Unleash Python and take your small readers on an adventurous ride through the world of programming

Python Projects for Kids

Jessica Ingrassellino

What you will learn from this book

- Start fiddling with Python's variables, build functions, and interact with users
- Build your own calculator using the math library
- Train Python to make logical decisions
- Work with moving 2D objects on screen
- Understand the pygame library and build your very own game!
- Write a cool program to manage inventories in your backpack

Who this book is written for

This book is for kids aged 10 and over. This book is intended for absolute beginners who lack any knowledge of computing or programming languages and want to get started in the world of programming.

In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'Welcome! Let's Get Started'
- A synopsis of the book’s content
- More information on Python Projects for Kids
About the Author

Jessica Ingrassellino is a multi-talented educator, business leader, and technologist. She received her EdD from Teachers College, Columbia University for music education with an emphasis on assessment.

Jessica is currently employed as the lead software engineer in testing at Bitly, New York City. She transitioned from a teaching career of 10 years to a technology career through a balance of freelance work and social media exposure. Jessica’s current work focuses on using Python to develop automated testing tools. She is an ASTQB certified quality assurance engineer with experience in testing web, mobile, and backend applications.

In addition to working at Bitly, Jessica remains committed to education and has founded http://www.teachcode.org/, a nonprofit that teaches computer programming skills to teachers and students in urban or underserved populations through Python and 2D game programming. This new initiative will give teachers the support they need through a standards-referenced curriculum, student-engaging activities, and access to experts in the field of technology.
Preface

As you can guess from the title, this book is designed to teach the basic concepts of Python to kids. This book uses several mini projects so that kids can learn how to solve problems using Python.

Python has grown to become a very popular language for programming web apps, analyzing data, and teaching people how to write code. Python is known for being a simple language to use because it is read much like natural languages, yet it is able to do data analysis very quickly, making it a great language to create websites that handle a lot of data. Another nice thing about Python that makes it fun to use is that people have been working on game libraries, such as pygame, so that people can create graphics programs with Python. The use of simple graphics to make short games is a fun way to learn programming constructs and is especially good for visual learners.

What this book covers

Chapter 1, Welcome! Let’s Get Started, discusses Python and setting up a Python development environment on Windows, Mac, and Linux operating systems.

Chapter 2, Variables, Functions, and Users, covers Python data types and functions, as well as how to program Python to get information from the user, store that information, and use it later.

Chapter 3, Calculate This!, uses Python to make a calculator that has multiple mathematical functions. We also learn about working in our file structure and the proper way to save code files.

Chapter 4, Making Decisions – Python Control Flows, covers the use of if, elif, and else, as well as the use of for and while loops, in order to help create programs that make decisions based on user actions.
Preface

Chapter 5, Loops and Logic, builds upon what we have learned in the previous chapters and allows us to build a number guessing game. We will build easy and difficult versions of the game.

Chapter 6, Working with Data – Lists and Dictionaries, explains how to use lists and dictionaries to store data. The differences between lists and dictionaries are explained, and we spend time building small lists and dictionaries as well.

Chapter 7, What's in Your Backpack?, allows us to use functions, loops, logic, lists and dictionaries to build a different kind of guessing game. We will also learn about nesting dictionaries and lists.

Chapter 8, pygame, talks about a popular graphical library that is used in Python to make small games. We will learn the fundamental aspects of this library and experiment with some code.

Chapter 9, Tiny Tennis, this game is a clone of a popular game. We will re-create the game using all of the skills that we have learned throughout the book. This is the major project of the book.

Chapter 10, Keep Coding!, shows you all the opportunities that will arise once you read this book.

Appendix, Quick Task Answers, has the answers to all the quick task questions within the chapters.
Welcome! Let's Get Started

If you've picked up this book, then you are taking your first step toward building amazing projects using code. Some of you might want to make games, while others might want to learn more about how all of your favorite websites and apps actually work. If you follow the exercises in this book, you'll be able to do the following:

- Create fun games to play with your family and friends
- Learn about the inner workings of your apps
- Learn how to take charge of your computer

Python projects for you

In this book, you will learn Python code. Specifically, you will learn how to design a computer program from the very beginning. It doesn't matter if you have never coded before because each exercise in this book is designed to get you ready to code. If you have coded before, you will find that this book has some really helpful exercises that can help make your code even better. Additionally, there are some more advanced projects toward the end of the book, which you should definitely take a look at!

What can you do with Python?

If you take a look at the Web and search for Python jobs, you will find that many of the highest paying jobs are in Python. Why?

Python is a very flexible and powerful language in the following ways:

- It can be used in order to go through millions of lines of data
- Python can search for information on a website without having to go to the website itself
- It is even used to host and design websites
Welcome! Let's Get Started

So, what will it take to learn Python? If you have never programmed, you will probably want to follow each lesson in order so that you can build the skills you need to make a game or another kind of computer program. The final project in this book will be a game. If you have some other programming experience, such as making modifications to your computer games, using programs such as Scratch or Logo or trying some of the free programming classes on the Internet, then you might decide to skim this book first to see what you already know. It is still recommended that you follow the contents of this book in the order they are presented, as each project builds on the projects that were explained in the previous chapter.

Why you should learn Python
Python teaches all of the basics of an object-oriented programming language, and it is still very powerful. In fact, many Internet companies, most notably Mozilla Firefox and Google, use Python in part or all of their products! Python has also been used to build Django, a free framework to make websites.

It has also been used to build many small video games by people learning about it as well as more advanced programmers. Finally, Python can be used to quickly read and analyze millions of lines of data very quickly! By learning Python, you will be prepared to build a variety of interesting projects, and you will gain the skills necessary to learn other programming languages if you choose to do so.

The prerequisites of Python
Before you get started, you need the following basic materials:

- A computer that can run Windows 7 or higher, Mac OS X 10.6 or higher, or Ubuntu 12.4 or higher. You may also use a Raspberry Pi as it comes preinstalled with Python, pygame, and the other software needed to complete the projects in this book.

- An Internet connection is necessary as some of the software you need to install on your computer might not be installed already. For example, Windows operating systems do not come with Python preinstalled, so an Internet connection will be needed; pygame is also not preinstalled on Windows, Mac, or Linux systems.

- Along with an Internet connection, you will also need a web browser, such as Firefox, Safari, Chrome, or Internet Explorer, which will allow you to visit the Python documentation pages.
Setting up your computer
There are many different computer operating systems, but the most common operating systems are Macintosh (Mac), Windows, and Linux. You should follow the installation steps that go with your operating system. There are some subtle but important differences between the systems.

For the projects in this book, we will be using Python 2.7. While there are higher versions than this (3.x), these versions do not work dependably with pygame on Windows, Mac, or Ubuntu Linux as yet. However, this book will be written to use conventions that work in both versions of Python so that projects are easily completed on Raspberry Pi (which uses Python 3.x that's been specially configured with pygame) with just a few modifications. These modifications will be duly noted.

For Mac and Ubuntu Linux users
Mac and Linux systems share enough similarities that people who use either Mac or Linux can follow the same set of instructions. These instructions will make note of any differences between Mac and Ubuntu Linux.

Python 2.7
At the time of writing, Mac OS X El Capitan comes with Python 2.7 preinstalled, so nothing extra needs to be done at this point.

Ubuntu Linux 15.10 has Python 2.7.10 installed by default, so users of this latest (as of writing this) version of Linux also need to do nothing extra at this point.

Terminal – the command line and the Python shell
Mac and Ubuntu Linux users have Python by default, but finding Python is tricky if you don’t know where to look. There is a program called Terminal on both Mac and Linux operating systems. This program allows you to exercise a lot of control over your computer in these ways:

- On a Mac, go to Finder | Applications | Utilities and click on Terminal. The terminal application will open up, and you should see a small, white window on your screen.
• Ubuntu users can search for terminal on their desktops, and the program will show up in their Start menu. When you click on the terminal, you will see a small, black window on your screen.

• The terminal also functions as a Python shell when a command is given to run Python. We will learn about this later.

Text editor
A text editor is a helpful tool for writing and editing Python programs. The terminal is a nice place to test snippets of Python code, but when we want to edit and save the code in order to use it over again, we will need a text editor. Although both Mac and Linux systems come with a text editor, there are some very nice, free editors that have good features. jEdit is one of these editors.

For Mac and Linux, go to http://www.jedit.org/ and download jEdit. Follow the installation instructions.

To successfully complete all of the exercises in this book, you will often need to keep both the terminal and text editor open at the same time on your screen.
This is what the text editor application, jEdit, looks like in Mac and Linux:

![jEdit interface](image)

For Windows users

Windows users, this setup might require help from your parents. Since Python is not installed by default on Windows, some system adjustments need to be made to successfully run Python on your computer. If you are feeling uncertain about performing these system changes yourself, make sure to ask for help:

1. First, you will need to download version 2.7.11 of Python.


   With Windows, you need to figure out if you are running 32-bit or 64-bit so that you can download the correct version of Python. To help you to determine which one is correct, visit [http://windows.microsoft.com/en-gb/windows/32-bit-and-64-bit-windows#1TC=windows-7](http://windows.microsoft.com/en-gb/windows/32-bit-and-64-bit-windows#1TC=windows-7).

   Download x86only if your computer is running 32-bit Windows. Most users will download the x86-64 version of Python.
Welcome! Let's Get Started

2. Choose the executable installer, and you will see the download progress.
3. When the download is complete, you will see a prompt to run Python. Click on Run.
4. An install prompt will come up, and when it does, look at the bottom of the window and click on the box next to Add Python 2.x to Path. Then, select Install Now.
5. Follow the installation instructions. Each step may take a few minutes. Once the installation is done, you will have an icon for Python 2.7.11, which you can find by searching for Python in the Windows search bar. This will open a special Python shell from where you can run and test the Python code.

Command prompt
In Windows 10, you will see a terminal called the command prompt. The command prompt is significantly different in Windows than it is on Mac or Linux.

To find the command prompt in Windows 10, perform these steps:

1. Go to the search bar at the bottom of the screen and search for cmd or command.
2. When you do, you will see the command prompt desktop app appear. Click on this app to open the command prompt, which looks like this:

![Command Prompt](image)

Text editor
In Windows, Notepad is the default text editor. However, Notepad++ is a much better substitute.
To get Notepad++, perform these steps:

1. Go to https://notepad-plus-plus.org/ and download the latest version.
2. Once the program has been downloaded, click on Run.

Notepad++ looks like this:

Write and run your first program in the command line

Now that you are set up, it is time to write your first line of code in Python! This line of code is almost a tradition for people who are programming for the first time, and it allows us to use one of the most basic, but most useful, functions in the Python language.

First, you need to start running a Python shell. On Mac or Linux, open your terminal and type this:

```
python
```

In the Mac or Ubuntu terminal, your resulting Python shell will look like this:

```
>>> 
```

In Windows, type `Python` in the search bar at the bottom of the page. Then, select Python 2.7.11 from your apps. You will also have a Python shell open:

```
>>> 
```

Once you see this symbol, your computer is now ready to work with the Python code. In your terminal or IDLE, type the following:

```
>>> print("Hello, world!")
```
Welcome! Let's Get Started

Once you have typed this, double-check to make sure that all of the spaces are exactly as they've been written. In Python, every space actually matters. Every punctuation mark matters. Once you have checked your code, hit Enter.

What is your result or the output of your code? If the output looks like the following image, then great! You typed all of your code properly so the computer will understand what you want it to do. The expected output will be similar to what is shown here:

```
$ python
Python 2.7.10 (default, Oct 14 2015, 16:09:02)
[GCC 5.2.1 20151010] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello, world!')
Hello, world!
>>> 
```

For Windows users, the output window will look like this:

```
C:\Python27\python.exe
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec 5 2015, 20:40:30) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello, world!')
Hello, world!
``` 

So, if your output does not look like the preceding code, you need to figure out what's wrong with it. Here are some of the reasons for this:

- Did you make a typing error?
- Did you forget to use parenthesis or round brackets () for the words 'Hello, world!'
- Did you forget to use the single quotation marks for Hello, world?

If you still have a problem, compare your code to the sample input code and fix any mistakes. Then, try to run the code again.
Python is what is called a *case-sensitive* language. Python cares about uppercase, lowercase, and whitespace. You need to watch what you type. You might get some strange messages from your computer if you make a typing mistake or a syntax error.

**Make yourself a work folder**

Before we get started on any large projects, we need to make a work folder. In the next chapter, you will start writing whole files of code that need to be run; therefore, we will need a place to put those files. Since you are setting up now, let's make a folder.

If you are very good at getting around your computer, you can put your folder wherever you want to.

If you are not too good at getting around your computer, you will probably want to put your folder on your desktop.

On both Mac and Windows machines, you can right-click somewhere on your desktop wallpaper, and a box will pop up with several options. One of the options will say **New**, and when you hover over **New**, you will get several other options. Choose **New Folder**, and a new folder icon will appear on your desktop. It will be named *untitled*, so you should give it a better name.

To find your folder in the Mac or Linux terminals, respectively, open one of them and perform these steps:

1. Run the `cd ..` command until you are at the root, which is often the name you have given to your computer. You may need to run the command three or four times.
2. Now, you will be able to run python programs by typing `python3.5 program.py`.

To find your folder in the Windows command line, open the command prompt and perform these steps:

1. Run the `cd ..` command until you are at the root or `C:\`. You may need to run the command three or four times.
2. Now, you will be able to run Python programs by typing `python program.py`. 
Welcome! Let's Get Started

A quick task for you
Now that you have finished this chapter, can you answer these questions?

Q1. What is a terminal (Mac/Linux) or command prompt (Windows)?
   1. A terminal is used to put data into or get data out of a computer without using the icons on the desktop.
   2. A terminal can be used to write computer programs.
   3. A terminal can be used to do complex work, such as giving hints on Python code.
   4. A terminal can do all of the above.

Q2. When you first open the terminal/command prompt, what do you need to do so that you can start reading and writing the Python code?
   1. Start typing the code.
   2. Type the word `python`.
   3. Wait for Python to start.
   4. None of the above; do something different.

Q3. How is the Python shell different from the command line?
   1. They are exactly the same.
   2. The command line cannot run Python commands.
   3. The Python shell is started by typing the word `python` into the command line.
   4. The Python shell can be used to test lines of Python code.

[оро] Compare your answers with those you find at the back of the book. [оро]
Summary

If you are reading this, it is because you have made it through some of the tricky work of getting ready to learn to program projects with Python. Congratulations! Setting it up is always tough. Hopefully, you learned a bit more about the tools on your computer, such as the text editor and terminal that every programmer uses to do their daily work. Also, you learned about the Python `print()` function, and you should now be able to print out messages in your Python shell. The fun is just beginning as we have so much more to learn!

In the next chapter, you will learn about the building blocks of Python programs. We’ll start with variables and learn about all the different kinds of information we can put in them. Then, we will build some functions that put these variables together and help us make blocks of code that have special jobs. Finally, we will even learn how to make a computer ask a user questions and store their answers so that our programs can become interactive!
Where to buy this book

You can buy Python Projects for Kids from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.