Lecture 1

- Find course web site @ markwilde.com/teaching

- have everyone introduce + say why they're interested in course

- see about changing time of course to MW only

- following course of Mark Newman, University of Michigan closely, but will deviate here + there

Course overview:

computation is an indispensable tool in modern physics. Nearly every problem is solved in some way w/ the aid of computers

such problems include:

numerical integration, inverting large matrices, solving nonlinear differential equations, etc.

Plan: ~ 1 month learn Python
~ 1 month numerical techniques
~ 1 month applications to physical problems (but this discussed throughout)

Why Python? quite popular now, easy to write & understand programs, powerful numerical libraries & visualization tools, it is free

should install python 3 (details in Appendix A of book)

Anyone need access to computer?

Simple example of Python code:

Rydberg formula for wavelengths of emission lines of hydrogen:

$$\frac{1}{\lambda} = R\left(\frac{1}{m^2} - \frac{1}{n^2}\right)$$

R - Rydberg constant $= 1.097 \times 10^{-2}\ nm^{-1}$

m & n are positive integers
For fixed m, n > m forms a series

Python program to generate λ's for 1st five lines of 1st three series

```
R = 1.097e-2
for m in [1, 2, 3]:
    print("Series for m =", m)
    for k in [1, 2, 3, 4, 5]
        n = m+k
        invlambda = R*(1/m**2 - 1/n**2)
        print("  ", 1/invlambda, "nm")
```

program already illustrates several key features of Python

- program flow indicated by indentation, ensuring programs are formatted well
- for loop, list
- assignment
- math calculations
- output

Now to Chapter 2...

Some Python basics

- Install from python.org & scipy stack
- Bring up IDLE
- Bring up "New File"

Write simple program

```
x=1
print(x)
```

Click Run Module

Python programming basics

program is a list of instructions which computer executes

quantities of interest are variables

```
>>>  x=1
```

assignment statement, tells computer to store value 1 in variable x

names can be longer, better to use longer names

types of variables:

integers - positive & negative numbers

float - floating point variable
real numbers

complex - complex numbers such as
$1 + 2j$
$-3.5 - 0.4j$
$j = \sqrt{-1}$ imaginary unit

good to have different types b/c of memory issues:
integers take less space than floats
& floats " " " complex

Also calculations w/ complex #s take longer to complete.

Also accuracy issues:

computers don't have infinite accuracy so sometimes calculations w/ integers can be better.

So best to use integers when quantity is genuinely an integer.

How to set type?

When you set its value, you set its type.

For example,

x = 1

x = 1.5

type is initially integer but then it becomes float. Although you can do this, best to avoid

tends to make programs confusing

best to make a given variable have just one type.

How to make x a float if it's value is 1? Just do

```
x = 1.0
```

could also write

```
x = float(1)
```

similar issue for complex numbers, could write

```
x = 1.5 + 0j
```

or

```
x = complex(1.5)
```

Another type of variable is "string"
(won't come up too much in physics programming)

like

x = "This is a string"

You can put numerical values in a string but then they are not interpreted as numbers

Also, you can put as much whitespace as you need in a given program. Can be helpful to make it readable.

go to test2.py + run some commands