[SQUEAKING]

[RUSTLING]

[CLICKING]

MICHAEL SCOTT ASATO CUTHBERT: Hello, everybody. One of the great things about making these videos is that we can introduce small concepts that otherwise might fall through the cracks. So I want to introduce the concept of ties and how we represent them in music21 and how you might represent them in your own projects.

So let's go ahead and import from music21 note, stream, and the new tie module. And we can go ahead and create a stream of a measure. We'll call it m. And let's create two notes. And doesn't really matter what you make them. Just make sure that they're the same pitch because otherwise we can't tie them. That's pretty good.

And we'll say we'll create a time signature. That's why I usually eventually will import star but 2/4. And let's put m.append(ts), m.append(n1). Actually, we can just put more than one thing in, and we'll show these two notes.

And now while it's doing that, we'll go ahead and get n1's. tie will be a new tie.tie object, and it'll be a start. n2's tie will be a tie.tie object, and it will be stop. And now when we show the measure, we have what we should have written as a half note, right?

But we can go ahead, and let's create another measure. And here's, of course, where we would be more likely to do this at, make a quarter rest. That's the default, but just to show it. And so we can, by the way, put objects in more than one stream. And so this is something that's going to happen sometimes. We'll put the first note in, and now we'll create m3 is a stream.measure and m3.append(n2).

Oh, you know I really do care about having all our measures appear, and p is a stream.part. We'll go ahead and put inside the stream m2 and append(m3). And let's go ahead and show that.

Great, so this shows how ties might work. Now, here's the thing. If we do for n in, well, p.recurse.notes, and we try to figure out how many notes there are. Well, it looks like there's three of them. But if we listen to this piece, obviously we're just going to hear two notes.

So what can we do? Well, we can create a new part. We'll say p2 equals part.stripTies. And now if we do for n-- well, actually, let's show this, and we'll show it as a text file. We'll see here that we only have two notes now.

But in this next measure, this note, the C only begins one quarter note in. So we've reduced-- what we've done is we've taken these two E's that are tied, and we've replaced them with one half note. In fact, we can look at this. That's our note and duration. It's a half note.

When we strip ties, we create things sometimes that don't really musically exist. We have a measure that has a time signature of 2/4, has a quarter rest, and then it has a half note in it. That really can't work. So you can't really do p2.show and expect something that looks right because that just wouldn't work.

But instead, in terms of iterating for n in p2.recurse, we'll say, .notesAndRests, [MUMBLES]. Let's print n's offset, and we can see that. And we can even do measure Number ha. This would totally have saved you time on problem set 3, wouldn't it have? Oops, we'll go back and call these number equals 1, measures are auto numbered.

And now when we go back here, we can see the rest is in measure one. Then note E is in measure one at offset 1. In fact, let's print it's-- we should see that E is in measure 1 at offset 1 is half note, and the C is in measure 2 at offset 1 is a quarter note.

I'd make this longer, but that's basically all there is to do with ties. Thanks for listening.