



CodeHS

Introduction to micro:bit Programming with Python Syllabus 1 Semester for Middle School (30-60 contact hours)

Course Overview and Goals

The CodeHS introduction to Physical Computing with micro:bit curriculum allows students to refresh their knowledge of basic programming concepts (control structures, variables, functions, etc.) in order to control a physical device. Students will perform basic physical tasks using LEDs, buttons, and sensors to see how computer programming gives physical devices the ability to interact with their environment.

Learning Environment: This course utilizes a blended classroom approach. The content is provided through a mix of web-based and physical exercises, with students writing and running code in the browser and then downloading code to their physical devices for further testing and exploration. Teachers utilize tools and resources provided by CodeHS to leverage time in the classroom and give focused 1-on-1 attention to students. Each unit of the course is broken down into lessons. Lessons consist of video tutorials, short quizzes, pseudocode exercises, physical explorations, example programs, and written programming exercises, adding up to over 30 hours of hands-on programming practice in total.

Programming Environment: Students write and run programs in the browser using the MakeCode editor and will download their programs to their micro:bit devices for further testing.

Technology Requirements: To complete all activities and exercises in this course, students must have access to the 3rd party sites and tools listed here: [Introduction to micro:bit Programming with Python Course Links](#)

More information: Browse the content of this course at <https://codehs.com/course/26839>

Prerequisites

This course is designed to reinforce understanding of computer science concepts by applying them to physical devices. It is assumed students have introductory knowledge of variables, control structures, and functions. Students should have completed (or be concurrently enrolled in) an introductory programming course.

The following courses would work as a concurrent prerequisite:

- Intro to Python with Tracy
- Intro to Programming with Karel
 - **Note:** This course does not cover variables, so teachers will likely need to supplement instruction in the physical computing course.
- Intro to Computer Science in Python
- Intro to Computer Science in JavaScript

Course Breakdown

Unit 1: Intro to Physical Computing with micro:bit (10 hours)

Browse the full content of this unit at <https://codehs.com/library/course/26839/module/38222>

Objectives / Topics Covered	● Intro to physical computing ● Goal Setting ● Comments ● Pseudocode ● Analog vs. digital ● Variables ● Connecting external components
Example Assignments / Labs	● 4 explorations ● 11 exercises total ● Example exercises: ○ Blinking Diamond ■ Blink LEDs between a large diamond shape and small diamond shape every half a second as long as the program is running ○ Moving Bright Box ■ Light the corners on the grid with the highest brightness for half a second. Light the corners of the inner box with a brightness of 150 for half a second. Light the middle LED with a brightness of 50 for half a second. Repeat until the program is manually ended. ○ Opposite Blinking External LEDs ■ Blink two external LEDs opposite one another. One LED should be lit while the other is off. After 1 second, the lit LED should turn off and the unlit LED should turn on. After 1 second, they should switch again. This should continue until the program is manually ended

Unit 2: Program Control with micro:bit (10-15 hours)

Browse the full content of this unit at <https://codehs.com/library/course/26839/module/38223>

Objectives / Topics Covered	● For loops ● While loops ● Variables ● Making sound ● If statements ● If/else statements ● Using buttons ● Using servo motors ● Operators (arithmetic, comparison, and logical) ● Using sensors (light, temperature, acceleration, distance) ● Functions and parameters
Example Assignments / Labs	● 4 explorations ● 17 exercises in total ● Example exercises:

	○ Twinkle Twinkle ■ Control your micro:bit to play Twinkle Twinkle Little Star! Use loops (maybe more than one!) to play the song until the program is manually stopped. ○ Servo Position by Button Press ■ Increase the servo position by 90 degrees each time Button A is pressed. Decrease the servo position by 90 degrees each time Button B is pressed. Repeat this motion until the program is manually ended. ○ Temperature Monitor ■ Create a temperature monitor. If the temperature is between 60 and 70 degrees fahrenheit, the middle LED should stay on. If the temperature is below 60 or above 70 degrees fahrenheit, the middle LED should blink on and off at a speed of a quarter second. Repeat this motion until the program is manually ended. ○ LED Arrow Following Servo ■ When button A is pressed, increase the servo position by 90 degrees. When button B is pressed, decrease the servo position by 90 degrees. Use functions to light an LED arrow on the screen that points in the direction of the servo (up, down, left, or right).
--	---

Unit 3: Advanced micro:bit (20-30 hours)

Browse the full content of this unit at <https://codehs.com/library/course/26839/module/38224>

Objectives / Topics Covered	● Challenges ● Explore a new sensor ● Build a step-by-step project ● Final project
Example Assignments / Labs	● Example exercises: ○ Explore a new sensor ■ Research a sensor we have not studied in this course and explore how it is used. In a group, develop and present a lesson to teach your peers about your chosen sensor, including exercises where they can practice using the sensor for themselves. ○ Build a step-by-step project ■ Find a ready-made project online. Follow the steps to recreate the project using your micro:bit and any needed materials. ■ Create an updated set of directions complete with pictures and tips from your experience. ○ Final project ■ Use your micro:bit to bring an idea to life using sensors and external components. ■ Present your project to peers, administration, and family!