## Contents

Let's Go

### 1 Getting Started with Python

1.1 Algorithms and algorithmic thinking  
   1.1.1 Algorithmic thinking  
   1.1.2 States  
   1.1.3 What does `a = a + 1` mean?  
   1.1.4 Symbolic versus numerical  

1.2 Launch Python  
   1.2.1 IPython console  
   1.2.2 Error messages  
   1.2.3 Sources of help  
   1.2.4 Good practice: Keep a log  

1.3 Python modules  
   1.3.1 `import`  
   1.3.2 `from ... import`  
   1.3.3 NumPy and PyPlot  

1.4 Python expressions  
   1.4.1 Numbers  
   1.4.2 Arithmetic operations and predefined functions  
   1.4.3 Good practice: Variable names  
   1.4.4 More about functions  

### 2 Structure and Control

2.1 Objects and their methods  

2.2 Lists, tuples, and arrays  
   2.2.1 Creating a list or tuple  
   2.2.2 NumPy arrays  
   2.2.3 Filling an array with values  
   2.2.4 Concatenation of arrays  
   2.2.5 Accessing array elements  
   2.2.6 Arrays and assignments  
   2.2.7 Slicing  

For general queries, contact webmaster@press.princeton.edu
2.2.8 Flattening an array 26
2.2.9 Reshaping an array 26
2.2.10 Lists and arrays as indices 26
2.3 Strings 27
  2.3.1 Formatting strings with the format() method 29
  2.3.2 Formatting strings with % 30
2.4 Loops 30
  2.4.1 for loops 30
  2.4.2 while loops 32
  2.4.3 Very long loops 32
  2.4.4 Infinite loops 32
2.5 Array operations 33
  2.5.1 Vectorizing math 33
  2.5.2 Reducing an array 35
2.6 Scripts 36
  2.6.1 The Editor 36
  2.6.2 Other editors 36
  2.6.3 First steps to debugging 37
  2.6.4 Good practice: Commenting 39
  2.6.5 Good practice: Using named parameters 42
  2.6.6 Good practice: Units 43
2.7 Contingent behavior: Branching 43
  2.7.1 The if statement 44
  2.7.2 On truth 45
2.8 Nesting 45

3 Data In, Results Out

3.1 Importing data 46
  3.1.1 Obtaining data 47
  3.1.2 Bringing data into Python 47
3.2 Exporting data 49
  3.2.1 Scripts 50
  3.2.2 Data files 50
3.3 Visualizing data 52
  3.3.1 The plot command and its relatives 52
  3.3.2 Manipulate and embellish 55
  3.3.3 Error bars 57
  3.3.4 3D graphs 57
  3.3.5 Multiple plots 57
  3.3.6 Subplots 59
  3.3.7 Saving figures 59
  3.3.8 Using figures in other applications 60
4 First Computer Lab

4.1 HIV example 61
   4.1.1 Explore the model 61
   4.1.2 Fit experimental data 62

4.2 Bacterial example 63
   4.2.1 Explore the model 63
   4.2.2 Fit experimental data 63

5 More Python Constructions

5.1 Writing your own functions 65
   5.1.1 Defining functions in Python 66
   5.1.2 Updating functions 68
   5.1.3 Arguments, keywords, and defaults 68
   5.1.4 Return values 69
   5.1.5 Functional programming 70

5.2 Random numbers and simulation 71
   5.2.1 Simulating coin flips 71
   5.2.2 Generating trajectories 72

5.3 Histograms and bar graphs 72

5.4 Contour plots and surfaces 74
   5.4.1 Generating a grid of points 74
   5.4.2 Contour plots 74
   5.4.3 Surface plots 75

5.5 Numerical solution of nonlinear equations 75
   5.5.1 General real functions 76
   5.5.2 Complex roots of polynomials 77

5.6 Solving systems of linear equations 77

5.7 Numerical integration 78
   5.7.1 Integrating a predefined function 78
   5.7.2 Integrating your own function 79
   5.7.3 Oscillatory integrands 79
   5.7.4 Parameter dependence 80

5.8 Numerical solution of differential equations 80
   5.8.1 Reformulating the problem 81
   5.8.2 Solving an ODE 81
   5.8.3 Parameter dependence 83

5.9 Vector fields and streamlines 83
   5.9.1 Vector fields 84
   5.9.2 Streamlines 85

6 Second Computer Lab

6.1 Generating and plotting trajectories 86
6.2 Plotting the displacement distribution 86
6.3 Rare events 88
   6.3.1 The Poisson distribution 88
   6.3.2 Waiting times 89

7 Still More Techniques 91
   7.1 Image processing 91
      7.1.1 Images are arrays of numbers 91
      7.1.2 Manipulating images 92
   7.2 Animation 93
      7.2.1 Creating animations 93
      7.2.2 Saving animations 94
         HTML movies 94
      7.2.3 Using an encoder 96
   7.3 Analytic calculations 97
      7.3.1 The SymPy package 97
      7.3.2 Wolfram Alpha 98

8 Third Computer Lab 100
   8.1 Convolution 100
      8.1.1 Python tools for image processing 101
      8.1.2 Averaging 102
      8.1.3 Smoothing with a Gaussian 102
   8.2 Denoising an image 103
   8.3 Emphasizing features 103

Get Going 105

A Installing Python 107
   A.1 Install Python and Spyder 107
      A.1.1 Graphical installation 107
      A.1.2 Command line installation 108
   A.2 Setting up Spyder 110
      A.2.1 Working directory 110
      A.2.2 Interactive graphics 110
      A.2.3 Script template 110
      A.2.4 Restart 111
   A.3 Acceleration 111
   A.4 Keeping up to date 111
   A.5 Installing FFmpeg 111

B Errors and Error Messages 113
B.1 Python errors in general  113
B.2 Some common errors  114

C  Python 2 versus Python 3  117
C.1 Division  117
C.2 User input  117
C.3 Print command  118
C.4 More assistance  118

D  Under the Hood  119
D.1 Assignment statements  119
D.2 Memory management  120
D.3 Functions  120
D.4 Scope  122
  D.4.1 Name collisions  123
  D.4.2 Variables passed as arguments  124
D.5 Summary  124

E  Answers to “Your Turn” Questions  126

Acknowledgments  131

References  133

Index  135