PHYS4038/MLiS and ASI/MPAGS

Scientific Programming in python

mpags-python.github.io

Steven Bamford
PHYS4038/MLiS

Course Introduction
Course information and materials

• Moodle page
  https://moodle.nottingham.ac.uk/course/view.php?id=106050

  ➔ https://mpags-python.github.io

• Slides and notebooks used in lectures

• Exercises and solutions

• ‘Engage’ lecture recordings available from Moodle page
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
Outline

- **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 University of Nottingham credits, towards a taught Masters or Undergraduate degree:

This is a 10 credit module.

- Code development – 60%
- Presentation on development – 20%
- Final report on development – 20%

All assessed work is performed individually.

You will be given a mark and feedback on each element.
Code development

- 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 – together 60% of module mark

1. hand-in by **28th October** – 5%
   • 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** – 15%
   • 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** – 40%
   • Complete working version of your code

Deadlines are 3pm on Wednesdays.
Presentation and report

• Develop your ability to communicate verbally and through writing:
  • scientific objectives
  • coding choices
  • tests and performance
  • results and implications
  • potential improvements

• Presentation – 20%
  • 5 – 10 minutes
  • In synchronous session in Week 10 (late November)

• Report – 20%
  • 2 – 3 sides of A4 (~1500 words plus figures)
  • hand-in by 16th December
Feedback

- I aim to provide:
  - feedback on each intermediate activity within ~ 1 week
  - final feedback within ~ 2 working weeks

- Code feedback will be given through GitHub
  - Introduced in Session 3

- Marks and feedback on presentation and report via Moodle
That’s it for today!

Next up:

• **Synchronous intro session**
  • Friday 2\textsuperscript{nd} October

• **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