### Phys105 – Week 2

# Comments on Week 1

- This week:
  - Comments on first week's Notebook
  - Arithmetic with Python
  - Functions in Python

- One common problem...
- Make sure you run all the code cells in the right order.
- E.g. if you haven't run:

#### [1]: import numpy as np

You will see something like the following if you try and use numpy:

```
NameError Traceback
<ipython-input-1-c9359eaab8fe> in <module>
    4 #
    5 # Define NumPy arrays, initially filled with
----> 6 xData = np.zeros(nPoints)
    7 yData = np.zeros(nPoints)
    8 #
```

NameError: name 'np' is not defined

## Comments on Week 1 and some hints

- When adding a picture in a Markdown cell, your Notebook must be able to find the picture.
- It will only look in the directory (or folder) in which it is located, unless you tell it to look elsewhere (e.g. by using a URL).
- The pictures available are:
  - Any of the \*.jpg files.
  - Any of the \*.png files.
- You can upload more pictures to your CoCalc space!

Туре	Name
ipynb	Phys105-Week01-Student.ipynb
\$	DUE_DATE.txt
	Python450pounds.jpg
	Week01-exercise02.png
	Phys105-Week01-Student.pdf
	LaTeX_symbols.pdf
	LineFitPlot.png
	GCT_CTA_01.png
	icon48.png
	<b>elephant</b> .jpg
	Cosine.png
	BallTraj.png

# Comments on Week 1 and some hints

- Can show line numbers in code cells to help with location of problems.
- Either:
  - Click View then Toggle Line Numbers.
- Or:
  - Make sure you haven't selected a Markdown cell in edit mode or a Code cell and then press *Shift l*.
- You will then see:

```
In [1]: 1 # <!-- Cell 16 -->
2 # nPoints is the number of data points in the fit
3 nPoints = 10
4 #
5 # Define NumPy arrays, initially filled with zeros,
6 xData = np.zeros(nPoints)
7 yData = np.zeros(nPoints)
8 #
```

- Python will do its best to tell you where and what the problem is.
- But sometimes the descriptions are a little cryptic and the location is not quite right...
- ...there are just too many ways we programmers can get it wrong!

# Arithmetic with Python

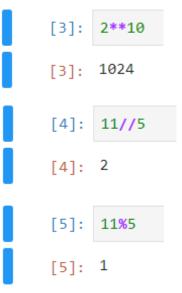
- One of the simplest things we can do using Python is to add, subtract, multiply and divide numbers.
- In a Jupyter Notebook, enter the sum you want to do in a Code cell and run the cell:

[1]: 3 - 6

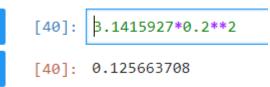
```
[1]: -3
```

- The answer appears below your code.
- In addition to the operators "+", "–", "\*" and "/", Python provides "\*\*", "//" and "%".

Look at examples:



- These examples use integers.
- Can also use real numbers ("floats"):



# Variables and functions Python

```
Can define variables:
```

```
[6]: r = 0.2
pi = 3.1415927
A = pi*r**2
round(A, 3)
[6]: 0.126
```

```
• And change their values:
```

```
[7]: r = 2.2 # m
pi = 3.1415927
A = pi*r**2 # m**2
round(A, 3)
```

[7]: 15.205

• Use print statement to provide intelligible output:

```
[9]: print("Radius of circle",r)
```

```
Radius of circle 0.2
```

Python provides many functions...

```
[11]: 1 theta = 0.3
2 $ = sin(theta)
3 print("sin",theta,"=",s)

NameError Tracel
<ipython-input-11-ee03b2a7a159> in <module>
1 theta = 0.3
----> 2 s = sin(theta)
3 print("sin",theta,"=",s)
```

NameError: name 'sin' is not defined

...but these are included in packages like numpy.

```
[12]: 1 import numpy as np
2 theta = 0.3
3 s = np.sin(theta)
4 print("sin",theta,"=",s)
```

```
sin 0.3 = 0.29552020666133955
```