Lecture 2

0 Queries!

Talk about comments. For query 4, explain multiple tests!

1 Expressions

1.1 Casting

```
x = 3.8
int(x)
x
```

1.2 Evaluate

Show them and then vote! **Note: floats have at least one decimal place of precision, do this on your midterms!**

2 Multiple Files

So far everythin we've done has been in one file. This is *very* unlikely for large scale projects. Programs could have 100s of files. In Python, each file is called a module.
Show `my_time.py`. What if I want to use these in a different file?
Make `use_time.py`

```
import my_time

print("The number of seconds in a day is", seconds_per_day)
```

This won't work! I didn't say where `seconds_per_day` came from! Add module name and .

3 Functions

3.1 Built-ins

Hopefully you've already seen the idea of functions in a math class at some point. Let's take a look at how to use and then define our own functions. Python already has a bunch of functions built in.

```
len("Monday")
abs(-10)
int(32.9)
float(1)
sqrt(4) # doesn't exist
# But!  There is a sqrt function in the math module
import math
math.sqrt(4)
# The math module has a lot more stuff
# some of which we will see later
```

3.2 Defining Functions

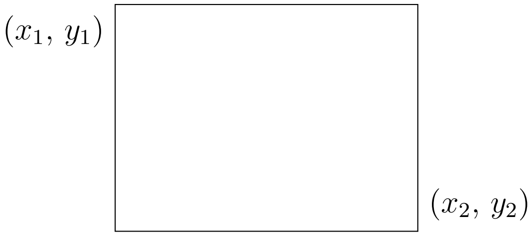
We can also define our own functions! Why should we? Write code once, test it, reuse it. Don't repeat code, put it in a function. In general, a function should do *one* thing and it should do that one thing really well.
Let's make $f(x) = 3x + 2$ in Python! (Put in `funcs.py`) ***TALK ABOUT INDENTATION!!***

```
def f(x):
    return 3*x + 2
```

Python functions can be a little more general than mathematical ones. Write `is_zero` function. Again, variable names can be verbose. Introduce == (and `!=` I guess?)

```
def is_zero(value):
    return value == 0
```

Write `area_of_rectangle(x1, y1, x2, y2)`.



```
def area_of_rectangle(x1, y1, x2, y2):
    width = abs(x2 - x1)
    height = abs(y1 - y2)
    return width * height
```

Then make a `use_funcs.py` and `test_funcs.py`