[SQUEAKING]

[RUSTLING]

[CLICKING]

MICHAEL CUTHBERT: Hello, everybody. Congratulations on getting problem set one in. Now the reward is your first unlocking video, so a topic part D. This is topic 1D. And you get to have certain tools that you can use to build future problems and solve future problems and build future problem sets using Music21.

So I'm in Jupiter. You probably want to be doing this-- well, you can do it in Jupiter or you can do it in your own favorite editor. By the way, PyCharm will really save you problems in this class. OK, so we're going to introduce Music21. You can import Music21 but, nah, not really. We're going to do from Music21 import pitch because you've unlocked your first pitch object and pitch module. pitch is lowercase here, so it represents a module in Music21, and we'll be calling things in it.

Now in the problem set, you were probably representing pitches as something like this or enharmonically equivalent, whatever. In Music21, you'll probably want to use our pitch object. So the pitch object is found in the pitch module. The pitch module is lowercase p and the Pitch object is capital P. And we can do the same thing-- A sharp four. And in Jupiter, we can just hit p and get that. You can also, if you're on the terminal, print p and you can see the representation is slightly different.

OK, so let's figure out what we know about this pitch object. So during the problem set, you were creating functions that worked on string or other list inputs. But now that we have an object, we're going to be calling methods and attributes directly on it. So one of the attributes of a pitch is its name. And the name, you can see, is a string-- that's why it has the quotes around it-- A sharp. Great. Well, where is its octave? Take a guess. p.octave.

OK, so this is nothing too surprising here. Let's grab ourselves a slightly different pitch. We'll say b equals pitch.Pitch B0. Let's say B here is that B0. And we can get the pitch B's name, which is b, and also get B's step, which is also B. So those two things look like they do the exact same thing. But go back to our P object and figure out for a second what the difference between name and step is.

OK, did you figure out that p's name has the sharp in it? The p step does not. So this says what diatonic step, what letter name is in the pitch itself? And are these things mutable? I actually made a mistake in the 4:00 to 5:00 class on the first day and said that Music21 objects were not mutable. They actually are. But most of the methods that are called on them try to not mutate by default.

So if you want to, you can change it. So now p is a G sharp four. p's octave stays at four. How do we get that G sharp four? You could do print p.name plus string of p.octave and you get that. Or there's something called name with octave, which is pretty nice.

Now let's create a B flat in octave six. So we'll call it B flat equals pitch.Pitch. Take a second on what goes here. OK, we're back. And then we get B flat in octave six there. Notice, by the way, again that we don't ever name variables with hyphens in them because it's trying to say, well, B minus flat. Yeah, that doesn't make any sense. So B flat is B minus six. And-- sorry, B-- yeah, B flat six. Take a second and try to figure out what if you wanted a B natural in octave negative six. How would you do this?

Great. Did you get really low B equals pitch.Pitch B? We said B four. And now really low B.octave equals negative six. Notice that the name with octave for this is the same as-- oops, where was it-- because B flat's name with octave? So probably, you're going to want most commonly to-- when you're actually working with your objects-- to do the two parts separately so that you can see the difference between that and that.

OK. So now we have a pitch and it's kind of like big whoop dee doo. You could have done this yourself. And why do we need to do anything with this? Well, let's create another pitch, and we'll make it a pitch that's pretty common. We call that middle C, right? Great.

So here's middle C, and we can get its midi number. And so it knows things like this. If we say p.octave equals five, now it's middle of treble clef. And if we can figure out what its midi number is, that's pretty fantastic. We can also do fun, interesting things.

What's its name in Spanish? What's its name? Let's say C double flat. Let's name it Spanish now. I think that that's pretty right. What's its name in German? OK. It's actually a little bit more fun if you work with German names. This and-- great. That's not part of the lecture in this class, but kind of a fun thing to do.

We worked on the distinction between midi and ps in your problem set. So let's recall what does ps stand for. Yeah, pitch space. So the pitch space is the same, in this case, as the midi number. They're exactly the same.

Some things we can do, though, is let's say you set it to 70.5. Does that have a name? Yep. That's kind of weird thing. You can get its full name-- B half flat. OK, so we're learning certain things that we can do with this. But p's ps is 70.5, but midi must be around numbers. So it's, I guess, just B natural.

The other thing is if we go to really low B-- was it really low B flat? No, really low-- It was just B natural. Really low B, and get its octave negative six. Get its name and get its ps. Wow, that's extremely low. That's, what? Is that 109 notes below middle C? But its midi number is constrained to be between zero and 127.

OK, these are just kind of fun things to play around with. You can do a directory on any Pitch object and see some of them. The things that are in here, ignore the ones with underscores for now. We can see that-- oh, there's the frequency. So let's create.

What's the only one that most of us know the frequency of on the top of our head? Great. A4 is 440. So we're-- well, an American made this, so it's not 442 or anything like that. So you'll see that not everything we'll be working with here. There's certain things-- the fundamentals and enharmonics and stuff like that-- that we'll get to. But let's see what some of these are methods and not attributes.

So let's do p.getLowerEnharmonic. And that gives us a G double flat. Maybe we'll make p an A flat, and now we can play around with-- oh, 415. Yep, that's lower. Enharmonic is G sharp. And the higher enharmonic of A flat-- think about it in your head. Great. B triple flat. OK.

By the way, p doesn't change. That's what I said about this being mostly non-mutable. So we have to assign that p triple flat to something else. Triple flat and flat. Yeah, it's so much. Great. And now we can do the B triple flat.name. And let's get its full name. Great.

OK. Going back to p, which is this A flat four. One of the other things we can do if we want to have mutable objects, we'll go lower enharmonic to get the G sharp. But let's-- we can do this particular attribute where we can see in place equals true. And now it doesn't return anything, but p has switched from A flat to G sharp. So you can do things however you would like.

Take a minute now to explore the system. And here are some questions that I'd like you to work on while you're exploring. Number one, what happens if you create a pitch without any arguments? What happens if you create a pitch with a number as an argument? Too big. And what happens if you create a pitch and leave off its octave?

OK, those are the pretty fast ones. Now a little bit more fun ones. How are accidentals stored on a Pitch object in Music21? Do all notes have accidentals? Take a few minutes and work through that on your own.

Great. Are you back? Let's see what our answers are. What happens if you create a pitch without any other arguments? We get that it's a C. And if we get what that note's midi number is-- 60. So it's a C4. OK, something a little bit strange.

What if you create a pitch with a number as its argument? Let's say 61. OK. Now we get a C sharp four. How does it know it's not a D flat? It doesn't. Doesn't really know that. Great.

Now let's see what happens if you create a pitch and leave off its octave. I think we might have already seen this, but C sharp, we'll call this. It's C sharp. And notice the difference here. There's no octave number shown when we do this. But when we look at its midi, it's still 61.

Well, what does this mean? Well, it means that certain things, we're going to end up treating octave as something that's implicit. So here, if we just say pitch is a C and midi is 60, let's get its lower enharmonic. That should be a B sharp. We'll call it bs or B sharp. B sharps are kind of bs.

And now let's look at its midi number. Oh, it's jumped all the way up because what it tries to do in this case is if you give it no octave, it tries to create everything in the octave around middle C. So that's kind of neat.

OK. The next question is a little bit harder. How are accidentals stored on a Pitch object in Music21? Well let's look at bs' accidental. Accidentally spelled that right. And it has this little character on it, which probably means it's some sort of a object thing. So we'll call it acc for accidental, and see that here. And we can get the type of it.

Oh, OK. It is its own object. And we can endure the accidental to see certain things that it knows. Its alteration, what is that? 1.2. That's the number of semitones above the accidental list object there. Great. And well, that's basically the one-- I mean you can do-- I think it has a name also.

Right. Well, that's a sharp. Dot name equals flat. OK, we're going to be able to change-- oops. Flat is string. Does this change? Hey, it does. Now, acc was an attribute on this B sharp object. So now let's see do we change anything about B sharp. Hey, now it's a B flat.

Name-- great. We can see that. We can also do unicode name and get-- OK, great. So we get to see the nice flats. If you ever want to print something out, you have all of that. The last question was do all-- sorry. Oh. That's a mistake. Do all pitches have accidentals? So let's try working on that.

Accidental, obviously, it does. Great. But what about C equals pitch.Pitch? Let's make a low C here. C2. Great. c.accidental. Oh, did I forget to print something? Well, let's fit the alter on it. Oh, we get an error.

So certain notes do not have accidental. Say, if accidental is set to none. But we can assign it explicitly an accidental. So we can call it pitch.accidental natural. And now we can see that it has an accidental with an alteration of zero. I think this is a good enough place to stop for a little bit with pitches. Take a little break, and I'll be right back with you with durations.