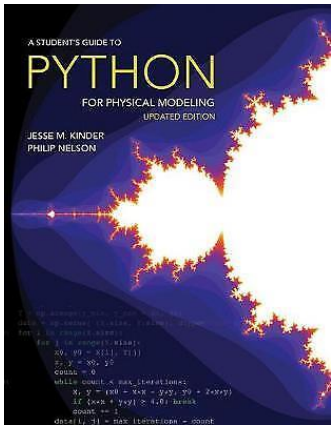
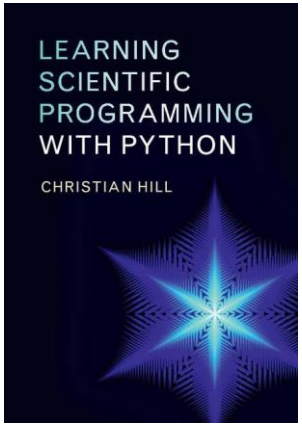
Phys105 – Introduction to Computational Physics

- Prof. Tim Greenshaw.
  - Office hours, Mon. 16:00-17:00.
  - Email green@liv.ac.uk
- Dr Carl Gwilliam.
  - Office hours, Tues. 16:00-17:00.
  - Email C.Gwilliam@liverpool.ac.uk
- Lectures:
  - Wednesday 15:00-16:00.
  - Online (MS Teams).
- Computer classes:
  - Two hours every second week.
  - Time depends on your bubble!
  - Online now, may return to the MOTC (Maths, room 101).

- Recommended textbooks:
  - "A Student's Guide to Python for Physical Modeling", Kinder and Nelsen (Princeton University Press).



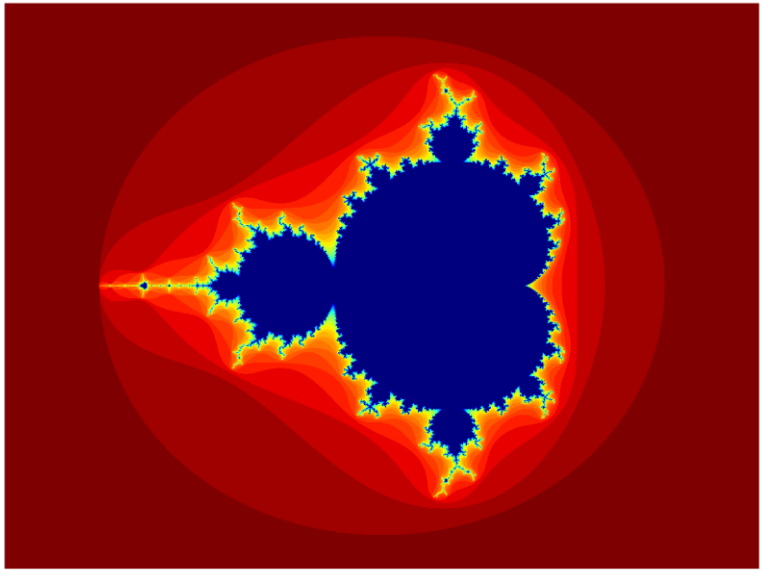
  - "Learning Scientific Programming with Python", Hill (Cambridge University Press).

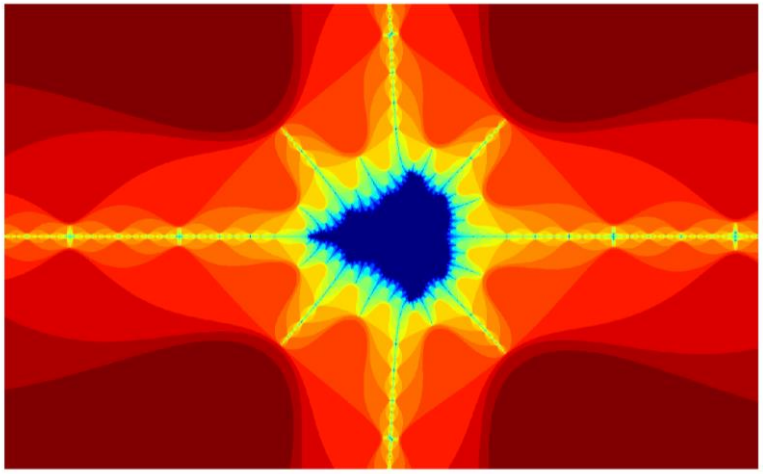


- Assessment:
  - Computer Classes.

Phys105 – Introduction to Computational Physics

- Outline syllabus:
  - Installing Python and necessary packages/tools.
  - Introduction to Python and Jupyter Notebooks.
  - First steps with Python.
  - Python and NumPy data structures.
  - Plotting data.
  - Generating random numbers and using these in Monte Carlo models.
  - Numerical solution of differential equations.
  - Introduction to computer algebra.





Aims and methods

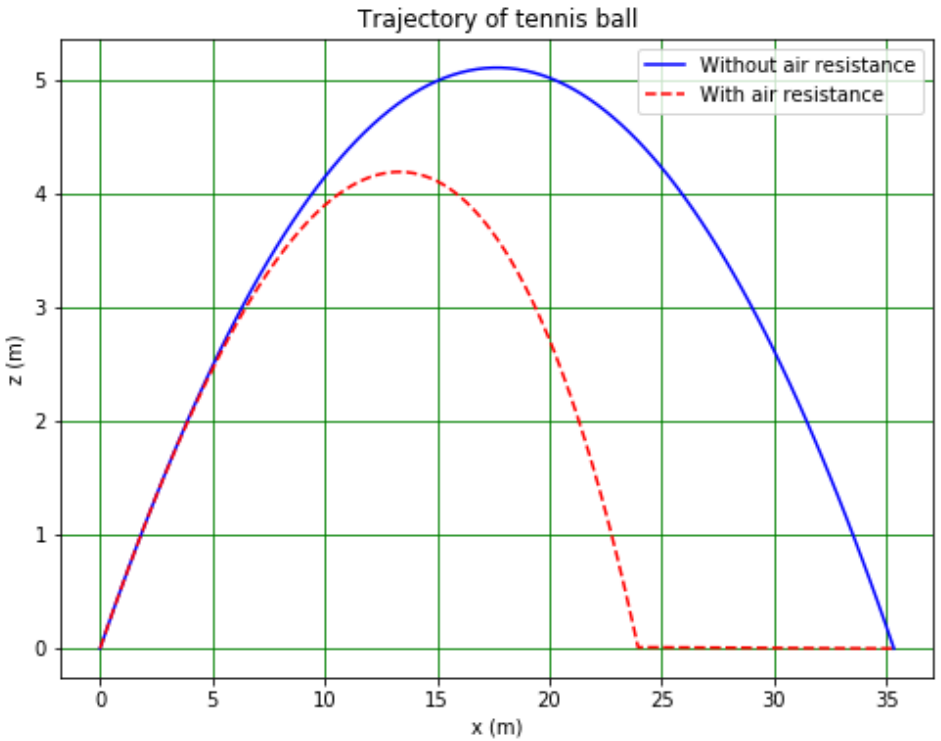
- Why do “Computational Physics”?
- An example.
- A tennis ball is thrown upwards from ground level at an angle of 30° to the horizontal and a speed of 20 m/s. Calculate how far it flies before its first bounce.
- We all know how to solve this problem using…

$$s = ut + \tfrac{1}{2}at^2 \qquad s = \tfrac{1}{2}(u+v)t$$

$$v^2 - u^2 = 2as \qquad a = \frac{v-u}{t}$$

- …but the agreement of our calculation with experiment, is poor!

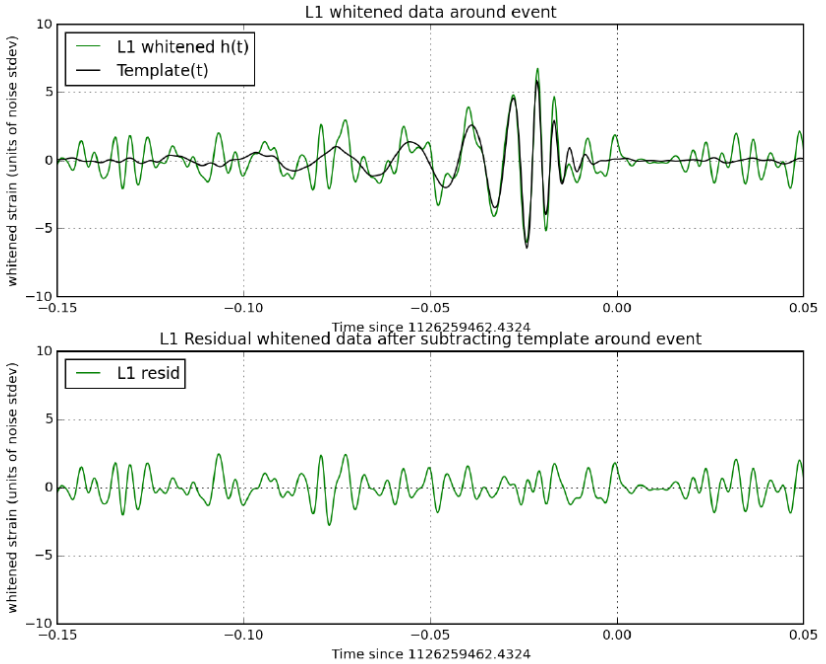
- To provide a realistic answer, need to include air resistance.
- No algebraic solution, but can be solved numerically using a computer.



Aims and methods

- Aim of Phys105 is to show how this problem can be solved, and also how computers can help with:
  - Analysing and presenting data.
  - Simulating the behaviour of stochastic systems.
  - Differentiating and integrating functions.
- Computer can also be used to do many other things we won't look at in this course, such as control equipment or capture data.
- We will use the Python programming language and work in Jupyter Notebooks.

- Python is one of the most widely used programming languages…
- … and Jupyter Notebooks are an excellent way of documenting scientific analysis.
- See e.g. https://losc.ligo.org/tutorials/

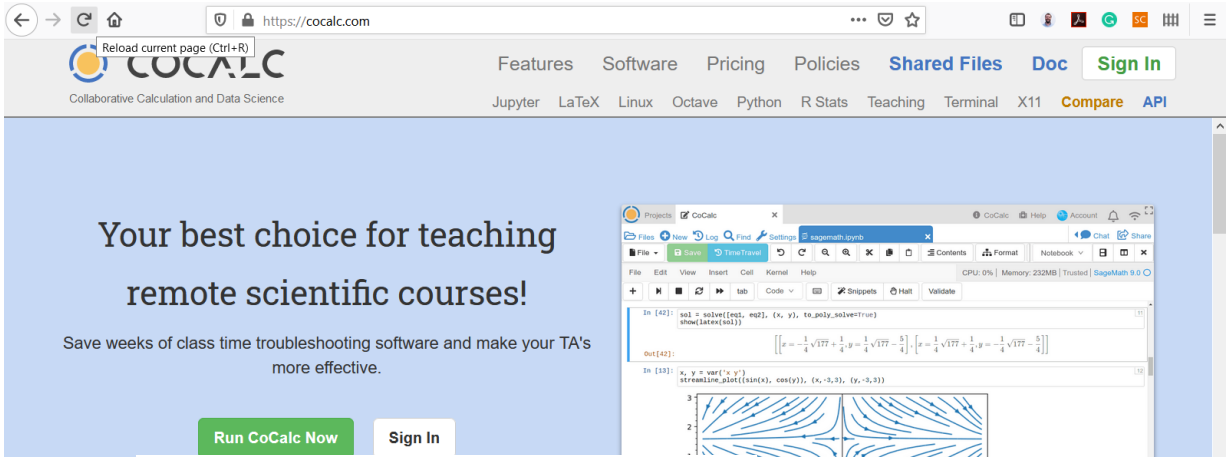


This lecture

- In this lecture we will:
  - See how to navigate to the course material on CoCalc.
  - Introduce Jupyter Notebooks.
  - Look at some Markdown examples.
  - Have a first look at using *least_squares* to fit a straight line to some data points.

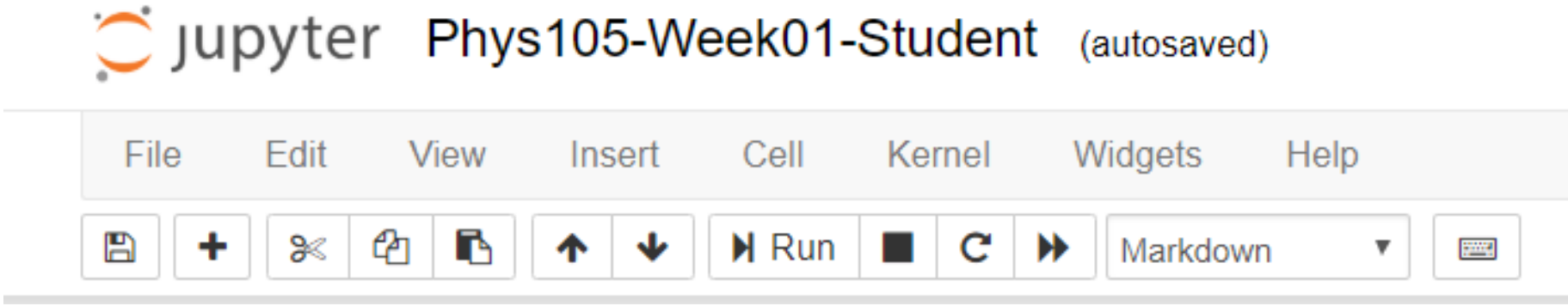
Finding the course material

- Start CoCalc at https://cocalc.com.
- Sign in using the ID you have created.



- Click **John Shaw - Phys105 Introduction to Computational Physics/Phys105**
- Then **Phys105 Introduction to Computational Physics**
- And **ComputerClassesStudent**
- And **Phys105-Week01**
- And finally on **Phys105-Week01-Student.ipynb**

Jupyter Notebooks



- Jupyter Notebooks consist of cells which can contain either computer code (Python) or text (with tables, hyperlinks, images, formulae…).
- When you create a new cell, its default type is Code.
- To change this to Markdown, select it (click in it) then click on *Cell* (menu bar), *Cell Type* and *Markdown*.
- Alternatively:
  - Select the cell.
  - Press *Esc*.
  - Type *m*.
- To change the cell back to code, use the menu or select the cell, press *Esc* and then type *y*.
- To delete a cell use the menu or press *Esc*, *d*, *d*.

Markdown cells

- Text can be entered directly into Markdown cells.
- Create italics by writing _italics_ or *italics*.
- Bold text is obtained using __bold__ or **bold**.
- To see the results run (or compile) the cell by selecting it and using *Run* in the menu bar…
- …or pressing *Shift* + *Enter*.
- These commands run the cell and select the next cell, or create a new cell if there is no next cell.
- Using *Ctrl* + *Enter* just runs the cell.

- A table as Markdown:

```
|Number | Angle (degrees)| Cosine of angle|
|-------|----------------|----------------|
|0      |0               |1               |
|1      |30              |0.866           |
|2      |45              |0.707           |
|3      |60              |0.5             |
|4      |90              |0.0             |
```

- The same table when it is compiled:

Number	Angle (degrees)	Cosine of angle
0	0	1
1	30	0.866
2	45	0.707
3	60	0.5
4	90	0.0

- Switch back to Markdown by double-clicking in the cell.

Code cells

- Code cells are used to write elements of a Python program, e.g.

```
In [ ]: 3 + 4
```

- When this cell is run, the code is executed:

```
In [1]: 3 + 4
Out[1]: 7
```

- Editing the cell and re-running it gives…

```
In [2]: 3 + 2
Out[2]: 5
```

- We will learn to develop Python programs from scratch in this module…
- …but the first thing we will look at is how to use *least_squares* and an existing program to fit a straight line to data, as this is needed for Practical Physics I (Phys106).

Comments

- You can enter comments into both Markdown and Code cells.
- These are used to explain what the Markdown or Python is doing.
- For Markdown, the format is:

```
This you see, <!-- this
you don't, --> and this you see again!
```

- When this is run, the result is:

This you see, and this you see again!

- For Code cells, use #…

```
In [6]: 3 + 4 + 5
Out[6]: 12
```

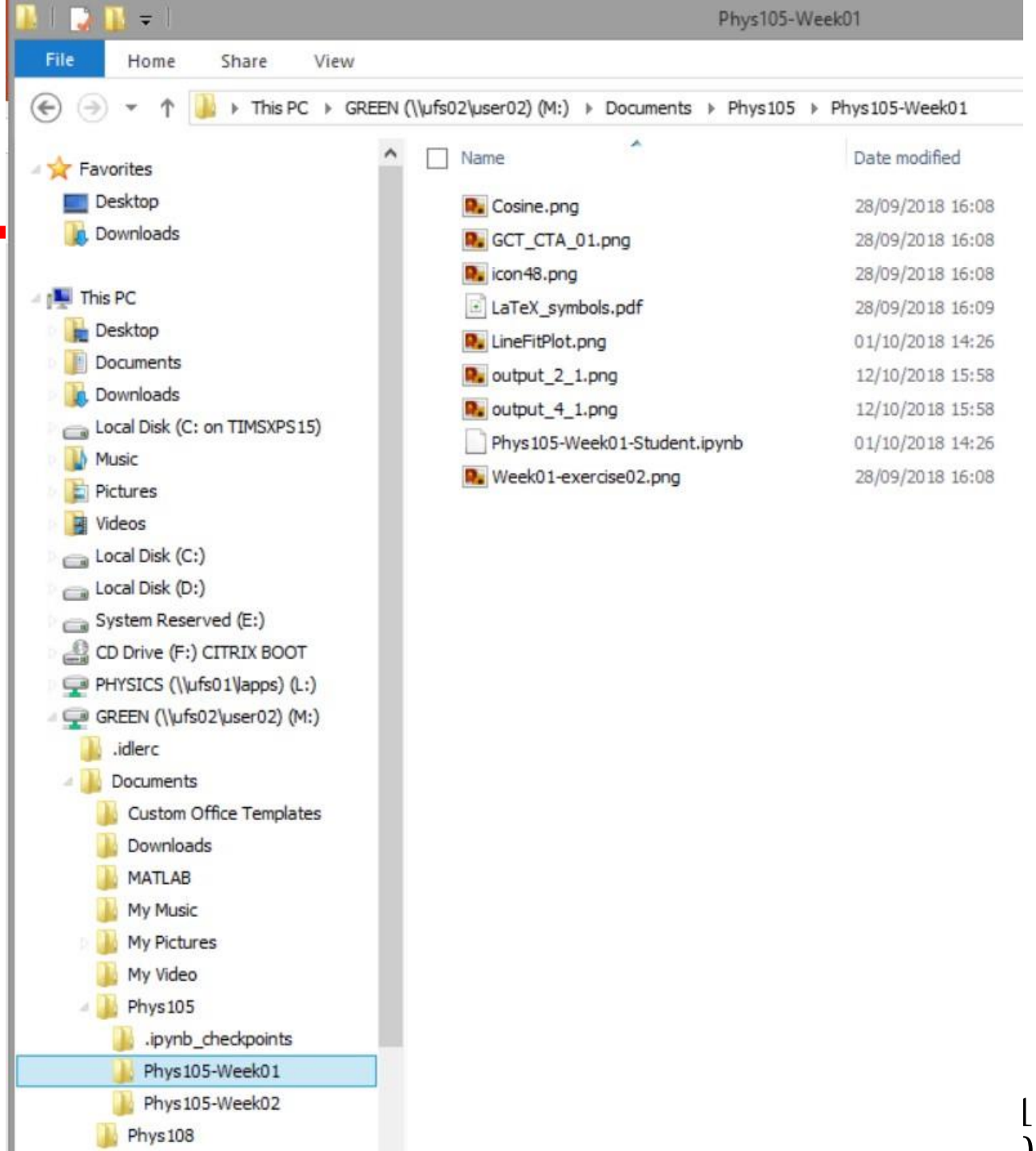
```
In [5]: 3 + 4 # + 5
Out[5]: 7
```

- …or three quotes.

```
In [8]: '''3 + 4 + 5'''
Out[8]: '3 + 4 + 5'
```

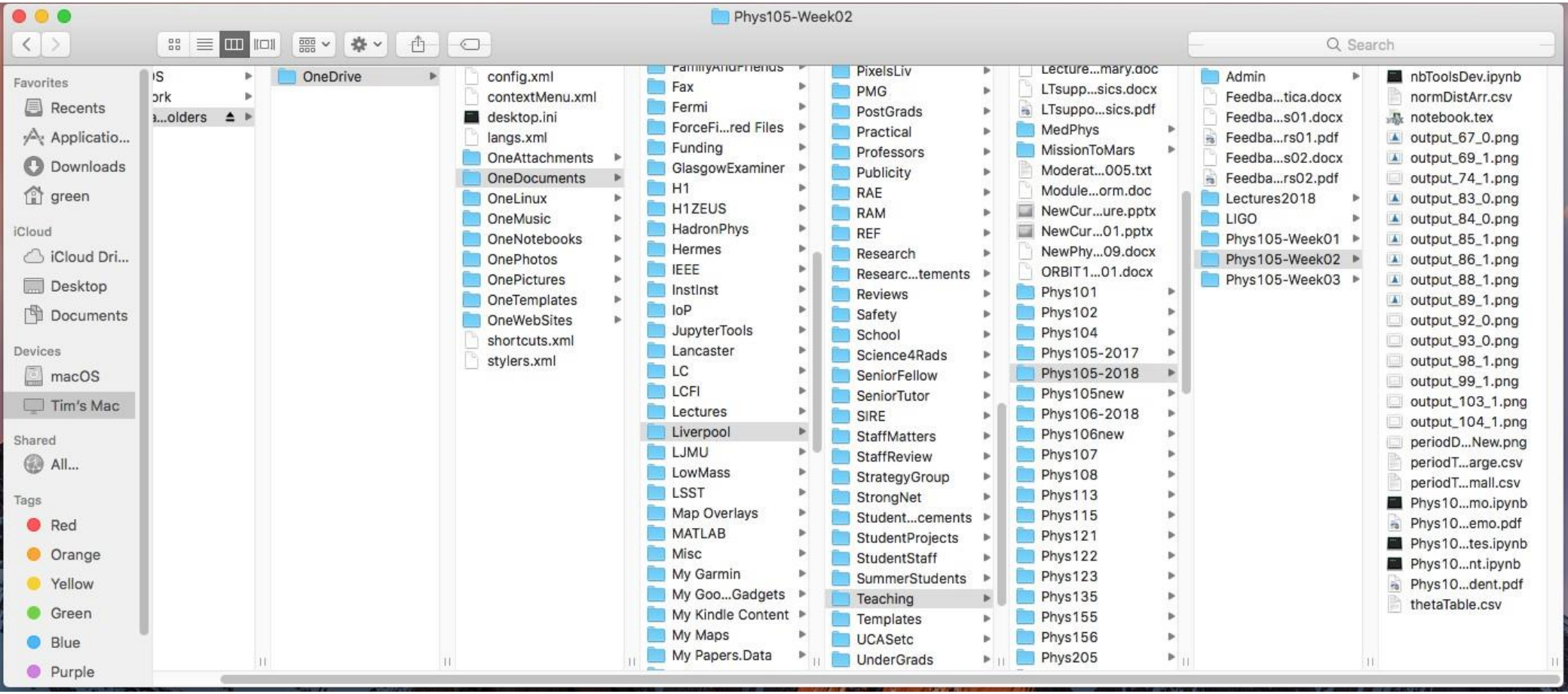
Directories on Windows

- Use *File Explorer*.
- Access via *Start*, *Windows System*, *File Explorer.*
- In *File Explorer*, click *View* and select *Navigation Pane*.
- Shows you all folders in “tree” on the left and selected folder’s contents on the right.
- Can move (or copy) files on the right into folders on the left…



Directories on a Mac

- Use *Finder*, can start from *Dock*.
- Choose columns view 



Directories on Linux

- Many different flavours of Linux!
- Example here is using CentOS 7.
- Select *Applications*, *Accessories*, *Files*.