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=95853

→ 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 sessions, every Monday this term, in George Green A13
  • 13:00 – 14:00 — lecture / workshop
    • mix of PowerPoint and Jupyter notebooks
    • have Python running and try things out as I talk
  • 14:00 – 15:00 — examples class
    • work on exercises and examples
    • ask any questions
    • make progress on coursework
    • help with debugging, etc.
Questions

• Talk to me:
  • During teaching sessions *(preferred)*
    • Specific questions, clarifications – just ask
    • Bigger issues – wait until end of lecture / start of examples class
  • Via email: steven.bamford@nottingham.ac.uk
  • Arrange a meeting
    • email me
    • office: CAPT A112b
Your backgrounds

• Your general programming experience?
  • languages
  • level
  • projects

• Your prior Python experience?

• Any particular things you want to be covered?
Provisional 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
  - Numpy
  - Using arrays wisely

- **Session 5**: Plotting with Python
  - Matplotlib (and others)

- **Session 6**: Scientific Python overview
  - Scipy and other tools
Provisional outline

• **Session 7:** Scientific Python examples
  • Filtering, interpolation, optimisation

• **Session 8:** Data handling
  • Efficiently storing and processing large amounts of data
    • PyTables, Pandas, Dask
    • Multiprocessing

• **Session 9:** Robust, fast & friendly code
  • Testing and timing
  • Wrapping external libraries and creating the fastest code
    • cython, numba, etc.
  • Web applications

• **Session 10:** Python for specialists
  • Python for astronomers
    • Astropy
  • Python for theorists
    • Symbolic algebra
  • Bayesian inference and Deep Learning in Python
    • MCMC with emcee
    • ANNs with keras
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
• Three stages – together 60% of module mark

1. hand-in by **1st November** – 5%
   • README describing what you intend your code to do
   • Rough outline of the code (classes, functions, snippets, comments, pseudocode)

2. hand-in by **15th 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 **13th December** – 40%
   • Complete working version of your code

Deadlines are 3pm on Fridays.
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 examples class on 25th November

• Report – 20%
  • 2 – 3 sides of A4 (~1500 words plus figures)
  • hand-in by 13th December
Feedback

• I aim to provide:

• feedback on each intermediate activity within ~ 1 week

• final feedback within ~ 2 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 (with MPAGS students):

• **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