



MISSION 2: Introducing CodeBot Lesson 2 Your First Program (Obj. 6-10)	Time Frame: 30-40 minutes						
<p><b>Project Goal:</b> Students will connect CodeBot to the computer and learn the basics of Python.</p> <p><b>Learning Targets</b></p> <ul style="list-style-type: none"> <li>• I can safely connect and disconnect the CodeBot using the USB cable.</li> <li>• I can create a new file.</li> <li>• I can write code using the conventions of comments and syntax.</li> </ul>	<p><b>Key Concepts</b></p> <ul style="list-style-type: none"> <li>• Python requires all objects – variables, peripherals, etc. – to be spelled exactly the same; capitalization matters!</li> <li>• Adding comments and blank lines in your code makes it easier to read.</li> <li>• Python programmers can import a library, or module, to access pre-defined functions.</li> </ul>						
<p><b>Assessment Opportunities</b></p> <ul style="list-style-type: none"> <li>• Mission 2 Lesson 2 Log assignment</li> <li>• Submit completed program <b><i>LightsOn</i></b></li> <li>• Submit the program with challenges</li> <li>• <a href="#">Mission 2 Obj 6-10 Review Kahoot!</a></li> </ul>	<p><b>Success Criteria</b></p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Create a new file</li> <li><input type="checkbox"/> Import the botcore library</li> <li><input type="checkbox"/> Use code to turn on an LED</li> <li><input type="checkbox"/> Use descriptive comments</li> </ul>						
<p><b>Teacher Materials in Learning Portal</b></p> <ul style="list-style-type: none"> <li>• Mission 2 Lesson 2 Slides</li> <li>• Mission 2 Lesson 2 Log</li> <li>• Mission 2 Lesson 2 Answer Key</li> </ul>	<p><b>Additional Resources</b></p> <ul style="list-style-type: none"> <li>• <a href="#">Mission 2 Obj 6-10 Review Kahoot!</a></li> <li>• <b><i>LightsOn_challenge</i></b> sample code (in learning portal)</li> </ul>						
<p><b>Vocabulary</b></p> <ul style="list-style-type: none"> <li>• <b>Comment:</b> Notes in a program code that don't get executed (more information in Mission 3)</li> <li>• <b>Import:</b> Provides access to a module, or library, of pre-defined Python objects and functions to use in your code</li> <li>• <b>Boolean:</b> A data type that has two possible values: True or False</li> </ul>							
<p><b>New Python Code</b></p> <table border="1"> <tbody> <tr> <td>from botcore import leds</td> <td>Import from botcore only the leds object and its functions</td> </tr> <tr> <td>leds.user_num(0, True) leds.user_num(7, False)</td> <td>Turn on one user LED (arguments are LED number 0-7, True=on, False=off) Turn off user LED #7</td> </tr> <tr> <td>leds.ls_num(0, True)</td> <td>Turn on a line sensor LED (arguments are LED number 0-4, True/False)</td> </tr> </tbody> </table>	from botcore import leds	Import from botcore only the leds object and its functions	leds.user_num(0, True) leds.user_num(7, False)	Turn on one user LED (arguments are LED number 0-7, True=on, False=off) Turn off user LED #7	leds.ls_num(0, True)	Turn on a line sensor LED (arguments are LED number 0-4, True/False)	
from botcore import leds	Import from botcore only the leds object and its functions						
leds.user_num(0, True) leds.user_num(7, False)	Turn on one user LED (arguments are LED number 0-7, True=on, False=off) Turn off user LED #7						
leds.ls_num(0, True)	Turn on a line sensor LED (arguments are LED number 0-4, True/False)						
<p><b>Real World Applications</b></p> <ul style="list-style-type: none"> <li>• Many electrical devices have single-color LEDs. Discuss devices that have them and what they are used for.</li> </ul>							
<p><b>Teacher Notes:</b></p> <ul style="list-style-type: none"> <li>• This is a good time to discuss common programming practices, such as descriptive variable names, adding comments and blank lines, use of capital letters, etc.</li> <li>• You may want to create a poster of Python commands to refer back to during the Mission Pack. Or you can encourage students to create their own “library” of Python codes.</li> </ul>	<p><b>Extensions / Cross-Curricular:</b></p> <ul style="list-style-type: none"> <li>• Make a poster or chart of Python commands.</li> <li>• Turn on the line sensor LEDs.</li> <li>• Combine user and line sensor LEDs to create a pattern, like turning on the even LEDs and leaving the odd LEDs off.</li> <li>• <b>LANGUAGE ARTS:</b> Students write about technology today and its impact.</li> <li>• <b>SCIENCE:</b> Have a lesson on LEDs and light.</li> </ul>						



### Preparing for the lesson:

Students will need a CodeBot and USB cable for this lesson.

Students can work with a partner (pair programming) or individually for the lesson.

- Look through the slides and workbook. Decide what materials you want to use for presenting the lesson. The slides can be converted to Google Slides. They can be projected on a large screen. The workbook (if used) can be printed or remain digital through your LMS and given to students.
- Be familiar with the mission log assignment and the questions they will answer. Prepare the assignment to give through your LMS (or print it for each student).
- If you have a word wall, or another form of vocabulary presentation, prepare the new terms.

### Lesson Tips and Tricks:

#### Teaching tip:

You can use a variety of discussion strategies to get the most engagement from your students. For example, you can have students write their answers before asking anyone for an answer. You can use one of many think-pair-share methods. You can have students write their answer and share with someone, and then have other students share answers they heard from their peers. You can randomly select students to answer.

#### Pre-Mission Warm-up: -- slides 2-3

Students can write in their log first and then share, or discuss first and then write in their log. The two warm-up questions are review from Lesson 1. You can modify them if you want to do a different warm-up.

- Question: What are some peripherals on CodeBot?
- Question: What is the difference between inputs and outputs?

#### Mission 2 Lesson 2 Activities:

Students need the CodeBot, a USB cable, and a computer or laptop with access to the Internet. The Chrome browser works best.

Each student will complete a Mission Log. Students can work in pairs or individually.

#### Teaching tip: Objective #6 -- slides 4-6

Students will click on the USB connector in the simulator to complete the goal. They should also connect the CodeBot physically. Students will answer a question in their mission log assignment.

#### Teaching tip: Objective #7 -- slides 7-8

Usually the CodeBot only needs to go through this connection process one time. But it is useful to see the message and know what to do if it happens periodically. The instructions in CodeSpace use CodeX in the images, but it is the same for CodeBot.

The button MUST be clicked first to connect. You can't just click OK.

The target should be USB CodeBot, not CodeX. The instructions in the slides and workbook will show CodeBot.

#### Teaching tip: Objective #8 -- slides 9-10

The name of the new file must match the name in the goal. No spaces should be used in filenames.



### **Teaching tip: Objective #9** -- slides 11-14

Students cannot copy and paste code from CodeTrek. They must type their own code, but they can use the code in CodeTrek to know what to type. Sometimes it gives the code, and sometimes it gives a **# TODO:** (to do). This is a good time to emphasize the purpose of **# TODO:**. Students should not type the # TODO, but substitute the actual code where the **# TODO:** is.

### **Teaching tip: Objective #10** -- slides 15-18

First students read about capitalization and punctuation. They answer two questions in the mission log. Then they learn more about the two lines of code: import and Boolean values. They answer two more questions in the mission log. Finally, students type in the two lines of code and turn on a single user LED.

### **Teaching tip: Objective #10 Challenges** -- slides 19-21

Three challenges are given to extend Objective 10 and give students a chance to be creative. The extensions are listed in the “Extensions / Cross Curricular” section above. The challenges can be omitted, depending on your students and the time frame. If you have time, I recommend having your students do the challenges.

### **Post-Mission Reflection:**

The post-mission reflection asks students to reflect on their experience so far with programming. You can change the question if there is something else you want to emphasize with your students. This can be a very good question to share-out with the class.

- What is something you learned about programming today?

You can use an extension or cross-curricular activity as post-mission activity. If students do the challenges (or extensions) you can have students do a gallery walk and appreciate the work and creativity of each other.

Optional: Start a poster for Python code.

End by collecting the Mission 2 Lesson 2 Log.

You can review with a [Mission 2 Obj 6-10 Kahoot!](#)

### **SUCCESS CRITERIA:**

- Create a new file
- Import the botcore library
- Use code to turn on an LED
- Use descriptive comments