

# A Brief Introduction to Python Programming

---

What is Python?  
How is it relevant to me?  
How do I get started?

Eli Smith  
Roxanne Johnson  
Steve Lindblad

# A Brief Introduction to Python Programming

---

Slides:

<https://www.github.com/elismith/farcon/>



# Eli Smith

Financial Analyst - Consultant

Microsoft Excel and Access

SQL, Javascript, Python

- Forecasting
- Performance measurement
- Reporting

Email: [esmith@browniegroup.com](mailto:esmith@browniegroup.com)

---

# What is Python?


A high-level,  
general-purpose,  
object-oriented  
programming  
language

A language  
used to teach  
computer  
science and  
programming

A thing used to build  
websites, automate  
workflows, build  
desktop applications  
and games

A way  
programmers  
share ideas  
with each other





# A way programmers share ideas with each other

Guido van Rossum  
2016 PyCon Keynote

- Benevolent Dictator for Life
- Released Python in 1989
- Maintained and extended by volunteers

Python Software Foundation

- [www.python.org](http://www.python.org)

PyMNtos - Twin Cities User Group

- [www.python.mn](http://www.python.mn)

PyCon - Portland OR, May 2017

- 2016 PyCon on Youtube -  
<https://www.youtube.com/channel/UCwTD5zJbsQGJN75MwbykYNw/videos>

Share your code

- <https://github.com/>

# What can Python do?

- Data wrangling
  - Extract data
  - Manipulate Word, Excel, CSV, JSON, HTML, PDF files
  - Correct field formatting
  - Fill in missing values
  - Workflow scripts
- Predictive modeling (scikit-learn)
- Machine learning (pandas)
- Web application framework (Django and Flask)
- Network and configuration management (Ansible)

## Python Strengths:

- Useful for rapid application development, scripting, and as glue to connect existing components
- Simple, easy-to-learn syntax emphasizes readability
- The standard library and interpreter are free
- Supports modules and packages, encouraging code reuse
- Debugging is easy: add a few print statements
- No separate compiler
- Python is biased to raise exceptions, which can be handled, vs. errors that cause crashes

## Weakness:

- Does not support true multi-core execution via multithreading

## Install Python locally or use a web-based interpreter?

### Local Installation

#### Official Release

- Learn to install and manage packages
- Learn to use the terminal (command line)
- Easy to manipulate local data files
- Official download

<https://www.python.org/downloads/>

#### Anaconda

- 100+ pretested extensions
- 720+ available extensions
- Includes R
- Includes Jupyter for Python in browser

<https://www.continuum.io/downloads>

### Web-based

- No need to install
- Skip the command line
- Upload data files
- Great for demos
- Identical setups for everyone

<https://www.pythonanywhere.com/>



Which version: Python 2.7 or the latest Python 3.5.2?

## 2.7

Older online resources (books, tutorials, examples, answers to stackexchange questions, etc.) may be written in 2.7.

Some Python libraries can only be used in Python 2.7

Existing code base may be written in 2.7

## 3.5.2

Represents the present and future of Python

Intentionally not backwards-compatible with 2.7

Useful changes and bug fixes

<https://docs.python.org/3/whatsnew/3.0.html>

Support for 2.7 will end in 2020

# Pro Tip 1

If you have one version of Python installed and need to install another version...

virtualenv may be the answer.

Source and docs:

<https://github.com/pypa/virtualenv>

Install virtualenv

```
$ [sudo] pip install virtualenv
```

Run virtualenv

```
$ virtualenv ENV
```

Install new version of Python in the directory ENV

The python in your new virtualenv is effectively isolated from the python that was used to create it

# Pro Tip 2

Command line operators...

Syntax error...

Missing parameter...

Floating point...



Google it<sup>1</sup>

Watch for:

[stackexchange.com](https://stackoverflow.com)

[python.org](https://python.org)

[blogs](#)

[wikipedia.org](https://wikipedia.org)

<sup>1</sup> 100% authentic pro tip - your results may vary

# Python 3.5.2 Documentation

[Tutorial](#)

start here

[Library Reference](#)

keep this under your pillow

[Language Reference](#)

describes syntax and language elements

[Python Setup and Usage](#)

how to use Python on different platforms

[Python HOWTOs](#)

in-depth documents on specific topics

[Installing Python Modules](#)

installing from Python Package Index & other sources

[Global Module Index](#)

quick access to all modules

[General Index](#)

all functions, classes, terms

[Glossary](#)

the most important terms explained

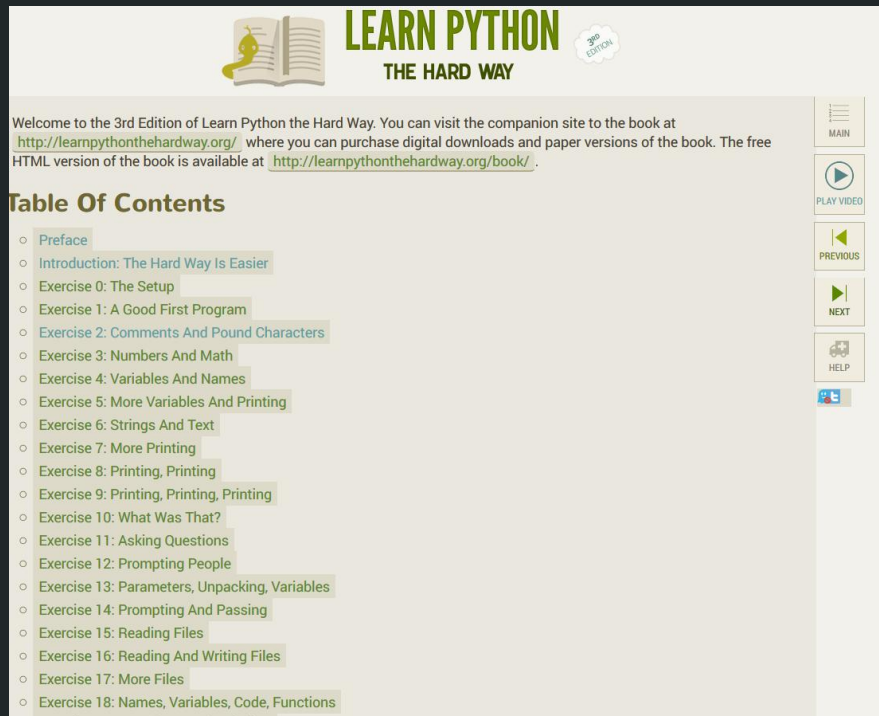
[FAQs](#)


frequently asked questions (with answers!)

Free Online Course

# Learn Python the Hard Way

[http://learnpythonthehardway.org  
book/](http://learnpythonthehardway.org/book/)



 **LEARN PYTHON**  
THE HARD WAY 3RD EDITION

Welcome to the 3rd Edition of Learn Python the Hard Way. You can visit the companion site to the book at <http://learnpythonthehardway.org/> where you can purchase digital downloads and paper versions of the book. The free HTML version of the book is available at <http://learnpythonthehardway.org/book/>.

### Table Of Contents

- Preface
- Introduction: The Hard Way Is Easier
- Exercise 0: The Setup
- Exercise 1: A Good First Program
- Exercise 2: Comments And Pound Characters
- Exercise 3: Numbers And Math
- Exercise 4: Variables And Names
- Exercise 5: More Variables And Printing
- Exercise 6: Strings And Text
- Exercise 7: More Printing
- Exercise 8: Printing, Printing
- Exercise 9: Printing, Printing, Printing
- Exercise 10: What Was That?
- Exercise 11: Asking Questions
- Exercise 12: Prompting People
- Exercise 13: Parameters, Unpacking, Variables
- Exercise 14: Prompting And Passing
- Exercise 15: Reading Files
- Exercise 16: Reading And Writing Files
- Exercise 17: More Files
- Exercise 18: Names, Variables, Code, Functions

Navigation: MAIN, PLAY VIDEO, PREVIOUS, NEXT, HELP

# Learn Python the Hard Way

<http://learnpythonthehardway.org/book/>

- PC, Mac or Linux
  - Covers basic setup:
    - Installing Python from python.org
    - Choosing a basic editor (notepad++)
    - Using the terminal
  - Requires no programming knowledge
  - Emphasis on typing, repetition
  - Examples are written in Python 2x, not 3x
-

Free Online Course:

# MIT 6.0 - Introduction to Computer Science and Programming

<http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00sc-introduction-to-computer-science-and-programming-spring-2011/unit-1/lecture-1-introduction-to-6.00/>

- [Video lectures](#)
  - [Subtitles/transcript](#)
  - [Online textbooks](#)
  - [Assignments and solutions](#)
  - [Exams and solutions](#)
  - [Recitation videos](#)
-

Free Online Course:

# Python Fundamentals

## Simeon Franklin

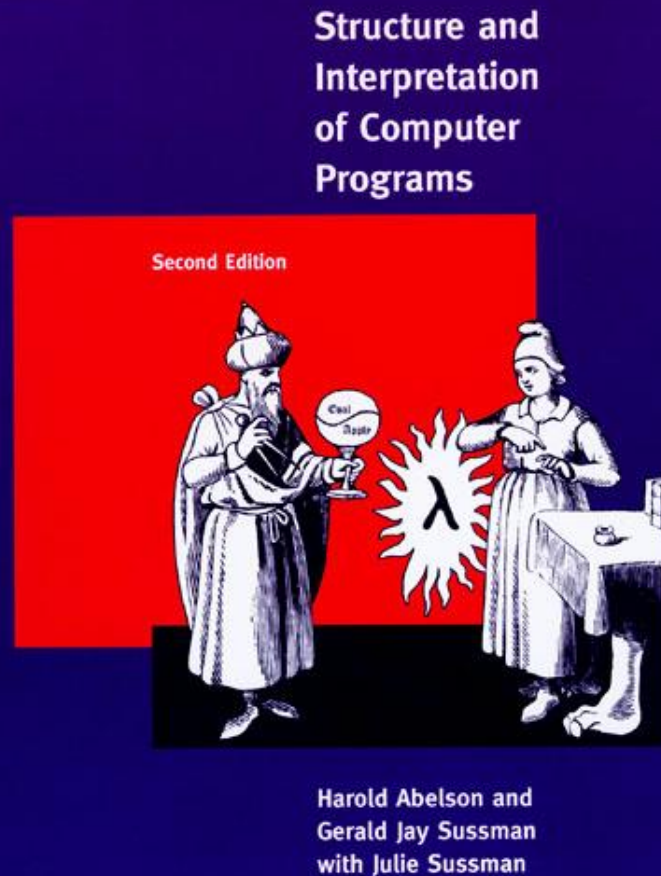
<https://www.youtube.com/watch?v=B9MyjMFokLc&list=PL26BA8B9FC33789FF>

- Recorded at a live 4-day course
  - 12 45-minute segments
  - Interactive (command-line) examples are easy to follow
  - Type along with the video, if you are fast
  - Live students are not noobs
-



# Structure and Interpretation of Computer Programs

<https://mitpress.mit.edu/sicp/full-text/book/book.html>



# Structure and Interpretation of Computer Programs

<https://mitpress.mit.edu/sicp/full-text/book/book.html>

Common elements that must appear in  
any powerful programming language

Techniques for building abstractions of  
the real world using:

- Procedures
- Data

Examples written in LISP (Scheme)

---

Level 1 complete!

Install Python (Anaconda)

<https://www.continuum.io/downloads>

Read the docs

<https://docs.python.org/3/>

Write some code

<http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00sc-introduction-to-computer-science-and-programming-spring-2011/unit-1/lecture-1-introduction-to-6.00/>

Join the community

<http://python.mn/>

# Roxanne Johnson

Research Analyst  
BlueGreen Alliance  
Foundation  
Python Beginner

---

Why might I want to learn Python? How is it relevant to me?

# How do I start?

*Finding the right learning  
resources*

# Learning Python for Data Analysis

0. Build on what you already know
1. Find content of resource by what kind of data you have
2. Find content of resource by analysis task
3. Find format of resource by project phase
4. Pull resources together to make a plan

# Data People: Learn Python!

Roxanne Johnson  
@roxanneminerals  
www.roxanneminerals.com  
john737@uwater.ca

View this document online: <http://bit.ly/1XCDQ94>

0

## Build on what you already know

### What tools do you currently use to work with data?

People who work with data have an advantage when learning programming. Proficiency or familiarity with other data tools can provide a conceptual foundation you can build on instead of starting at the very beginning.

The path laid out in orange is a sample of what my journey could have looked like.

<b>Excel/ Spreadsheets</b>	Using formulas in Excel can be similar to programming. You can even create named functions in Excel if you wanted to. Using Pivot tables and other features can help you identify analyses you may want to try in Python. You may even identify tasks that are difficult to do in Excel that may inspire you to try Python!
<b>Databases and queries (SQL)</b>	Structured Query Language (SQL) is a programming language, but it is a different programming paradigm so the style is different, which might be challenging. It may be useful to read on how the languages are structured differently.
<b>Web-based tool with a graphic interface (Tableau)</b>	Drag-and-drop menus can be a great way to identify the steps and order of an analysis. Once you can visualize what exactly you want to do with your data, you can translate that into Python to see how it could work.
<b>Math/statistical software (SAS, SPSS, STATA, MATLAB)</b>	If you use a domain-specific programming language, you may be able to take an analysis you already know (or look at someone else's) in the language you know and try using Python to see how it could work.
<b>GIS and online mapping tools (MapBox, CartoDB)</b>	Similar to the drag-and-drop menu idea, spatial analysis software allows you to visualize what you want to do with your data. Creating GIS allows you to add Python to automate tasks. Additionally, GIS software builds a database to store geospatial data.
<b>Other Programming language</b>	If you already know another programming language, this tool may not be that useful to you! You are welcome to use it anyway :)

1

## Find content of resource by the kind of data you have

### What kind of data do you have?

The kind of data, the domain you work in, and the file format you have will affect what kinds of analysis you might want or be able to do. Identifying some keywords to use in searches will help you find the right tools.

<b>Data Keywords</b> Tabular Time Series Categorical Survey Geospatial Statistics Relational Database Inventory	<b>Content Keywords</b> Financial Economic Education Manufacturing Membership Company
---	---

## Why This Tool?

I have been a research analyst for about five years, and actively trying to learn Python (and R) for data analysis for about two years. I've found it extremely challenging and I've not many others with similar experiences. Some of the challenges I have identified:

- Not knowing where to start or what the big picture looks like, no clear view of what's possible or a map of how to get there
- The vast number of resources is daunting. It is difficult to find resources that are both at my skill level and relevant to what I want to learn
- Not knowing what I don't know, lacking the awareness that I don't understand a core concept, and not being able to articulate questions or search terms

2

## Find content of resource by analysis task

### What task(s) do you want to do with data?

There are several tasks you may want to do with your data, and this may impact what kinds of tools you want to learn about using the resources you find.

#### Some Basics

- There are a few things you will have to learn to start working with data using Python:
  - Conceptual challenge: if you are used to opening a spreadsheet to view your data, accessing it with Python may seem unintuitive.
  - Downloading and installing Python. Version 2 or 3? Anaconda distribution?
  - Opening and running Python, integrated development environments (IDEs), Jupyter Notebooks
  - Navigating file systems, paths, and the command line
  - Basic syntax plus built-in data structures: lists, dictionaries, arrays
  - Reading in data from a file, writing to a file
  - How to identify, obtain, load libraries that are good for the data task you want
  - General library use: how do they work? (dot notation)

#### Gather, Obtain, Collect, Scraper it

- Learn about:
  - Real-world data in a usable format, open data
  - File formats, accessibility, machine readability
  - Identify potential issues with readability and format
  - Collecting your own data and storing it in a usable format
  - Collection methods (survey) and coding (scraping), survey design
  - Data structure
  - Scraping data from websites: Regular Expressions

#### Useful Tools

Libraries: os, and csv, pandas, NumPy, BeautifulSoup (HTML/XML), Colander, storing, using data Data Science Framework: University of Chicago Center for Data Science & Public Policy

#### Munge, Clean, Wrangle, Prepare, Format it

- Learn about:
  - Data hygiene/cleanliness. What potential problems will they be your problems? How do you deal with them?
  - Formatting/preparing data. Why would you want to? What are the benefits of one format over another? Using a tool like Tableau requires that your data be formatted a certain way. You could use Python to format it.
  - Many people use Python to group their data by employee in another language or with another tool.

#### Useful Tools

Libraries: pandas, re, csv, BeautifulSoup, Regular Expressions

#### Analyze, Explore, Look for Patterns, Play with it

- Learn about:
  - Types of analysis: Statistical, summarizing, grouping, calculating new metrics
  - Exploratory vs. explanatory analysis
  - Modeling: predictive models and machine learning

#### Useful Tools

Libraries: pandas, NumPy, SciPy, matplotlib (plotting), seaborn (work with color lines), Seaborn (statistical visualizations), pprint (pretty prints), seaborn (machine learning). If you use pivot tables in Excel, try groupby in pandas. Single comparison: choose a specific data analysis and predictive modeling in Python, R, and Excel.

#### Visualize it with Charts, Graphs, or Maps

- Learn about:
  - What types of charts are good for the data you have and story you want to tell?
  - What data structures are best for storing your data and what you want to do with it?

3

## Find format of resource by project phase

### Which phase(s) of the data analysis project are you working on?

There are many types of resources, and some are better than others for the phase you're working on in your project. Some resources are very broad, others very specific. Some help you learn big concepts and ideas, others very specific use of a tool, identifying where you are in a project can help you look for the most useful resource.

#### Idea and Exploration

Thinking up a data analysis project: what kind of question you might want to answer or explore, looking for what data is out there, thinking about what tools you have access to use in this phase you want broad resources that will introduce you to what's possible and existing.

#### Scoping and Planning

Come up with a good question or problem statement. You want to make sure it's doable. Do you have the data? Do you already know the tool fairly well? How much time do you have? In this phase you want to narrow down what you can do from what you want to do. Resources should help you set an obtainable goal. May require some exploratory analysis.

#### Implementation and Troubleshooting

Actually doing the data analysis could include exploratory and/or explanatory analysis, developing and running models. In this phase you are executing your ideas and dealing with challenges as they arise. Good resources will allow you to quickly learn how to use a tool and easily find solutions to specific technical questions and problems.

#### Presentation, Reporting, and Evaluation

Presenting your findings and results, possibly your methodology. In this phase you want resources to help with formatting, or maybe places to talk about what you did and get feedback on how to share it with others.

#### General Skill Building

You may want to explore or build skills in Python without a specific project in mind.

4

## Pull resources together to make a plan

### Planning my First Data Analysis in Python

#### What tools do I already know?

What kind of data do I have? (file format, domain, keyword)

#### What task(s) do I want to do?

What phase(s) of the analysis am I working on?

What are some terms I can look up?

What are some Python tools I might learn more about?

What types of resources might I look for?

## About this Project

### Problem

Non-programmers interested in learning programming as a tool to help them work with data need a way to identify appropriate learning resources because there's so much out there it can be daunting.

### Solution

My solution is an interactive tool like a "Choose Your Own Adventure" book that uses information on why a learner wants to learn Python for data analysis and what specific task they want to work on to recommend ways to search for resources. It will also direct the learner to some actual resources for learning Python.

## Types of Resources

### Google

Google can be really useful if you're just searching something really broad or finding something really specific.

Tip: Take some time to find the right search terms. Copying and pasting your messages into Google can help you with troubleshooting.

### One-on-One Help

There are probably others out there who can help you with your project. Asking for help is a great way to be corrected with other Python people near you!

Tip: Be clear with your ask: what do you want from this person? Write down specific questions to discuss. Remember that they may not remember what it's like to try to learn a new concept.

### Websites, Tech Blogs, and GitHub

Other people out there may have done an analysis similar to yours. They may have written up a guide on how to do it, where they struggled, and they may post their code! You may find others with a similar background and struggles that you can learn from.

Tip: Find people you like, who have the same interests, or who are doing interesting things. Or make your own!

### Documentation

The information that comes with libraries and programming books can show you how to use them and give you details about how they work.

Tip: These documents assume you have a basic knowledge of Python syntax and can be hard to read. Sometimes you are a beginner. Look for examples to see how the code needs to be written.

### Books

There are a lot of guides, how-to books, and reference books out there.

Tip: Identify what kind of book you are looking for and evaluate it on content and readability. Is what you want to learn to do it easy to find or hard? Does it walk you through examples and does it come with code you can download?

### Event (Hackathon, Conference)

Events are a great place to get ideas or work as part of a team on a project. They are also a great place to make connections with people who have similar interests.

Tip: Once you know a skill a hackathon can be a great place to try to use, however it may be difficult to start learning a skill or a new topic to do a project.

### Local Python User Group

Your local user group is a great place to meet other Python users and get feedback on a Python project you do.

Tip: Presenting a project to your local user group is a great way to get feedback on tests you used, hear about other ways to do it, and learn ways to make your code more readable, Pythonic, and efficient.

### Online Class

There are lots of different places to take free online classes. This is a great way to build domain knowledge and skills but may not be a great resource for the middle of a project you want to get done!

Tip: Look for reviews to find well-reviewed classes. There are some bad ones out there. Read the reviews to make sure of the content you want to see. Use the information about your goal for the class. Find others to join you in a cohort and meet up!

### In Person Class, Workshop

Live classes and workshops can be a great way to learn with access to an instructor who is familiar with learning barriers and can answer questions.

Tip: Try to find a class that is the right skill level for you. Look at the class materials or read the syllabus. Possible: "You want something that is challenging but also accessible. You may want to try and build some basic skills before the workshop to prepare."

## My Ongoing Journey

While the orange path is nice and neat, my actual path to learning Python looks like a game of Chutes and Ladders, inspiring me to make this tool. My hope is that it will help other non-programming data people:

- See what's possible and be inspired to try it
- Create a mental map of the landscape that is learning Python for data analysis, so that they can navigate it more easily to get from where they are now to where they want to be
- Identify places they might get stuck and help them search for the right resources to get through it

I hope to take this prototype and feedback from sharing it to inform another iteration of this tool. Stay tuned!

Roxanne Johnson, PyCon 2016





# Steve Lindblad

Retirement Actuary (Aon Hewitt)

Python enthusiast

Email: [splindblad@gmail.com](mailto:splindblad@gmail.com)



# Sample Program—Most popular baby names

- Data on the most popular baby names from the Social Security Administration
  - <https://catalog.data.gov/dataset/baby-names-from-social-security-card-applications-national-level-data>
    - ZIP file containing one text data file for each year of birth 1880–2015
- Each file is can be viewed individually, but what if we want to create a list of all of the most popular baby names by year?
- Quick Python script to automate this: topnames.py
  - <https://github.com/splindblad/topnames>
- Let's walk through the key features to get a flavor for the power and syntax of Python



Thank You!