Week4 Lecture 2.3

So now, we're going to combine everything in

the previous lectures and we're going to actually make our first program.

So before we make this program,

we need to talk a little bit about documentation.

We talked about mnemonic variables as a friendly thing for humans.

Another thing that's a friendly thing in programs for humans,

and you will tend to realize that the human that you're being

friendly to when you're writing code and you're doing it well, is yourself.

So good variable names are going to help you a lot.

They might help your teaching assistant but they're also going to help you.

Comments. Comments are a way for us to add text that's ignored by Python to our programs.

So it's a great way to give a little idea what's going on in code,

maybe there's a couple of lines coming up that are 10

lines long that's a little complex and you want to say,

"Oh, all this is really doing is reversing

the order of these two things," or who knows what.

Or you might document who wrote the code and you can actually just take a line of

code and put a pound sign in front of it instead of deleting the line of code.

Sometimes, you do that for debugging.

You add debugging code and then when you don't want the debugging code,

you just put a pound sign in case you want

to take the pound sign out to turn the code back on.

So I use it as a way to turning off and on lines of code.

So here is a word frequency that we keep coming

back to and what I've added here is I've added four comments.

I've added comments that basically help us understand what's going on,

and remember I call these like paragraphs.

Our first paragraph has human-readable code, human-readable text.

Get the name of the file and open it.

That's what's going to happen in the next two lines.

These next five lines are count the word frequency,

maybe I should say make a histogram.

So make a histogram and you can just read this.

So, again, don't think of this as

like the teacher's telling you you've got to write comments.

Think of this as like, "I'll write comments so I remember what I was doing here.

Why did I write this code?"

So write the comments to help yourself out.

Now, here, the last thing,

we're all done and we're printing this stuff out, okay?

So that's comments, it's an important part of documenting

your code so that you can figure it out later.

Again, the human who's going to like you for doing all this stuff is you. All right.

So this is you and you're giving a gift to the future you, right?

So you are writing comments so that the future you can read the comments.

Because in a day or a week or a month,

you won't quite remember what you were doing so go ahead and write comments.

But don't write them because I said so.

Write only the ones that you find useful and don't do something like x equals 1,

pound sign, put one in x.

That's silly, everyone can figure out what x equals 1 means.

So you don't put silly comments in just for silliness,

you just put them in especially when you

have to understand what's going on here

so that people don't have to read quite so detailed.

One of things they would do is they would read this and check to see

if it really did what you said it was going to do to help you debug it,

for example. So that's comments.

Documentation, very important.

Okay. So now,

we're going to do our first programming code.

Now, the pattern that's going to happen in here is

a pattern that I call and I was taught 25,

30, many years ago.

Input processing and output are the essential things that computers do.

They gather some input maybe from a file or from a web service.

They do some work to it and then they produce something.

In, work, out.

The work is the hard part usually.

So this program is our first program that demonstrates all three of those things;

input, processing, and output.

It's a three line program with one comment.

So the problem that this program is trying to solve,

for those of you who have traveled in the US and traveled everywhere else,

is that the ground floor in hotels in most of

the world is the zero floor and the ground floor in United States is the one floor.

So you might find yourself in a European elevator asking

what is the equivalent United States floor to this floor that I'm sitting on.

So if I'm on floor seven,

what would be this floor if I was in the United States?

That's the problem we're going to solve.

Now, it's probably not going to make us

rich if we build an app for this but perhaps someone can get

the European elevator converter app

into some app store and maybe you will get wealthy after all.

But for now it just is sufficient to

teach us about a full blown program that does input, processing, and output.

This is simple a program as I could make.

So let's take a look at it.

A comment, convert elevator floors,

that has nothing to with Python,

has to do with you or me reading about this.

Then input.

Well, remember, input prints out the prompt and then pauses and waits.

Then we type and then we type the Enter key, right?

Then this zero, this is a string.

The string ends up being input in the variable inp.

Then it continues to the next line and it works on the right-hand side.

We're going to pull this string zero,

we have to convert it to an integer,

otherwise, we can't add one to it.

If we just inp plus one, we'd get a trace back.

But we say int of inp, convert it to an integer.

Now, if you mess this up and put Bob in, then this thing is going to blow up because int can't convert Bob to an integer.

But because we have a zero here, we're okay.

So then we add one to it and store that in usf, mnemonic variable name,

United States floor, inp, the input we got from the user,

I'm using good variable names here.

So we store this in and so we got that

becomes one and then we print out US Floor comma usf,

remember the comma produces this little space down here.

So we have our input, processing, and output in a way that builds us an application.

Okay? Now, there's lots of things, it will have to do.

Most programs aren't one line long and there's a lot of work that we're going to have to learn, but this gets us a start.

So we've talked a lot of stuff.

We talked about constants, we've talked about variables, we've talked about reserved words, we've talked about type, we talked about mnemonic variable names which are both wonderful and a little confusing at the same time, operators,

operator precedence, focused a little bit on division where we talked about Python 2 versus Python 3,

type conversion and comments and then writing an entire program.

So up next, we're going to start talking about conditional execution,

using the if and the else and other of the reserved words.

This is where some of the intelligence starts to seem to come into computers.

So up next, conditional execution.