PHYS4038/MLiS and ASI/MPAGS

Scientific Programming in

mpags-python.github.io

Steven Bamford

University of Nottingham
ASI/MPAGS

Course Introduction
Course prerequisites

• To make the most of this course, you should have:
  • Some programming experience (in any language)
  • Access to a computer with Python installed
    • Anaconda recommended – see course webpage

• Ideally you should also have:
  • Some current or upcoming need of a scripting language
  • A piece of real or toy analysis on which you can try out using Python
Course aims

To give you…
- experience of using a modern scripting language
- introduction to all essential Python syntax
- practical advice about scientific programming
- knowledge of the main scientific modules for Python
- the ability to do basic data analysis tasks in Python
  (e.g. data manipulation, plotting, …)
- knowledge of some specific tools for scientific computing
  (e.g. signal processing, optimisation, …)
- an overview of Python's full capabilities

Not to…
- teach programming in general (but I will try to help!)
- cover every aspect of Python
Course structure

• Ten weeks
• About one hour of recorded lecture videos each week
• Watch at your own pace, try out examples
• Work on exercises and coursework
• Synchronous online session via MS Teams – Fridays at 10am
  • Ask any questions
  • Exercise solutions
  • Help with debugging
• Talk to me:
  • During synchronous Teams sessions (*preferred*)
  • Via Slack channel (you will receive an invitation)
  • Email: steven.bamford@nottingham.ac.uk
Outline

• **Session 1**: Introduction to Python
  • Why Python is (mostly) awesome
  • Writing and running Python
  • Language basics

• **Session 2**: Introduction to Python, continued
  • More language basics
  • Good programming practice

• **Session 3**: Staying organised
  • Managing your environment with conda and pip
  • Version control with GitHub

• **Session 4**: Numerical Python and Plotting
  • Numpy
  • Using arrays wisely
  • Matplotlib (and others)

• **Session 5**: Scientific Python
  • Scipy and other tools
  • Filtering, interpolation, optimisation
• **Session 6**: Data handling
  • Efficiently storing and processing large amounts of data
    • PyTables, Pandas, Dask
    • Multiprocessing

• **Session 7**: Python for specialists
  • Python for astronomers
    • Astropy
  • Python for theorists
    • Symbolic algebra

• **Session 8**: MSc presentations (no lecture / no PhD students)

• **Session 9**: Bayesian inference and Deep Learning in Python
  • MCMC with emcee
  • ANNs with keras

• **Session 10**: Robust, fast & friendly code
  • Testing and timing
  • Wrapping external libraries and creating the fastest code
    • cython, numba, etc.
  • Web applications
Assessment

For those taking this module for MPAGS credits

- Assessed by development of a Python program relevant to your interests
  - put course material into practice
  - opportunity to become familiar with Python
  - get feedback on your coding

- Your code should...
  - be written as an executable module (.py file) or Jupyter notebook (.ipynb)
  - do something meaningful: analyse real data or perform a simulation
  - define at least two user functions (but typically more)
  - make use of appropriate specialist modules
  - produce at least one informative plot
  - comprise >~ 50 lines of actual code
    - excluding comments, imports and other ‘boilerplate’
  - contain no more than 1000 lines in total
    - if you have written more, please isolate an individual element
Code development

- Three stages (first two optional for MPAGS students)

1. hand-in by **28th October** (optional for feedback)
   - README describing what you intend your code to do
   - Rough outline of the code (classes, functions, snippets, comments, pseudocode)

2. hand-in by **18th November** (optional for feedback)
   - Rough version of your code, may be incomplete, have bugs, although try to make it reasonable and easy to understand!

3. hand-in by **16th December** (required for MPAGS credits)
   - Complete working version of your code

Deadlines are 3pm on Wednesdays.
That’s it for today!

Next up:

• **Session 1**: Introduction to Python
  • Why Python is (mostly) awesome
  • Writing and running Python
  • Language basics

• **Session 2**: Introduction to Python, continued
  • More language basics
  • Good programming practice