



Overview

In this assignment, you'll learn to program by creating a dance party—no coding experience required! You'll start with block-based programming to make dancers move, respond to events, and create effects using AI. You'll further explore the foundations of programming by making music with code. You'll learn core programming concepts like variables, loops, conditionals, and functions through hands-on activities. By the end, you'll see how your visual blocks translate into real JavaScript code, giving you a glimpse into how professional programmers build interactive experiences.

Assignment Preparation

Make a copy of this document to record your answers. Rename your copy of the document to **lastname_programming**, where **lastname** is your last name. You will **place all answers and screenshots** in this Google Document. The steps below will tell you what to put in your answer document - use the spaces provided to insert your answers accordingly.

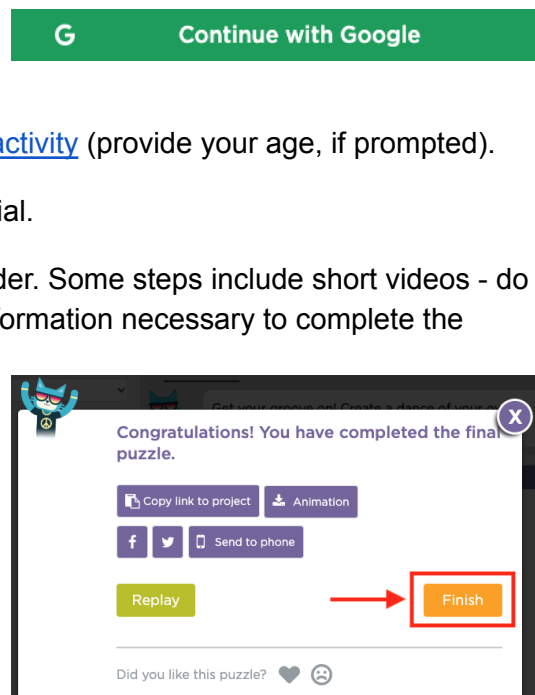
Part 1: Block-Based Programming (Dance Party)

Learning to program may sound like an overwhelming or complex task, but at a foundational level, it can be quite simple. Block-based programming provides an easy way to learn some programming basics that's fun for all ages!

Dance Party


DO THE FOLLOWING:

- [Sign into code.org](https://code.org) using your school provided Google Account.
- Once signed in, go to Code.org's [Dance Party activity](#) (provide your age, if prompted).
- Watch the video and proceed to follow the tutorial.
- Do all 10 steps of the Dance Party tutorial in order. Some steps include short videos - do not skip these! The videos provide important information necessary to complete the activity.
- For Step 10, make a simple dance party (you will make a more elaborate dance party later)



and then click *Finish*. This will bring up the Congratulations window shown here.

- Click *Finish* on the Congratulations window. This will bring you to the Certificate page.
- Enter your first and last name on the Certificate page and click *Submit* to generate your personalized certificate of completion.
- Provide a screenshot of the resulting certificate showing your first and last name.

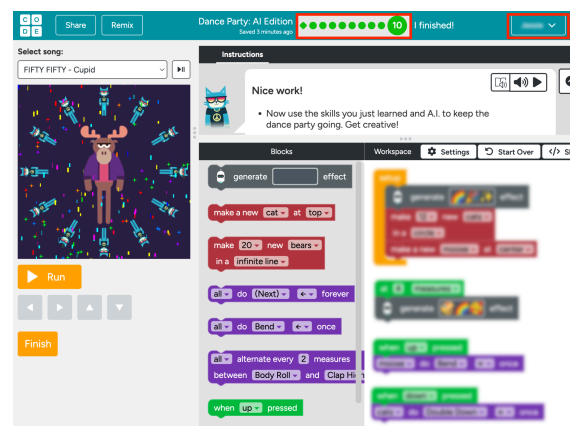
	[Insert screenshot here]
-----------------------------------------------------------------------------------	--------------------------


Dance Party: AI Edition

Combine the power of programming and AI to create a new dance party.

DO THE FOLLOWING:

- While still logged in, go to Code.org's [Dance Party: AI Edition](#) tutorial.
- Watch the video and proceed to follow the tutorial.
- Do all 10 steps of the Dance Party tutorial in order. Some steps include short videos - do not skip these.
- For Step 10, make a simple dance party using AI (you will make a more elaborate dance party later).
- Run your program and take a screenshot showing your completed code from Step 10.
- The screenshot must demonstrate that you completed all steps and must include your username.
- Provide a screenshot showing your code and username (see example).



	[Insert screenshot here]
-------------------------------------------------------------------------------------	--------------------------

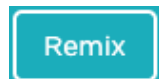
Part 2: The Final Dance - Testing & Refining

When programming, it is best to have a vision of the final product (i.e. the final dance), but to program and test one step at a time. Often as you program, your vision may change - that is expected, it is called *agile programming*.

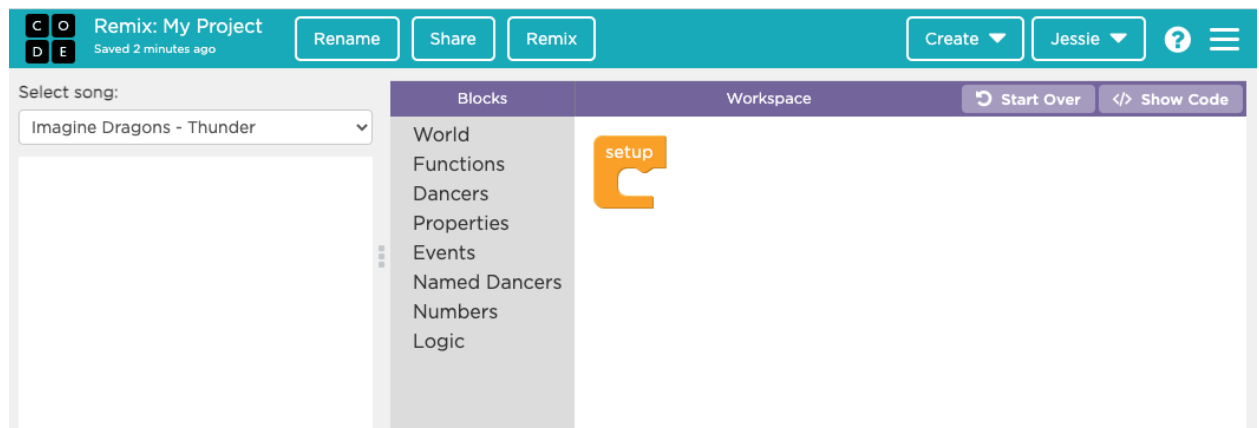
▶ Watch the video on [Systematically Testing & Refining: Dance Party Example](#) [4:59]

DO THE FOLLOWING:

- Return to [Step 10 of the Dance Party: AI Edition](#) tutorial and click the Remix button at the top left of the screen to create a saved project.



- Clear all blocks by dragging them from the workspace back to the blocks column to delete them, keeping only the *setup* block.



- Drag blocks into the blank workspace to program your own original Dance Party!
- Your final program should meet the following criteria (be sure to test and refine your program as you build):
 - ☐ At least 12 blocks, with at least 5 different kinds of blocks (more are encouraged).
 - ☐ Use the AI generate effect block at least 2 times
 - ☐ Use at least 2 different dancers
 - ☐ Change at least one property of a dancer.
 - ☐ Use at least 2 event blocks
 - ☐ Use at least 1 repeat block
 - ☐ Be creative and have fun!

- When you are done with your dance party program, click the Share button at the top left of the screen.



→ When prompted, click the *Copy link to project* button.

→ Paste the link to your original Dance Party program below:

 Copy link to project

Link:	
-------	--

Part 3: Connection & Reflection

In addition to systematically testing and refining your program, it is also important to get feedback from your users and consider how you might incorporate it into your program.

DO THE FOLLOWING:

→ Show your final Dance Party program (from Step 2 above) to at least one other person and ask them to suggest a change based on one of the following:

- number of dancers
- size of the dancers
- music selection
- Background

→ What is the suggested change?

--

→ How would you incorporate their feedback to make the suggested change to your program?

--

→ In 2-3 sentences, explain the steps you took to systematically test and refine your program (use specific examples from your program in your response).

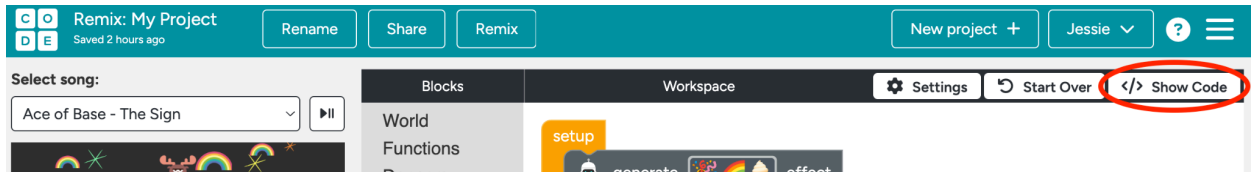
--

Real World Connection

The Dance Party program that you wrote used the Blockly programming language. Blockly is an easy-to-use way to generate code in the Javascript programming language, which is the most widely-used programming language in the world. Javascript programming in the real world does not use blocks; it uses text-based programming.


DO THE FOLLOWING:

- With your Dance Party program still open, click on the *Show Code* button in the upper right corner to view the text-based Javascript code.



⚠ If your program is no longer open, go to <https://studio.code.org/projects> to view a list of your projects to re-open it.

- Provide a screenshot showing your Javascript Dance Party code.

 [Insert screenshot here]

- In 3-5 sentences, explain in your own words what you see in the code, including how Blockly relates to Javascript. You don't have to understand the code, but you are to look at it and find something that you recognize from your program and mention that in your answer.

Part 4: Programming in Music Lab

Explore foundational programming concepts through music!


Jam Session Tutorial

DO THE FOLLOWING:

- While still logged in, go to Code.org's [Music Lab: Jam Session](#) tutorial.
- Watch the video and proceed to follow the steps of the tutorial.
- Do all 17 steps of the Jam Session tutorial in order.



- For Step 17, experiment with what you learned to create a simple song (you will make a more elaborate version later).
- Run your program and take a screenshot showing your completed code from Step 17.
- The screenshot must demonstrate that you completed all steps and must include your username.
- Provide a screenshot showing your code and username (see example).

	[Insert screenshot here]
-----------------------------------------------------------------------------------	--------------------------

Make Music with Code

 Watch [Make Music, Learn Code](#) [2:02 min]

DO THE FOLLOWING:

- While still logged in to code.org with your Google Account, go to Code.org's [Music Lab Expansion Tutorial](#).
- Do all 15 steps of the Music Lab Tutorial in order (some steps are repeated from the Jam Session tutorial, that's ok - complete them anyway).
- For Step 15, continue on to make a more elaborate song using what you learned from both tutorials. You can recreate an existing song or mix some beats to create something completely new - be creative and have fun!
- Give your music program a name by clicking the Rename button at the top left of the screen, enter a name, and click Save.

Rename
- Your final music program should meet the following criteria (be sure to test and refine your program as you build):
 - ☐ At least 20 blocks, with at least 5 different kinds of blocks (more are encouraged).
 - ☐ 5 different functions (i.e. intro, verse, chorus, etc.)
 - ☐ Use the play together block at least 2 times
 - ☐ Use the repeat block at least 1 time
 - ☐ Use an AI generated beat at least 1 time
 - ☐ Add at least one trigger

Share

- When you are done with your music program, click the Share button at the top left of the screen.
- When prompted, click the *Copy link to project* button.
- Paste the link to your music program below:

Link:	
-------	--

Part 5: Core Programming Concepts

In Musics Lab, you solved a computational problem by using the computer to program your own dance party and original song. Make connections from what you have learned in code.org's Music Lab activities to the core programming concepts below.

DO THE FOLLOWING:

Concept #1 - Algorithms

▶ Watch: [What is an Algorithm and Why Should You Care?](#) [5:27 min]

📖 Read: [What is an Algorithm?](#)

- Reflect on your original music program from above.
- Describe the algorithm you used in your own words by explaining the step-by-step process your program follows from start to finish.

--

- What happens if you rearrange the order of your music blocks? Does the sequence matter? Explain why.

--

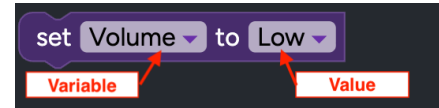
- Think about a simple everyday task you do (like making a sandwich or getting ready for class). Explain how that task is similar to the algorithm in your program.

--

Concept #2 - Variables

📖 Read: [What is a Variable?](#)

- One example of a variable used in Music Lab is the Volume. The Volume can get different values like Full, Medium, or Low.



- Provide another example of a variable and its value that you used in Music Lab.

Variable: (ex: <i>Volume</i>)		Value: (ex: <i>Low</i>)	
-----------------------------------	--	-----------------------------	--

- Explain what setting the variable to the value does in your program.

Concept #3 - Loops

📖 Read: [What is a Loop?](#)

- Describe an example from Music Lab where you used repetition (a loop) in your music program.

- If a chorus repeats 4 times, is it better to copy-paste the code 4 times or use a loop? Why?

Concept #4 - Conditionals

📖 Read: [What is a Conditional?](#)

- Describe how you used a trigger in your music program. Explain how this relates to the “if-then” concept of conditional decision making?

Concept #5 - Functions

 Read: [What is a Function?](#)

- Describe how you used a function in your music program? Include the function name, what the function does, and how you called the function to use it in your program.

--

- If you were to develop a music app (like Spotify), what functions might you create in your music app program? List 3 example functions and briefly describe their purpose.

	Function Name	Purpose
1		
2		
3		