

Lesson 1: Flappy Code

47 minutes

Overview

In this lesson, learners get an introductory experience with computer science and create a game using basic block code. This lesson has been designed for learners in the middle grades, ages 10-16, but can be adapted for younger or older learners using the differentiation suggestions provided.

Purpose

This lesson introduces the core CS concepts of coding and programming (using blocks), as well as simple debugging techniques.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2011)

- ▶ **CPP** - Computing Practice & Programming
- ▶ **CT** - Computational Thinking

Common Core English Language Arts Standards

- ▶ **SL** - Speaking & Listening

Common Core Math Standards

- ▶ **MP** - Math Practices

ISTE Standards for Students

- ▶ **1** - Students demonstrate creative thinking, construct knowledge, and develop innovative products and processes using technology.
- ▶ **4** - Students use critical thinking skills to plan and conduct research, manage projects, solve problems, and make informed decisions using appropriate digital tools and resources.
- ▶ **6** - Students demonstrate a sound understanding of technology concepts, systems, and operations.

Agenda

Getting Started (5 minutes)

Setting the Stage

Activity (35 minutes)

Objectives

Students will be able to:

- Define “coding” and “computer science”
- Identify key computer science vocabulary
- Identify places to go to continue learning computer science and coding
- Make connections between computer science concepts and the real world

Preparation

One Week Before Your Hour of Code

- Review the [Hour of Code Educator Guide](#) and [Best Practices from Successful Educators](#) in order to begin to plan your Hour of Code event.
- [Register your Hour of Code event](#) if you'd like to receive swag or classroom support.
- Review and complete the online tutorial yourself: [Make a Flappy Game](#)
- Be sure to test it first before asking your students to complete it. Check your technology and decide if you need to troubleshoot anything in advance of your Hour of Code.

One Day Before Your Hour of Code

- Write the words “coding”, “programming” and “debugging” on

Flappy Code

Wrap Up (5 minutes)

Debrief

Celebrate

Next Steps

Assessment (2 minutes)

Speak Up

Extended Learning

Beyond an Hour of Code

the board or add them to your word wall if you have one.

- Each student who completes the activity should receive a certificate.
- Print one for everyone in advance to make this easier at the end of your Hour of Code.

Vocabulary

- **Debugging** - Finding and fixing problems in an algorithm or program.
- **Program** - An algorithm that has been coded into something that can be run by a machine.
- **code** - (v) to write code, or to write instructions for a computer.

Teaching Guide

Getting Started (5 minutes)

Setting the Stage

Welcome students to class and very briefly introduce the day's activity.

Say: "Today we're going to spend one hour learning to code. Has anyone here heard the term "code" before? What does it mean?"

Students might mention that a "code" is like a secret message, or that it's related to computers in some way.

💡 Teaching Tip

One way to introduce the Hour of Code if you are not very familiar with coding yourself is to show one of our inspirational videos. Choose one you think your students will find inspiring, and share it now. For young learners, we suggest "Anybody Can Learn".

Explain that in computer science, "code" means a set of instructions that a computer can understand. Let students know that today, they are going to practice "coding," "programming" and "debugging".

Define:

- **Coding** means to write code, or to write instructions for a computer.
- **Programming**, similarly, means to write code or instructions. Today, you will program with blocks on the computer (if you're using an online tutorial) or with pen and paper (if you're using an unplugged activity).
- **Debugging** means to check code for mistakes and try to fix errors.

Ask students to think about what their ideal next step might be after high school graduation. Ask students to raise their hand if they believe that next step will require them to use a computer. Everyone should raise their hand, because computers are all around us...in our phones, in our cars, even in our microwaves and dishwashers!

Ask students to guess how many K12 schools there are in the United States of America. The answer is around 100,000. Now ask students to guess how many of those K12 schools offer a course in computer science. The answer is less than 1 in 4 schools. That means that less than 25% of the US population has the opportunity to learn computer science in school. Ask students: how does this impact a young person's ability to land their dream job or get into their ideal college, if they have little to no experience with computers? And this problem exists not just in the US, but all over the world.

Explain that across the globe, young people just like your students are participating in Hour of Code as a way to bring awareness to this injustice -- the lack of access to computer science education. Your class will be taking part, not just to learn the basics of CS and have a little fun -- but also to take a stand and use your voice to speak up about the importance of CS for all.

Activity (35 minutes)

Flappy Code

Challenge your students to complete the [Make a Flappy Game](#) tutorial.

Depending on the age and ability of your students, you might consider:

- For middle and high school students, we suggest you choose tutorials that allow free play and creation of apps or games. You can have students work independently or in pairs or small groups. There are many other tutorials on [code.org/learn](#) that allow for free creation, including [Playlab](#) and our new beta tool [AppLab](#).
- For younger students, we suggest you break your class into pairs or very small groups (three to four students each) and ask each group to work together to complete the tutorial using [pair programming](#).
- For adult learners, the [Write your first computer program](#) tutorial works extremely well either as an independent challenge or a pair programming activity.

💡 Teaching Tip

Be sure to play through your chosen tutorials yourself, before asking your students to attempt them. That way you'll know what to expect and can make decisions about whether to let students choose their own tutorial, or if you want to assign tutorials based on student needs.

 1-10

1 2 3 4 5 6 7 8 9 10

If a group or individual finishes early, they can attempt another tutorial by visiting [code.org/learn](#).

Wrap Up (5 minutes)

Debrief

Facilitate a "Turn and Talk." Ask students to share their game, app or final product with a friend for feedback. Then switch and check out your partner's app or game. Ask a few groups to share out their experience: what did you learn about how apps and games work today? How do you feel having had the opportunity to study computer science?

Celebrate

Explain that you are spending one hour coding today, because this week is CS Education Week, and millions of other students across the globe have also been learning one Hour of Code this week. Congratulate students on being part of this world wide movement.

Give each student a [certificate](#) with his or her name on it.

Next Steps

Let students know that if they enjoyed today's activity, they have many options for continuing to code. Encourage students to visit code.org/learn for a list of options, or, if you're planning any of the extension activities that follow, tell students what's coming next in your classroom.

Assessment (2 minutes)

Speak Up

Challenge students to share some aspect of their Hour of Code experience on social media using #hourofcode as a way to lend their voices to this world wide movement. Students can share their game, images, videos or just their thoughts.

Extended Learning

Beyond an Hour of Code

After your Hour of Code ends, there are many ways to continue teaching computer science in your K-5 classroom. Here are some ideas:

- Teach the [Code Studio Computer Science Fundamentals](#) courses. These six courses are designed for young learners. Students work their way through a series of puzzles that teach them to code, and educators have access to engaging lesson plans that help make the learning come alive. Code.org offers free professional development for these courses, online or in-person.
- Invite a computer science expert to talk to your class about his or her work. Don't know any volunteers, reach out to a few on the [volunteer map](#).



This work is available under a [Creative Commons License \(CC BY-NC-SA 4.0\)](#).

If you are interested in licensing Code.org materials for commercial purposes [contact us](#).