Chapter 7.1

So welcome to Chapter 7.

This is the payoff.

I hope at the end of this chapter you're like, I get it.

I understand why we had to learn all that other stuff.

Because we're reading files.

从 ::20 开始播放视频并学习脚本0:20

So, everything we've done up until now has been sort of just if a tree falls

in the forest, if a variable gets set in a computer and no one reads it,

did it really happen?

Or if a tree falls in the forest and no one hears it, did it really happen?

So that's kind of what we've been doing so far.

We've set x equals 3 and then asked when we print it back out,

did we do this, whatever.

So we actually haven't done anything that touches the real world,

I guess we did help people get out of lost elevators,

but other than that, we haven't done much interesting at all.

And so, that's because what we are doing so far is practicing, sending

instructions to the central processing unit and getting stuff back and forth.

And so all the variables, like your x and y and fruit and letter,

they all live in here. The code gets run in here.

And so we're just kind of doing this over and over and over again.

This is the first chapter.

And of course, we did go read some stuff from input.

And it comes back in.

And we did print some stuff to output.

So that we've done that.

But now, we're going to start working here.

Because this is where the permanent storage is.

And later we're going to talk to the network, talk to the data band log,

and later we'll talk to database.

And so these are the kind of permanent storage things.

And so everything we've done up to now has either been to the keyboard and the screen, or

just inside the computer.

Because that's what we've been learning how to do basically things.

But now we're going to do things to something, and

that is to a file or to something.

That's where it gets interesting.

And so we have these files and the files we're going to play with initially

are what are called flat text files. They are files that consist of lines.

And the file we're going to play with is called mbox.txt or mboxshort.txt.

And you'll notice that this first line is the exact line that I played with at

the end of the strings chapter.

This turns out to be a standard format called a mailbox format.

And this is when you sort of take a mail program and you export a folder or

a subfolder of that mail program into a flat file.

This is the format that you get.

And so it turns out that it's somewhat useful and it's quite common to have to

write programs to scan through email and look through stuff in that email.

So that's what we're going to do.

从 :2:36 开始播放视频并学习脚本2:36

And so these text files, these flat text files,

meaning that they're not like a word processing document, or a PDF, or

something like that, they're just a set of lines that a program can read.

Every program that you've written when you're writing Python programs are also

flat text files, they just happen to end in .py

And so here is a sequence of lines.

It's some characters, and then a newline.

It's like you hit the Enter key, and then there's more characters, and

then you hit the Enter key, and then it goes back.

It's like a typewriter, type type type type type, Enter key, type type type.

And then it goes back, etc.

And so it's a series of lines...

And so before we can work with a file inside Python,

we want to tell Python that we're going to work with that file.

This is an act we call opening.

It's not actually reading the file.

It's just making the file available to the code that we're going to write.

And so the function that we use is a built-in function in Python, we pass in

the name of the file and whether we're going to read it or write it.

If we leave this out, it's going to be read.

And then it gives us back what's called a file handle.

So an example bit of code is open mbox.txt, so we can read it.

And then give me back the file handle.

File handle is not the data, it's just a way to get at the data.

And we'll see this later, we'll end up with these handles or connections, or

sockets, or whatever.

That they're not actually the data, but a thing we can manipulate.

So, this handle is something that's sort of

a porthole between your program and this file that's sitting on the disk.

And we can open it, we can read from it, we can write to it if we want.

And then we can close the handle and the handle goes away but

the handle is like our connection.

And so if you just open a file, this is for

read because I forgot that second parameter.

I didn't put it in.

I print it out and it is not the actual lines.

It says this is a file, and it's a thing.

And that's what its name is, we're reading it.

And as we interpret the data coming out of the file,

we're going to use the UTF-8 character set to do that.

So it's telling us something but it is not the actual file.

There could be hundreds of thousands of lines of data in that file.

从 :4:58 开始播放视频并学习脚本4:58

If you try to open a file that doesn't work,

that's not there, not surprisingly, we get a traceback, okay?

从 :5:7 开始播放视频并学习脚本5:07

And later we'll see how to use a try and except to deal with the fact that

sometimes files don't exist and you want to know that.

Now, an important character that we haven't played with much,

that's going to kind of rule your life when you deal with files,

is what's called the newline character.

从 :5:23 开始播放视频并学习脚本5:23

And so here is the newline character.

In a string you go blah blah blah blah blah, \n.

从 :5:30 开始播放视频并学习脚本5:30

And so if you print this out one way, which means you just type it here.

It shows you the \n, but

if you run it on a print statement, it actually interprets it and treats that as

a nonprinting character that basically moves to the beginning of the next line.

So that's what the newline does. The newline is not two characters.

We represent it as two characters in a string, but inside the string,

inside Python, it's really just a single character.

So if we say x newline y and we store that in stuff.

So newline.

We print it and it shows this, but if we ask how many characters are in it, it says

there are three characters because that's character one, two, and three. Actually

It's zero, one, and two.

But whatever, that's three characters.

It's not four characters.

So the newline is one character, not two.

But we represent it when we're typing into Python, we represent it as \n.

So, if we think of this file as a sequence of lines,

we sort of visualize it this way.

That's line one, line two, line three, line four.

But actually what's going on is it's actually a long string of characters,

which are punctuated by newlines.

And so these newlines are really in the file.

Go back to the beginning.

It's like hitting the Enter button.

This is that, it remembers the Enter button.

It's not like it's magic,

there is an actual character that represents the Enter button.

And these sometimes on different operating systems,

there are different versions of this.

There's the UNIX, the MS-DOS, but at the end of the day, in these files,

there is a character that says go back to the beginning of the next line.

And even a blank line ends up, so it's like newline, and then

it's newline again and that gives you a blank line.

So, if you take a look here,

this blank line has really got a character that you don't see.

It's a nonprinting character.

Remember when we talked about whitespace?

Newlines are whitespace.

It's things you don't see, but exist.

Right? Things you don't see.

You just have to believe that they're there.

You have to believe there's a newline there on that little line.

从 :7:43 开始播放视频并学习脚本7:43

So after a while, you don't need to imagine these.

Just remember that newlines are there and

the other thing that's interesting that the print function, we print something out.

It automatically sticks a newline at the end for us and there's a way to tell

print not to do that, but by default print actually prints out... and

then it puts the newline there on purpose.

Okay, so that is how files are structured and

where we store them inside the computer.

And up next, we're going to take a look at how we read files in Python.