



## Coding Sonic Pi: online course with pi-top [4]

- Course includes pi-top [4] Complete & Sensor Foundation Kit
- 12 one hour sessions
- One-to-one tutoring
- For ages 11 to 17, beginner to intermediate level
- Tutors from top universities and with CR checks

**This course teaches kids how to program the Sonic Pi: “the Live Coding Music Synth for Everyone”.**

They’ll connect physical components such as buttons and sound & light sensors to the pi-top [4], which they can use to activate and control their code and extend their learning into the real world. We learn by doing!

Students will learn the fundamentals of coding music as they write their own songs. Once they’ve mastered coding basics such as variables, if/then statements and constructors using coding blocks, they’ll progress to writing their own code in Python.

### Standards

Content aligns to CSTA Level 2 Standards:  
2-CS-02; 2-CS-03; 2-DA-07; 2-AP-11; 2-AP-12; 2-AP-13; 2-AP-14; 2-AP-16; 2-AP-17; 2-AP-19

### Course objectives

#### Outcome:

- Students will learn how to use computers to make things and change things - not just to consume content. They will learn about how computer programs are made, how they work and how to program them.

#### Essential Questions:

- What makes a good program/code?
- How can programs make the things I do *easier*?
- How can programs make the things I do *better*?

Students will understand/know:

- What programming and coding mean
- How programs are written and how they run
- How troubleshooting works
- How to diagram/flow-chart processes

Students will be able to:

- Built and prototype physical computing projects
- Write and run Python code

## Learning Plan

### Day 1: Intro to pi-top, Sonic Pi

*Overview*

Getting started: setting up our pi-top and exploring Sonic Pi.

*Activities*

- Set up pi-top [4]
- Watch Sonic Pi video
- Launch Sonic Pi
- Explore

### Day 2: Sonic Pi day

*Review*

Yesterday we set up your new pi-top [4] and learned about Sonic Pi. What were you able to do inside of Sonic Pi that you thought was interesting? What can we do with a tool like Sonic pi?

*Overview*

Programming basics. What is a program? What is code? What can programs do for us? What can they do for us in Sonic Pi?

*Activities*

- Play a note
- Play a rest
- Change the note and the rest
- Write a song

### Day 3: Intro to Python & Sonic Pi

*Review*

Yesterday we used Sonic Pi to play music and make songs. How is this different from other ways of making music? How is this different from other ways of playing music?

### *Overview*

Today we will go one step further down the path of making music on our computer by removing the Sonic Pi program from the equation, and programming the music in a more common programming language - Python. Python is a language that can do a lot more than just play music - it is used to build software applications, control robots, and much more.

### *Activities*

- Code a note, a rest, and a song in Python
- How can this change what we do with our music?

## Day 4: Python + Sonic Pi + hardware

### *Review*

Yesterday we learned to code music in Python. We talked about how this was different from coding music in Sonic Pi.

### *Overview*

Today we will learn how to incorporate another Python ability into our music making. We'll program physical buttons that we can use to make things happen with our music.

### *Activities*

Use a button to:

- Play notes
- Hold notes
- Change notes

## Day 5: Advanced Python + hardware

### *Review*

Yesterday we learned to use physical hardware to control music making in real time.

### *Overview*

Today we'll add a second button, lights, and new code to make even more complex musical tools

### *Activities*

- Code a loop, add lights that play with each note
- Start and stop the loop with a button
- Code two separate loops that are each controlled by a separate button

## Day 6: Introduce potentiometers

### *Review*

Yesterday we added more buttons, lights, and new code (loops) to our projects to make our most complex musical creations yet.

### *Overview*

Today we will switch out our buttons for knobs and find out how that can change the way we interact with our music, and even change our music.

### *Activities*

- Introduce potentiometers
- Use potentiometers to change the pitch of a note
- Use potentiometer to control one of two loops

## Day 7: Potentiometers & loops

### *Review*

Yesterday we learned how to use potentiometers to send signals to the pi-top in a range of values. We learned how to make different things happen at different values - like changing the pitch of a note, or setting specific ranges to play certain loops, light lights, etc.

### *Overview*

Today we will try to code multiple loops that can all be activated from different points on our potentiometers.

### *Activities*

- Code 6 loops
- Loops 1-3 are controlled by the left side of the pi-top; 4-6 are controlled by the right side

## Day 8: Libraries

### *Review*

Yesterday we built an elaborate sound machine that could play six different loops based on two knobs and two buttons. It even lit up!

### *Overview*

Today we will learn how to store even more information that we can call up whenever we need it - no more having to program loops each time we want them! Today we learn libraries.

### *Activities*

- Build a library with two loops in it
- Build a code where each button press calls up a different loop from the library

## Day 9: Call libraries from the OLED screen

### *Review*

Yesterday we learned how to store code in libraries that could be used in other code we are writing. This saves us from having to re-code something every time we want to use it.

### *Overview*

Today we will call up loops from our libraries using the pi-top [4] OLED screen and programmable buttons, then play them with a button.

### *Activities*

- Build a program to pull loops from the library
- Display them on the OLED screen
- Select loops

## Day 10: Use the OLED screen to assign values to components

### *Review*

Yesterday we learned how to use the OLED screen. We used it to access loops from the library of code we have built.

### *Overview*

Today we will create a program that can pull loops from libraries using the OLED and programmable buttons, then assign them to a value on the potentiometer or to a button.

### *Activities*

Build the assigned program:

- Connect a loop library to the OLED/programmable buttons
- Assign selected loops to different positions on the potentiometer (0-33%; 34-66%; 67-100%)

## Day 11: Ultrasonic sensor

### *Review*

Our musical creations are becoming elaborate and crazy. Yesterday we learned how to take the loops we can call from a library and assign them to buttons using just our physical computing hardware.

### *Overview*

The only answer is to move forward. Combine everything you have learned to build something ludicrous. To help, let's add the ultrasonic sensor.

### *Activities*

- Quick dive into an ultrasonic theramin!
- Experiment all day!

## Day 12: Ultrasonic sensor

### *Review*

This is it. We've learned as much as we can! We started by setting up a pi-top and running songs in Sonic Pi, then we added buttons, loops, knobs, lights, libraries, screens, more buttons, and an ultrasonic sensor.

### *Overview*

Today you'll build something totally your own!

### *Activities*

- Take what you've made so far and put it in a cardboard console
- Rearrange components
- Write the most elaborate song you can
- Play!