Create Performance Task: Practice #1
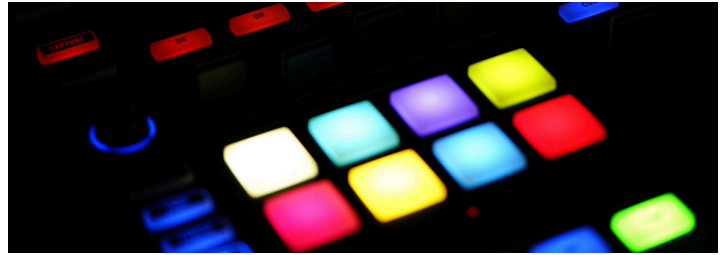
For this project, you will practice writing a program that meets the requirements for the Create Performance Task, which is part of the AP exam. This project will include multiple lists and a function with a parameter.

Practice #1 Instructions

Step #2 – Decide on a topic for your first two lists. Since the Diamondbacks are in the World Series, I selected the team as a topic. Then I created a list for the position and another list for the players. You will do something similar, but you can pick your own topic. Make sure you match the information in the two lists. Suggestions: Player/position, musician/song, artist/painting, author/book, family member/name, class/period or teacher. The list should have at least 6 items, but you can have more. (Slide 7)

Example of two lists on one topic:

```
dbacks_pos = ["pitcher", "catcher", "1st Base",
          "2nd Base", "3rd Base", "shortstop"]
dbacks_player = ["Merrill Kelley", "Gabriel Moreno", "Christian Walker",
          "Ketel Marte", "Evan Longoria", "Geraldo Perdomo"]
```

Name of list	Data in the list

Step #3 – Create an intro() and ending() function. Use print statements. The intro should let the user know what the buttons will do. For example, the R button will scroll to the right. The L button will scroll to the left. The U button will get a random item from the list. The D button will quit the program. The ending should just be a message so the user knows the program has ended. (Slides 8-10)

Step #4 – Write the code for your main program. Start by calling the intro function and initializing an index variable. Then start your infinite loop. Inside the loop, add code for each button, printing the information from both lists when scrolling to the right or left, or getting a random item. (Slides 11-14)

```
display.clear()
display.print(dbacks_pos[index])
display.print(dbacks_player[index])
```

Step #5 – Decide on another topic for your third and fourth lists. Going with the Diamondbacks theme, I chose the Texas Rangers. But you can use any other topic. As you think about your topic, decide on two matching lists. The lists should have the same number of items as the first list. If your first two lists have 6 items, your next two lists should also have 6 items. (Slides 15-16)

Example of second topic:

```
rangers_pos = ["cather", "3rd base", "1st base",
               "shortstop", "2nd base", "outfielder"]
rangers_player = ["Mitch Garver", "Josh Jung", "Nathaniel Lowe",
                  "Corey Seager", "Marcus Semian", "Adolis Garcia"]
```

Step #6 – Make a new function for the code that is needed for both topics and their lists. Copy code from the infinite while loop and paste it in the new function. (Slides 17-21)

Step #7 – Add a parameter to your function that will let the user select a topic. (Slides 22-25)

Step #8 – Test and debug

Challenge (optional) – Include the topic in your display. (Slide 27)

To turn in the assignment, download your code (FILE-DOWNLOAD), which will be a text file. Add your name in the filename. Then submit the file through Google Classroom or the class LMS.

Wrap Up: Practice #1
Use your program code to include snippets that show the following:

Name of a list used in the code:	
Code snippet where list is created:	
Code snippet where list is used:	
Code snippet of function with a parameter:	
Code snippet of parameter used in an if statement:	
Code snippet where function is called:	
What is one possible value of the	

argument, passed to the function?	
What will happen if this value is passed?	
What is another value of the argument, passed to the function?	
What will happen if this value is passed?	

SUCCESS CRITERIA:
- ❏ Create a new program file for the Create PT Practice #1
- ❏ Import the modules needed for the program
- ❏ Create two lists on one topic
- ❏ Create functions for intro and ending
- ❏ Write code for the main program that will include if statements for four buttons
- ❏ Create two more lists for another topic
- ❏ Create a function for the code that is needed to print data from either topic
- ❏ Add a parameter with the "status" variable
- ❏ Use the parameter in an if statement
- ❏ Test and debug the program so that it runs as expected