begin reading, but by the time you reach the last page, you will be ready to explore your own projects and help other people with Scratch.

The projects start with simpler concepts and get progressively more complicated in terms of programming concepts and design. You will learn how to make multiple-scene stories, think through the logic of a fast-paced arcade game called *Breakout*, interact with a snarky fortune teller, and more. The book's projects tend to demonstrate a programming concept first and then discuss the concept in more detail.

You will receive a balanced introduction to Scratch and universal programming concepts as you create digital stories, animations, and games. With a firm grasp on the fundamentals, you'll be ready to take on more advanced topics and projects.

## What this book covers

*Chapter 1, Welcome to Scratch 2.0*, introduces Scratch and the various types of projects covered in the book.

*Chapter 2, A Quick Start Guide to Scratch*, takes us on a tour of the online Scratch community. In this chapter, we will create our first Scratch animation while learning basic programming concepts such as loops.

*Chapter 3, Creating an Animated Birthday Card*, will guide us through how to use Scratch's built-in paint editor to draw bitmap and vector images. To create the card, we will learn important programming concepts such as project initialization, object naming, and event coordination.

*Chapter 4, Creating a Scratch Story Book*, will guide us through how to build a joke book and coordinate scene changes as a way to navigate through the book. The chapter introduces sound and coordinates as a way to move sprites.



*Chapter 5, Creating a Multimedia Slideshow*, will guide us through how to create a personalized slideshow by uploading files from our computer. We will also work on resizing images and recording slide narrations that can be played on demand.

*Chapter 6, Making an Arcade Game – Breakout (Part I)*, remixes the classic Pong game into our own brick-busting version called Breakout. We'll clone sprites, estimate direction, and create custom variables to develop the framework of the game.

*Chapter 7, Programming a Challenging Gameplay – Breakout (Part II)*, builds on our Breakout game from the previous chapter. Here, we make the gameplay more challenging by programming the ball speed and reducing the paddle size based on the gameplay. Important concepts include custom procedures, Boolean values, and cloud data.

*Chapter 8, Chatting with a Fortune Teller*, deals with our game of fortune, where a fortune teller will provide a random fortune in response to the user's typed question. We will work with lists, track intervals with `mod`, and split words apart to identify individual words.

*Chapter 9, Turning Geometric Patterns into Art Using the Pen Tool*, combines all the programming concepts we've learned so far to draw art using simple math equations, polygons, and string art. The projects will show you how to take user-defined values and turn them into shapes. This chapter also explains how to apply color and shades to Scratch projects.

*Appendix A, Connecting a PicoBoard to Scratch 1.4*, emphasizes on projects that use a computer's webcam and the PicoBoard, which is an add-on device capable of running on Scratch 1.4 on the Raspberry Pi. The PicoBoard project incorporates an experiment that measures the resistance of warming water using a thermistor and generates graphs for it.

## What you need for this book

To create projects using the Scratch 2 project editor, you need a relatively recent web browser (Chrome 7 or later, Firefox 4 or later, or Internet Explorer 7 or later) with Adobe Flash Player Version 10.2 or later installed. Scratch 2 is designed to support a screen resolution of 1024 x 768 or larger. If your computer doesn't meet these requirements, you can try downloading and installing Scratch 1.4, which you can still use to share projects to the Scratch 2 website.

An offline Scratch 2 editor is also available. You can also still use Scratch 1.4. Note that you can have both Scratch 1.4 and 2 on your computer.

The software to download are as follows:

- ◆ The Scratch 2 offline editor can be downloaded from the following link:  
<http://scratch.mit.edu/scratch2download>
- ◆ The Scratch 1.4 editor can be downloaded from the following link:  
[http://scratch.mit.edu/scratch\\_1.4](http://scratch.mit.edu/scratch_1.4)

## Who this book is for

The author approaches the content in this book with the belief that we are all teachers and that you are reading this book not only because you want to learn, but also because you want to share your knowledge with others. Motivated students can pick up this book and teach themselves how to program because the book takes a simple, strategic, and structured approach to learning Scratch.

Parents can grasp the fundamentals so that they can guide their children through introductory Scratch programming exercises. It's therefore perfect for homeschool families. Teachers of all disciplines from Computer Science to English can also quickly get up to speed with Scratch and adapt the projects for use in the classroom.

## Conventions

In this book, you will find several headings that appear frequently.

To give clear instructions of how to complete a procedure or task, we use:

### **Time for action – heading**

- 1.** Action 1
- 2.** Action 2
- 3.** Action 3

Instructions often need some extra explanation so that they make sense, so they are followed with:

### ***What just happened?***

This heading explains the working of tasks or instructions that you have just completed.

You will also find some other learning aids in the book, including:

### **Pop quiz – heading**

These are short multiple-choice questions intended to help you test your own understanding.



## Have a go hero – heading



These practical challenges give you ideas for experimenting with what you have learned.

You will also find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "Conditional statements are used to check whether a statement is `true` or `false`. For example, `if 4 > 0` is a conditional statement."

**New terms** and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "The top of the page contains the **Create**, **Explore**, and **Discuss** links."

 Warnings or important notes appear in a box like this. 

 Tips and tricks appear like this. 

## Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to [feedback@packtpub.com](mailto:feedback@packtpub.com), and mention the book title through the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on [www.packtpub.com/authors](http://www.packtpub.com/authors).

## Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

## Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at <http://www.packtpub.com>. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

## Downloading the color images of this book

We also provide you a PDF file that has color images of the screenshots/diagrams used in this book. The color images will help you better understand the changes in the output. You can download this file from: [https://www.packtpub.com/sites/default/files/downloads/07240T\\_coloredimages.pdf](https://www.packtpub.com/sites/default/files/downloads/07240T_coloredimages.pdf).

## Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website, or added to any list of existing errata, under the Errata section of that title.

## Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at [copyright@packtpub.com](mailto:copyright@packtpub.com) with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

## Questions

You can contact us at [questions@packtpub.com](mailto:questions@packtpub.com) if you are having a problem with any aspect of the book, and we will do our best to address it.



---

# 1

## Welcome to Scratch 2.0

*I assume you're reading this book because you want to learn how to create interactive stories, animations, and games using Scratch, or you want to learn Scratch so that you can teach someone else how to program. It matters not whether your classroom is in a middle school, a home school environment, an after-school workshop, or a weekend coding project with your son or daughter. We are all teachers. That's the perspective of this book. You'll learn how to create projects using Scratch so that you can teach someone else, but no programming knowledge is expected.*

In this chapter, we will:

- ◆ Review what Scratch is and how we can use it
- ◆ Learn more about the types of projects we will create in this book
- ◆ Explore an example project from the Scratch website and review the project editor

Whether you're 8 or 80, the Scratch programming language provides a beginner-friendly computer programming environment that enables you to create digital projects. Success with Scratch comes quickly. You won't find any quirky syntax to learn, and you won't make any typing mistakes that prevent your program from running.

Creating a project in Scratch is as easy as snapping the color-coded blocks together. This environment allows us to see the positive results quickly. In addition to this, Scratch helps turn passive users into creators.

You'll find comfort in Scratch's building-block approach to create animations, games, and stories. After using Scratch, programming will make sense. It will seem easy. It will bring a smile to your face, and you'll be able to cope with technical concepts in the future.

## About Scratch

Mitch Resnick and the Lifelong Kindergarten Group at the Massachusetts Institute of Technology (MIT) in the Media Laboratory developed Scratch as a teaching language primarily for 8 – 16 year olds, but there's nothing stopping the rest of us from enjoying the Scratch experience and sharpening our creative minds.

## Encouraging everyone to think programmatically

The natural reaction of people is to see Scratch as a means of teaching computer science and integrating it into classrooms of all levels. There are teachers who use Scratch across a variety of subjects as seen on the ScratchEd site. The ScratchEd site caters to the educational community and aggregates a lot of Scratch resources, including lesson plans and tips. However, the approach and thoroughness of the included material varies greatly. You can check out ScratchEd at <http://scratch.ed.media.mit.edu/>.

While writing this book, I did not set out to write a computer science textbook. It's quite simply a tutorial for people who want to learn how to use Scratch to create stories, animations, games, or art. It primarily addresses the parents, home school families, and teachers who may not be programmers themselves but want a fun way to help their children become more digitally literate. Everyone, however, can use this tutorial to learn Scratch, and many young students have worked through the Version 1.4 of Scratch of this book. I expect young scratchers will be more than capable of working through the projects in this edition.

My underlying belief is that knowing how to program can benefit everyone, but not everyone needs to be a programmer. The mental work required to create a program inherently develops an understanding of how computers work, sharpens our critical thinking skills, and gives us lots of practice at solving problems.

There's also an increasingly popular idea that sometimes we want to create applications for personal use. Of course, system administrators have always created custom scripts to help automate repetitive tasks. However, modern applications such as Scratch or the **MIT App Inventor** (originally developed by Google) make it incredibly easy to create programs for personal use or with the intention of sharing it with a small group of friends, which counters the perspective that learning to program is synonymous with wanting to be a professional programmer or an application developer.

When you have a little bit of programming knowledge, you'll approach non-programming problems in a different way. For example, I've used programming as a marketer to manage search engine optimization and keyword research on business websites. I've also used my programming knowledge to write automated software tests.

Bottom line, programming becomes a tool in your problem-solving toolbox. This is the key to understanding how I approach this Scratch tutorial. I want the computer scientists to come along for the ride, but I'm catering to a broader audience.

## Sample Scratch uses

I couldn't begin to suggest every possible way for you to use Scratch; that's why we have an imagination. However, here are a few ideas to get you started:

- ◆ Use Scratch to teach yourself or your students how to program. That's the obvious one.
- ◆ Use Scratch to demonstrate Math concepts. Scratch can also demonstrate the  $x$  and  $y$  coordinate system in an interactive way.
- ◆ Use Scratch to inspire your kids to read and write. Find a story and animate each scene or encourage them to animate the story. Turn their haiku into a Scratch project.
- ◆ Have a child who only wants to play video games? Make a deal. Your child can only play the games he creates or remixes with Scratch.

As you work through the examples in this book, write down your project ideas no matter how hard, easy, obvious, or silly they seem. The next one might be your best idea yet.

## Computational thinking

Learning: we do it for life. We should help our children develop skills that will help them keep learning and solving problems in an increasingly digital environment. Using Scratch, we will learn how to design, think, collaborate, communicate, analyze, and program in a computer language.

You may frame the Scratch approach as computational thinking. According to Wikipedia:

*Computational thinking is a problem solving method that uses computer science techniques. The term computational thinking was first used by Seymour Papert in 1996.*

By the time we make our cat dance for the first time, we'll forget all about the academic research and theories behind Scratch. Instead, we'll focus on having fun and creating the next project.



- [The Climb: Tragic Ambitions on Everest here](#)
  - [The Wild Garden \(Expanded Edition\) for free](#)
  - [download Just for Now \(Sea Breeze, Book 4\) pdf, azw \(kindle\), epub](#)
  - [Understanding Figurative Language: From Metaphor to Idioms \(Oxford Psychology Series\) here](#)
  - [Beginning Microsoft SQL Server 2012 Programming book](#)
  - [download online The Mime Order \(The Bone Season, Book 2\)](#)
- 
- <http://thermco.pl/library/The-Somme--The-Darkest-Hour-on-the-Western-Front.pdf>
  - <http://conexdx.com/library/The-Wild-Garden--Expanded-Edition-.pdf>
  - <http://bestarthritiscare.com/library/Just-for-Now--Sea-Breeze--Book-4-.pdf>
  - <http://damianfoster.com/books/Understanding-Figurative-Language--From-Metaphor-to-Idioms--Oxford-Psychology-Series-.pdf>
  - <http://dadhoc.com/lib/Beginning-Microsoft-SQL-Server-2012-Programming.pdf>
  - <http://nexson.arzamaszev.com/library/The-Mime-Order--The-Bone-Season--Book-2-.pdf>