

TECHNOLOGY: CODING – A LANGUAGE FOR THE FUTURE

According to the website Code.org, in the United States alone there are more than half a million coding jobs available. The reason is that few students are learning to code, but the demand for this skill is only increasing. As technology continues to grow, more jobs will require coding skills. In this week’s Teacher’s Corner, we will explore the technology component of STEM (Science, Technology, Engineering, and Math) by learning the basics of coding. Along the way, students will practice their listening and reading skills by coding in an English-only environment.

This week’s activity takes a constructivist approach to teaching and learning. In constructivism, students are encouraged to work through problems and discover answers. This style of learning requires teachers to take a less direct role in the classroom. Instead of directing students and telling them what to do, teachers support students in their learning. In this week’s activity, students should know that mistakes are part of the learning process. Our goal as teachers will be to encourage them to think through the problems and to keep trying – not provide answers to them.

For more on constructivism, check out [Constructivism in Theory and in Practice](#).

LEVEL

Intermediate to Advanced

LANGUAGE FOCUS

Reading, listening (primary focus); speaking (secondary focus)

GOALS

During this activity students will do the following:

- Practice listening skills while learning the importance of code
- Practice reading on-screen instructions while coding

MATERIALS

- Teacher: whiteboard/chalkboard, markers/chalk, computer and printer
- Students: pencils or pens, notebooks
- Computer lab: computers for each student (if possible) or one computer per small group of students (2-3 students per group)

PREPARATION

1. Read through all the materials carefully.
2. This activity uses the websites [Code.org](https://code.org) and [Scratch](https://scratch.mit.edu). Before using this activity in class, visit the websites to become familiar with them. If you are new to coding, try some of the basics yourself!

ACTIVITY PART ONE: CODE.ORG

This activity begins with an “Hour of Code” on the Code.org website. The Hour of Code allows students to practice the basics of coding with videos and tutorials that guide them through the process. After they have completed their hour of code, students can print out a certificate of completion.

1. Begin the class by having students go to a computer. Students can work individually or in small groups.
2. On the whiteboard/chalkboard write down the website Code.org and have the students navigate to that webpage.

3. Once the students are on the Code.org webpage, have them click on “Students.”
 - a. **Note:** The “Students” area is to the lower left of the screen and colored in purple.
Students may need to scroll down the page to see the student area.
4. On the student page, have the students click on the Hour of Code start button.
 - a. **Note:** The Hour of Code option is the second option on the screen.
5. Once the students are on the Hour of Code page, they can choose which Hour of Code they would like to do. The webpage offers a variety of options from famous characters to famous video games, but they all teach the same content.
6. After the students have selected their Hour of Code, have them begin the activity. Each activity begins with a short video that students should watch.
7. After watching the video, students should follow the on-screen directions to begin building their code.
8. Have the students work on their code for as long as time permits. Students who do not finish their code in class can work on it after class as a homework assignment.
9. If time permits at the end of class, have the students share what they did and what they learned by presenting their work to the class for an optional speaking activity.

Note: On the first day of incorporating this activity in your classroom, you may wish to walk students through the process and conduct this as a whole-class learning experience and demonstration. You could display your own computer and have all students choose the same activity in order to become familiar with it.

ACTIVITY PART TWO: SCRATCH

This second activity can be completed as an extension of the first activity or as a stand-alone activity.

1. Begin the class by having students go to a computer. Students can work individually or in small groups.
2. On the whiteboard/chalkboard write down the website scratch.mit.edu and have the students navigate to the Scratch webpage.
3. Once the students are on the Scratch webpage, have them click on “Try It Out.”
4. On the Try It Out page, have the students click on the “Getting Started with Scratch” section of the website on the right side of the webpage.
 - a. **Note:** To begin the tutorial, have the students click on the blue “Start Moving” button.
5. Have students follow the tutorial while they create a small project on Scratch.
 - a. For students who already know code or who are eager to learn more, encourage them to explore the Scratch webpage. Scratch is less structured than Code.org, but it allows students to make more complex projects.

Note: As mentioned for the first activity, you may wish to walk students through the process and conduct this as a whole-class learning experience and demonstration on the first day. You could display your own computer screen and have all students work as a whole class on the first project before they work in independent groups.



Optional activity: If students show interest and excitement over coding, have them work in teams to build an English language learning game on Scratch. Once the games are completed, students can share them with the class. For examples of English language learning games on Scratch, check out:

<https://scratch.mit.edu/search/projects?q=english+learning>

americanenglish.state.gov