Introduction to Python for Computational Science and Engineering

Hans Fangohr

Jan 21, 2022

CONTENTS

1 Introduction **3**
- 1.1 Computational Modelling 3
- 1.2 Why Python for scientific computing? 5
- 1.3 Python version 7
- 1.4 These documents 8
- 1.5 Your feedback 9

2 A powerful calculator **11**
- 2.1 Python prompt and Read-Eval-Print Loop (REPL) 11
- 2.2 Calculator 11
- 2.3 Integer division 13
- 2.4 Mathematical functions 14
- 2.5 Variables 16
- 2.6 Impossible equations 18

3 Data Types and Data Structures **21**
- 3.1 What type is it? 21
- 3.2 Numbers 21
- 3.3 Sequences 24
- 3.4 Passing arguments to functions 37
- 3.5 Equality and Identity/Sameness 43

4 Introspection **47**
- 4.1 dir 47
- 4.2 type 51
- 4.3 isinstance 51
- 4.4 help 52
- 4.5 Docstrings 59

5 Input and Output **61**
- 5.1 Printing to standard output (normally the screen) 61
- 5.2 Reading and writing files 67
- 5.3 Further reading 69

6 Control Flow **71**
- 6.1 Basics 71
- 6.2 If-then-else 74
- 6.3 For loop 75
- 6.4 While loop 75
- 6.5 Relational operators (comparisons) in `if` and `while` statements 76
- 6.6 Exceptions 77

7 **Functions and modules** **81**
7.1 Introduction . . . 81
7.2 Using functions . . . 81
7.3 Defining functions . . . 82
7.4 Default values and optional parameters . . . 84
7.5 Modules . . . 85
7.6 Further Reading . . . 89

8 **Functional tools** **91**
8.1 Anonymous functions . . . 91
8.2 Map . . . 92
8.3 Filter . . . 93
8.4 List comprehension . . . 94
8.5 Reduce . . . 95
8.6 Why not just use for-loops? . . . 98
8.7 Speed . . . 98
8.8 The `%%timeit` magic . . . 100

9 **Common tasks** **103**
9.1 Many ways to compute a series . . . 103
9.2 Sorting . . . 106

10 **From Matlab to Python** **109**
10.1 Important commands . . . 109

11 **Python shells** **111**
11.1 IDLE . . . 111
11.2 Python (command line) . . . 111
11.3 Interactive Python (IPython) . . . 111
11.4 Spyder . . . 113
11.5 Editors . . . 113

12 **Symbolic computation** **115**
12.1 SymPy . . . 115
12.2 Related tools . . . 133

13 **Numerical Computation** **135**
13.1 Numbers and numbers . . . 135

14 **Numerical Python (numpy): arrays** **143**
14.1 Numpy introduction . . . 143

15 **Visualising Data** **151**
15.1 Matplotlib – plotting y=f(x), (and a bit more) . . . 151
15.2 The pylab interface . . . 157
15.3 The matplotlib.pyplot interface . . . 164
15.4 Visual Python . . . 171
15.5 Visualising higher dimensional data (VTK) . . . 176
15.6 Further tools and developments . . . 178
15.7 Jupyter Notebooks . . . 181

16 **Numerical Methods using Python (scipy)** **183**
16.1 Overview . . . 183
16.2 SciPy . . . 183
16.3 Numerical integration . . . 185

16.4 Solving Ordinary Differential Equations (ODEs) . . . 187
16.5 Root finding . . . 196
16.6 Interpolation . . . 198
16.7 Curve fitting . . . 199
16.8 Fourier transforms . . . 201
16.9 Optimisation . . . 202
16.10 Other numerical methods . . . 204
16.11 scipy.io: Scipy-input output . . . 204

17 Pandas - Data Science with Python **207**
17.1 Motivational example (Series) . . . 207
17.2 Pandas `Series` . . . 209
17.3 Create Series from list . . . 214
17.4 Plotting data . . . 215
17.5 Missing values . . . 218
17.6 Series data access: explicit and implicit (`loc` and `iloc`) . . . 219
17.7 Data Frame . . . 222
17.8 Example: European population 2017 . . . 226
17.9 Further reading . . . 238

18 Python packages and environments **239**
18.1 Introduction . . . 239
18.2 Python virtual environments . . . 240
18.3 Python Package Index (PyPI) . . . 242
18.4 Anaconda . . . 251
18.5 Managing many different environments - pyenv . . . 253

19 Where to go from here? **255**
19.1 Advanced programming . . . 255
19.2 Compiled programming language . . . 255
19.3 Testing . . . 255
19.4 Simulation models . . . 256
19.5 Software engineering for research codes . . . 256
19.6 Data and visualisation . . . 256
19.7 Version control . . . 256
19.8 Parallel execution . . . 256
19.9 Acknowledgements . . . 256

20 Change history **259**