

Program Kit

At WisCode Literati, we believe that teaching our communities how to code is essential for the future! Learning to code allows individuals to interact and compete in a highly digital society by teaching computational thinking, critical decision-making, experimentation, troubleshooting, and cause and effect.

We offer various kits and resources to help librarians and educators offer coding and problem-solving programs to their communities. WisCode Literati was started by a group of librarians interested in problem solving, technology, and learning.

Beginner Coding Camp with Code.org

This program uses the Course 2 curriculum on Code Studio at Code.org and requires no prior programming experience. During a six-week session, students learn about sequence, loops, debugging, conditionals, events, and functions (Course 3). In each class, students work collaboratively on an “unplugged” activity, without the use of devices, and then complete a self-guided tutorial using Blockly on the code.org site for that day’s topic.

Why?

This program is really focused on helping students understand the principles behind coding and how it makes computer programs work. Code.org offers 4 courses in Code Studio (course 4 is currently in beta) and provides complete lesson plans for each course, including the “unplugged” activities for each topic. It is easy to tweak the lesson plans to best fit your group of students and it is easy to move between the courses online to make it easier or more challenging for individual students. Since code.org uses Blockly, which is a visual programming language, students are able to focus on how they have to think and problem solve to make a program work rather than on having to know a specific coding language.

Who?

We limited our coding camp to 10 students (small groups are definitely best) and started with grades 3-5, but it is definitely flexible for different age groups as well as abilities. It can even be used with pre-readers if you use the Course 1 level. It is possible to run this program with one person, but the more

Dream Big. Learn Code.

WisCode Literati | www.wiscode.org | literati@wiscode.org | @wiscodeliterati



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

assistants you have, the more one-on-one help you can give to students as they work through the online component. We used volunteers as assistants and it worked great.

What?

Code.org can be used on any kind of computer or device and everything can be done with pair programming, so you could run the program with as few as one device per pair of students

How?

Code.org can be used on any kind of computer or device and everything can be done with pair programming, so you could run the program with as few as one device per pair of students. You can set up a teacher account, which gives you access to the teacher dashboard. This allows you to set up accounts for each of the students in each section of the class that you run without requiring an email address from them. Each student is assigned a secret word and a URL for the specific sections they're in. They can access their account at home to continue their progress using that URL and their secret word.

The basic outline we followed for our beginner coding camp was:

- Week 1: Unplugged Activity: graph paper programming
Online: sequence with maze and artist
- Week 2: Unplugged Activity: iteration dance
Online: loops with maze, artist, and bee
- Week 3: Unplugged Activity: relay programming
Online: debugging with bee and artist
- Week 4: Unplugged Activity: conditionals with cards
Online: conditionals with bees
- Week 5: Unplugged Activity: big event
Online: create a flappy bird game and create a story in play lab
- Week 6: Unplugged Activity: functional suncatchers
Online (Course 3): functions with artist and bee

We followed the lesson plans from code.org pretty closely, but made some adjustments to the unplugged activities to make them better suited to our

Page 2

Dream Big. Learn Code.

WisCode Literati | www.wiscode.org | literati@wiscode.org | @wiscodeliterati



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

environment and the group of students we were working with. There are assessment worksheets that are provided with the lesson plans for the unplugged activities, which we only sometimes used if we felt they offered a kind of clarification the kids wouldn't get from the activities themselves. We much preferred to encourage the collaboration that was required for the activities.

The program instructor typically leads the class in doing the unplugged activity as a group, usually about 15-20 minutes. The remaining time is spent with the students individually working through the stages online, with the instructor and assistants walking around to help individual students as needed.

Extras

We originally offered this program as an eight-week session with each class running 1-1/2 hours in length. We found that this was too long. It was hard for students to be able to attend each class in an 8-week period, so we adjusted that to 6 weeks, which worked so much better. It was great to have 1-1/2 hours for some of the topics, but it was hard for students to stay focused for that long. We found that they understood the concepts just as well with an hour class and they stayed engaged the entire time. They can also work on things at home if they don't finish a stage of the online component.

A few things we picked up when we offered our first few coding camps:

- Make sure the students read the directions when doing the online puzzles. Some kids have a tendency to just skip over the instructions, but they usually give them specific pieces of information that they need to solve the puzzle. It was also good to remind them what the goal of each puzzle was. The course often requires them to solve a puzzle a certain way based on that goal and with the fewest blocks possible in order to move on.
- It was helpful to guide students toward taking it one step at a time when they got frustrated. Ask them what the first thing they would do is and then ask what they would do next. Once they got into the habit of breaking it down step by step, they realized how capable they were of

solving the puzzles. It was also helpful for them to run programs with the code they had so far, even if it was incomplete, when they got stuck on a step and they would usually see what they had to do next right away.

- Trial and error is okay and is really part of the process of coding. Students learned a whole lot more when they made an error in their code and figured out what to try in its place themselves. We really tried to keep ourselves from guiding the students in a certain direction, instead asking them questions about what they wanted to accomplish, what blocks they would need, how they could write the code with fewer blocks, etc.. We did find at times that we needed to point out certain patterns in code to help them see how they could write their code in an easier way, but, other than that, we tried to be as hands-off as possible.

If you're near the Milwaukee area, a free code.org professional development workshop is offered periodically at Marquette University (<https://code.org/professional-development-workshops>).

This project was made possible in part by the Institute of Museum and Library Services (www.ims.gov) with grant 15-30-7402-15-112-LSTA.

Contributor

This kit was created by Jen Fait, Head of Youth Services at Kenosha Public Library.