Chapter 8.3

So now we're going to talk about how strings and lists work together and

this is one of the more common things that we're going to do.

We're going to read some data, we're going to bust it into pieces,

and we're going to look for little fragments of pieces.

从 ::21 开始播放视频并学习脚本0:21

And this is one of people's favorite methods in strings,

and that is the split method.

So we're going to come up with a way to split a string.

And the simple thing we're going to do is we're going to find the spaces and

split it into pieces.

And there is a built-in function in Python called split.

So it's a member of abc, so abc.split, and this returns a list.

So it basically takes a string and gives us back a list.

And the list is with three words. The spaces are gone,

they're chopped up, and we have them.

Now, when we're done we see in that particular string there were three words

and we can pull out the first word by printing stuff sub 0.

So this is a really convenient thing.

Sometimes we want the second word or the third word or who knows,

maybe we want to write a loop that loops through all the words.

It's very, very, very easy to do.

And so now we have stuff and now we have a three,

we're going to loop through them and then we have an iteration variable w

that is going to go through the three words in the list.

So the code to make this work is here's my string, split it, and

then write a simple for loop

that's horizontally going to go through and

look at each of the three words in that.

So it only took three lines of code to do that.

So this loop runs three times and each time through it's with three words.

And this pattern, when you're lost in a future assignment,

come back and look at this slide.

because this is the pattern for how we read a line, break it into pieces, and

then look at each word in the line is a split and then a for loop.

从 :1:50 开始播放视频并学习脚本1:50

Okay. A couple of subtle details about split is that splits

by default on whitespace and it treats more than one space as a single space.

And so even though there's extra spaces here,

when we split this line we still get four things, a lot of spaces.

The words that you would expect to get out of that.

So it's kind of intelligent, an extra space doesn't freak split out.

The second thing about split is that it doesn't have to work with spaces.

And so if you get data like this, maybe from a login system or

from your accounting system or something and they have some weird delimiter,

you can tell split to use a different delimiter.

If we do it just this way, it's looking for spaces and it finds none.

So we get back a list of the words as defined by spaces.

And it doesn't realize these semicolons are what we want it to split

on in the first place.

And we just got a list and that list had one string in it and

that's what happens if there are no spaces in the string.

On the other hand, you can tell split to split based on

a different character other than whitespace.

So when you tell it to split on a different character,

it doesn't do this fancy thing of compressing multiple spaces.

It doesn't split on spaces anymore, it only splits on what you told it to split.

So looks, chops, looks, chops, looks, chops.

And so what we get is a list first, second, and third.

And again, this is when you need to do this because you got some

data that's really weird.

And we'll see situations where we use split.

Instead of looking where we go finding a character and we do a find and

we do another find, we use split and then just grab things.

从 :3:22 开始播放视频并学习脚本3:22

Okay, so this is an example of how we would use split as we parse mail data,

and we are going to do a lot of parsing of mail data.

And so, so here we go.

Again, these lines you are are going to get used to. Open the file. Loop through the file.

Strip the white space off the right-hand side of the thing.

Check to see if it starts with From.

From space, in this case. If it's not, continue.

So this is the skip code.

Skip the lines we're not interested in. And then we're going to split it.

从 :3:51 开始播放视频并学习脚本3:51

And what we're interested in this particular thing is here's these

From lines.

We're only going to read the From lines because theres a lot more in this

mbox-short.

We're looking at what day of the week this thing happened in.

And so you can see these lines.

They always have spaces, so we just chop it based on spaces, chop, chop.

And they'll always be the same thing and so that will be word 0, word 1,

and word 2.

And so word sub 2 is always going to be the day of the week.

And so this is going to skip the lines, skip the lines that don't have From.

As soon as we see From, we split it, and we pull out the second word.

And this is really simple code.

The split makes our life a lot easier.

We don't have to search for a space, we don't have to search for another space.

If you're doing this with just find, you'd have to do something like search for

the first space, then in another line of code you'd search for the second space.

And then you'd search for the third space and

then you'd use string slicing to pull this out.

Yuck. Just split it, go grab the third word, and you're done.

So now what we're going to do is we're going to show you a pattern that I call

double split.

And that's you split something and then you split it again.

And we're going to go back to the problem that we were solving earlier.

And that is we want to pull out this information, right.

And the last time we did it, we did a find.

And then we found the space afterwards, and then we used string slicing.

And that still would work.

That's not a bad way to do it, but

let's look at how we could do this with splitting.

So the first thing we're going to do is we're going to split this into a split

based on the spaces, using the normal split.

And so, now we have words 0, 1, 2, 3, 4, 5, and 6.

And now we're going to grab out words sub 1, okay?

So that's not the host name, that's the email address, okay?

So that pulls out the email address in two really elegant lines.

And if you go to the find strategy, we had to use variables,

we had to remember what the variables mean, you have to draw a little picture.

从 :5:51 开始播放视频并学习脚本5:51

It's hard. Now it's like oh, it's the second thing, just grab it.

It's the second thing. And so that is what it would be.

Then what we're going to do is we're going to do a double split. We're going to

do a second split.

And so we're going to put this variable words sub 1, which is this one right here,

into the email variable and now we are going to split this,

this thing right here, based on at sign

because we know that the email consists of two parts.

One is before the at sign, the second is after the at sign.

And so now what we get is a list.

And so you see that it split that based on the at sign.

We have the second, so this is the 0 piece, and this is the 1 piece.

I tend to use the word "pieces" on this all the time.

If you see my code

I use pieces because when I split it, it gets split into pieces.

And that's why I name my variable pieces.

And now, I split pieces sub 1.

It's not words sub 1, but it's pieces sub 1.

So words was each of these words and

pieces was the pieces after we split that into pieces.

And pieces sub 1 is the second part

of the pieces that came after we split with an at sign.

And so you look at this, compare it to what we did before and

like I understand what that is, and I could write that code quickly.

从 :7:3 开始播放视频并学习脚本7:03

And it's predictable, and I don't have to remember what those variables are.

I don't have to draw myself any pictures, I just go like chop, chop.

This is the 1. Chop.

This is the 1 of that little second piece of pieces.

从 :7:16 开始播放视频并学习脚本7:16

So, that gives us sort of a good start on lists, our first real data structure.

The concept of a collection and where we put multiple things into one variable.

These collections have internal organizing mechanisms.

Lists are organized sequentially with the bracket operator as the lookup operator.

And then we did some operations on strings, on lists

we sorted the lists and we've used split.

And so in the next two chapters,

we're going to refine all of these techniques and add new techniques to them.