

Plugged

Lesson plans for online (plugged) Hour of Code activities.

Lesson 1: Write your first computer program

In this lesson, learners of all ages get an introductory experience with coding and computer science in a safe, supportive environment. This lesson has been designed for young learners, ages 4-10, but can be adapted for older learners using the differentiation suggestions provided.

Lesson 2: Code with Anna and Elsa

In this lesson, learners of all ages get an introductory experience with coding and computer science in a safe, supportive environment. This lesson has been designed for learners in the middle grades, ages 10-13, but can be adapted for younger or older learners using the differentiation suggestions provided. Students should have a basic understanding of simple geometry and drawing angles.

Lesson 3: Make a Flappy game

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.

Lesson 4: Playlab

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.

Lesson 5: Star Wars: Building a Galaxy with Code

In this lesson, learners of all ages get an introductory experience with coding and computer science in a safe, supportive environment. This lesson has two versions.

Lesson 6: MINECRAFT Hour of Code

In this lesson, learners of all ages get an introductory experience with coding and computer science in a safe, supportive environment. This lesson works well for any students old enough to read (ages 6+). Younger learners will probably not finish the tutorial, but will have lots of fun working through the puzzles for an hour. High school students will mostly finish the tutorial and have some time to play on the free play level at the end.

Lesson 7: Intro to App Lab

Tutorial Summary: This tutorial is designed to quickly introduce the App Lab programming environment as a powerful tool for building and sharing apps. The tutorial itself teaches students to create and control buttons, text, images, sounds, and screens in JavaScript using either blocks or text. At the end of the tutorial students are given time to either extend a project they started building into a "Choose Your Own Adventure", "Greeting Card", or "Personality Quiz" app. They can also continue on to build more projects featured on the code.org/applab page.

Lesson 8: Dance Party

In this lesson, learners of all ages get an introductory experience with coding and computer science in a safe, supportive environment. This lesson has been designed for learners of all ages but does require reading. This activity requires sound as the tool was built to respond to music.



This curriculum 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**.

Lesson 1: Write your first computer program

Overview

In this lesson, learners of all ages get an introductory experience with coding and computer science in a safe, supportive environment. This lesson has been designed for young learners, ages 4-10, but can be adapted for 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.

Agenda

Assessment (2 minutes)

Wrap Up (5 minutes)

Debrief
Celebrate
Next Steps

Extended Learning

Beyond an Hour of Code

Getting Started (5 minutes)

Setting the Stage

Activity (30-45 minutes)

Differentiation Suggestions

Adjustments for K-2 Teachers

View on Code Studio

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

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: **Write your first computer program**

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

Print one or more of the **Exit Ticket examples** at the end of this lesson plan, or create your own.

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

- **code** - (v) to write code, or to write instructions for a computer.
- **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.

Teaching Guide

Assessment (2 minutes)

Give each student an **“Exit Ticket”** and ask them to summarize what they learned today, how they felt, or what they experienced. Students can draw, write or express themselves in any way they feel comfortable. Collect an “Exit Ticket” from each student on their way out of the room. If you like, you can post the “Exit Tickets” on a bulletin board or some place prominent in the school as a reminder of your Hour of Code.

Wrap Up (5 minutes)

Debrief

Ask students to reflect on the day’s activities. What did they learn about coding? Programming? Debugging? How do they feel about computer science and code after spending one hour exploring?

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.

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 four 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 coming 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 local computer scientists? Try signing up for a virtual classroom visit through **Code.org’s and Skype’s Guest Speakers in Computer Science** program.

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.

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.

💡 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 “**The Hour of Code is Here** .”

Activity (30-45 minutes)

Challenge your students to complete the **Write your first computer program** tutorial.

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

- 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 learners in the middle grades, we find that working independently on tutorials works well. Sometimes it helps to allow students to choose their own tutorial. If students aren’t interested in Angry Birds, they can get a similar experience with the **Code with Anna and Elsa** or **Make a Flappy Game** tutorials.
- For older or adult learners, the **Write your first computer program** tutorial works extremely well either as an independent challenge or a pair programming activity.

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

💡 Teaching Tip

You’ll want to spend very little time front-loading or introducing your Hour of Code. With young learners, it’s best to jump in as quickly as possible. Too much explanation or lecture at the beginning tends to spoil the fun, and fun is the whole point!

Differentiation Suggestions

Adjustments for K-2 Teachers

- Have the first level of the activity already pulled up on students’ computer screens when they students start the activity.
- If possible, have students sit away from the computers while explaining the directions to the activity.
- Explain the 3 commands that the bird can do: move forward, turn right, and turn left.
- Do a live example of the first 3 levels. Try putting tape on the floor to make it look like the bird’s maze. Model the commands yourself and then pick 1 or 2 student volunteers to model for the class.
- Have students pair program by sitting 2 students at the same computer. Have the person controlling the mouse and keyboard first be a “1” and the other student who makes suggestions, points out errors, and asks questions be a “2.” Every 5 minutes, have the students switch roles.
- Practice clicking and dragging blocks before attempting to solve any of the puzzles. Also, practice dragging blocks to the trash can.

💡 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.

Standards Alignment

CSTA K-12 Computer Science Standards (2011)

- ▶ **CD** - Computers & Communication Devices
- ▶ **CI** - Community, Global, and Ethical Impacts
- ▶ **CPP** - Computing Practice & Programming

ISTE Standards for Students

- ▶ **1** - Creativity and Innovation
- ▶ **4** - Critical Thinking, Problem Solving, and Decision Making
- ▶ **6** - Technology Operations and Concepts

Common Core English Language Arts Standards

- ▶ **L** - Language
- ▶ **RI** - Reading Informational
- ▶ **SL** - Speaking & Listening

Common Core Math Standards

- ▶ **MP** - Math Practices
- ▶ **OA** - Operations And Algebraic Thinking



This curriculum 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**.

Lesson 2: Code with Anna and Elsa

Overview

In this lesson, learners of all ages get an introductory experience with coding and computer science in a safe, supportive environment. This lesson has been designed for learners in the middle grades, ages 10-13, but can be adapted for younger or older learners using the differentiation suggestions provided. Students should have a basic understanding of simple geometry and drawing angles.

Purpose

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

Agenda

Assessment (2 minutes)

Getting Started (5 minutes)

Setting the Stage

Activity (30-45 minutes)

Wrap Up (5 minutes)

Debrief

Celebrate

Next Steps

Extended Learning

Beyond an Hour of Code

[View on Code Studio](#)

Objectives

Students will be able to:

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

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: **Coding with Anna and Elsa**

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.

Review the supplementary lesson plan, **“The Drawing Machine”** to determine whether or not you'd like to lead this lesson in advance of your Hour of Code.

One Day Before Your Hour of Code

Write the words “coding”, “programming” and “debugging” on the board or add them to your word wall if you have one.

Write the words “Computer Science” in the middle of your board or on piece of paper at the center of a bulletin board. This will serve as your **“mind map”** for the Getting Ready and Assessment activities.

Have protractors available for students new to measuring angles

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

- **code** - (v) to write code, or to write instructions for a computer.
- **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.

Teaching Guide

Assessment (2 minutes)

Ask students to add their “Whip Around” sticky notes or note cards to your “Computer Science” mind map on their way out the door. Try to populate the board with lots of great ideas about what CS is and why it matters.

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.

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 name some jobs they have heard of that are related to coding. Students might mention things such as “programmer”, “computer scientist”, “software developer,” or “engineer”. Capture student responses on your “Computer Science” board, making a mind map of the information your students share.

Say: “You’re right, folks! There are no right or wrong answers here...just about any job these days involves some sort of knowledge of code. While there are many, many careers that require some knowledge of coding, learning to code is something anyone can do. And we’re going to do it today. The things we’re going to do today may not seem immediately like those, but everything you learn today could lead into making the next Angry Birds or Twitter.”

Activity (30-45 minutes)

Challenge your students to complete the **Code with Anna and Elsa** tutorial.

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

- 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**. Students in grades K-5 may struggle with the mathematics in the tutorial (it requires a basic understanding of angle measurement). If your students are on the younger side, consider delivering the supplementary lesson plan, **“The Drawing Machine”** to introduce the

💡 Teaching Tip

If you have time or if your students need a bit more background information, you might consider delivering **“The Drawing Machine”** before your Hour of Code event. This brief lesson introduces students to the idea of computer-assisted drawing, which will set them up for more success with the **Code with Anna and Elsa tutorial**.

💡 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 learners in the middle grades, we suggest **“The Hour of Code is Here.”**

basics before you facilitate your Hour of Code.

- For older students, we find that working independently on tutorials works well. Sometimes it helps to allow students to choose their own tutorial. If students aren't interested in Frozen, they can get a similar experience with the **Write Your First Computer Program** tutorial.
- For adult learners, **Code with Anna and Elsa** works extremely well either as an independent challenge or a pair programming activity.

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

💡 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.

Wrap Up (5 minutes)

Debrief

Give each student a few sticky notes or notecards.

Facilitate a quick **“Whip Around”** activity:

- Pose a prompt that has multiple answers such as “Share back something you really liked about the Hour of Code activity you completed” or “Share some skills you learned today.”
- Have students write down as many responses as possible, one idea per sticky note or note card. “Whip” around the room, calling on one student at a time. Have students share one of their responses. When called on, students should not repeat a response; they must add something new.
- After completing the whip around, have students discuss which ideas and themes showed up most in their responses.

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.

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 four 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 coming alive. Code.org offers free professional development for these courses, online or in-person.
- Research some of the careers in coding you identified today. Find resources on planning career research projects on [Sharemylesson.com](https://sharemylesson.com).
- **Invite a computer science expert to talk to your class about his or her work** . Don't know any local computer scientists? Try signing up for a virtual classroom visit through **Code.org's and Skype's Guest Speakers in Computer Science** program.

Standards Alignment

CSTA K-12 Computer Science Standards (2011)

- ▶ **CPP** - Computing Practice & Programming

ISTE Standards for Students

- ▶ **1** - Creativity and Innovation
- ▶ **4** - Critical Thinking, Problem Solving, and Decision Making
- ▶ **6** - Technology Operations and Concepts

Common Core English Language Arts Standards

- ▶ **L** - Language
- ▶ **RI** - Reading Informational

Common Core Math Standards

- ▶ **G** - Geometry
- ▶ **MD** - Measurement And Data
- ▶ **MP** - Math Practices
- ▶ **NBT** - Number And Operations In Base Ten



This curriculum 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**.

Lesson 3: Make a Flappy game

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.

Agenda

Assessment (2 minutes)

Speak Up

Getting Started (5 minutes)

Setting the Stage

Activity (30-45 minutes)

Wrap Up (5 minutes)

Debrief

Celebrate

Next Steps

Extended Learning

Beyond an Hour of Code

View on Code Studio

Objectives

Students will be able to:

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

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 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

- **code** - (v) to write code, or to write instructions for a computer.
- **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.

Teaching Guide

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.

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.

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.

💡 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.**"

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 (30-45 minutes)

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.

If a group or individual finishes early, they can attempt another tutorial by visiting **code.org/learn**.

💡 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.

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.

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 four 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 coming 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 local computer scientists? Try signing up for a virtual classroom visit through **Code.org's and Skype's Guest Speakers in Computer Science** program.

Standards Alignment

CSTA K-12 Computer Science Standards (2011)

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

ISTE Standards for Students

- ▶ **1** - Creativity and Innovation
- ▶ **4** - Critical Thinking, Problem Solving, and Decision Making
- ▶ **6** - Technology Operations and Concepts

Common Core English Language Arts Standards

- ▶ **SL** - Speaking & Listening

Common Core Math Standards

- ▶ **MP** - Math Practices



This curriculum 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**.

Lesson 4: Playlab

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.

Agenda

Wrap Up (5 minutes)

Debrief
Celebrate
Next Steps

Assessment (2 minutes)

Speak Up

Extended Learning

Beyond an Hour of Code

Activity (30-45 minutes)

Getting Started (5 minutes)

Setting the Stage

View on Code Studio

Objectives

Students will be able to:

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

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: **Playlab**

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 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

- **code** - (v) to write code, or to write instructions for a computer.
- **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.

Teaching Guide

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 four 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 coming 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 local computer scientists? Try signing up for a virtual classroom visit through **Code.org's and Skype's Guest Speakers in Computer Science** program.

Activity (30-45 minutes)

Challenge your students to complete the **Playlab** 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 the Make a Flappy Game] (<https://studio.code.org/s/playlab/stage/1/puzzle/1>) 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.

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

💡 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.

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.

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".

💡 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**."

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.

Standards Alignment

CSTA K-12 Computer Science Standards (2011)

- ▶ **CL** - Collaboration
- ▶ **CPP** - Computing Practice & Programming

- ▶ **CT** - Computational Thinking

ISTE Standards for Students

- ▶ **1** - Creativity and Innovation
- ▶ **2** - Communication and Collaboration
- ▶ **4** - Critical Thinking, Problem Solving, and Decision Making
- ▶ **6** - Technology Operations and Concepts

Common Core English Language Arts Standards

- ▶ **W** - Writing

Common Core Math Standards

- ▶ **MP** - Math Practices



This curriculum 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**.

Lesson 5: Star Wars: Building a Galaxy with Code

Overview

In this lesson, learners of all ages get an introductory experience with coding and computer science in a safe, supportive environment. This lesson has two versions.

Option 1: Blocks

The first option uses drag-drop blocks. This version works best for:

- Students on mobile devices without keyboards
- Younger students (6+ because the tutorial requires reading)
- International students

We recommend this for international students because JavaScript syntax is not translated and for the first Hour of Code, the translated blocks provide a better introduction.

Option 2: JavaScript

This option teaches the same basic concepts, but because it uses both drag-drop blocks and JavaScript, the students need to be able to type on a keyboard. For older students on computers, learning JavaScript can be fun and provide an additional challenge. This version of the tutorial is also great if you have some students in your class who have already learned some coding. It is recommended for ages 11+.

Purpose

This lesson introduces the core CS concepts of coding and programming including sequencing and events. The JavaScript version of the tutorial also lets the students experience typing code.

Agenda

Getting Started (5 minutes)

Setting the Stage

Activity (30-45 minutes)

Wrap Up (5 minutes)

Debrief

Celebrate

Next Steps

Extended Learning

Beyond an Hour of Code

Assessment (2 minutes)

View on Code Studio

Objectives

Students will be able to:

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

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.

Review and complete the online tutorial yourself: **Star Wars: Building a Galaxy with Code**

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

Write the words “Computer Science” in the middle of your board or on piece of paper at the center of a bulletin board. This will serve as your **“mind map”** for the Getting Ready and Assessment activities.

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

- **code** - (v) to write code, or to write

instructions for a computer.

- **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.

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.

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.

💡 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 learners in the middle grades, we suggest "**Change the World: Hour of Code 2015**."

Ask students to name some jobs they have heard of that are related to coding. Students might mention things such as "programmer", "computer scientist", "software developer," or "engineer". Capture student responses on your "Computer Science" board, making a mind map of the information your students share.

Say: "You're right, folks! There are no right or wrong answers here...just about any job these days involves some sort of knowledge of code. While there are many, many careers that require some knowledge of coding, learning to code is something anyone can do. And we're going to do it today. The things we're going to do today may not seem immediately like those, but everything you learn today could lead into making the next Angry Birds or Twitter."

Activity (30-45 minutes)

Challenge your students to complete the **Star Wars: Building a Galaxy with Code** tutorial.

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

- For younger students, we suggest you break your class into pairs and ask each pair to work together to complete the tutorial using **pair programming**.
- For older students, we find that working independently on tutorials works well. Sometimes it helps to allow students to choose their own tutorial. If students aren't interested in Star Wars, they can get a similar experience with the **Write Your First Computer Program** tutorial.
- For adult learners, **Star Wars: Building a Galaxy with Code** works extremely well either as an independent challenge or a pair programming activity.

At the end of the tutorial, the students have the opportunity to build a Star Wars game. If students finish early, encourage them to try each other's games and then make their own games more interesting or challenging.

You can also let a group or individual attempt another tutorial by visiting code.org/learn.

Wrap Up (5 minutes)

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.

Debrief

Give each student a few sticky notes or notecards. Facilitate a quick **“Whip Around”** activity:

- Pose a prompt that has multiple answers such as “Share back something you really liked about the Hour of Code activity you completed” or “Share some skills you learned today.”
- Have students write down as many responses as possible, one idea per sticky note or note card. “Whip” around the room, calling on one student at a time. Have students share one of their responses. When called on, students should not repeat a response; they must add something new.
- After completing the whip around, have students discuss which ideas and themes showed up most in their responses.

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.

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 four 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 coming alive. Code.org offers free professional development for these courses, online or in-person.
- Research some of the careers in coding you identified today. Find resources on planning career research projects on [Sharemylesson.com](https://www.sharemylesson.com).
- **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](#).

Assessment (2 minutes)

Ask students to add their “Whip Around” sticky notes or note cards to your "Computer Science" mind map on their way out the door. Try to populate the board with lots of great ideas about what CS is and why it matters.



This curriculum 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**.

Lesson 6: MINECRAFT Hour of Code

Overview

In this lesson, learners of all ages get an introductory experience with coding and computer science in a safe, supportive environment. This lesson works well for any students old enough to read (ages 6+). Younger learners will probably not finish the tutorial, but will have lots of fun working through the puzzles for an hour. High school students will mostly finish the tutorial and have some time to play on the free play level at the end.

Purpose

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

Agenda

Activity (30-45 minutes)

Getting Started (5 minutes)

Setting the Stage

Wrap Up (5 minutes)

Debrief

Celebrate

Next Steps

Assessment (2 minutes)

Extended Learning

Beyond an Hour of Code

View on Code Studio

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

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: **MINECRAFT Hour of Code**

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

Print one or more of the **Exit Ticket examples** at the end of this lesson plan, or create your own.

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

- **code** - (v) to write code, or to write instructions for a computer.
- **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.

Teaching Guide

Activity (30-45 minutes)

Challenge your students to complete the **MINECRAFT** tutorial.

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

- For younger students, we suggest you break your class into pairs and ask each group to work together to complete the tutorial using **pair programming**.
- For learners in the middle grades, we find that working independently on tutorials works well. Sometimes it helps to allow students to choose their own tutorial. If students aren't interested in MINECRAFT, they can get a similar experience with the **Code with Anna and Elsa** or **Star Wars** tutorials.
- For older or adult learners, the **MINECRAFT** tutorial works extremely well either as an independent challenge or a pair programming activity.

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

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.

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.

Wrap Up (5 minutes)

Debrief

Ask students to reflect on the day's activities. What did they learn about coding? Programming? Debugging? How do they feel about computer science and code after spending one hour exploring?

Celebrate

💡 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.

💡 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 "**The Hour of Code is Here**."

💡 Teaching Tip

You'll want to spend very little time front-loading or introducing your Hour of Code. Especially with young learners, it's best to jump in as quickly as possible. Too much explanation or lecture at the beginning tends to spoil the fun, and fun is the whole point!

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)

Give each student an **"Exit Ticket"** and ask them to summarize what they learned today, how they felt, or what they experienced. Students can draw, write or express themselves in any way they feel comfortable. Collect an "Exit Ticket" from each student on their way out of the room. If you like, you can post the "Exit Tickets" on a bulletin board or some place prominent in the school as a reminder of your Hour of Code.

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 four 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 coming 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 local computer scientists? Reach out to a volunteer on our **volunteer map**



This curriculum 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**.

Lesson 7: Intro to App Lab

Overview

Tutorial Summary: This tutorial is designed to quickly introduce the App Lab programming environment as a powerful tool for building and sharing apps. The tutorial itself teaches students to create and control buttons, text, images, sounds, and screens in JavaScript using either blocks or text. At the end of the tutorial students are given time to either extend a project they started building into a "Choose Your Own Adventure", "Greeting Card", or "Personality Quiz" app. They can also continue on to build more projects featured on the code.org/applab page.

Age Appropriateness: The tutorial is designed for students over 13. Because it allows students to upload custom sounds and images, young students should not use this without supervision. To protect students privacy, if your students are under 13, they will not be able to use this tutorial unless you first set up accounts for them in a section you manage.

Checking Correctness: This tutorial will not tell students whether they completed the level correctly. Encourage students to use the target images and directions provided in every level to know if they are on the right track. If students want to move on past a particularly tricky level they can simply click "Finish" and continue on.

Have fun completing your Hour of Code with App Lab!

Purpose

This lesson is your students' first step with App Lab. The tutorial itself should be an approachable introduction to a powerful tool with broad functionality. Students should have a positive experience during the tutorial and more importantly should be motivated to keep building in App Lab. They are only scratching the surface of what the tool is capable of here, so afterwards they should use the code.org/applab page or the project gallery to get inspired for more complex types of projects they want to make.

Agenda

Warm Up

Build Excitement!

Activity

General Support

Setting Properties - Levels 1-4

Make It Interactive - Levels 5-7

Images and Sounds - Levels 8-9

Design Mode - Levels 10-13

[View on Code Studio](#)

Objectives

Students will be able to:

- Build and share their own apps in App Lab using features like buttons, text, images, sound, and screens.
- Continue building projects in full App Lab using the resources provided on the code.org/applab page.

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

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

Print one or more of the **Exit Ticket examples** at the end of this lesson plan, or create your own.

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.

Share Your App - Levels 14-15

Wrap Up

Celebrate and Keep Going!



Teaching Guide

Warm Up

Build Excitement!

Motivate: Explain to students the goals of today's activity. They are going to start using a new tool that will let them quickly make apps they can instantly send to themselves or friends to use.

Video: The first level of this activity is a video that should both motivate students to complete the activity and explain how it will work. Consider watching it as a class and quickly debriefing afterwards to answer questions.

Activity

General Support

General Support: As a teacher your role is primarily to support students as they make their way through the tutorial. Here are a few tips that should help students regardless of the level they're working on

- **Collaborate with Neighbors:** Encourage students to check in with a neighbor when they're getting stuck. Since this tutorial includes videos and students may be wearing headphones it can get easy to "go into a bubble". Help break those barriers by actively pairing students if they seem like they need help.
- **Read the Instructions:** The instructions usually provide helpful tips on how to complete the level. Before completing a level for a student ensure they've actually looked closely at the target image and read all the text there.
- **Stuck? Click here:** Each level includes a GIF showing exactly how to complete the level. If students want help they can and should use these GIFs. If they only want to use them as a hint, just have them close the GIF once they've seen the first part.
- **Move On and Come Back:** Sometimes students will benefit more from coming back to a tricky level. Except for the "Design Mode" sequence, all levels are independent of one another, though they do rely to some degree on previously introduced concepts.

Setting Properties - Levels 1-4

These levels are all about learning to use the `setProperty()` block. The dropdowns in this block make it easy to know exactly what it is capable of changing. That said, there's a couple tips that can help

- **When to use Double Quotes:** The `setProperty()` block automatically changes the final dropdown when you select what properties you want to change, including whether they use double quotes. When in doubt, students should first change the first two dropdowns, then use the last one as a model for what values work there. The most common error is failing to use double quotes around a color name.
- **Hover to Read IDs:** By hovering over an element in your app you can read its ID. This will help students when they're trying to change multiple elements on their screen.

Make It Interactive - Levels 5-7

This sequence introduces the `onEvent()` block. Here's some helpful tips if students are getting stuck.

- **onEvents Don't Go Inside One Another:** Students just starting out may try to put one block inside of the others. This is never the intended behavior for this tutorial. Even though this is mentioned in the videos, a quick reminder might help get kids unstuck.
- **Check Your IDs:** You need to change the "id" property in `onEvent()` so that it detects events with the correct element.
- **You Can Use Multiple Blocks in an `onEvent()`:** If you want multiple things to happen when you click the same button, just add more blocks to the same `onEvent()`. You should never have a program that has two `onEvent()` blocks for the combination of element (e.g. "button1") and event type (e.g. "click").

📍 Images and Sounds - Levels 8-9

This section has a single level that has students add an animal to a soundboard. This level is a little more involved than the previous ones, so expect that students may need to either rewatch the video beforehand or read instructions carefully to complete all the steps.

💡 Teaching Tip

Finding Images: Students do not need to search for images or sounds online in order to complete this lesson, but they may wish to. You'll likely want to prepare in advance to advise responsible and appropriate searching for images.

- **Images Use** `setProperty()`: To add an image to a screen element students can use the "image" property. There is no new block.
- **Link to Images:** Students can copy the URL of images they find directly into the `setProperty()` block in order to add them to their apps. There's no need to download them to their computers and upload them to App Lab if they don't wish to.

Design Mode - Levels 10-13

In the last sequence students are working on the same project for three bubbles in a row. They are now learning to add screen elements themselves which means that the total number of things they can do in App Lab has grown a lot. Assume that some students will spend some time exploring at this point as they try out all the new tools.

- **Using Good IDs:** An important part of programming in App Lab is giving your elements good IDs. Up to this section students have had their IDs created for them, so they haven't had a chance to practice this skill. This is a useful reminder for the teacher to reinforce during this section.

Share Your App - Levels 14-15

This section is very open-ended. The tutorial itself is designed to give students ample time to keep working on this project, either making Choose Your Own Adventure, or one of their own creation.

- **Try the Samples:** Students are provided 3 sample apps that should help them brainstorm their own ideas.
- **Encourage Sharing:** If students have cell phones with a data plan they can quickly text a link to their projects to their own phone or a friend's. If your school policy allows it, encourage them to do so here.
- **Encourage Creativity:** Compared to other activities in this lesson, this section asks students to be much more creative. Ask students to think "what will your story be about?" or do a quick group brainstorm so that classmates can hear ideas from one another.

Wrap Up

Celebrate and Keep Going!

Print Certificates: Print certificates for students to celebrate their achievement.

code.org/applab: At the end of the lesson encourage students to head to the App Lab home page where they can get ideas for more ways to keep their adventure with App Lab going.



This curriculum 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**.

Lesson 8: Dance Party

Overview

In this lesson, learners of all ages get an introductory experience with coding and computer science in a safe, supportive environment. This lesson has been designed for learners of all ages but does require reading. This activity requires sound as the tool was built to respond to music.

Purpose

This lesson introduces the core CS concepts of coding and event programming (using blocks).

Agenda

Getting Started (5 minutes)

Setting the Stage

Activity (30-45 minutes)

Music Filtering General Support Level by Level

Wrap Up (5 minutes)

Debrief Celebrate Next Steps

Go Viral

Assessment (2 minutes)

View on Code Studio

Objectives

Students will be able to:

- Develop programs that respond to timed events
- Develop programs that respond to user input
- Create dance animations with code

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 get classroom support, discounted posters, or participate in the robotics give-away (US only).

Review and complete the online tutorial yourself: **Dance Party**

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.

Review the supplementary lesson plan, **"The Big Dance Party"** to determine whether or not you'd like to lead this lesson in advance of your Hour of Code.

One Day Before Your Hour of Code

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.

(optional) Encourage students to bring in headphones to use during the activity.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Spotify Playlist (all ages)**

Vocabulary

- **code** - (v) to write code, or to write instructions for a computer.
- **Event** - An action that causes something to happen.
- **Program** - An algorithm that has been coded into something that can be run by a machine.

Teaching Guide

Getting Started (5 minutes)

Setting the Stage

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

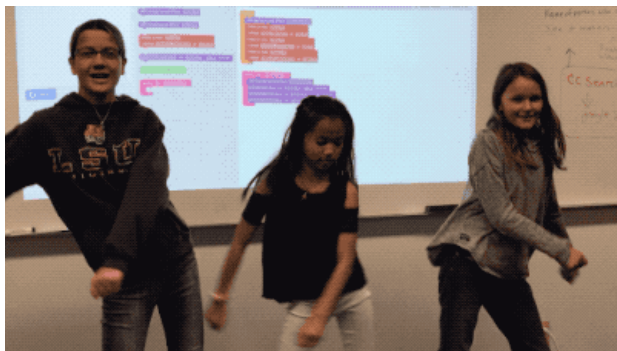
Remarks

Today we're going to do something really creative. What's your favorite way to be creative?

Encourage students to share the ways they express creativity, such as with art, dance, music, writing.

Explain that today we're going to be creative with "code", which just means means a set of instructions that a computer can understand. Just like choosing which type of colors of paint to use, or what kinds of words you write with can be express creativity, choosing what code you write and how people interact with it can be an opportunity to express your creativity too!

Get up and dance: Announce to the class that today we're going to see how we can combine coding with dancing in a creative way. Ask your kids to floss, dab, or do a creative dance move of their own for 10 seconds to get them in the mood. You can play a song from this **Spotify Playlist (all ages)** to help kick things off. Capture your class's moves on video.



Teaching Tip

If you have time and would like to prepare your students with an unplugged activity, consider delivering "**Dance Party: Unplugged**" before your Hour of Code event. This brief lesson introduces students to the idea of events triggering different dance moves.

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 learners in the middle grades, we suggest **The Hour of Code is Here**.

Activity (30-45 minutes)

Music Filtering


This tutorial features songs from popular artists. To get a preview of the song list in this tutorial, check out this **Spotify Playlist**. We are using radio-safe versions of all songs and for students under 13, we limit the music to this filtered list **Spotify Playlist (all ages)**. If you would like to use the filtered list with older students, you can share **this link** with your classroom.

General Support

General Support: As a teacher your role is primarily to support students as they make their way through the tutorial. Here are a few tips that should help students regardless of the level they're working on

- **Collaborate with Neighbors:** Encourage students to check in with a neighbor when they're getting stuck. Since this tutorial includes videos and students may be wearing headphones it can get easy to "go into a bubble". Help break those

barriers by actively pairing students if they seem like they need help. Depending on the age and ability of your students, you might consider formally breaking your class into pairs to complete the tutorial using **pair programming**.

- **Read the Instructions:** The instructions usually provide helpful tips on how to complete the level. Before completing a level for a student ensure they've actually looked closely at the target image and read all the text there.
-  **Stuck? Look for hints:** Each level includes a hint with further details about how to pass the level. If students are trying to pass puzzles as quickly as possible, they might find they just need to wait a certain amount of time for all of the code to play out before moving on. Many levels contain a second hint with suggestions of ways to get even more creative by extending the task.
- **Move On and Come Back:** Sometimes students will benefit more from coming back to a tricky level. All levels are independent of one another, though they do rely to some degree on previously introduced concepts.
- **Take your time and experiment.** Even though the first couple levels tell students exactly what they need to do to pass, they should also feel free to try out their own ideas.

Level by Level

Level 1

- Drag the red `make a new` block from the toolbox on the left to the workspace on the right. Connect it inside the `setup` block.
- You have now written your first program. Make sure to press Run to see what happens. You should hear music and see a character start to move in the display area.

Level 2

- The green blocks are event blocks. These blocks start a new sequence of code and do not need to be connected inside the `setup` block.
- Connecting the purple block under the green event block should make the character perform a dance move after the number of measures you indicate.

Level 3

- Make sure to bring out a second green event block. You should have a `after 4 measures` block and a `after 6 measures` block in your workspace. Both should have purple block connected underneath.

Level 4

- Drag out the `set background effect` block.
- This block can be connected inside `setup`, under an event block, or in both places!

Level 5

- With the right code, you should see the dancer cycle through a different move every 2 measures.
- Make sure there is a purple `do forever` block connected inside the `every 2 measures` block.
- Make sure the `do forever` block is set to either (Next), (Previous), or (Random). Otherwise, the dancer will just perform the set move repeatedly.

Level 6

- Be careful to avoid putting the red `make a new` block inside the `every 2 measures` block. This will cause your program to create multiple identical dancers at the same location.

Level 7

- Many students will be familiar with the idea that you can make something seem to be further away by drawing it a smaller scale. In this level, you'll create this effect by making the two dancers on the ends smaller.
- It is important to make sure that the teal `set` block is placed somewhere in the program **after** the dancer has been created. To solve this puzzle, place a `set backup_dancer2 size to 50` block somewhere under the `make a new robot called backup_dancer2` block.

Level 8

- As with the previous level, make sure to only use the `set tint to (color)` block after you have made the dancer in your program. For example, placing it as the first step in the `setup` area of your program will have no effect.

Level 9

- Computer science allows to process input information into interesting types of output. In this level, students can explore how a dancer's properties can be updated automatically based on the actual sound of the music.

🔗 Level 10-11

- These levels are about making the dance interactive. Try out the new `when pressed` event to make the dancers respond to key presses.
- You can use the arrow keys on your computer keyboard or click the orange buttons under the display area.
- Note that the **code students write in these levels is not checked for correctness**. This means they will always pass the level, even if they do not change the program. Encourage students to determine for themselves whether the code is doing what they expect before moving on.

💡 Teaching Tip

By this point in the lesson you may notice that the instructions are less prescriptive. Encourage students to be creative and explore the new blocks that are introduced. From this point on student code is **not checked for correctness** in order to encourage experimentation instead of solving a specific task.

Level 12

- The top six blocks in this tool box look familiar but all work in a different way. Instead of creating or controlling one dancer, they work with groups of dancers.
- Like the previous levels, **code is not checked for correctness here**. Students should feel free to experiment with groups in ways that are interesting to them.

Level 13

- This last level is very open-ended. The tutorial itself is designed to give students ample time to keep working on their own dance.
- **Encourage Sharing:** If students have cell phones with a data plan they can quickly text a link to their projects to their own phone or a friend's. If your school policy allows it, encourage them to do so here.
- **Encourage Creativity:** Creativity is important throughout this lesson, but this is true here more than anywhere else!

Wrap Up (5 minutes)

Debrief

- Pose a prompt that has multiple answers such as "What is something you enjoyed about today's activity?" or "What is the connection between creativity and 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.

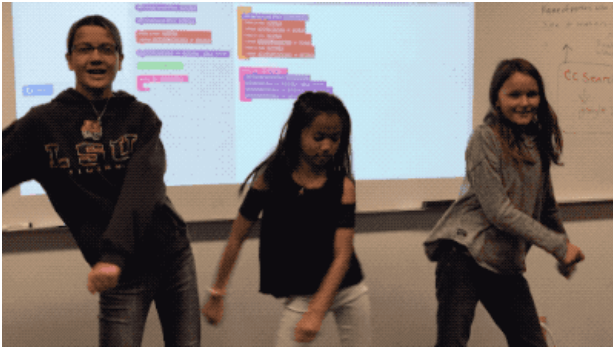
Next Steps

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.

Go Viral

The Hour of Code is about creativity, and we can't wait to see what you create! Please share student creations, photos, and videos on social media! Teachers, record your classroom coding a dance, or dancing the dance. Make your video special by adding an **ending clip**.

Be sure to include #HourOfCode and tag us on Facebook, Twitter and Instagram. Bonus points for tagging the artist whose music your students used. Code.org will re-share our favorite posts to our millions of followers.



Of course, make sure to respect your school's social media policy

Assessment (2 minutes)

Ask students to add their "Whip Around" sticky notes or note cards to your "Computer Science" mind map on their way out the door. Try to populate the board with lots of great ideas about what CS is and why it matters.



This curriculum 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**.