Crack the Code Volunteer Handbook

# Description:

Bring the whole family for a fun introduction to computer programming!  
All ages (kids under 12 must be accompanied by an adult)  
90 minutes

# Why Coding?

In today’s world, just about everything relies on coding (computer programming) of some kind. Just like math or reading, coding has become a foundational skill – something that’s important to understand whether or not you plan to have a career in computer science. Learning about coding also helps kids develop great problem solving and logic skills, perseverance, and critical thinking.

# The Program:

This fun, noisy program will feature various coding activities appropriate for ages 3 to 93. Volunteers will facilitate the activities at each station, and encourage parents and children to explore coding together.

There are two types of activities: offline (no screen) and online (requires computers). Families with younger children should focus on the offline activities. For older kids, teens and adults, the focus will be on trying fun self-directed online coding lessons with help from volunteers. Kids and parents are encouraged to attempt these together.

The setup of each activity will depend on your library’s size and available spaces

## Offline Activities

|  |  |  |
| --- | --- | --- |
| **Activity** | **Ages** | **Setup** |
| Code-a-pillar toys | 3-6 | Floor space with small obstacles like toys or stacks of board books |
| Robot Turtles board game | 3-8 | Tables and chairs |
| littleCodr card game | 4+ | Medium to large open area (some obstacles like furniture are fine but the more open the better) |
| IF-THEN game (optional) | 4+ | Large open area |

## Online Activities

|  |  |  |
| --- | --- | --- |
| **Activity** | **Ages** | **Setup** |
| Hour of Code modules | 6 and up | PCs or Chromebooks, with at least two chairs per computer. Chromebooks are preferred since there’s no login needed. |

As a volunteer, you will help with setup and cleanup, and run the activity in between. More information about each station is included in the next section.

# Code-a-pillar (ages 3-6)

## Volunteer Role

Your job is to encourage parents and children to play with the Code-a-pillar; you can show kids how it works, set up small obstacle courses, and tidy up the area when necessary to keep it safe for preschoolers. There is only one Code-a-pillar so encourage kids to work together or take turns.

Note: the large button on top is to start the program – it’s not actually the “on” switch. There is a small black power switch on the bottom of the head (it will make a lot of noise when it’s on).

**Share the message:** Toys like Code-a-pillar are a great way for kids to begin learning basic coding skills, like creating a set of instructions (an algorithm), problem solving (how do I get from point A to point B?) and debugging (figuring out what’s wrong when something doesn’t work the way they wanted, and fixing it).

## Activity Details

* Code-a-pillar is a fun toy designed to teach young children about problem solving, algorithms, and debugging on a basic level. It’s quite noisy and for this program, that’s ok!
* Children take the caterpillar apart and rearrange the segments to program it; the simple instructions include moving forward, turning left or right, flashing lights, or playing music.
* An **algorithm** is a set of instructions that a computer follows; **debugging** is finding a mistake in that set of instructions and fixing it. For example, if you wanted the caterpillar to turn left and it turned right instead, you could remove that segment and replace it – and you just debugged your program.
* The Code-a-pillar comes with a “start” and “target” disc; encourage children to try to get from one to the other. Start with a short, straight path, and then make it more complicated by introducing obstacles (like other toys, or a stack of board books) that they have to steer around.
* Help kids work through problems, but don’t solve the problems for them. For example, if the toy turned the wrong direction, ask, “Which way did it go? Where do you want it to go? Which piece looks like it goes that way?” instead of saying, “You need to add this piece”.



# Hour of Code (ages 6 and up)

## Volunteer Role

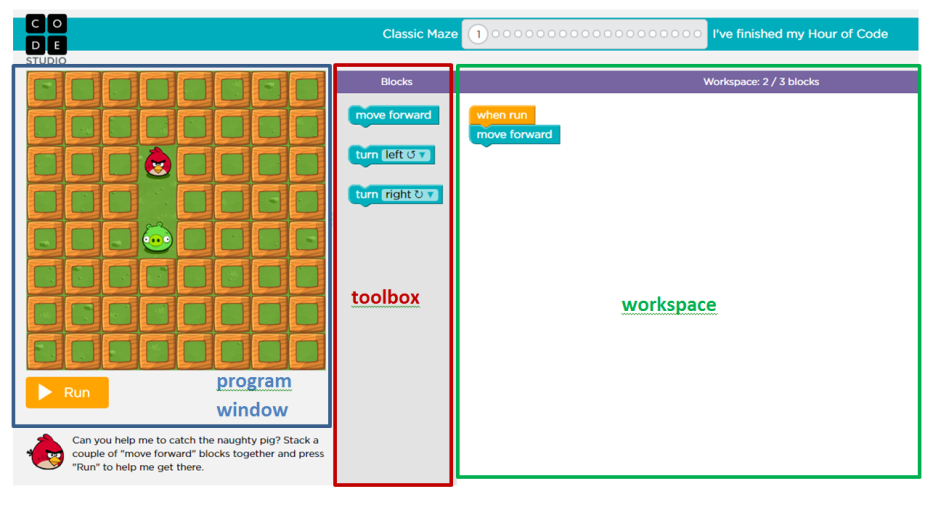
Set each computer up by navigating to [code.org/learn](https://code.org/learn). They will need to have the volume turned on (there are short tutorial videos). Ask your staff contact for help checking this. Kids and parents can choose their own lesson from the ones provided, but may need some help choosing. Encourage parents to work through the lessons with their children so they can learn together.

Reset the computer every time someone leaves by returning to [code.org/learn](https://code.org/learn).

**Share the message:** Anyone can learn to code! In today’s digital world, coding is a foundational skill like reading or math; everyone should know a little bit about computer science.

## Activity Details

* These self-directed online coding lessons were created by Code.org, a nonprofit organization dedicated to helping every child learn computer science. Don’t worry though – they’re a lot of fun, and you don’t need to know anything about coding to get started!
* Choose a lesson from the eight provided; “Classic Maze” (the Angry Birds one) is a great place to start.
* These modules use Blockly, a drag**-**and-drop programming language. The screen is split into three parts: the program window, where the program runs; the “Blocks” or toolbox section, which has all of the pieces of code that kids can use; and the workspace, where they build their program.
* Kids drag code blocks from the toolbox and snap them together in the workspace. The program will always run from top to bottom; when they are ready, they just click on the yellow “Run” button at the bottom left of the program window.



#### Parents can help their children by:

* Talking through problems, not doing it for them; for example, if the bird didn’t move far enough, parents should ask questions like “how many more spaces do we need to go?” instead of saying “you need one more move block”.
* Encouraging persistence - these modules are totally forgiving, so they can keep trying until they get it right. Coding is all about persistence and solving problems.
* **Celebrating success!** These modules are set up as short lessons or puzzles, so every time they finish one successfully, parents should high five or tell their child he or she did a good job.

# The IF-THEN Game

## Volunteer Role

Read the instructions for the IF-THEN game (below) and be prepared to explain them to kids and parents. Start with the lowest difficulty and yourself (or another volunteer) as the Programmer. As players join (and figure it out), you can ask “Who wants to be the Programmer?” and let someone else take over.

**Share the message:** this game helps kids understand the coding concept of conditionals – actions that only happen when a statement is true. An IF-THEN statement is one example of a conditional: IF this statement is true, THEN do this action.

## Activity Details

This game is similar to Simple Simon. Start with one “Programmer” (this can be a child or an adult); everyone else is a “Computer” that must follow the Programmer’s directions.

The Programmer makes an IF-THEN statement by saying something like “If I \_\_\_\_\_\_, then you \_\_\_\_\_\_”. For example, “IF I touch my nose, THEN you touch your ears.” The programmer then does any action he or she wants and the Computers try to follow along.

#### Difficulty Levels

This game is suitable for kids from age 3 and up, and you can adjust the difficulty based on their ages.

**Easy – IF I do this, THEN you do this**

The Programmer and the Computers do the same action at the same time. Monkey-see, monkey-do.

Example: “IF I touch my nose, THEN you touch your nose”.

**Medium – IF I do this, THEN you do that**

In this variation, the Computers do something different than the Programmer – but only when the IF statement is true. This is harder because what you see and what you hear are different.

Example: “IF I turn in circles, THEN you jump up and down.”

**Hard – IF-THEN-ELSE**

A more complex version of the IF-THEN statement, IF-THEN-ELSE statements also tell the computer what to do when the condition is false. So the Computers should always be doing something, but it will change depending on what the Programmer is doing.

Example: “IF I turn in circles, THEN you turn in circles, ELSE jump up and down.” So – if the Programmer is turning in circles, everyone is turning in circles; if the Programmer is just standing still, all of the Computers should be jumping up and down.

# littlecodr and Robot Turtles (ages 4+)

## Volunteer Role

Read the instructions for each game (littlecodr and Robot Turtles), or watch the videos showing how to play, and be prepared to teach kids and parents how to play.

In each of these games, kids get to be the programmer (the boss!) and a parent or older child acts as the computer, following the instructions they are given.

Your job is to encourage parents and children to try the games, and show them how to play. Games are more appealing if they’re already set up (not sitting inside the box). You can also act as the computer so that kids and parents can work on the coding part together.

**Share the message:** Both of these games teach children basic coding skills, like creating a set of instructions (an algorithm), problem solving (how do I get from point A to point B?) and debugging (figuring out what’s wrong when something doesn’t work the way they wanted, and fixing it).

## Activity Details

#### littlecodr

* littlecodr is a card game; kids use simple action cards to lay out a program, which the computer (or robot) must then follow. The fun part is that the computer is a real person, moving through the real world, so kids have to figure out things like how to get Dad around the couch (or how to make Dad run into the couch…).
* Wildcards act as variables, which the kids can define (“spin in a circle”, “moo like a cow”…)
* Mission cards provide challenges for kids to solve.
* Littlecodr gameplay video (0:48): <https://www.youtube.com/watch?v=WbpKH-bykG8>

#### Robot Turtles

* Robot Turtles is a board game where everyone can win!
* Every player chooses a turtle, and has their own jewel which they must try to reach.
* One person (usually the adult) is the “Turtle Mover”, who moves the turtles according to each player’s instructions.
* How to Play video (2:14): <https://www.youtube.com/watch?v=RHjB9XQodzE>