

Mission 5: Remix

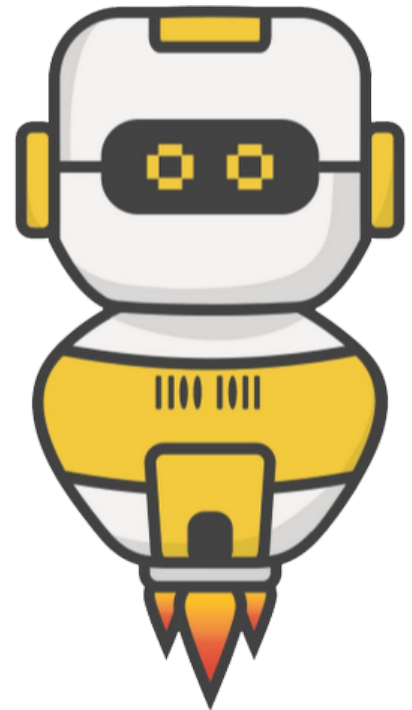
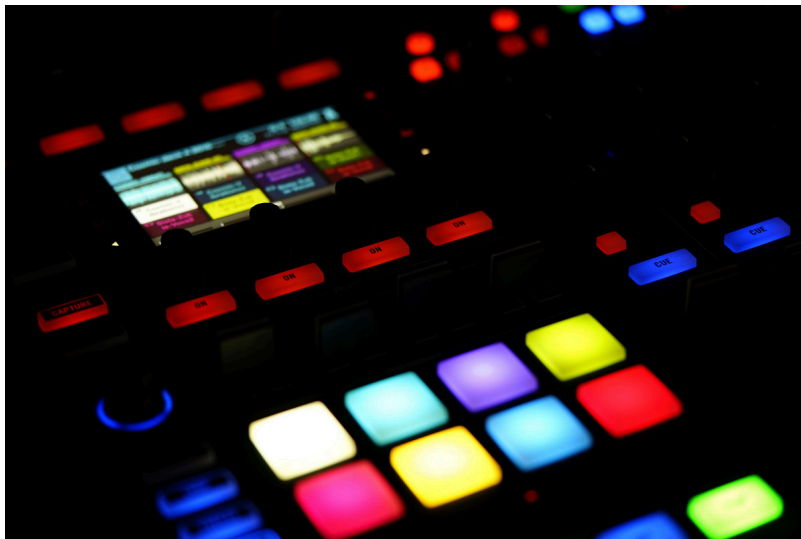
Student Workbook





Mix it up with sound!

This assignment will let you be creative and come up with your own program for the CodeX to run.



Go to the Mission 5 Remix Log and fill out the Pre-Remix preparation.



Time for a project remix

A remix can be:

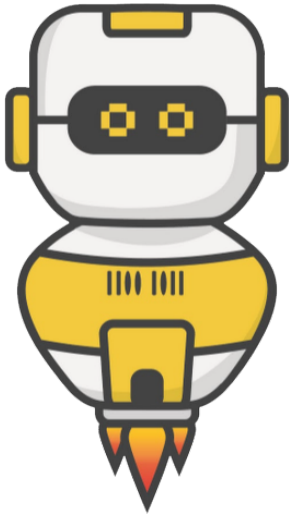
- A new program created by adding new code to a program you already created
- You can combine parts of two or more programs in a remix
- Use a similar idea in a different way

Creating a remix will let you:

- Master the skills and concepts practiced in the mission
- Be creative
- Remember code from earlier programs and missions
- Work with your peers
- Design an original program and write the code all on your own

Step #1: Review the mission

- Review your programs from Mission 2 through 5
 - What do the programs do?
 - What skills were used or concepts learned?



DO THIS:

- Open your project from Mission 5 - Music1
- Review what the program does
- Review the programming concepts and skills you learned
- Fill out the information in the remix log

```
Music1 x Music1_Remix1 x
1 # Student Name
2 from codex import *
3
4 # Display MUSIC pic
5 display.show(pics.MUSIC)
6 # Play Africa song
7 audio.mp3("sounds/africa")
8
```

Step #2: Brainstorm ideas

- Read through remix suggestions
 - Six ideas are on the next pages. You can use any of these ideas or come up with your own.
- Use your creativity to come up with your own idea for a project
- Decide with your partner what project you will do



Mild Remix #1A

Add a voice prompt for the game created in Mission 4 (Display). Either replace each `display.print()` and audibly prompt the user which button must be pressed, or include both the display and the audio.

[Video of Remix #1A](#)



Mild Remix #1B

Open the Mission 4 program (Display). Add an audio file to the if statements to indicate correct or wrong. You can use “okay” and “no” or any of the audio files.

[Video of Remix #1B](#)



Medium Remix #2A

Combine the two mild remixes into one program -- use an audio prompt for the button to press and an audio file for correct or wrong. Also, add print statements at the beginning for an intro and a print statement at the end.

[Video of Remix #2A](#)



Medium Remix #2B

Create turn signals for a bicycle. Press button L to turn left, button R to turn right. Blink to pixels to indicate the turn, and use an audio file to say the direction you are turning.

EXTENSION: use a while True: loop to use the turn signals more than once.

```
while True:
    if statement:
        Indented code for left signal
    if statement:
        Indented code for right signal

* be careful with indenting
* you will need to physically stop the code
```

[Video of Remix #2B](#)

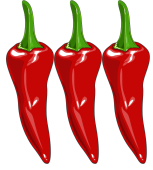
[Video of Remix #2B with Extension](#)



Spicy Remix #3A

Program each button to do a different task. For example, BTN_A could light pixels and display an image. BTN_B could display an image and play audio. BTN_U could cause pixels to blink random colors. And so forth. Make sure to use at least two audio clips.

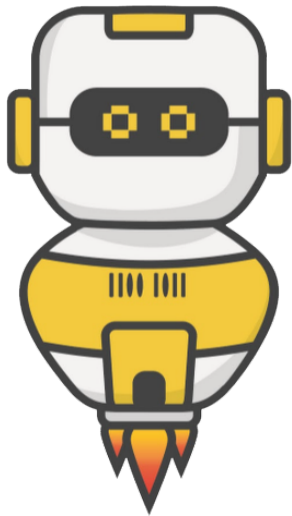
[Video of Remix #3A](#)



Spicy Remix #3B

Program at least three buttons to display a short poem and play an audio file.

[Video of Remix #3B](#)



DO THIS:

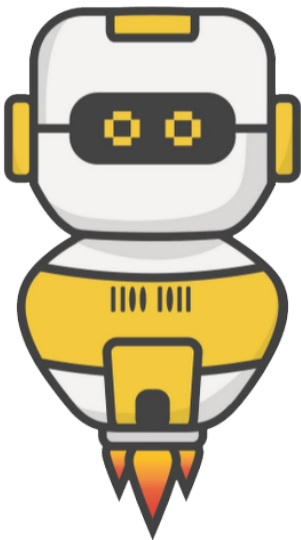
- Decide with your partner what project you will do
- Fill out the information in the Mission 5 Remix Log for [Step #2](#)

Remix Step 2: Describe what your remix project will do:

Step #3: Make a plan

Now that you have an idea for your remix, you need a plan.

- What variables will you need?
 - NOTE: You do not have to use a variable for this remix
- What colors will you use?
- What images will you display?



DO THIS:

- Fill out the information in the Mission 5 Remix Log for **Step #3**

Remix Step 3: Plan your code. What variables will you use in the project?


You do not need to fill out every line if you don't need that many variables.

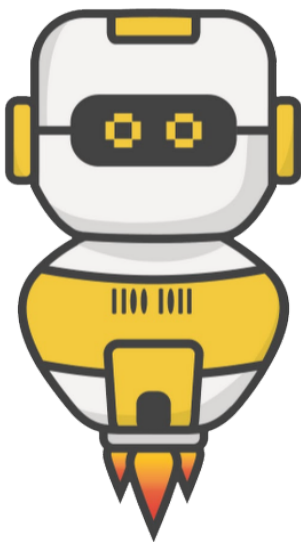
Variable Name	What it will be used for

Image to use	What it will be used for

Sound to use	What it will be used for

Step #4: Code your project

- **IMPORTANT:** In CodeSpace, go to the sandbox:  It is above the toolbox in the lower left corner.
- You can leave your projects open as a guide: **Display, Music1, Pixel1**, remix projects, etc.



DO THIS:

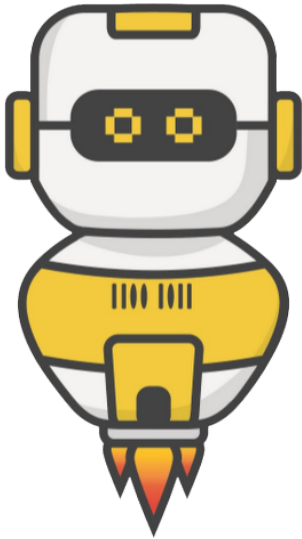
- Start with a new file and give it a descriptive name (**Remix5**)
- Import your modules
- Define your variables
- Write your code, testing frequently

Reminders!

- Don't try to write all the code at one time
- Mistakes happen, so find them early
- Type just a few lines of code and then run the program
- If there is an error, fix it before continuing
- Use the debugger and your other programs for help

Step #5: Documentation

You should always make your code readable and easy to follow.

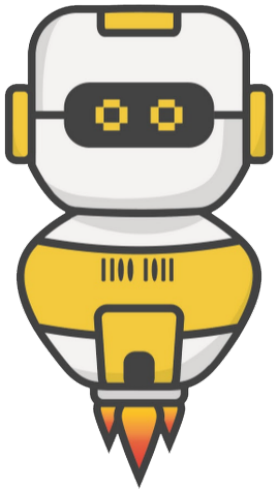


DO THIS:

- Add blank lines where needed to divide sections of code
- Add a comment at the top with your name and the name of the program
- Add a few more comments to sections of your code that explain what they do

Step #5: Get feedback

Getting feedback and reflecting on your code can help you make the program even better.



DO THIS:

- Show your code to another student
- Have him/her fill out the feedback form on your Mission 5 Remix Log
- Give yourself some feedback
- Is there something you want to change or improve or add?
- Fill out the feedback form on your Mission 5 Remix Log

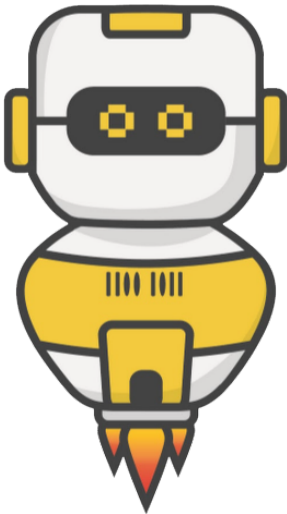
Modify your code to make your project even better

Congratulations!

Now you have your own remix!
Great job! Share your project with
your friends.

By completing this remix you have:

- learned more about programming
- used skills and concepts from the missions
- been thinking!
- and problem solving
- and much more!



DO THIS:

- Complete the Mission 5 Remix Log
- Don't forget to clear your CodeX by running your **Clear** program