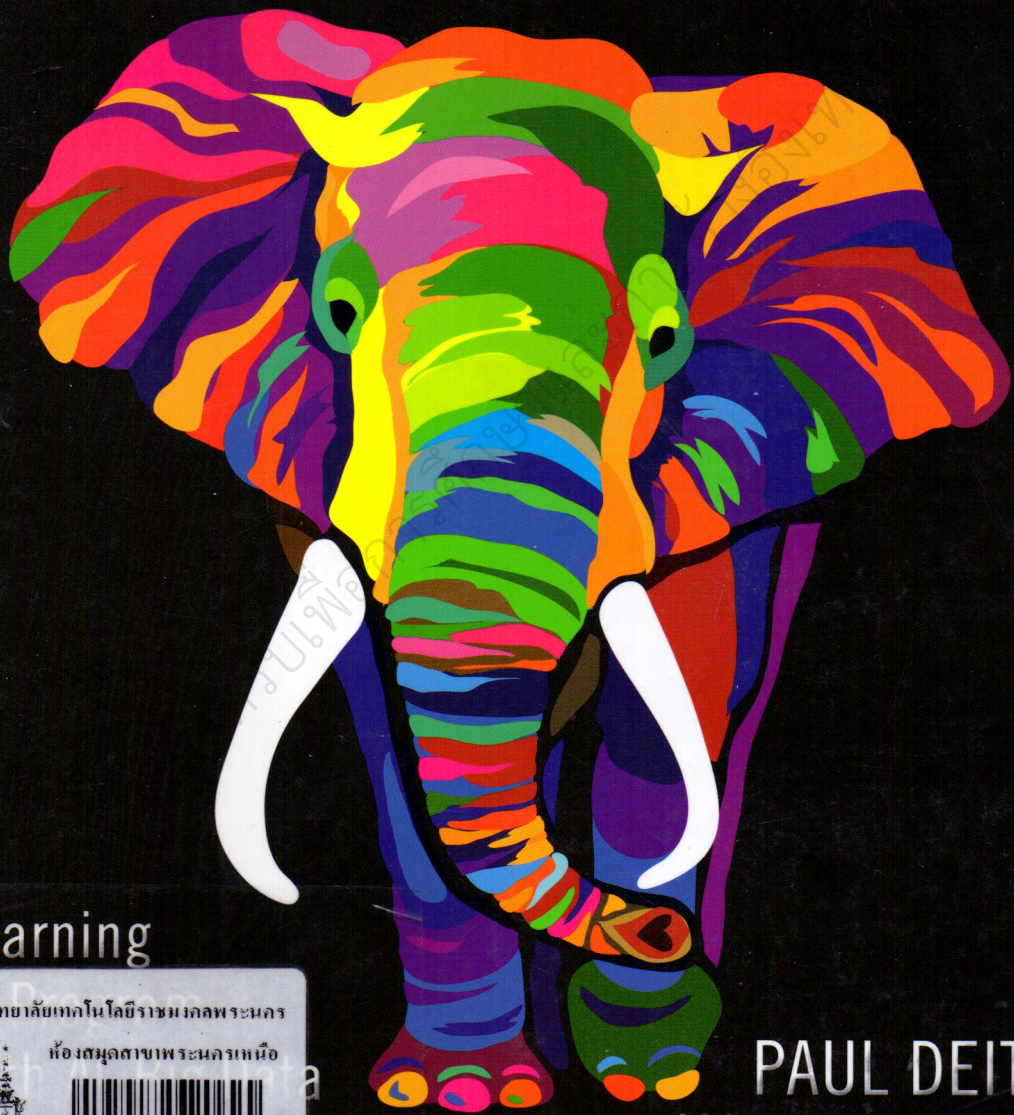


# Intro to Python<sup>®</sup>

for Computer Science and Data Science



Learning

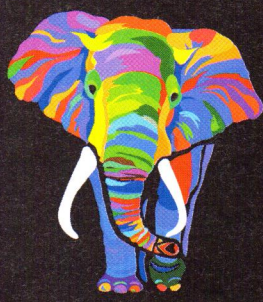
มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

ห้องสมุดสถาบันพระนครเหนือ



502006132

PAUL DEITEL  
HARVEY DEITEL



# Contents

**Preface** **xix**

**Before You Begin** **xliv**

**1 Introduction to Computers and Python** **I**

1.1	Introduction	2
1.2	Hardware and Software	3
1.2.1	Moore's Law	4
1.2.2	Computer Organization	4
1.3	Data Hierarchy	6
1.4	Machine Languages, Assembly Languages and High-Level Languages	9
1.5	Introduction to Object Technology	10
1.6	Operating Systems	13
1.7	Python	16
1.8	It's the Libraries!	18
1.8.1	Python Standard Library	18
1.8.2	Data-Science Libraries	18
1.9	Other Popular Programming Languages	20
1.10	Test-Drive: Using IPython and Jupyter Notebooks	21
1.10.1	Using IPython Interactive Mode as a Calculator	21
1.10.2	Executing a Python Program Using the IPython Interpreter	23
1.10.3	Writing and Executing Code in a Jupyter Notebook	24
1.11	Internet and World Wide Web	29
1.11.1	Internet: A Network of Networks	29
1.11.2	World Wide Web: Making the Internet User-Friendly	30
1.11.3	The Cloud	30
1.11.4	Internet of Things	31
1.12	Software Technologies	32
1.13	How Big Is Big Data?	33
1.13.1	Big Data Analytics	38
1.13.2	Data Science and Big Data Are Making a Difference: Use Cases	39
1.14	Intro to Data Science: Case Study—A Big-Data Mobile Application	40

**2 Introduction to Python Programming** **49**

2.1	Introduction	50
2.2	Variables and Assignment Statements	50

2.3	Arithmetic	52
2.4	Function <code>print</code> and an Intro to Single- and Double-Quoted Strings	56
2.5	Triple-Quoted Strings	58
2.6	Getting Input from the User	59
2.7	Decision Making: The <code>if</code> Statement and Comparison Operators	61
2.8	Objects and Dynamic Typing	66
2.9	Intro to Data Science: Basic Descriptive Statistics	68
2.10	Wrap-Up	70

### 3 Control Statements and Program Development 73

3.1	Introduction	74
3.2	Algorithms	74
3.3	Pseudocode	75
3.4	Control Statements	75
3.5	<code>if</code> Statement	78
3.6	<code>if...else</code> and <code>if...elif...else</code> Statements	80
3.7	<code>while</code> Statement	85
3.8	<code>for</code> Statement	86
	3.8.1 Iterables, Lists and Iterators	88
	3.8.2 Built-In <code>range</code> Function	88
3.9	Augmented Assignments	89
3.10	Program Development: Sequence-Controlled Repetition	90
	3.10.1 Requirements Statement	90
	3.10.2 Pseudocode for the Algorithm	90
	3.10.3 Coding the Algorithm in Python	91
	3.10.4 Introduction to Formatted Strings	92
3.11	Program Development: Sentinel-Controlled Repetition	93
3.12	Program Development: Nested Control Statements	97
3.13	Built-In Function <code>range</code> : A Deeper Look	101
3.14	Using Type <code>Decimal</code> for Monetary Amounts	102
3.15	<code>break</code> and <code>continue</code> Statements	105
3.16	Boolean Operators <code>and</code> , <code>or</code> and <code>not</code>	106
3.17	Intro to Data Science: Measures of Central Tendency— Mean, Median and Mode	109
3.18	Wrap-Up	111

### 4 Functions 119

4.1	Introduction	120
4.2	Defining Functions	120
4.3	Functions with Multiple Parameters	123
4.4	Random-Number Generation	125
4.5	Case Study: A Game of Chance	128
4.6	Python Standard Library	131
4.7	<code>math</code> Module Functions	132
4.8	Using IPython Tab Completion for Discovery	133

4.9	Default Parameter Values	135
4.10	Keyword Arguments	136
4.11	Arbitrary Argument Lists	136
4.12	Methods: Functions That Belong to Objects	138
4.13	Scope Rules	138
4.14	import: A Deeper Look	140
4.15	Passing Arguments to Functions: A Deeper Look	142
4.16	Function-Call Stack	145
4.17	Functional-Style Programming	146
4.18	Intro to Data Science: Measures of Dispersion	148
4.19	Wrap-Up	150

## 5 Sequences: Lists and Tuples 155

5.1	Introduction	156
5.2	Lists	156
5.3	Tuples	161
5.4	Unpacking Sequences	163
5.5	Sequence Slicing	166
5.6	del Statement	169
5.7	Passing Lists to Functions	171
5.8	Sorting Lists	172
5.9	Searching Sequences	174
5.10	Other List Methods	176
5.11	Simulating Stacks with Lists	178
5.12	List Comprehensions	179
5.13	Generator Expressions	181
5.14	Filter, Map and Reduce	182
5.15	Other Sequence Processing Functions	185
5.16	Two-Dimensional Lists	187
5.17	Intro to Data Science: Simulation and Static Visualizations	191
	5.17.1 Sample Graphs for 600, 60,000 and 6,000,000 Die Rolls	191
	5.17.2 Visualizing Die-Roll Frequencies and Percentages	193
5.18	Wrap-Up	199

## 6 Dictionaries and Sets 209

6.1	Introduction	210
6.2	Dictionaries	210
	6.2.1 Creating a Dictionary	210
	6.2.2 Iterating through a Dictionary	212
	6.2.3 Basic Dictionary Operations	212
	6.2.4 Dictionary Methods keys and values	214
	6.2.5 Dictionary Comparisons	216
	6.2.6 Example: Dictionary of Student Grades	217
	6.2.7 Example: Word Counts	218

6.2.8	Dictionary Method update	220
6.2.9	Dictionary Comprehensions	220
6.3	Sets	221
6.3.1	Comparing Sets	223
6.3.2	Mathematical Set Operations	225
6.3.3	Mutable Set Operators and Methods	226
6.3.4	Set Comprehensions	228
6.4	Intro to Data Science: Dynamic Visualizations	228
6.4.1	How Dynamic Visualization Works	228
6.4.2	Implementing a Dynamic Visualization	231
6.5	Wrap-Up	234

## 7 Array-Oriented Programming with NumPy 239

7.1	Introduction	240
7.2	Creating arrays from Existing Data	241
7.3	array Attributes	242
7.4	Filling arrays with Specific Values	244
7.5	Creating arrays from Ranges	244
7.6	List vs. array Performance: Introducing %timeit	246
7.7	array Operators	248
7.8	NumPy Calculation Methods	250
7.9	Universal Functions	252
7.10	Indexing and Slicing	254
7.11	Views: Shallow Copies	256
7.12	Deep Copies	258
7.13	Reshaping and Transposing	259
7.14	Intro to Data Science: pandas Series and DataFrames	262
7.14.1	pandas Series	262
7.14.2	DataFrames	267
7.15	Wrap-Up	275

## 8 Strings: A Deeper Look 283

8.1	Introduction	284
8.2	Formatting Strings	285
8.2.1	Presentation Types	285
8.2.2	Field Widths and Alignment	286
8.2.3	Numeric Formatting	287
8.2.4	String's format Method	288
8.3	Concatenating and Repeating Strings	289
8.4	Stripping Whitespace from Strings	290
8.5	Changing Character Case	291
8.6	Comparison Operators for Strings	292
8.7	Searching for Substrings	292
8.8	Replacing Substrings	294

8.9	Splitting and Joining Strings	294
8.10	Characters and Character-Testing Methods	297
8.11	Raw Strings	298
8.12	Introduction to Regular Expressions	299
	8.12.1 re Module and Function <code>fullmatch</code>	300
	8.12.2 Replacing Substrings and Splitting Strings	303
	8.12.3 Other Search Functions; Accessing Matches	304
8.13	Intro to Data Science: Pandas, Regular Expressions and Data Munging	307
8.14	Wrap-Up	312

## **9 Files and Exceptions** **319**

9.1	Introduction	320
9.2	Files	321
9.3	Text-File Processing	321
	9.3.1 Writing to a Text File: Introducing the <code>with</code> Statement	322
	9.3.2 Reading Data from a Text File	323
9.4	Updating Text Files	325
9.5	Serialization with JSON	327
9.6	Focus on Security: <code>pickle</code> Serialization and Deserialization	330
9.7	Additional Notes Regarding Files	330
9.8	Handling Exceptions	331
	9.8.1 Division by Zero and Invalid Input	332
	9.8.2 <code>try</code> Statements	332
	9.8.3 Catching Multiple Exceptions in One <code>except</code> Clause	335
	9.8.4 What Exceptions Does a Function or Method Raise?	336
	9.8.5 What Code Should Be Placed in a <code>try</code> Suite?	336
9.9	<code>finally</code> Clause	336
9.10	Explicitly Raising an Exception	339
9.11	(Optional) Stack Unwinding and Tracebacks	339
9.12	Intro to Data Science: Working with CSV Files	342
	9.12.1 Python Standard Library Module <code>csv</code>	342
	9.12.2 Reading CSV Files into Pandas <code>DataFrames</code>	344
	9.12.3 Reading the Titanic Disaster Dataset	346
	9.12.4 Simple Data Analysis with the Titanic Disaster Dataset	347
	9.12.5 Passenger Age Histogram	348
9.13	Wrap-Up	349

## **10 Object-Oriented Programming** **355**

10.1	Introduction	356
10.2	Custom Class <code>Account</code>	358
	10.2.1 Test-Driving Class <code>Account</code>	358
	10.2.2 <code>Account</code> Class Definition	360
	10.2.3 Composition: Object References as Members of Classes	361
10.3	Controlling Access to Attributes	363

10.4	Properties for Data Access	364
10.4.1	Test-Driving Class Time	364
10.4.2	Class Time Definition	366
10.4.3	Class Time Definition Design Notes	370
10.5	Simulating “Private” Attributes	371
10.6	Case Study: Card Shuffling and Dealing Simulation	373
10.6.1	Test-Driving Classes Card and DeckOfCards	373
10.6.2	Class Card—Introducing Class Attributes	375
10.6.3	Class DeckOfCards	377
10.6.4	Displaying Card Images with Matplotlib	378
10.7	Inheritance: Base Classes and Subclasses	382
10.8	Building an Inheritance Hierarchy; Introducing Polymorphism	384
10.8.1	Base Class CommissionEmployee	384
10.8.2	Subclass SalariedCommissionEmployee	387
10.8.3	Processing CommissionEmployees and SalariedCommissionEmployees Polymorphically	391
10.8.4	A Note About Object-Based and Object-Oriented Programming	391
10.9	Duck Typing and Polymorphism	392
10.10	Operator Overloading	393
10.10.1	Test-Driving Class Complex	394
10.10.2	Class Complex Definition	395
10.11	Exception Class Hierarchy and Custom Exceptions	397
10.12	Named Tuples	399
10.13	A Brief Intro to Python 3.7’s New Data Classes	400
10.13.1	Creating a Card Data Class	401
10.13.2	Using the Card Data Class	403
10.13.3	Data Class Advantages over Named Tuples	405
10.13.4	Data Class Advantages over Traditional Classes	406
10.14	Unit Testing with Docstrings and doctest	406
10.15	Namespaces and Scopes	411
10.16	Intro to Data Science: Time Series and Simple Linear Regression	414
10.17	Wrap-Up	423

## **11 Computer Science Thinking: Recursion, Searching, Sorting and Big O** **431**

11.1	Introduction	432
11.2	Factorials	433
11.3	Recursive Factorial Example	433
11.4	Recursive Fibonacci Series Example	436
11.5	Recursion vs. Iteration	439
11.6	Searching and Sorting	440
11.7	Linear Search	440
11.8	Efficiency of Algorithms: Big O	442
11.9	Binary Search	444
11.9.1	Binary Search Implementation	445

11.9.2	Big O of the Binary Search	447
11.10	Sorting Algorithms	448
11.11	Selection Sort	448
11.11.1	Selection Sort Implementation	449
11.11.2	Utility Function <code>print_pass</code>	450
11.11.3	Big O of the Selection Sort	451
11.12	Insertion Sort	451
11.12.1	Insertion Sort Implementation	452
11.12.2	Big O of the Insertion Sort	453
11.13	Merge Sort	454
11.13.1	Merge Sort Implementation	454
11.13.2	Big O of the Merge Sort	459
11.14	Big O Summary for This Chapter's Searching and Sorting Algorithms	459
11.15	Visualizing Algorithms	460
11.15.1	Generator Functions	462
11.15.2	Implementing the Selection Sort Animation	463
11.16	Wrap-Up	468

## **12 Natural Language Processing (NLP) 477**

12.1	Introduction	478
12.2	TextBlob	479
12.2.1	Create a TextBlob	481
12.2.2	Tokenizing Text into Sentences and Words	482
12.2.3	Parts-of-Speech Tagging	482
12.2.4	Extracting Noun Phrases	483
12.2.5	Sentiment Analysis with TextBlob's Default Sentiment Analyzer	484
12.2.6	Sentiment Analysis with the <code>NaiveBayesAnalyzer</code>	486
12.2.7	Language Detection and Translation	487
12.2.8	Inflection: Pluralization and Singularization	489
12.2.9	Spell Checking and Correction	489
12.2.10	Normalization: Stemming and Lemmatization	490
12.2.11	Word Frequencies	491
12.2.12	Getting Definitions, Synonyms and Antonyms from WordNet	492
12.2.13	Deleting Stop Words	494
12.2.14	n-grams	496
12.3	Visualizing Word Frequencies with Bar Charts and Word Clouds	497
12.3.1	Visualizing Word Frequencies with Pandas	497
12.3.2	Visualizing Word Frequencies with Word Clouds	500
12.4	Readability Assessment with <code>Textastic</code>	503
12.5	Named Entity Recognition with <code>spaCy</code>	505
12.6	Similarity Detection with <code>spaCy</code>	507
12.7	Other NLP Libraries and Tools	509
12.8	Machine Learning and Deep Learning Natural Language Applications	509
12.9	Natural Language Datasets	510
12.10	Wrap-Up	510

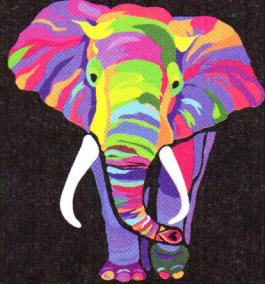


<b>13 Data Mining Twitter</b>	<b>515</b>
13.1 Introduction	516
13.2 Overview of the Twitter APIs	518
13.3 Creating a Twitter Account	519
13.4 Getting Twitter Credentials—Creating an App	520
13.5 What's in a Tweet?	521
13.6 Tweepy	525
13.7 Authenticating with Twitter Via Tweepy	525
13.8 Getting Information About a Twitter Account	527
13.9 Introduction to Tweepy Cursors: Getting an Account's Followers and Friends	529
13.9.1 Determining an Account's Followers	529
13.9.2 Determining Whom an Account Follows	532
13.9.3 Getting a User's Recent Tweets	532
13.10 Searching Recent Tweets	534
13.11 Spotting Trends: Twitter Trends API	536
13.11.1 Places with Trending Topics	536
13.11.2 Getting a List of Trending Topics	537
13.11.3 Create a Word Cloud from Trending Topics	539
13.12 Cleaning/Preprocessing Tweets for Analysis	541
13.13 Twitter Streaming API	542
13.13.1 Creating a Subclass of StreamListener	543
13.13.2 Initiating Stream Processing	545
13.14 Tweet Sentiment Analysis	547
13.15 Geocoding and Mapping	551
13.15.1 Getting and Mapping the Tweets	552
13.15.2 Utility Functions in tweetutilities.py	556
13.15.3 Class LocationListener	558
13.16 Ways to Store Tweets	559
13.17 Twitter and Time Series	560
13.18 Wrap-Up	560
<b>14 IBM Watson and Cognitive Computing</b>	<b>565</b>
14.1 Introduction: IBM Watson and Cognitive Computing	566
14.2 IBM Cloud Account and Cloud Console	568
14.3 Watson Services	568
14.4 Additional Services and Tools	572
14.5 Watson Developer Cloud Python SDK	573
14.6 Case Study: Traveler's Companion Translation App	574
14.6.1 Before You Run the App	575
14.6.2 Test-Driving the App	576
14.6.3 SimpleLanguageTranslator.py Script Walkthrough	577
14.7 Watson Resources	587
14.8 Wrap-Up	589

<b>15</b>	<b>Machine Learning: Classification, Regression and Clustering</b>	<b>593</b>
15.1	Introduction to Machine Learning	594
15.1.1	Scikit-Learn	595
15.1.2	Types of Machine Learning	596
15.1.3	Datasets Bundled with Scikit-Learn	598
15.1.4	Steps in a Typical Data Science Study	599
15.2	Case Study: Classification with k-Nearest Neighbors and the Digits Dataset, Part 1	599
15.2.1	k-Nearest Neighbors Algorithm	601
15.2.2	Loading the Dataset	602
15.2.3	Visualizing the Data	606
15.2.4	Splitting the Data for Training and Testing	608
15.2.5	Creating the Model	609
15.2.6	Training the Model	610
15.2.7	Predicting Digit Classes	610
15.3	Case Study: Classification with k-Nearest Neighbors and the Digits Dataset, Part 2	612
15.3.1	Metrics for Model Accuracy	612
15.3.2	K-Fold Cross-Validation	616
15.3.3	Running Multiple Models to Find the Best One	617
15.3.4	Hyperparameter Tuning	619
15.4	Case Study: Time Series and Simple Linear Regression	620
15.5	Case Study: Multiple Linear Regression with the California Housing Dataset	625
15.5.1	Loading the Dataset	626
15.5.2	Exploring the Data with Pandas	628
15.5.3	Visualizing the Features	630
15.5.4	Splitting the Data for Training and Testing	634
15.5.5	Training the Model	634
15.5.6	Testing the Model	635
15.5.7	Visualizing the Expected vs. Predicted Prices	636
15.5.8	Regression Model Metrics	637
15.5.9	Choosing the Best Model	638
15.6	Case Study: Unsupervised Machine Learning, Part 1—Dimensionality Reduction	639
15.7	Case Study: Unsupervised Machine Learning, Part 2—k-Means Clustering	642
15.7.1	Loading the Iris Dataset	644
15.7.2	Exploring the Iris Dataset: Descriptive Statistics with Pandas	646
15.7.3	Visualizing the Dataset with a Seaborn pairplot	647
15.7.4	Using a KMeans Estimator	650
15.7.5	Dimensionality Reduction with Principal Component Analysis	652
15.7.6	Choosing the Best Clustering Estimator	655
15.8	Wrap-Up	656

<b>16</b>	<b>Deep Learning</b>	<b>665</b>
16.1	Introduction	666
16.1.1	Deep Learning Applications	668
16.1.2	Deep Learning Demos	669
16.1.3	Keras Resources	669
16.2	Keras Built-In Datasets	669
16.3	Custom Anaconda Environments	670
16.4	Neural Networks	672
16.5	Tensors	674
16.6	Convolutional Neural Networks for Vision; Multi-Classification with the MNIST Dataset	676
16.6.1	Loading the MNIST Dataset	677
16.6.2	Data Exploration	678
16.6.3	Data Preparation	680
16.6.4	Creating the Neural Network	682
16.6.5	Training and Evaluating the Model	691
16.6.6	Saving and Loading a Model	696
16.7	Visualizing Neural Network Training with TensorBoard	697
16.8	ConvnetJS: Browser-Based Deep-Learning Training and Visualization	700
16.9	Recurrent Neural Networks for Sequences; Sentiment Analysis with the IMDB Dataset	701
16.9.1	Loading the IMDB Movie Reviews Dataset	702
16.9.2	Data Exploration	703
16.9.3	Data Preparation	705
16.9.4	Creating the Neural Network	706
16.9.5	Training and Evaluating the Model	709
16.10	Tuning Deep Learning Models	710
16.11	Convnet Models Pretrained on ImageNet	711
16.12	Reinforcement Learning	712
16.12.1	Deep Q-Learning	713
16.12.2	OpenAI Gym	713
16.13	Wrap-Up	714
<b>17</b>	<b>Big Data: Hadoop, Spark, NoSQL and IoT</b>	<b>723</b>
17.1	Introduction	724
17.2	Relational Databases and Structured Query Language (SQL)	728
17.2.1	A books Database	730
17.2.2	SELECT Queries	734
17.2.3	WHERE Clause	734
17.2.4	ORDER BY Clause	736
17.2.5	Merging Data from Multiple Tables: INNER JOIN	737
17.2.6	INSERT INTO Statement	738
17.2.7	UPDATE Statement	739
17.2.8	DELETE FROM Statement	739

17.3	NoSQL and NewSQL Big-Data Databases: A Brief Tour	741
17.3.1	NoSQL Key-Value Databases	741
17.3.2	NoSQL Document Databases	742
17.3.3	NoSQL Columnar Databases	742
17.3.4	NoSQL Graph Databases	743
17.3.5	NewSQL Databases	743
17.4	Case Study: A MongoDB JSON Document Database	744
17.4.1	Creating the MongoDB Atlas Cluster	745
17.4.2	Streaming Tweets into MongoDB	746
17.5	Hadoop	755
17.5.1	Hadoop Overview	755
17.5.2	Summarizing Word Lengths in <i>Romeo and Juliet</i> via MapReduce	758
17.5.3	Creating an Apache Hadoop Cluster in Microsoft Azure HDInsight	758
17.5.4	Hadoop Streaming	760
17.5.5	Implementing the Mapper	760
17.5.6	Implementing the Reducer	761
17.5.7	Preparing to Run the MapReduce Example	762
17.5.8	Running the MapReduce Job	763
17.6	Spark	766
17.6.1	Spark Overview	766
17.6.2	Docker and the Jupyter Docker Stacks	767
17.6.3	Word Count with Spark	770
17.6.4	Spark Word Count on Microsoft Azure	773
17.7	Spark Streaming: Counting Twitter Hashtags Using the pyspark-notebook Docker Stack	777
17.7.1	Streaming Tweets to a Socket	777
17.7.2	Summarizing Tweet Hashtags; Introducing Spark SQL	780
17.8	Internet of Things and Dashboards	786
17.8.1	Publish and Subscribe	788
17.8.2	Visualizing a PubNub Sample Live Stream with a Freeboard Dashboard	788
17.8.3	Simulating an Internet-Connected Thermostat in Python	790
17.8.4	Creating the Dashboard with Freeboard.io	792
17.8.5	Creating a Python PubNub Subscriber	794
17.9	Wrap-Up	798



## Preface

*“There’s gold in them thar hills!”<sup>1</sup>*

For many decades, some powerful trends have been in place. Computer hardware has rapidly been getting faster, cheaper and smaller. Internet bandwidth (that is, its information carrying capacity) has rapidly been getting larger and cheaper. And quality computer software has become ever more abundant and essentially free or nearly free through the “open source” movement. Soon, the “Internet of Things” will connect tens of billions of devices of every imaginable type. These will generate enormous volumes of data at rapidly increasing speeds and quantities.

Not so many years ago, if people had told us that we’d write a college-level introductory programming textbook with words like “Big Data” and “Cloud” in the title and a graphic of a multicolored elephant (emblematic of “big”) on the cover, our reaction might have been, “Huh?” And, if they’d told us we’d include AI (for artificial intelligence) in the title, we might have said, “Really? Isn’t that pretty advanced stuff for novice programmers?”

If people had said, we’d include “Data Science” in the title, we might have responded, “Isn’t data already included in the domain of ‘Computer Science’? Why would we need a separate academic discipline for it?” Well, in programming today, the latest innovations are “all about the data”—*data science*, *data analytics*, *big data*, relational *databases* (SQL), and NoSQL and NewSQL *databases*.

So, here we are! Welcome to *Intro to Python for Computer Science and Data Science: Learning to Program with AI, Big Data and the Cloud*.

In this book, you’ll learn hands-on with today’s most compelling, leading-edge computing technologies—and, as you’ll see, with an easily tunable mix of computer science and data science appropriate for introductory courses in those and related disciplines. And, you’ll program in Python—one of the world’s most popular languages and the fastest growing among them. In this Preface, we present the “soul of the book.”

Professional programmers often quickly discover that they like Python. They appreciate its expressive power, readability, conciseness and interactivity. They like the world of open-source software development that’s generating an ever-growing base of reusable software for an enormous range of application areas.

Whether you’re an instructor, a novice student or an experienced professional programmer, this book has much to offer you. Python is an excellent first programming language for novices and is equally appropriate for developing industrial-strength applications. For the novice, the early chapters establish a solid programming foundation.


We hope you’ll find *Intro to Python for Computer Science and Data Science* educational, entertaining and challenging. It has been a joy to work on this project.

---

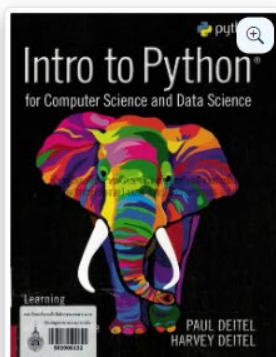
1. Source unknown, frequently misattributed to Mark Twain.

สามารถยืมและติดตามหนังสือใหม่ได้ที่ ระบบห้องสมุดอัตโนมัติ Walai Autolib

<https://lib.rmutp.ac.th/bibitem?bibid=b00108260>

 **Intro to Python for computer science and data science : blearning to program with AI, big data and the cloud / Paul Deitel and Harvey Deitel.**  
Deitel, Paul.

My list 



Subject [Python \(Computer program language\).](#)

#### Details

Added Author [Deitel, Harvey.](#)  
Published New York : Pearson Education, 2020.  
Detail xlviii, 831 p. : ill ; 23 cm.  
ISBN 9780135404676

 13  32  0

 MARC

 Export

 Save

 Share

สำหรับการศึกษาระดับปริญญาตรีและการอ้างอิงเท่านั้น