Unit 2: Putting it all Together

Mission 7: Personal Billboard

Intro and Discussion Points:

Have you ever made a sign to post on a door or wall? How about a name badge to wear? Or a cap or t-shirt with a message or slogan on it? In this project, students will build a device that lets them display images or text, and even use the CodeX's buttons to select what is displayed to suit a particular occasion or mood.

Emphasize that this is a project with *practical* applications. Imagine if everyone at a party or meeting wore CodeX badges like this, to show their name and interests. Or what if you made this into a cap for a sports team, to give out to all the fans before a big game?

UNIT 2 : Putting it all Together	**MISSION 7: Personal Billboard**	**# DAYS:** 3
UNIT GOALS: Students will synthesize skills to create more complex programs.	**ADDITIONAL MATERIALS:** ● none	**VOCABULARY:** ● List ● Data types (image, string) ● Conditional statements

FOCUS CSTA STANDARDS:1B-AP-09, 1B-AP-10, 2-AP-11, 3A-DA-09, 3A-AP-14

LEARNING TARGETS:
- I can create a list to make my code more efficient.
- I can distinguish between string and image data types.
- I can apply an if/else conditional statement to a new program.

SUCCESS CRITERIA:
- ❏ Program the buttons to select from a series of images to show.
- ❏ Change the code to make it easy to add lots more images.
- ❏ Mix text messages with a selection of images.

KEY CONCEPTS:
- An infinite loop is a good way to continuously check for button presses. Many programs follow this pattern, with a **"main loop"** that monitors and acts on events as they occur.
- You can use a number variable to track the **state** of the program. In this project the buttons just add +1 or -1 to the state variable, and the remaining code sets the display based on that number.
- Comparison operators! In python, the **double-equals** sign is used to compare for equality. (Single equal is used for *assignment* to variables.)
- Codespace lets you **inspect variables in the debugger**.
- Python's list is a powerful way to hold a collection of objects.
- New data type – **strings**!
- Smoothly scroll messages on the CodeX with display.scroll()
- You can check the **type** of a variable, and deal with it accordingly.

DISCUSS REAL WORLD APPLICATIONS:
Have you ever scrolled through a menu system on a website or video game? Software applications of all kinds deal with lists or collections of different types of data. What if you made it so instead of button-presses, the selections just advanced on their own? The list could be animation sequences, or the set-list for a band – you decide! The code you've learned to write in this project has thousands of real world applications.

ASSESSMENT STRATEGIES:
5.5 Checkpoint - Spend time focusing on the functions of a list..
Remix suggestions (set aside 0.5-1 period to complete):
- Add music to your selections. You may want to study the music.play() tool information, and use the wait=False parameter.
  - ex: music.play(music.NYAN, wait=False)
- Make the selection advance at a controlLCD speed when you hold a button down.
- When both buttons are pressed at the same time, go back to the first selection in list.

TEACHER NOTES:
Always refer to **Answer Keys by Mission** if you get stuck. All coding solutions are available, in alphabetical order.