FIRIA LABS

MISSION 1 & MISSION 2 (Welcome to CodeSpace, Introducing CodeX)

Time: 45 minutes

Overview:

These first missions are all about getting to know the **Codespace** user interface and the **CodeX** hardware. Before students finish, they will plug in the hardware and write some code to make it do something!

In this early stage, the most important guidance is to **carefully read** the instructions. The answer to most problems is right there on the screen!

Objectives:

- I can safely connect and disconnect the CodeX to my computer.
- I can create a new file and name it according to its purpose.
- I can write code using conventions of capitalization and punctuation specific to Python.

Standards:

1B-CS-01 Describe how internal and external parts of computing devices function to form a system.

1B-CS-02 Model how computer hardware and software work together as a system to accomplish tasks.

1B-CS-03 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. .

CSP Framework:

Computational Thinking Practices:

5.A Explain how computing systems work.

5.C Describe the impact of a computing innovation.

Key Concepts:

- Follow instructions in the **Objective panel** carefully.
- The **text editor** is where you type in the code!
- Your code is saved to the file **name** you create.
- Look for "tool icons" to collect coding-tools in your **Toolbox.**
- The **CodeX** is a powerful general-purpose computing device you can use to build an infinite number of cool projects.

Preparation:

Create a class on your dashboard and have access to the join code (see link for help)

Decide:

- Will you use the programming journal or vocab canvas
- What will you do for wrap-up
- What will you do for formative assessment?

Links:

- Codespace: https://sim.firialabs.com/
- Getting started (for students)
- Assignment (make a copy or put into your LMS)
- Programming Journal (make copy for each student)
- Introduction to the vocabulary canvas
- Daily reflection form (Here is the one I use. You will want to make your own)

Agenda:

- Warm-up (5 minutes)
- Mission 1 (10 minutes)
- Mission 2 (15 minutes)
- Wrap-up & Assessment (10 minutes)

Vocabulary:

- **Bug:** When your program doesn't do what you intended it to do
- **Debugging:** the process of understanding what the computer is actually doing and then changing the code to do what you want it to do
- **CPU:** Central Processing Unit or the brain of the computer
- **Peripheral:** A device that interacts with the CPU (common peripherals are LED lights, display screen, buttons, mouse, keyboard, and printer)

Teaching Guide

Warm-up (5 minutes)

🧑‍🤝‍🧑 **Discuss** – Use a discussion strategy, like journaling, working at boards, selecting random students, or a form of think-pair-share.

> 🧑‍🤝‍🧑 **SAY:** electronic devices have similar circuit boards inside. The tools and techniques you are learning apply to all the electronic devices you use every day! Can you name a few devices you use everyday that might contain computer chips or "microcontrollers," such as the one on the CodeX.

Possible answers:

- Microwave oven
- Cell phone
- Automobile
- Watch or fitness tracker
- Video game controller
- Refrigerator
- Home thermostat
- Coffee maker
- Bread machine
- Alarm system
- Fuel pumps
- Automatic garage doors
- Electronic locks

🧑‍🤝‍🧑 **Discuss:**

> How are our lives impacted by electronic devices? How were related tasks done before computer technology was invented?

Activity – Mission #1 (10 minutes)

💻 Randomly group students into pairs for pair programming. (Alternatively, students can work individually.)

Students log in to one computer. Two computers can be used if they want to see instructions on one computer and work on the other computer. However, the assignment document requires snippets, so it will need to be open on the same computer as CodeSpace.

Follow the instructions for students to join the class and have the CodeX curriculum available in CodeSpace.

Each pair opens the assignment. Follow the instructions in Mission 1 and the Assignment document (#1-#4)

Activity – Mission #2 (15 minutes)

💻 Students remain with same partners. They continue to Mission 2 and the Assignment document (#5-#12).

💡 **Teaching tip:**

> Remind students to be careful when plugging the USB-C cable into the CodeX; they should focus on pushing it and pulling it straight and not rocking it back and forth! With a "bare circuit board" like the CodeX, students should have clean hands before handling, as food particles and liquids can cause the circuits to malfunction. Static electricity can also damage the CodeX, so you'll need to show students a way to "ground" themselves prior to handling it. An example reliable grounding point is a light-switch screw or metal wall plate.

💡 **Teaching tip:**

We recommend waiting to hand out the CodeX until they are first shown on the screen in Codespace. The on-screen instructions will inform students about proper care and handling of the CodeX. Each student will need a CodeX and a USB-C cable to complete this first project.

✅ Assignment is complete and ready to turn in. If paired, both students should include their names on the document. The assignment document instructs students to paste a snippet of their code in the document. You can pass off completed programs a different way, if you like.

Wrap-Up (10 minutes)

If this lesson is completed in one class period, the following can be used as a wrap-up. If you are on a block schedule and continuing to the next lesson, a wrap-up isn't necessary.

Introduce clearing the CodeX. You can use this short slide deck as a guide.

The programming journal can be introduced at the end of this lesson (or any time thereafter).

If you are starting the programming journal with this lesson, each student needs a copy to keep their own programming journal. The journal is divided into three sections: vocabulary, meaningful notes, and debugging chart. At this stage, only the vocabulary section is needed.

Formative Assessment:

- Daily reflection journal or Google form (this is my suggestion for daily reflection)
- Exit ticket on vocabulary (bug, CPU, peripheral)
- Group review on vocabulary (bug, CPU, peripheral)
- Students create a vocabulary canvas with vocabulary words (bug, CPU, peripheral).
  This concept is adapted from Code.org AP CSP curriculum. The vocabulary canvas slide deck is created by Code.org.

SUCCESS CRITERIA:

❏ Identify major parts of the CodeSpace interface: Mission Bar, Objective Panel, text editor, CodeTrek, Toolbox, and Lesson Navigation Controls

❏ Successfully connect and disconnect the CodeX using the USB-C cable.

❏ Identify major parts of the CodeX: USB connector, LCD Grid, CPU

❏ Write a program, run it, and save it to the CodeX