

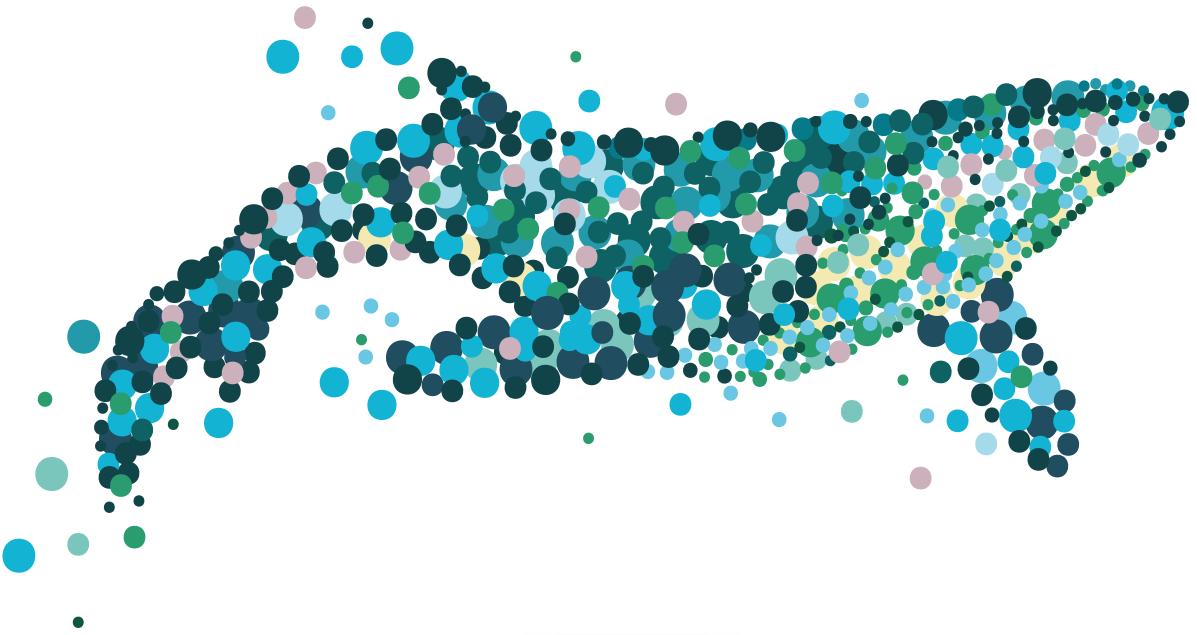
GLOBAL
EDITION



Intro to Python®

for Computer Science and Data Science

Paul Deitel • Harvey Deitel



Learning to Program with AI, Big Data and the Cloud



Digital Resources for Students

Your eBook provides 12-month access to digital resources that may include VideoNotes (step-by-step video tutorials on programming concepts), source code, and more. Refer to the preface in the textbook for a detailed list of resources.

Follow the instructions below to register for the Companion Website for Paul and Harvey Deitel's *Intro to Python®: for Computer Science and Data Science*, Global Edition.

1. Go to www.pearsonglobaleditions.com.
2. Enter the title of your textbook or browse by author name.
3. Click Companion Website.
4. Click Register and follow the on-screen instructions to create a login name and password.

ISSPCD-WAHOO-DIARY-KALPA-FRACK-OOSSE

Use the login name and password you created during registration to start using the online resources that accompany your textbook.

IMPORTANT:

This access code can only be used once. This subscription is valid for 12 months upon activation and is not transferable.

For technical support, go to <https://support.pearson.com/getsupport>.

Intro to Python for Computer Science and Data Science: Learning to Program with AI, Big Data and The Cloud, Global Edition

Table of Contents

Cover

Digital Resources for Students

Intro to Python® for Computer Science and Data Science

Half Title

Copyright

Title Page

Dedication

Contents

Preface

Before You Begin

1. Introduction to Computers and Python

1.1 Introduction

1.2 Hardware and Software

1.2.1 Moores Law

1.2.2 Computer Organization

1.3 Data Hierarchy

1.4 Machine Languages, Assembly Languages and High-Level
Languages

1.5 Introduction to Object Technology

Table of Contents

1.6 Operating Systems

1.7 Python

1.8 Its the Libraries!

1.8.1 Python Standard Library

1.8.2 Data-Science Libraries

1.9 Other Popular Programming Languages

1.10 Test-Drives: Using IPython and Jupyter Notebooks

1.10.1 Using IPython Interactive Mode as a Calculator

1.10.2 Executing a Python Program Using the IPython Interpreter

1.10.3 Writing and Executing Code in a Jupyter Notebook

1.11 Internet and World Wide Web

1.11.1 Internet: A Network of Networks

1.11.2 World Wide Web: Making the Internet User-Friendly

1.11.3 The Cloud

1.11.4 Internet of Things

1.12 Software Technologies

1.13 How Big Is Big Data?

1.13.1 Big Data Analytics

1.13.2 Data Science and Big Data Are Making a Difference: Use Cases

1.14 Case StudyA Big-Data Mobile Application

1.15 Intro to Data Science: Artificial Intelligenceat the Intersection of CS and Data Science

2. Introduction to Python Programming

2.1 Introduction

2.2 Variables and Assignment Statements

2.3 Arithmetic

2.4 Function print and an Intro to Single- and Double-Quoted Strings

2.5 Triple-Quoted Strings

Table of Contents

- 2.6 Getting Input from the User
- 2.7 Decision Making: The if Statement and Comparison Operators
- 2.8 Objects and Dynamic Typing
- 2.9 Intro to Data Science: Basic Descriptive Statistics
- 2.10 Wrap-Up
- 3. Control Statements and Program Development
 - 3.1 Introduction
 - 3.2 Algorithms
 - 3.3 Pseudocode
 - 3.4 Control Statements
 - 3.5 if Statement
 - 3.6 ifelse and ifelifelse Statements
 - 3.7 while Statement
 - 3.8 for Statement
 - 3.8.1 Iterables, Lists and Iterators
 - 3.8.2 Built-In range Function
 - 3.9 Augmented Assignments
 - 3.10 Program Development: Sequence-Controlled Repetition
 - 3.10.1 Requirements Statement
 - 3.10.2 Pseudocode for the Algorithm
 - 3.10.3 Coding the Algorithm in Python
 - 3.10.4 Introduction to Formatted Strings
 - 3.11 Program Development: Sentinel-Controlled Repetition
 - 3.12 Program Development: Nested Control Statements
 - 3.13 Built-In Function range: A Deeper Look
 - 3.14 Using Type Decimal for Monetary Amounts
 - 3.15 break and continue Statements

Table of Contents

3.16 Boolean Operators and, or and not

3.17 Intro to Data Science: Measures of Central Tendency Mean,
Median and Mode

3.18 Wrap-Up

4. Functions

4.1 Introduction

4.2 Defining Functions

4.3 Functions with Multiple Parameters

4.4 Random-Number Generation

4.5 Case Study: A Game of Chance

4.6 Python Standard Library

4.7 math Module Functions

4.8 Using IPython Tab Completion for Discovery

4.9 Default Parameter Values

4.10 Keyword Arguments

4.11 Arbitrary Argument Lists

4.12 Methods: Functions That Belong to Objects

4.13 Scope Rules

4.14 import: A Deeper Look

4.15 Passing Arguments to Functions: A Deeper Look

4.16 Function-Call Stack

4.17 Functional-Style Programming

4.18 Intro to Data Science: Measures of Dispersion

4.19 Wrap-Up

5. Sequences: Lists and Tuples

5.1 Introduction

Table of Contents

5.2 Lists

5.3 Tuples

5.4 Unpacking Sequences

5.5 Sequence Slicing

5.6 del Statement

5.7 Passing Lists to Functions

5.8 Sorting Lists

5.9 Searching Sequences

5.10 Other List Methods

5.11 Simulating Stacks with Lists

5.12 List Comprehensions

5.13 Generator Expressions

5.14 Filter, Map and Reduce

5.15 Other Sequence Processing Functions

5.16 Two-Dimensional Lists

5.17 Intro to Data Science: Simulation and Static Visualizations

5.17.1 Sample Graphs for 600, 60,000 and 6,000,000 Die Rolls

5.17.2 Visualizing Die-Roll Frequencies and Percentages

5.18 Wrap-Up

6. Dictionaries and Sets

6.1 Introduction

6.2 Dictionaries

6.2.1 Creating a Dictionary

6.2.2 Iterating through a Dictionary

6.2.3 Basic Dictionary Operations

6.2.4 Dictionary Methods keys and values

6.2.5 Dictionary Comparisons

Table of Contents

6.2.6 Example: Dictionary of Student Grades

6.2.7 Example: Word Counts

6.2.8 Dictionary Method update

6.2.9 Dictionary Comprehensions

6.3 Sets

6.3.1 Comparing Sets

6.3.2 Mathematical Set Operations

6.3.3 Mutable Set Operators and Methods

6.3.4 Set Comprehensions

6.4 Intro to Data Science: Dynamic Visualizations

6.4.1 How Dynamic Visualization Works

6.4.2 Implementing a Dynamic Visualization

6.5 Wrap-Up

7. Array-Oriented Programming with NumPy

7.1 Introduction

7.2 Creating arrays from Existing Data

7.3 array Attributes

7.4 Filling arrays with Specific Values

7.5 Creating arrays from Ranges

7.6 List vs. array Performance: Introducing %timeit

7.7 array Operators

7.8 NumPy Calculation Methods

7.9 Universal Functions

7.10 Indexing and Slicing

7.11 Views: Shallow Copies

7.12 Deep Copies

7.13 Reshaping and Transposing

Table of Contents

7.14 Intro to Data Science: pandas Series and DataFrames

7.14.1 pandas Series

7.14.2 DataFrames

7.15 Wrap-Up

8. Strings: A Deeper Look

8.1 Introduction

8.2 Formatting Strings

8.2.1 Presentation Types

8.2.2 Field Widths and Alignment

8.2.3 Numeric Formatting

8.2.4 Strings format Method

8.3 Concatenating and Repeating Strings

8.4 Stripping Whitespace from Strings

8.5 Changing Character Case

8.6 Comparison Operators for Strings

8.7 Searching for Substrings

8.8 Replacing Substrings

8.9 Splitting and Joining Strings

8.10 Characters and Character-Testing Methods

8.11 Raw Strings

8.12 Introduction to Regular Expressions

8.12.1 re Module and Function fullmatch

8.12.2 Replacing Substrings and Splitting Strings

8.12.3 Other Search Functions; Accessing Matches

8.13 Intro to Data Science: Pandas, Regular Expressions and Data Munging

8.14 Wrap-Up

Table of Contents

9. Files and Exceptions

9.1 Introduction

9.2 Files

9.3 Text-File Processing

9.3.1 Writing to a Text File: Introducing the with Statement

9.3.2 Reading Data from a Text File

9.4 Updating Text Files

9.5 Serialization with JSON

9.6 Focus on Security: pickle Serialization and Deserialization

9.7 Additional Notes Regarding Files

9.8 Handling Exceptions

9.8.1 Division by Zero and Invalid Input

9.8.2 try Statements

9.8.3 Catching Multiple Exceptions in One except Clause

9.8.4 What Exceptions Does a Function or Method Raise?

9.8.5 What Code Should Be Placed in a try Suite?

9.9 finally Clause

9.10 Explicitly Raising an Exception

9.11 (Optional) Stack Unwinding and Tracebacks

9.12 Intro to Data Science: Working with CSV Files

9.12.1 Python Standard Library Module csv

9.12.2 Reading CSV Files into Pandas DataFrames

9.12.3 Reading the Titanic Disaster Dataset

9.12.4 Simple Data Analysis with the Titanic Disaster Dataset

9.12.5 Passenger Age Histogram

9.13 Wrap-Up

10. Object-Oriented Programming

Table of Contents

10.1 Introduction

10.2 Custom Class Account

10.2.1 Test-Driving Class Account

10.2.2 Account Class Definition

10.2.3 Composition: Object References as Members of Classes

10.3 Controlling Access to Attributes

10.4 Properties for Data Access

10.4.1 Test-Driving Class Time

10.4.2 Class Time Definition

10.4.3 Class Time Definition Design Notes

10.5 Simulating Private Attributes

10.6 Case Study: Card Shuffling and Dealing Simulation

10.6.1 Test-Driving Classes Card and DeckOfCards

10.6.2 Class CardIntroducing Class Attributes

10.6.3 Class DeckOfCards

10.6.4 Displaying Card Images with Matplotlib

10.7 Inheritance: Base Classes and Subclasses

10.8 Building an Inheritance Hierarchy; Introducing Polymorphism

10.8.1 Base Class CommissionEmployee

10.8.2 Subclass SalariedCommissionEmployee

10.8.3 Processing CommissionEmployees and SalariedCommissionEmployees Polymorphically

10.8.4 A Note About Object-Based and Object-Oriented Programming

10.9 Duck Typing and Polymorphism

10.10 Operator Overloading

10.10.1 Test-Driving Class Complex

10.10.2 Class Complex Definition

10.11 Exception Class Hierarchy and Custom Exceptions

10.12 Named Tuples

Table of Contents

10.13 A Brief Intro to Python 3.7s New Data Classes

10.13.1 Creating a Card Data Class

10.13.2 Using the Card Data Class

10.13.3 Data Class Advantages over Named Tuples

10.13.4 Data Class Advantages over Traditional Classes

10.14 Unit Testing with Docstrings and doctest

10.15 Namespaces and Scopes

10.16 Intro to Data Science: Time Series and Simple Linear Regression

10.17 Wrap-Up

11. Computer Science Thinking: Recursion, Searching, Sorting and Big O

11.1 Introduction

11.2 Factorials

11.3 Recursive Factorial Example

11.4 Recursive Fibonacci Series Example

11.5 Recursion vs. Iteration

11.6 Searching and Sorting

11.7 Linear Search

11.8 Efficiency of Algorithms: Big O

11.9 Binary Search

11.9.1 Binary Search Implementation

11.9.2 Big O of the Binary Search

11.10 Sorting Algorithms

11.11 Selection Sort

11.11.1 Selection Sort Implementation

11.11.2 Utility Function print_pass

Table of Contents

11.11.3 Big O of the Selection Sort

11.12 Insertion Sort

11.12.1 Insertion Sort Implementation

11.12.2 Big O of the Insertion Sort

11.13 Merge Sort

11.13.1 Merge Sort Implementation

11.13.2 Big O of the Merge Sort

11.14 Big O Summary for This Chapters Searching and Sorting Algorithms

11.15 Visualizing Algorithms

11.15.1 Generator Functions

11.15.2 Implementing the Selection Sort Animation

11.16 Wrap-Up

12. Natural Language Processing (NLP)

12.1 Introduction

12.2 TextBlob

12.2.1 Create a TextBlob

12.2.2 Tokenizing Text into Sentences and Words

12.2.3 Parts-of-Speech Tagging

12.2.4 Extracting Noun Phrases

12.2.5 Sentiment Analysis with TextBlobs Default Sentiment Analyzer

12.2.6 Sentiment Analysis with the NaiveBayesAnalyzer

12.2.7 Language Detection and Translation

12.2.8 Inflection: Pluralization and Singularization

12.2.9 Spell Checking and Correction

12.2.10 Normalization: Stemming and Lemmatization

12.2.11 Word Frequencies

12.2.12 Getting Definitions, Synonyms and Antonyms from WordNet

12.2.13 Deleting Stop Words

Table of Contents

12.2.14 n-grams

12.3 Visualizing Word Frequencies with Bar Charts and Word Clouds

12.3.1 Visualizing Word Frequencies with Pandas

12.3.2 Visualizing Word Frequencies with Word Clouds

12.4 Readability Assessment with Textatistic

12.5 Named Entity Recognition with spaCy

12.6 Similarity Detection with spaCy

12.7 Other NLP Libraries and Tools

12.8 Machine Learning and Deep Learning Natural Language Applications

12.9 Natural Language Datasets

12.10 Wrap-Up

13. Data Mining Twitter

13.1 Introduction

13.2 Overview of the Twitter APIs

13.3 Creating a Twitter Account

13.4 Getting Twitter CredentialsCreating an App

13.5 Whats in a Tweet?

13.6 Tweepy

13.7 Authenticating with Twitter Via Tweepy

13.8 Getting Information About a Twitter Account

13.9 Introduction to Tweepy Cursors: Getting an Accounts Followers and Friends

13.9.1 Determining an Accounts Followers

13.9.2 Determining Whom an Account Follows

13.9.3 Getting a Users Recent Tweets

Table of Contents

13.10 Searching Recent Tweets

13.11 Spotting Trends: Twitter Trends API

13.11.1 Places with Trending Topics

13.11.2 Getting a List of Trending Topics

13.11.3 Create a Word Cloud from Trending Topics

13.12 Cleaning/Preprocessing Tweets for Analysis

13.13 Twitter Streaming API

13.13.1 Creating a Subclass of StreamListener

13.13.2 Initiating Stream Processing

13.14 Tweet Sentiment Analysis

13.15 Geocoding and Mapping

13.15.1 Getting and Mapping the Tweets

13.15.2 Utility Functions in tweetutilities.py

13.15.3 Class LocationListener

13.16 Ways to Store Tweets

13.17 Twitter and Time Series

13.18 Wrap-Up

14. IBM Watson and Cognitive Computing

14.1 Introduction: IBM Watson and Cognitive Computing

14.2 IBM Cloud Account and Cloud Console

14.3 Watson Services

14.4 Additional Services and Tools

14.5 Watson Developer Cloud Python SDK

14.6 Case Study: Travelers Companion Translation App

14.6.1 Before You Run the App

14.6.2 Test-Driving the App

14.6.3 SimpleLanguageTranslator.py Script Walkthrough

14.7 Watson Resources

Table of Contents

14.8 Wrap-Up

15. Machine Learning: Classification, Regression and Clustering

15.1 Introduction to Machine Learning

15.1.1 Scikit-Learn

15.1.2 Types of Machine Learning

15.1.3 Datasets Bundled with Scikit-Learn

15.1.4 Steps in a Typical Data Science Study

15.2 Case Study: Classification with k-Nearest Neighbors and the Digits Dataset, Part 1

15.2.1 k-Nearest Neighbors Algorithm

15.2.2 Loading the Dataset

15.2.3 Visualizing the Data

15.2.4 Splitting the Data for Training and Testing

15.2.5 Creating the Model

15.2.6 Training the Model

15.2.7 Predicting Digit Classes

15.3 Case Study: Classification with k-Nearest Neighbors and the Digits Dataset, Part 2

15.3.1 Metrics for Model Accuracy

15.3.2 K-Fold Cross-Validation

15.3.3 Running Multiple Models to Find the Best One

15.3.4 Hyperparameter Tuning

15.4 Case Study: Time Series and Simple Linear Regression

15.5 Case Study: Multiple Linear Regression with the California Housing Dataset

15.5.1 Loading the Dataset

15.5.2 Exploring the Data with Pandas

15.5.3 Visualizing the Features

15.5.4 Splitting the Data for Training and Testing

Table of Contents

15.5.5 Training the Model

15.5.6 Testing the Model

15.5.7 Visualizing the Expected vs. Predicted Prices

15.5.8 Regression Model Metrics

15.5.9 Choosing the Best Model

15.6 Case Study: Unsupervised Machine Learning, Part
1 Dimensionality Reduction

15.7 Case Study: Unsupervised Machine Learning, Part 2 k-Means
Clustering

15.7.1 Loading the Iris Dataset

15.7.2 Exploring the Iris Dataset: Descriptive Statistics with Pandas

15.7.3 Visualizing the Dataset with a Seaborn pairplot

15.7.4 Using a KMeans Estimator

15.7.5 Dimensionality Reduction with Principal Component Analysis

15.7.6 Choosing the Best Clustering Estimator

15.8 Wrap-Up

16. Deep Learning

16.1 Introduction

16.1.1 Deep Learning Applications

16.1.2 Deep Learning Demos

16.1.3 Keras Resources

16.2 Keras Built-In Datasets

16.3 Custom Anaconda Environments

16.4 Neural Networks

16.5 Tensors

16.6 Convolutional Neural Networks for Vision; Multi-Classification with
the MNIST Dataset

16.6.1 Loading the MNIST Dataset

Table of Contents

- 16.6.2 Data Exploration
- 16.6.3 Data Preparation
- 16.6.4 Creating the Neural Network
- 16.6.5 Training and Evaluating the Model
- 16.6.6 Saving and Loading a Model
- 16.7 Visualizing Neural Network Training with TensorBoard
- 16.8 ConvnetJS: Browser-Based Deep-Learning Training and Visualization
- 16.9 Recurrent Neural Networks for Sequences; Sentiment Analysis with the IMDb Dataset
 - 16.9.1 Loading the IMDb Movie Reviews Dataset
 - 16.9.2 Data Exploration
 - 16.9.3 Data Preparation
 - 16.9.4 Creating the Neural Network
 - 16.9.5 Training and Evaluating the Model
- 16.10 Tuning Deep Learning Models
- 16.11 Convnet Models Pretrained on ImageNet
- 16.12 Reinforcement Learning
 - 16.12.1 Deep Q-Learning
 - 16.12.2 OpenAI Gym
- 16.13 Wrap-Up
- 17. Big Data: Hadoop, Spark, NoSQL and IoT
 - 17.1 Introduction
 - 17.2 Relational Databases and Structured Query Language (SQL)
 - 17.2.1 A books Database
 - 17.2.2 Select Queries
 - 17.2.3 Where Clause
 - 17.2.4 Order by Clause

Table of Contents

- 17.2.5 Merging Data from Multiple Tables: Inner Join
- 17.2.6 Insert into Statement
- 17.2.7 Update Statement
- 17.2.8 Delete from Statement
- 17.3 NoSQL and NewSQL Big-Data Databases: A Brief Tour
 - 17.3.1 NoSQL Key/Value Databases
 - 17.3.2 NoSQL Document Databases
 - 17.3.3 NoSQL Columnar Databases
 - 17.3.4 NoSQL Graph Databases
 - 17.3.5 NewSQL Databases
- 17.4 Case Study: A MongoDB JSON Document Database
 - 17.4.1 Creating the MongoDB Atlas Cluster
 - 17.4.2 Streaming Tweets into MongoDB
- 17.5 Hadoop
 - 17.5.1 Hadoop Overview
 - 17.5.2 Summarizing Word Lengths in Romeo and Juliet via MapReduce
 - 17.5.3 Creating an Apache Hadoop Cluster in Microsoft Azure HDInsight
 - 17.5.4 Hadoop Streaming
 - 17.5.5 Implementing the Mapper
 - 17.5.6 Implementing the Reducer
 - 17.5.7 Preparing to Run the MapReduce Example
 - 17.5.8 Running the MapReduce Job
- 17.6 Spark
 - 17.6.1 Spark Overview
 - 17.6.2 Docker and the Jupyter Docker Stacks
 - 17.6.3 Word Count with Spark
 - 17.6.4 Spark Word Count on Microsoft Azure
- 17.7 Spark Streaming: Counting Twitter Hashtags Using the pyspark-notebook Docker Stack

Table of Contents

17.7.1 Streaming Tweets to a Socket

17.7.2 Summarizing Tweet Hashtags; Introducing Spark SQL

17.8 Internet of Things and Dashboards

17.8.1 Publish and Subscribe

17.8.2 Visualizing a PubNub Sample Live Stream with a Freeboard
Dashboard

17.8.3 Simulating an Internet-Connected Thermostat in Python

17.8.4 Creating the Dashboard with Freeboard.io

17.8.5 Creating a Python PubNub Subscriber

17.9 Wrap-Up

Index

Reviewer Comments