



# The Blender Python API

Precision 3D Modeling and Add-on  
Development

---

Chris Conlan

Apress®

# The Blender Python API

Precision 3D Modeling and Add-on  
Development



Chris Conlan

Apress®

***The Blender Python API: Precision 3D Modeling and Add-on Development***

Chris Conlan  
Bethesda, Maryland  
USA

ISBN-13 (pbk): 978-1-4842-2801-2  
DOI 10.1007/978-1-4842-2802-9

ISBN-13 (electronic): 978-1-4842-2802-9

Library of Congress Control Number: 2017944928

Copyright © 2017 by Chris Conlan

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Cover image designed by Freepik

Managing Director: Welmoed Spahr  
Editorial Director: Todd Green  
Acquisitions Editor: Susan McDermott  
Development Editor: Laura Berendson  
Technical Reviewer: Justin Mancusi  
Coordinating Editor: Rita Fernando  
Copy Editor: Kezia Endsley  
Composer: SPi Global  
Indexer: SPi Global

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://www.springeronline.com). Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail [rights@apress.com](mailto:rights@apress.com), or visit <http://www.apress.com/rights-permissions>.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at [www.apress.com/9781484228012](http://www.apress.com/9781484228012). For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*For my teachers and colleagues at the University of Virginia Department of Statistics.*

# Contents at a Glance

<b>About the Author .....</b>	<b>xiii</b>
<b>About the Technical Reviewer .....</b>	<b>xv</b>
<b>Introduction .....</b>	<b>xvii</b>
<b>■ Chapter 1: The Blender Interface.....</b>	<b>1</b>
<b>■ Chapter 2: The bpy Module.....</b>	<b>11</b>
<b>■ Chapter 3: The bmesh Module.....</b>	<b>27</b>
<b>■ Chapter 4: Topics in Modeling and Rendering.....</b>	<b>43</b>
<b>■ Chapter 5: Introduction to Add-On Development.....</b>	<b>65</b>
<b>■ Chapter 6: The bgl and blf Modules.....</b>	<b>87</b>
<b>■ Chapter 7: Advanced Add-On Development.....</b>	<b>105</b>
<b>■ Chapter 8: Textures and Rendering .....</b>	<b>123</b>
<b>Index.....</b>	<b>137</b>

# Contents

<b>About the Author .....</b>	<b>xiii</b>
<b>About the Technical Reviewer .....</b>	<b>xv</b>
<b>Introduction .....</b>	<b>xvii</b>
<b>■ Chapter 1: The Blender Interface.....</b>	<b>1</b>
<b>The Default Blender Interface .....</b>	<b>1</b>
3D Viewport .....	3
Header Menu .....	3
Properties Window.....	3
Tool Shelf and Tool Properties .....	3
Timeline .....	3
<b>The Scripting Interface.....</b>	<b>3</b>
Text Editor.....	5
Command Log.....	5
Interactive Console .....	6
<b>Customizing the Interface .....</b>	<b>6</b>
<b>Starting Blender from the Command Line (for Debugging) .....</b>	<b>7</b>
<b>Running Our First Python Script.....</b>	<b>8</b>
Finding the Function.....	8
Testing the Function .....	8
Writing the Script.....	8
<b>Conclusion.....</b>	<b>9</b>

■ **Chapter 2: The bpy Module** ..... **11**

**Module Overview**..... **11**

    bpy.ops ..... 11

    bpy.context ..... 11

    bpy.data ..... 12

    bpy.app ..... 12

    bpy.types, bpy.utils, and bpy.props ..... 12

    bpy.path ..... 12

**Selection, Activation, and Specification** ..... **12**

    Selecting an Object..... 13

    Activating an Object..... 14

    Specifying an Object (Accessing by Name) ..... 15

**Pseudo-Circular Referencing and Abstraction** ..... **16**

**Transformations with bpy**..... **17**

**Visualizing Multivariate Data with the Minimal Toolkit**..... **20**

    Visualizing Three Dimensions of Data ..... 21

    Visualizing Four Dimensions of Data ..... 22

    Visualizing Five Dimensions of Data..... 24

    Discussion ..... 26

**Conclusion**..... **26**

■ **Chapter 3: The bmesh Module** ..... **27**

**Edit Mode** ..... **27**

**Selecting Vertices, Edges, and Planes**..... **28**

    Switching Between Edit and Object Modes Consistently ..... 28

    Instantiating a bmesh Object..... 29

    Selecting Parts of a 3D Object..... 30

**Edit Mode Transformations**..... **31**

    Basic Transformations ..... 31

    Advanced Transformations ..... 32

Note on Indexing and Cross-Compatibility ..... 34

Global and Local Coordinates ..... 35

Selecting Vertices, Edges, and Faces by Location ..... 37

Checkpoint and Examples ..... 39

Conclusion ..... 42

■ **Chapter 4: Topics in Modeling and Rendering** ..... **43**

    Specifying a 3D Model ..... 43

        Specifying Meshes ..... 43

        Specifying Textures ..... 44

    Common File Formats ..... 45

        Wavefront (.obj and .mtl) ..... 45

        STL (STereoLithography) ..... 46

        PLY (Polygon File Format) ..... 47

        Blender (.blend) Files and Interchange Formats ..... 48

    Minimal Specification of Basic Objects ..... 48

        Definition of a Cube ..... 48

        Naive Specification ..... 48

        Using Indices to Share Vertices and Normals ..... 51

        Using Coplanar Vertices to Reduce Face Count ..... 52

        Using Face Vertices to Simplify Indices ..... 53

        Representing a Cube as a Primitive ..... 55

        Summary ..... 55

    Common Errors in Procedural Generation ..... 56

        Concentric Normals ..... 56

        Flipped Normals ..... 60

        Z-Fighting ..... 61

    Conclusion ..... 63

■ <b>Chapter 5: Introduction to Add-On Development</b> .....	<b>65</b>
A Simple Add-On Template.....	65
Components of Blender Add-Ons .....	69
The bl_info Dictionary .....	69
Operators and Class Inheritance (bpy.types.Operator) .....	70
Panels and Class Inheritance (bpy.types.Panel) .....	71
Register() and Unregister() .....	72
Scene Properties and bpy.props.....	74
Precision Selection Add-On Example .....	79
Code Overview for Our Add-On.....	79
The poll() Classmethod .....	84
EnumProperty Variables .....	85
Preparing Our Add-On for Distribution.....	85
Conclusion.....	85
■ <b>Chapter 6: The bgl and blf Modules</b> .....	<b>87</b>
Instantaneous Drawing .....	87
Handlers Overview .....	87
Clock Example .....	88
Managing Handlers.....	89
Types of Handlers .....	89
Persistent Handlers .....	90
Handlers in blf and bgl .....	91
Example Add-On.....	92
Drawing Lines and Text .....	99
Declaring Button-Activated Drawing Functions.....	100
Declare Main Drawing Function .....	101
Declaring the Operator with Handlers .....	101
Declaring the Panel with Dynamic Drawing .....	101
Extending our bgl and blf Template.....	101
Conclusion.....	103

<b>■ Chapter 7: Advanced Add-On Development</b> .....	<b>105</b>
Developing in Blender’s Filesystem .....	105
Creating an Add-on in the Filesystem.....	107
Using F8 to Reload Add-Ons .....	109
Important Takeaway .....	109
Managing Imports.....	109
IDEs for In-Filesystem Development .....	110
Lightweight (Notepad++, Gedit, and Vim) .....	110
Midweight (Sublime Text, Atom, and Spyder) .....	110
Heavyweight (Eclipse PyDev, PyCharm, and NetBeans) .....	111
Compiling Blender as a Python Module.....	111
Summary .....	111
Best Practices for External Data.....	111
Using File Interchange Formats.....	112
Using Hardcoded Python Variables.....	112
Algorithmic Manipulation of Primitives.....	113
Summary .....	115
Advanced Panel Creation .....	116
Panel Organization.....	116
Panel Icons .....	119
Conclusion.....	121
<b>■ Chapter 8: Textures and Rendering</b> .....	<b>123</b>
Vocabulary of Textures .....	123
Types of Influence in Blender .....	123
Types of Textures in Blender.....	125
Adding and Configuring Textures .....	126
Loading Textures and Generating UV Mappings .....	126
Textures Versus Materials in Blender .....	129
UV Coordinates and Loops.....	129
Another Note on Indexing and Cross-Compatibility.....	130

<b>Removing Unused Textures and Materials .....</b>	<b>130</b>
<b>Rendering Using Blender Render .....</b>	<b>131</b>
Adding Lights.....	131
Adding Cameras .....	132
Rendering an Image .....	133
<b>Conclusion.....</b>	<b>136</b>
<b>Index.....</b>	<b>137</b>

# About the Author



**Chris Conlan** began his career as an independent data scientist specializing in trading algorithms. He obtained his degree in statistics from the University of Virginia where he established himself as an expert in automated trading. His passion for intuitive data visualization introduced him to various 3D modeling and virtual reality suites that he hopes to better integrate into the lives of data scientists. He is currently managing development of private technology companies in high-frequency Forex, machine vision, and precision 3D modeling.

# About the Technical Reviewer



**Justin Mancusi** attended the University of Virginia, where he obtained degrees in computer science and mathematics. In the past, he has worked as an independent consultant at the intersection of computing and statistics. He is experienced in a breadth of computational topics including advanced optimization, computational statistics, and stochastic processes.

# Introduction

This text details the development and use of 3D modeling tools in Blender's Python API. We challenge the perception of Blender as purely an artist's tool by building precise data-driven models. Simultaneously, we teach you how aid and enable artists by deploying custom tools in the familiar Blender environment.

The knowledge presented in this text is the result of a deep understanding of not only Blender's documentation and source code, but also of the source code of add-ons written by Blender's core developers. The author has discovered many useful functionalities that are, as of the time of writing, undocumented. Thankfully, we as users can stay on the cutting edge by listening to and learning from those developers. This text unifies well-documented introductory material and undocumented advanced material to create a powerful reference.

This book is packed with code examples and screenshots of powerful scripts and add-ons. We include scripts to automate precise tasks that would otherwise be very difficult to implement by hand. In addition, we build add-ons that augment Blender's existing functionalities with new tools, objects, and customization options.

## Definitions

*3D modeling* is the art of manipulating data to create 3D representations of objects and environments. 3D artists use the following tools and techniques to build 3D models.

- *Manual modeling* involves the artist interacting with a software interface. This can be:
  - Using a 3D modeling suite (Blender, Maya, or 3ds Max) to create and edit objects by hand
  - Playing video games with 3D building elements (Minecraft, Fallout 4, or Sims)
  - Manually inputting data into a 3D object file (.obj, .stl, or .glTF)
- *Automated Modeling* involves algorithmically generating 3D models. This can be:
  - Procedural generation of environments and characters in video games
  - Generating detailed models of buildings from architectural specifications
  - Producing 3D-printed art from fractal algorithms

- *Primitives* are the basic building blocks of 3D models. Though there are no strict rules on what constitutes a primitive, these can be:
  - Simple closed shapes like planes, cubes, and pyramids
  - Simple curved shapes like spheres, cylinders, and cones
  - Complex shapes like tori (plural of torus), Bezier curves, Nurbs surfaces

*3D models* are data representations of objects and environments. 3D models have the following components.

- *Data formats* allow models to differentiate and specialize by application. Every type of 3D model has a format by which it is specified. These include:
  - Suite-specific formats like `.blend` for Blender, `.3ds` for 3ds Max, and `.ma` for Maya
  - Renderer-specific formats like `.babylon` for BabylonJS, `.json` geometry descriptor for 3JS, and `.glsl` for OpenGL shaders
  - Minimalistic interchange formats like `.obj` and `.stl`
- *Vertices and faces* define the points and the surfaces connecting those points in 3D space.
  - Vertices are triplets of real numbers 3D space, or traditional  $(x, y, z)$  coordinates of each point of the object.
  - Faces are triplets of integers, where  $(i, j, k)$  represents the triangle in 3D space formed by the  $i$ -th,  $j$ -th, and  $k$ -th vertex.

## Prerequisite Knowledge for This Book

This book covers Blender version 2.78c running Python 3.5.2. Most examples run on Blender 2.70 and greater, and the concepts apply to Blender generally. Nonetheless, it is recommended that readers use Blender 2.78c to best follow along. As we discuss the history and development of Blender and the Python language, we will point out programming practices that are not likely to work on past and future versions.

We assume a basic working knowledge of Blender and Python 3. Familiarity with any version of Blender 2.60 or greater is sufficient. Similarly, pure Python 2 programmers will have no problem following along.

## Material Overview

This text introduces knowledge and sequentially builds on it to create more and more complete and complex software solutions. We introduce and discuss the following major topics.

### Chapter 1: The Blender Interface

There are many individual interfaces that make up Blender. The core interfaces are highly scriptable because almost every possible user interaction is tied directly to a Python function. We establish some familiarity with those parts of the interface especially important for Python programming.

The Blender interface will act as both the deployment and development environment for your software. We discuss unique considerations for programming and testing Python while remaining in the Blender interface.