EasyCode Sample Lessons: Grades 3-8

Let's Get Started: Challenges 0 – 5

This lesson introduces students to the fascinating world of computers and to the EasyCode platform. Some of your students may be familiar with the terms "coding" and "programming" and feel comfortable working with computers. For others, learning to code may be an intimidating experience. As educators, our goal is to help students learn and explore a variety of subjects, including computer science. We want to create an environment where students can learn from their mistakes and build their foundational knowledge to create something new. (Estimated time: 45 minutes)

Primary Objectives

- Students can define coding and computer programming.
- Students become familiar with the EasyCode platform.
- Students complete challenges 0 - 5 on the EasyCode platform.

Warm Up (8 Minutes)

Discussion - 2 minutes

Discuss the following with your students:

- How many of you have ever used a computer?
- Have you ever created something on a computer, like a presentation, a drawing, or maybe even a game?
- Let two or three students tell the class what they created.

Explain - 1 minute

In order for all of our favorite applications and games to work on a computer, we have to give instructions to the computer. Computers can't think for themselves, they do whatever we tell them to do. Giving instructions to the computer is called computer programming or coding.

Warm-up Activity - 3 minutes

Play a short game with your students to illustrate instructions. Place an object somewhere visible in the classroom. Ask your students to give you instructions to guide you from where you are standing in class to the object.

What instructions did the students use, e.g., step, turn right, turn left?

Discussion - 2 minutes

Do computers speak the same language as humans?

Computers have their own languages; they can't understand human language as we understand it. HTML, JavaScript, and Python are just a few of the languages computers speak. Each language is different, but they all have something in common: they require a certain way of thinking, clear instructions, and structure. Basically, learning a coding language is just like learning a new language.

Activity (25 minutes)

Accessing EasyCode for the first time - 3 minutes

Today you will start learning basic coding principles through a game called EasyCode. The language we will learn is called CoffeeScript.

Help the students get in to the assignment:

1. Log in to Learning.com.
2. Go to the Let's Get Started assignment.
3. Click the EasyCode button.

Walk-through - 12 minutes

Click the EasyCode button to go to your EasyCode account.

Walk your students through the basic appearance of EasyCode:

Game Overview

1. Click on the "play now" button on the homepage.
2. Watch the short introduction animation.
3. Read the instructions out loud.
4. EasyCode is built of levels called challenges. This is what a challenge looks like.
5. The editor on the right is where you'll write your code. You can also use the buttons at the bottom for easy access.
6. On the left is the stage, this is where you'll see your code come to life. Your goal is to complete every challenge by helping the monkey catch the banana.
7. The monkey on the lower left corner is Gordo, he will give you instructions and sometimes even hints if you're stuck. At any time, you can click on Gordo to see the instructions again.

Challenges and Stars

1. In every challenge, you will execute the code by clicking on the "run" button to see what the starting code will do.
2. The code on the right says step 15, so when we'll click on "run" the monkey will step 15 steps forward.
3. Click on run.
4. We completed the first challenge. After every completed challenge, you'll get a star score rating your solution. 3 stars is the highest score and is rewarded for catching all the bananas, implementing new learned topics, and writing short code. If you get less than 3 stars, a hint will help you get them all. You can try to solve a challenge as many times as you want, it will not affect your stars score!
5. Click on replay to see your solution again.
6. Edit the solution to change it from step 15 to step 5 step 10.
7. Click run again to execute your solution again. Show the students that this solution only got 2 stars, and draw their attention to the hint that tells them how to get the 3rd star.
8. Click replay again, fix the solution to get 3 stars, and execute it again by clicking run.

Tips for the Challenges

1. Let's move on to the next challenge, click on next challenge.
2. Read the instructions out loud.
3. The code on the right says: step 10. Let's click on run and see what happens.
4. The monkey didn't walk far enough, and the hint told us to try: step 15. Let's change the number 10 to 15, and click run again.
5. It's always a good idea to use the code that was there. Before we try to change the code, click run to see what happens, read the hint, and then try to solve. It doesn't matter that you ran code that doesn't solve the challenge. Besides, the code is there as a starting point, so don't delete it.
6. Another good strategy for when you're stuck is to start again from the beginning. You can reset your code by clicking on the reset button.
7. Click the replay button, and then click the reset button to show your students how to reset the code to what it was in the beginning.
8. Solve the challenge again by editing the code and clicking run to execute the solution.
9. Show your users how to go back to challenge 0 by clicking on the map (top right corner) and clicking on challenge 0. Note that unlike you as a teacher, your students will not be able to skip forward beyond the first challenge that they have not solved yet.
10. Open challenge #2 and show the ruler animation. Follow the instructions to measure the distance between the monkey and the banana, and then use that distance to fix the code. Make sure your students understand how to use the ruler.

Play Time - 10 minutes

All students should complete challenges 0 to 5 with at least two stars. Use the teacher dashboard to keep track of students' achievements. To get to a group dashboard, click the 3-bar menu icon to the left of the map in the upper right corner of the screen and choose Groups. Then, click on the group name in the list of groups to see the group dashboard.

If students are having trouble confusing right and left, draw their attention to the watch on the monkey's left wrist. Tell them that turning in that direction is left.

Wrap-up (8 minutes)

Discussion - 5 minutes

Ask the students the following questions:

- What instructions did you learn today?
- What did you like most about EasyCode?
- Besides instructions, what else did you learn today?
- How do you get 3 stars in a EasyCode challenge? Does it matter how many times I tried to solve the challenge? Explain that in programming, the best solutions often start as less than the best solutions that the programmer continues to try to improve.
- What do you do when you are stuck? (the following two questions are related)
  - In a challenge, how do you display the instructions again?
  - In a challenge, how do you reset the code to what it was in the beginning?

Review - 3 minutes

Open challenge 6 and have the class solve it with you as a group. They will solve it by themselves in the next lesson.

Extension Assignment (20 minutes)

Assignment

Have the students choose another location at their school such as the cafeteria, gym, library, computer lab, office, etc. Tell the students they will create instructions using the code they learned today to tell another student how to get from the classroom to another location in the school. They can only use the code they used today in the challenges. Then, they will give the instructions to a fellow student to test their code.

Show an example of such a sequence on the whiteboard as a model.

Part 1

1. Click the button below to goto to EasyCode and complete challenges 0 to 5.
2. Try to get 3 stars on challenges 0 to 5 if you have not already.

Part 2

1. Choose somewhere other than your classroom in your school (gym, library, cafeteria, etc.).
2. In the space below and using only code you used today, create directions from your classroom to the other place.
   - Start by leaving blanks for the number of steps.
   - When instructed by your teacher, figure out exactly how many steps you need and add them to your directions.
   - Save your directions.
3. Turn in your assignment.

EasyCode Sample Lessons: Grades 3-8

Turn Around: Challenges 6 - 10

In this lesson, students will continue to explore the EasyCode platform by completing five more challenges. Prior to class, use the teacher dashboard to make sure all of your students have completed the first five challenges with three stars. *(Estimated Time: 45 minutes)*

Primary Objectives

- Students will identify the different ways to use “turn” instructions.
- Students will complete challenges 6-10 on EasyCode.
- Students will learn to use “turn” using degrees.
- Students will learn use “step” for backwards movement.
- Students will learn what a program, function, argument, statement and object are and how to they relate to coding.

Warm-up (29 minutes)

Review - 5 minutes

Start with a brief discussion with the class of what was learned in the previous lesson:

- What is coding?
- What instructions have we used so far? (step, turn)
- What is a programming language and which one do we use in EasyCode? (CoffeeScript)

Activity - 8 minutes

Ask for three volunteers, giving each of them a role: one is the *Programmer*, one is the *Computer*, and the third is the *Character*. Now ask the *Programmer* to instruct the *Computer* to lead the *Character* to an object you placed in the classroom. Make sure the students use instructions properly (step with a number, turn left or turn right). As they go, write the instructions on the board to remind the other students of what they have learned.

Repeat this activity with another group of three volunteers.

Ask the students, “Why do you need both a computer and a character? Why can’t one person be both?”

If we compare programming to the human body, then the programmer is the brain that sends instructions to the different parts of the body. The computer is responsible for making sure that the different parts of the body (the *Characters*) execute the instructions exactly as instructed.

Discussion: Statements, Objects, Functions and Arguments - 7 minutes

Introduce your students to the term *statement*: an element which expresses some action to be carried out. A computer program is a set of instructions which are simple tasks provided to the computer. These instructions are

called statements. The instructions the *Programmer* gave earlier to the *Computer* are statements. Statements can be anything from a simple line of code to a complex set of conditions and formulas.

- Ask the class to give you an example for a statement and write it on the whiteboard. Some examples might be: step 10, step 15, turn right, turn left

Objects are everything in the scene we can interact with, like the bush, bridge, banana, and turtle.

Each object has a set of actions it can do, like step or turn for the monkey. These actions are called *functions*, and the input we add to them is called an *argument*. For example in turn 10, the argument is 10.

- Ask what is the function in this statement (step or turn)
- Ask what is the argument (10, 15, right or left)

Discussion: Turning with Degrees - 5 minutes

This lesson is about turning and walking backwards. There are three ways to make a character turn. The first is to use turn right or turn left like we learned in the first lesson. In this lesson, we are introducing another way to turn.

Instead of turning right or left, we can turn by degrees. If your students have basic knowledge of degrees, such as a 360 degree turn or a 90 degree turn, then make a quick review of that knowledge. Otherwise, provide a short introduction to degrees. Optionally use a protractor.

Check your students' understanding of turning with degrees: Ask all of them to stand up and instruct them to turn 90, turn 120, and turn 360.

Repeat the explanation of turning by degrees: turn followed by a number turns the monkey by that number of degrees. For example, turn 90 turns the monkey the same as turn left.

Discussion: Walking Backwards - 4 minutes

Understanding the concept of walking backwards is pretty easy. If we want to go forward 15 steps, we type step 15, and if we want to go backwards, we type step -15. -15 will be read by the computer in this context as 15 steps backwards. If appropriate for level of your students, this may be a good opportunity to talk about negative numbers on the number line.

Check your students' understanding of walking backwards: Stand with your back to the door and ask, "If I were the monkey what would be the right instruction to get me to the door?". Emphasize that it should be one instruction, and not involve turning. Make sure their answer includes step -(some value).

Repeat the explanation of stepping backwards: to step backwards a number of steps, add the minus sign (-) before the number. (*For example: step -10.*) The computer reads -10 in this context just like "10 steps backwards".

Activity (16 minutes)

Walk-through - 5 minutes

Click the EasyCode button to go to your EasyCode account.

Open challenge #7 and show the animation about angles. Use the ruler to measure the distance between the monkey and the banana, and show that ruler is also a protractor: it shows the number 45 which is the angle the monkey has to turn in order to face the banana.

Show that this is the same number in the code. Make sure your students understand how to use the ruler as a protractor.

In level 8, the students can use either turn left or turn 90 to get three stars. Some of your students will probably use turn left. Make sure to emphasize that they can also use turn 90 for the same result.

Open the challenge map and show your students the skill mode tab. Explain that in skill mode students can play through more challenges to perfect their coding skills.

These extra challenges are great practice and they only unlock after we complete certain challenges. Hover over a locked challenge to show the unlocking tip. The first skill challenges will open for your students after they complete challenge 6.

Let students know that if they finish early, they can go to skill mode and complete unlocked challenges.

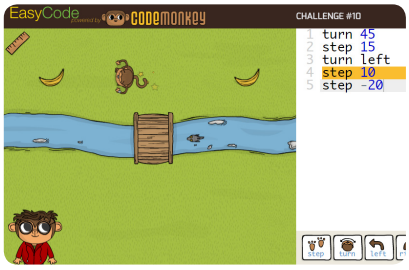
Access EasyCode - 1 minute

Ask the students to:

1. Log in to Learning.com.
2. Go to the Turn Around assignment.
3. Click the EasyCode button.

If a student is having trouble remembering his or her login information, use your **Learning.com** roster or provide students with log in cards.

Play time - 10 minutes



All students should complete challenges 6-10 with at least two stars. Use the teacher dashboard to keep track of students' achievements. Keep in mind that students might find turning with degrees difficult. You may need to provide extra help in levels 7 and 8. Challenge #10 is an assessment challenge that covers everything your students recently learned in EasyCode.

Assignment (5 minutes)

Ask your students to pick a new location at the school and write new instructions for going from the classroom to that location without using degrees instead of right or left.

Part 1

1. Click the button below to goto to EasyCode and complete challenges 6 to 10.
2. Try to get 3 stars on challenges 6 to 10 if you have not already.
3. If you have extra time, complete as many of the skill challenges as you can from the Skill Mode tab, trying to get 3 stars on each.

Part 2

1. Choose another location other than your classroom in your school (gym, library, cafeteria, etc.)
2. In the space below and using only code you used today, create directions from your classroom to the other place using degrees instead of right or left.
    - Start by leaving blanks for the number of steps.
    - When instructed by your teacher, figure out exactly how many steps you need and add them to your directions.
    - Save your directions.
3. Turn in your assignment.

EasyCode Sample Lessons: Grades 3-8

I Have a Plan: Challenges 11 - 15

This lesson revolves around planning. Everything we do in the physical world has to be planned, even if we sometimes do things automatically. We can cross the road without checking if it's clear, but that may result in a very dangerous outcome. Computers are the same. If we want to create a game or a program, we have to plan ahead and organize our instructions in the correct order.

(Estimated Time: 45 minutes)

Primary Objectives

- Students will review what they learned in the previous lesson.
- Students will discuss the concept of planning and its importance in coding.
- Students will understand the importance of accuracy when providing instructions.

Warm Up (20 minutes)

Discussion - 2 minutes

From the previous lesson's objectives, review the first two ways to turn. Ask for two volunteers and instruct each of them to explain and demonstrate one of the ways to turn that were learned in the previous lesson:

- Direction (e.g. turn right, turn left)
- Degree (e.g. turn 45, turn 30, turn 180)

Write the responses, with the examples, on the board as a reference for all students.

Explain - 3 minutes

Introduce the third way to turn: by using turnTo. When using turnTo the computer identifies that there is another object present, besides our beloved monkey. When its name is called, it knows which way to turn.

Warm-up Activity 1 (Whole Group) - 5 minutes

To check for understanding of turn to, play a short game similar to Simon Says. Give instructions to your students to "turn to" a specific place or a specific student. They should only turn when you say "turn to" and not when you say "turn".

Warm-up Activity 2 (Small Group) - 5 minutes

In this lesson, students will learn about the importance of planning. Have students work in small groups to answer the prompt, "What do you do in the morning to get ready for school?"

Have students use a bubble map to brainstorm answers to the question. The main center bubble would contain the question. Once they have brainstormed, have students work together to create an ordered list of what they do in the morning to get ready for school.

Discussion - 5 minutes

Ask students to think about why they placed the items in that order and as a whole discuss the reasons.

It is necessary for students understand the importance of planning. We plan our day and the order in which we do things. Sometimes, we do this without thinking and sometimes we plan every step.

Explain to your students that when we write code, we have to consider that computers read the code from TOP to BOTTOM, and we have to think ahead about the order of instructions. When we have just one object, this isn't a big problem (in our case, the monkey is the object). But what happens when we want to control another object? How do we know who should be instructed to go first?

In this lesson's challenges, your students will meet our trusty turtle and will have to use his help to get more bananas. In order to do so, they will have to think ahead and plan how to write the code.

Activity (15 minutes)

Access EasyCode - 1 minute

Ask the students to:

1. Log in to Learning.com.
2. Go to the I Have a Plan assignment.
3. Click the EasyCode button.

If a student is having trouble remembering his or her login information, use your **Learning.com** roster or provide students with log in cards.

Walk-through - 6 minutes

Click the EasyCode button to go to your EasyCode account.

Open challenge #12 and show the animation. It explains how to use objects on the screen. After the animation, walk your students through the following steps:

1. Hover over the bridge and show that the word bridge appears on the screen.
2. The name of that object is bridge.
3. Highlight the word banana in the editor.
4. Click on the bridge and show how the word banana is replaced by the word bridge.
5. Move the cursor by clicking on row 3 after the word turnTo.
6. Click the banana and show how the word banana is entered into the code.
7. Move the cursor to line 4 and write step 10.
8. Run the solution.
9. Click replay to go back to your solution.
10. Delete all the code to start from blank.
11. Now you will demonstrate how to use even more clicking instead of typing.
12. Hover over the block *step* at the bottom of the editor, show how a description shows up.

13. Show the descriptions that show up when hovering over every block.
14. By clicking step, turnTo, bridge, and banana, reach the following solution: turnTo bridge, step 10, turnTo banana, step 10
15. Make sure you have only used the keyboard for typing the number and jumping to the next line.
16. Make sure your students understand how to use clicking and hovering for object on the stage (banana, bridge) and for blocks at bottom (turnTo, step).

Play Time - 8 minutes

All students should complete challenges 11-15 with at least two stars. Use the teacher dashboard to keep track of students' achievements.

Wrap-up (5 minutes)

Discussion - 4 minutes

Open level 14 and ask your students: "How did you plan what to write in your code?"

Make sure to lead them to the correct answer, explaining the right train of thought needed when planning the code. We should first think about what steps should be taken to achieve our goal (in this case, get the banana), and then break the steps into separate statements, while deciding what should come first (should the turtle or monkey go first?). If we tell the monkey to move before the turtle is in the right place, he is going to fall in the water, and monkeys don't like water.

Review - 1 minute

Use this opportunity to remind your students that a program is a set of instructions, or simple tasks provided to a computer. These instructions are called statements. Statements can be anything from a single line of code to a complex mathematical equation.

Assignment (5 minutes)

After completing challenges 11 to 15, have the students reflect on what they have learned in the response area of the assignment and turn it in.

Part 1

1. Click the button below to goto to EasyCode and complete challenges 11 to 15.
2. Try to get 3 stars on challenges 11 to 15 if you have not already.
3. If you have extra time, complete as many of the skill challenges as you can from the Skill Mode tab, trying to get 3 stars on each.

Part 2

After completing challenges 11 to 15, reflect on what you have learned. Use the space below to write a paragraph explaining the importance of and the process to follow when planning the code.

EasyCode Sample Lessons: Grades 3-8

Turtle Lake: Challenges 16 – 20

In the previous 3 lessons your students have learned how to move around using code. They have actually mastered the foundation to programming, as they are now able to write a block of code that will carry out the instructions they intend to give the computer. We will take the current lesson to practice and reinforce this knowledge, and to deepen their understanding of what is actually going on. (Estimated time: 45 minutes)

Primary Objectives

- Students will practice using functions with different objects (monkey, turtle).
- Students will apply what they have learned: step, turn, turnTo, step- and activate code by clicking on run as they complete challenges 16-20.

Warm Up (5 Minutes)

Discussion - 5 minutes

Recall with your students that in EasyCode we are writing code in a programming language called CoffeeScript. As they experienced in the previous lessons, the code has to be written in a particular way in order for the computer to do what we are trying to achieve.

Explain that this is because a programming language, just like any language, has its own rules on how things can be said or written. In programming this is called the syntax of the language. There might be more than one correct way to say or write a certain thing in CoffeeScript, just like in English or any language. An important difference between programming languages and other languages is this: In a spoken language, sometimes we can say something incorrectly but still be understood. However, with the computer even the slightest mistake will definitely cause our code to fail. So we always have to pay attention to syntax and be very accurate. For example, if we forget a dot or a space in turtle.step 10 we will get turtle step 10 or turtle.step10, and the code will not do the right thing.

Activity (35 minutes)

Access EasyCode - 1 minute

Today you will start learning basic coding principles through a game called EasyCode. The language we will learn is called CoffeeScript.

Ask the students to:

1. Log in to Learning.com.
2. Go to the Turtle Lake: Challenges 16-20 assignment.
3. Click the link to launch EasyCode.

If a student is having trouble remembering his or her login information, use your Learning.com roster or provide students with log in cards.

Walk-through - 5 minutes

Click the EasyCode button to go to your EasyCode account.

1. Open challenge #15 and click reset to reset the code to what it was initially (blank).
2. Use typing only, no clicking, and enter the following code: turtle step 10
3. Click run to execute the code. Read out loud the error message that appears. Explain that the dot (.) is important. In this example the computer was able to guess what we meant, but this is not always the case.
4. Edit the code to this code: turtle.step 10 step15
5. Execute it and read the error message with the students.
6. Repeat the same with the following modification to the 2nd line (capital S): turtle.step 10 Step 15
7. And with the following (without breaking between lines): turtle.step 10 step 15
8. Conclude that spelling, punctuation, capitals and new lines are part of the syntax and are essential for our code to do what we want.
9. Finally, run a 3-star solution: turtle.step 10 step 15
10. When it completes, click replay and edit it to the following: turtle.step 10 monkey.step 15

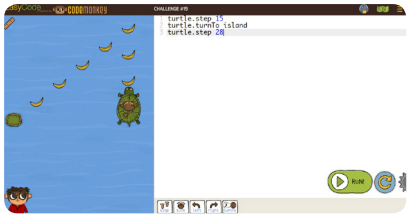
Conclude with your students that step and monkey.step can be used interchangeably, because the computer assumes we are referring to the monkey. When we refer to the turtle or any other object, we must use its name.

Play Time - 29 minutes

All students should complete challenges 16 to 20 with at least two stars. Use the teacher dashboard to keep track of students achievements.

Note that challenge #16 is a tricky one to achieve three stars. Make sure your students do not stay on this challenge for too long and encourage them to keep going and come back to it if they have time left. At the end of the lesson, you can open a discussion regarding this challenge and try to solve it together with your students in order to get those sneaky three stars.

In challenge #19 there are different ways to make the monkey turn the right way after catching 3 bananas. One way is by using the island: turtle.turnTo island



One way is by using any of the bananas along that path:
turtle.turnTo bananas[3]

In both cases, hovering and/or clicking will do the trick. Remind your students that hovering over an object shows its name, and clicking enters that name into the editor. If your students ask you about the meaning of something like bananas[3] just tell them that it is the way to access a particular banana and we will get back to it later on.

Note that challenge #20 is an assessment challenge that covers everything your students have recently learned.

Encourage students who finish early to open skill mode on the map and complete unlocked challenges.
As part of this lesson, students are asked to answer the following questions to reflect on what they have learned:

- What is the importance of being accurate with our syntax when we are coding? (Answer may include: if we are not accurrate, the computer may not be able to understand our commands or may do the wrong command.)
- How is a programming language different from a spoken language? (Answers may include: with spoken language, you can often have grammatical errors and still be understood; with programming language, you have to be exact in order for the computer to understand you command.)
- What is the name of the programming language that we are using in EasyCode? (CoffeeScript)

Wrap-up (10 minutes)

Discussion - 10 minutes

Revisit challenge #16 and discuss what challenges your students had when trying to get their three stars. As a group, work together to find a solution for getting the three stars.

Next, open skill challenge 2-7 from the class score book and solve it with your class. Ask them to explain how they plan the solution for this challenge. You can even invite a student to solve this challenge in front of the class. The trick in this challenge is similar to the one in challenge 16: tell the monkey to walk backwards in order to have less lines of code, and to get the third star.

Ask students if this challenge seems similar to one they ve solved before. (As noted above, this is similar to challenge 16.) Explain that it is fairly common to use references from old projects when programming, or even full blocks of code. Encourage students to go back to old challenges to get inspiration or help when needed.