



Python with CodeX Units Overview

Pre-Mission Work (5-10 hours)

Build a foundation for programming by utilizing some unplugged activities. If your students come with no Computer Science background, it is important to start by building a foundation of computational thinking. Dedicate some time for students to learn basic terms, such as algorithm, program, and debug. See the Firia Labs collection of Unplugged Activities at <https://learn.firialabs.com/curricula/cs-unplugged>.

Mission 0 (Set up class)

When you are ready for your students to start the CodeX missions, you will need to set up a class for them in CodeSpace and give them a class join code.

Unit 1: Getting Started (8-20 hours)

Students will learn about the programming environment, the CodeX, and basic commands for programming the CodeX using Python. Students start by turning on the RGB pixels, then displaying images on the LCD and finally by playing mp3 files.

Mission 1: Welcome	30-60 minutes
Mission 2: Introducing CodeX	30-60 minutes
Mission 3: Light Show	45-60 minutes
Mission 3 Remix	1-3 hours
Mission 4: Display Games	1-2 hours
Mission 4 Remix	1-3 hours
Mission 5: Micro Musician	30-60 minutes
Mission 5 Remix	1-3 hours
Unit 1 Remix and Assessment	2-5 hours

Unit 2: Inputs and Outputs (7-19 hours)

Students continue their programming journey by updating variables and using lists. Students first learn to increment and decrement a variable to adjust the speed of a flashing image. Then they learn about lists and how they can manage complexity. They learn the basics of lists, like how to create a list, access an item in the list, and scroll through a list using a variable for an index. Finally, students learn about random numbers and functions that are in the random module and how they can be used with lists.

Mission 6: Heartbeat	45-90 minutes
Mission 6 Remix	1-3 hours
Mission 7: Personal Billboard	60-120 minutes
Mission 7 Remix	1-3 hours
Mission 8: Answer Bot	45-90 minutes
Mission 8 Remix	1-3 hours
Unit 2 Remix and Assessment	2-5 hours

Unit 3: Functions and Sensors (9-20 hours)

Students learn about and use functions, an added level of abstraction. The programs also use a variety of inputs, like an internal clock, the accelerometer and the light sensor, and different ways of showing output.

Mission 9: Game Spinner	60-120 minutes
Mission 9 Remix	1-3 hours
Mission 10: Reaction Tester	45-90 minutes
Mission 10 Remix	1-3 hours
Mission 11: Spirit Level	45-90 minutes
Mission 11 Remix	1-3 hours
Mission 12: Night Light	45-90 minutes
Mission 12 Remix	1-3 hours
Unit 3 Remix and Assessment	2-5 hours



OPTIONAL UNITS – More advanced math and programming concepts

Elementary students may not be ready for these units.

Unit 4: Graphics and Sounds (6-15 hours)

Students learn about creating graphics on CodeX. First, students learn more about the draw features and create a graphical user interface. Then they use the draw features to create line art. These projects use a bit of math, but do not let that keep you from completing these engaging programs.

Mission 13: Sounds Fun	1-2 hours
Mission 13 Remix	1-3 hours
Mission 14: Line Art	1-2 hours
Mission 14 Remix	1-3 hours
Unit 4 Remix and Assessment	2-5 hours

Unit 5: Python Applications (7-14 hours)

Students put all their learning and skills together to make a vintage arcade game.

Mission 15: Handball	1-2 hours
Mission 16: Breakout	1-2 hours
Final Project and Assessment	5-10 hours

Possible Pacing Guides

Elementary classrooms have a unique schedule compared to middle and high school computer science classes. Computer science is more likely an elective or “specials” class that doesn’t meet every day. The pacing for completing the missions will be different and unique for each classroom. The overview includes suggested time for each mission. Some missions can be completed in one period, and some can be spread out over two or more periods. Don’t feel like all missions need to be completed. Don’t rush through any of the missions, and take time for remix projects. The main goal for elementary students working through this Mission Pack should be to enjoy programming and want to continue learning.

Recommendations:

1. If your students have no background in computer science, start with some fun unplugged activities.
2. Start with Missions 1, 2 and 3.
3. After Mission 3, and every mission following, give students time to experiment and complete their own remix project. A remix gives students a chance to practice the new concepts and have fun in an engaging way.
4. An assignment (mission log) is available for each mission. If the instructions in CodeSpace are difficult to follow, a slide deck and student workbook are provided that chunk the material into manageable instructions.
5. A review Kahoot for each Mission is available and can be used for an assessment. Also, completed programs for missions and remixes can be used for assessment.
6. If computer science or STEM is an elective and not necessarily for a grade, you can skip the review, unit remix and unit assessment.
7. After Mission 8, skip Mission 9 and come back to it later. Missions 10, 11 and 12 are shorter than Mission 9 and are very engaging. They use sensors and the CodeX clock for really fun projects.
8. End with Mission 9 and a final project.
9. If you have time in the school year:
 - a. Look at some cross-curricular projects
 - b. Continue with Mission 13. Just be prepared to give extra help or supplement with extra lessons to help students with more advanced math and topics.