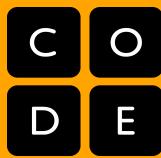


# Computer Science Discoveries

## Curriculum Overview



**Why Computer Science? Every 21st century student should have the opportunity to learn computer science. The basics of computer science help nurture creativity and problem-solving skills, preparing students for a future in any field or career.**

### What is Computer Science Discoveries?

Computer Science Discoveries (CS Discoveries) is an introductory computer science course designed for 6th - 10th grade students. The curriculum emphasizes problem-solving, creation, collaboration, while introducing students to the many ways computer science impacts their lives.

### K-12 curriculum pathway

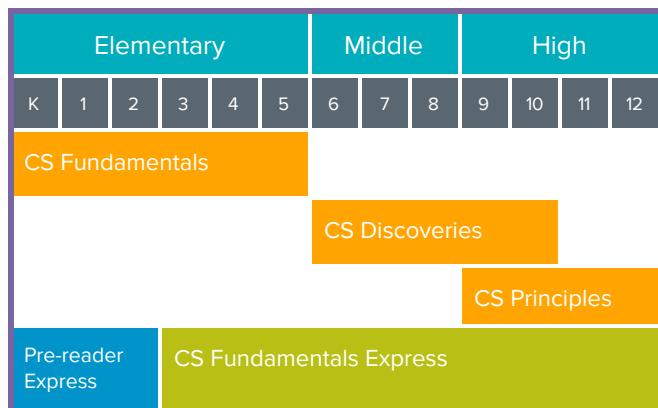
CS Discoveries fits naturally between our CS Fundamentals course (for K-5th grade) and our CS Principles course (for high school). This allows districts, teachers, and students to complete a K-12 pathway of CS courses that build on each other and cohesively flow together.

### Flexible Implementation

We built the CS Discoveries curriculum for students in grades 6-10, so teachers can implement in either middle school or high school classrooms. The two semesters spiral upon each other, allowing the course to be taught as a single semester (Units 1-3), two sequential semesters, a full-year course, or even integrated into existing technology classes.

### Designed for Equity

We designed this course from the ground up to be accessible and engaging for all students, regardless of background or prior experience. By providing students opportunities to engage with culturally and personally relevant topics in a wide variety of CS related fields, we hope to show all students that CS can be for them.



### Professional Learning Program

Teachers implementing CS Discoveries as a semester or full-year course can apply to participate in a one-year professional learning program. The program has two phases:

- **Summer:** Teachers attend a 5-day in-person, conference style workshop designed to introduce CS concepts from the curriculum and core teaching practices. (Travel may be required.)
- **School Year:** Teachers engage in continued learning through professional development workshops focused on supporting their first year of implementation.



**"I do not have a computer science background. I would change nothing about the training. It was an incredible experience, and I felt valued and respected."**



**"It was absolutely rich and the most meaningful training I have ever attended in 16 years of teaching."**

## Curriculum Features:

- Daily instructional lesson plans that include inquiry- and equity-based pedagogy and background content
- Formative and summative assessments, exemplars and rubrics
- Videos for students and teachers including concept tutorials, instructional guides, and lesson tips
- Code.org — a learning platform that organizes lesson plans and activities with student and teacher dashboards

## Curriculum tools:

- **App Lab:** JavaScript programming environment on Code.org, designed for creating event driven web apps with block-to-text workspace and debugging capabilities
- **Game Lab:** JavaScript programming environment on Code.org, designed for creating object oriented sprite-based games and animations with block-to-text workspace and debugging capabilities
- **Circuit Playground:** Adafruit's new Arduino-based microcontroller that has a number of components and sensors built right onto the board and is used as the hardware for physical computing lessons. Using the Maker Toolkit, students can program their Circuit Playground boards right from App Lab with easy-to-understand JavaScript commands and blocks
- **Web Lab:** HTML/CSS programming environment on Code.org used for website development



## CS Discoveries unit overview

### Semester 1: Exploration and Expression

<b>Unit 1</b> Problem Solving	Explore the problem-solving process and the different ways humans and computers solve problems
<b>Unit 2</b> Web Development	Discover the languages powering the web. Build your own websites in HTML and CSS using Web Lab
<b>Unit 3</b> Animations and Games	Learn the powerful constructs underlying programming languages. Build interactive animations and games in JavaScript using Game Lab

### Semester 2: Innovation and Impact

<b>Unit 4</b> The Design Process	Follow a design process to identify and empathize with problems faced by a target audience. Prototype an app to help solve that problem using App Lab.
<b>Unit 5</b> Data and Society	Develop binary representations of different kinds of information. Collect, analyze, visualize, and make automated decisions using data.
<b>Unit 6</b> Physical Computing	Explore the relationship between hardware and software while building interactive projects on Adafruit's Circuit Playground.

Learn more about professional learning! <https://code.org/professional-learning>

For curriculum, videos, support documents, and more, visit: <https://code.org/csd>

Code.org is a 501(c)3 non-profit dedicated to expanding participation in computer science education by making it available in more schools and increasing participation by women and underrepresented students of color. The Code.org vision is that every student in every school should have the opportunity to learn computer programming.

