

# *Learning Python*



## **Student Workbook**

October, 2015

HTML Workbook Version 2.7-3.5

Copyright © Mark Lutz, 1997—2015

# Introduction

This is the root page of the class workbook. The workbook contains all the material presented during the class, source code for examples and lab exercises, and links to related information on the web. Usage tips:

## Navigation

- Always **start here**, and click on the titles below to go to lecture unit pages.
- To go to **lab exercises**, click either the links at the end of each lecture unit page, or the exercises link near the end of this page.
- To **return here**, use your browser's "back" button, or create a shortcut to this file on your desktop.

## General

- As of October 2015, this workbook should render well in all browsers (*Internet Explorer* is no longer preferred).
- Copy the Workbook folder to a **hard drive** or USB stick if pages open too slowly from a CD or server copy.

## Other tips

- For **reference material**, see Python's manuals, or the ebook copy of *Python Pocket Reference* in Extras.
- See also the distribution package's top-level "**README.txt**" file for more usage notes.
- This workbook is mostly a **conversation starter**—the class goes off-page often, and is driven by your input.

*The usual first question:* for pointers on which version of Python to install and use for the class (2.X or 3.X), see the Preface below, or wait for the first lab session.

# Contents

## Preface

PYTHON 2.X OR 3.X?  
ABOUT THIS CLASS

COURSE TOPICS  
DAILY SCHEDULE

## Part I: Python Language

---

### 1: General Python Introduction

SO WHAT'S PYTHON?  
WHY DO PEOPLE USE PYTHON?  
SOME QUOTABLE QUOTES  
A PYTHON HISTORY LESSON  
ADVOCACY NEWS  
WHAT'S PYTHON GOOD FOR?  
WHAT'S PYTHON NOT GOOD FOR?  
THE COMPULSORY FEATURES LIST  
PYTHON PORTABILITY  
ON APPLES AND ORANGES  
SUMMARY: WHY PYTHON?

### 2. Using the Interpreter

HOW PYTHON RUNS PROGRAMS  
HOW YOU RUN PROGRAMS  
CONFIGURATION DETAILS  
MODULE FILES: A FIRST LOOK  
THE IDLE INTERFACE  
OTHER PYTHON IDES  
TIME TO START CODING  
*LAB SESSION 1*

### 3. Types and Operators

A FIRST PASS  
THE 'BIG PICTURE'  
NUMBERS  
DYNAMIC TYPING INTERLUDE  
STRINGS  
LISTS  
DICTIONARIES  
TUPLES  
FILES  
GENERAL OBJECT PROPERTIES  
SUMMARY: PYTHON'S TYPE HIERARCHIES  
BUILT-IN TYPE GOTCHAS  
*LAB SESSION 2*

### 4. Basic Statements

GENERAL SYNTAX CONCEPTS

ASSIGNMENT  
EXPRESSIONS  
PRINT  
IF SELECTIONS  
PYTHON SYNTAX RULES  
DOCUMENTATION SOURCES INTERLUDE  
TRUTH TESTS  
WHILE LOOPS  
BREAK, CONTINUE, PASS, AND THE LOOP ELSE  
FOR LOOPS  
COMPREHENSIONS AND ITERATIONS  
LOOP CODING TECHNIQUES  
COMPREHENSIVE LOOP EXAMPLES  
BASIC CODING GOTCHAS  
PREVIEW: PROGRAM UNIT STATEMENTS  
*LAB SESSION 3*

## 5. Functions

FUNCTION BASICS  
SCOPE RULES IN FUNCTIONS  
MORE ON “GLOBAL” (AND “NONLOCAL”)  
MORE ON “RETURN”  
MORE ON ARGUMENT PASSING  
SPECIAL ARGUMENT MATCHING MODES  
ODDS AND ENDS  
GENERATOR EXPRESSIONS AND FUNCTIONS  
FUNCTION DESIGN CONCEPTS  
FUNCTIONS ARE OBJECTS: INDIRECT CALLS  
FUNCTION GOTCHAS  
OPTIONAL CASE STUDY: SET FUNCTIONS  
*LAB SESSION 4*

## 6. Modules

MODULE BASICS  
MODULE FILES ARE A NAMESPACE  
NAME QUALIFICATION  
IMPORT VARIANTS  
RELOADING MODULES  
PACKAGE IMPORTS  
ODDS AND ENDS  
MODULE DESIGN CONCEPTS  
MODULES ARE OBJECTS: METAPROGRAMS  
MODULE GOTCHAS  
OPTIONAL CASE STUDY: A SHARED STACK MODULE  
*LAB SESSION 5*

## 7. Classes

OOP: THE BIG PICTURE  
CLASS BASICS  
A MORE REALISTIC EXAMPLE  
USING THE CLASS STATEMENT

USING CLASS METHODS  
CUSTOMIZATION VIA INHERITANCE  
SPECIALIZING INHERITED METHODS  
OPERATOR OVERLOADING IN CLASSES  
NAMESPACE RULES: THE WHOLE STORY  
OOP EXAMPLES: INHERITANCE AND COMPOSITION  
CLASSES AND METHODS ARE OBJECTS  
ODDS AND ENDS  
NEW STYLE CLASSES  
CLASS GOTCHAS  
OPTIONAL CASE STUDY: A SET CLASS  
SUMMARY: OOP IN PYTHON  
*LAB SESSION 6*

## 8. Exceptions

EXCEPTION BASICS  
FIRST EXAMPLES  
EXCEPTION IDIOMS  
EXCEPTION CATCHING MODES  
CLASS EXCEPTIONS  
EXCEPTION GOTCHAS  
*LAB SESSION 7*

## 9. Built-in Tools Overview

THE SECRET HANDSHAKE  
DEBUGGING OPTIONS  
INSPECTING NAME-SPACES  
DYNAMIC CODING TOOLS  
TIMING AND PROFILING PYTHON PROGRAMS  
FILE TYPES AND PACKAGING OPTIONS  
DEVELOPMENT TOOLS FOR LARGER PROJECTS  
SUMMARY: PYTHON TOOL-SET LAYERS  
*LAB SESSION 7*

## Part II: Python Applications

---

## 10. System Interfaces

SYSTEM MODULES OVERVIEW  
RUNNING SHELL COMMANDS  
ARGUMENTS, STREAMS, SHELL VARIABLES  
FILE TOOLS  
DIRECTORY TOOLS  
FORKING PROCESSES  
THREAD MODULES AND QUEUES  
THE SUBPROCESS AND MULTIPROCESSING MODULES  
IPC TOOLS: PIPES, SOCKETS, SIGNALS  
FORK VERSIS SPAWNV

LARGER EXAMPLES  
*LAB SESSION 8*

## 11. GUI Programming

PYTHON GUI OPTIONS  
THE TKINTER 'HELLO WORLD' PROGRAM  
ADDING BUTTONS, FRAMES, AND CALLBACKS  
GETTING INPUT FROM A USER  
ASSORTED TKINTER DETAILS  
BUILDING GUIS BY SUBCLASSING FRAMES  
REUSING GUIS BY SUBCLASSING AND ATTACHING  
ADVANCED WIDGETS: IMAGES, GRIDS, AND MORE  
LARGER EXAMPLES  
TKINTER ODDS AND ENDS  
*LAB SESSION 8*

## 12. Databases and Persistence

OBJECT PERSISTENCE: SHELVES  
STORING CLASS INSTANCES  
PICKLING OBJECTS WITHOUT SHELVES  
USING SIMPLE DBM FILES  
SHELVE GOTCHAS  
ZODB OBJECT-ORIENTED DATABASE  
PYTHON SQL DATABASE API  
PERSISTENCE ODDS AND ENDS  
*LAB SESSION 9*

## 13. Text Processing

STRING OBJECTS: REVIEW  
SPLITTING AND JOINING STRINGS  
REGULAR EXPRESSIONS  
PARSING LANGUAGES  
XML PARSING: REGEX, SAX, DOM, AND ETREE  
*LAB SESSION 10*

## 14. Internet Scripting

USING SOCKETS IN PYTHON  
THE FTP MODULE  
EMAIL PROCESSING  
OTHER CLIENT-SIDE TOOLS  
BUILDING WEB SITES WITH PYTHON  
WRITING SERVER-SIDE CGI SCRIPTS  
PYTHON: PYTHON FOR JAVA SYSTEMS  
ACTIVE SCRIPTING AND COM  
OTHER INTERNET-RELATED TOOLS  
*LAB SESSION 10*

## 15. Extending Python in C/C++

- PYTHON INTEGRATION MODEL
- REVIEW: PYTHON TOOL-SET LAYERS
- WHY INTEGRATION?
- INTEGRATION MODES
- A SIMPLE C EXTENSION MODULE
- C MODULE STRUCTURE
- BINDING C EXTENSIONS TO PYTHON
- DATA CONVERSIONS: PYTHON ↔ C
- C EXTENSION TYPES
- USING C EXTENSION TYPES IN PYTHON
- WRAPPING C EXTENSIONS IN PYTHON
- WRITING EXTENSIONS IN C++
- SWIG EXAMPLE (PP)
- PYTHON AND RAPID DEVELOPMENT
- LAB SESSION 11*

## 16. Embedding Python in C/C++

- GENERAL EMBEDDING CONCEPTS
- RUNNING SIMPLE CODE STRINGS
- CALLING OBJECTS AND METHODS
- RUNNING STRINGS: RESULTS & NAME-SPACES
- OTHER CODE STRING POSSIBILITIES
- REGISTERING PYTHON OBJECTS AND STRINGS
- ACCESSING C VARIABLES IN PYTHON
- C API EQUIVALENTS IN PYTHON
- RUNNING CODE FILES FROM C
- PRECOMPILING STRINGS INTO BYTE-CODE
- EMBEDDING UNDER C++
- MORE ON OBJECT REFERENCE COUNTS
- INTEGRATION ERROR HANDLING
- AUTOMATED INTEGRATION TOOLS
- LAB SESSION 12*

## 17. Advanced Topics

- UNICODE TEXT AND BINARY DATA
- MANAGED ATTRIBUTES
- DECORATORS
- METACLASSES
- CONTEXT MANAGERS
- PYTHON 3.X CHANGES
- LAB SESSION 13*

## 18. Resources

- INTERNET RESOURCES
- PYTHON BOOKS
- PYTHON CONFERENCES AND SERVICES
- AND FINALLY

## Labwork and Examples

---

### Laboratory Exercises

- LAB 1: USING THE INTERPRETER
- LAB 2: TYPES AND OPERATORS
- LAB 3: BASIC STATEMENTS
- LAB 4: FUNCTIONS
- LAB 5: MODULES
- LAB 6: CLASSES
- LAB 7: EXCEPTIONS AND BUILT-IN TOOLS
- LAB 8: SYSTEM INTERFACES AND GUI
- LAB 9: PERSISTENCE
- LAB 10: TEXT PROCESSING AND THE INTERNET
- LAB 11: EXTENDING PYTHON IN C/C++
- LAB 12: EMBEDDING PYTHON IN C/C++
- LAB 13: DECORATORS AND METACLASSES

### Selected Exercise Solutions

- LAB 1: USING THE INTERPRETER
- LAB 2: TYPES AND OPERATORS
- LAB 3: BASIC STATEMENTS
- LAB 4: FUNCTIONS
- LAB 5: MODULES
- LAB 6: CLASSES
- LAB 7: EXCEPTIONS AND BUILT-IN TOOLS

[Exercise solution files directory](#)

[Lecture example files directory](#)

[Final exam \(optional\)](#)