



# How to Code a Dress-Up Game in Scratch

Have you ever used an online dress-up game to make an avatar? Did you know that every single accessory had to be programmed separately to appear on que, fit with the other options, and disappear when replaced? These seemingly-simple games are full of the same basic coding concepts that are used in complex applications, from the app you're using to read this PDF to the computers that help launch rockets into space!

**Put your learners' creativity to the test with this intro-to-coding activity.** Using our provided assets and Scratch, a free browser app by MIT, all your students need is a computer, internet access and this guide to program their own game.

Since Scratch is designed for ages 8-16, this activity is easy to adapt to suit any grade level. Try adding it into a research project, or have your students use what they learned to make something new!



## HANDS-ON STEM EDUCATION

For over 30 years, PCS Edventures has inspired students to develop a passion for Science, Technology, Engineering and Mathematics (STEM), focusing our efforts on making learning and discovery a fun and interactive process for grades K-12.

- Classroom
- After-School
- Home Learning

# How to Implement

## Remote

Scratch is compatible with all modern browsers on computers and tablets (except for Internet Explorer), so this activity is best for online classes where students have device access.

Try breaking your students into groups. In each group, assign one student control over the Scratch game. Then, with that student sharing their screen, the group can work together to understand the steps and troubleshoot problems.

You may also choose to complete the game as a class. Share your screen, or choose a student to share their screen. Then edit the Scratch file based on the class's feedback.

## In-Person

Depending on your preferences and the number of devices available, you may decide to complete this activity together as a class, break the students into groups, or have students complete this activity independently.

Try printing out these instructions for each student so that they can look at both the instructions and the game at the same time.

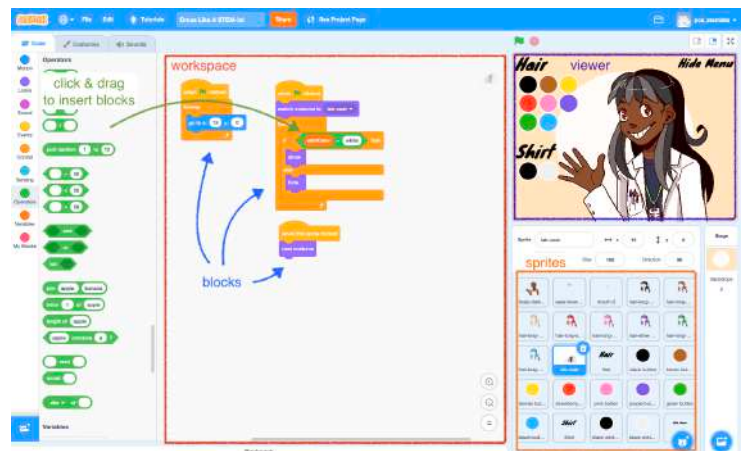
# Resources

Scratch has a number of resources to help you get started, including:

- [Help setting up a teacher account](#)
- [Additional educator resources](#)
- [Scratch Tutorial](#)

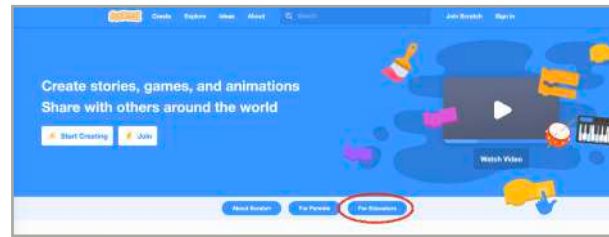
# Vocabulary

- **Sprite:** An image that you can use in your scratch game.
- **Block:** A statement that tells your Scratch game what to do.
- **Workspace:** The blank space in the center of the screen where you place your code blocks.
- **Viewer:** The window in the top-right of your screen where you can preview your game.



## Teachers: How to Make a Scratch Account for Yourself and Your Students

1. Visit [scratch.mit.edu](https://scratch.mit.edu). Click on the For Educators button on the home page.



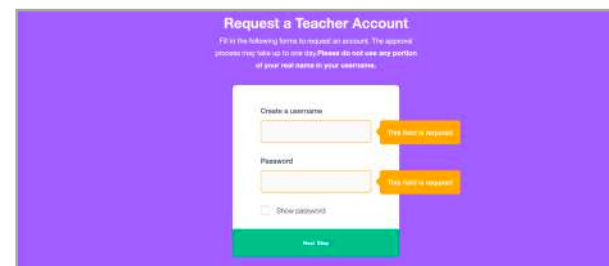
2. Click on **Teacher Accounts**.



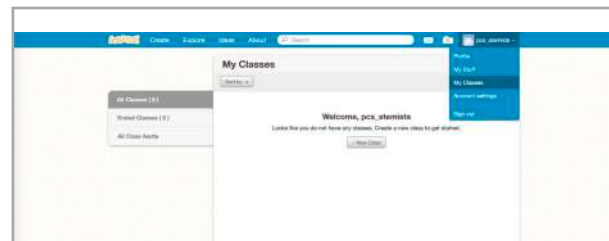
3. Choose **Request Account**.



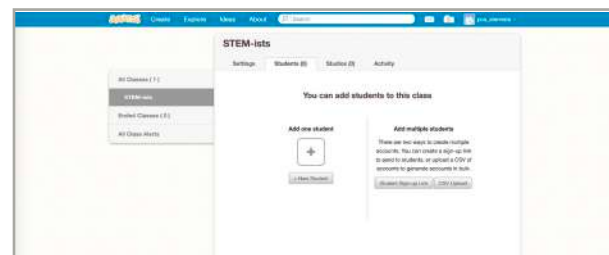
4. Complete the application process. Once finished, check your inbox for an email from [no-reply@scratch.mit.edu](mailto:no-reply@scratch.mit.edu) and confirm your email address.



5. Once your teacher account is approved, create a student account for each student or group. Navigate to the **My Classes** page, then create a new class.



6. Click the **+ New Student** button to add students individually, or use the student sign-up link feature to have students create their own accounts. Visit the [Scratch Teacher Guide](#) for more information on adding student accounts.

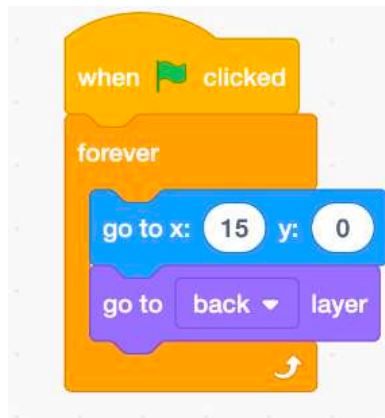


# How to Make a Dress-Up Game with Scratch

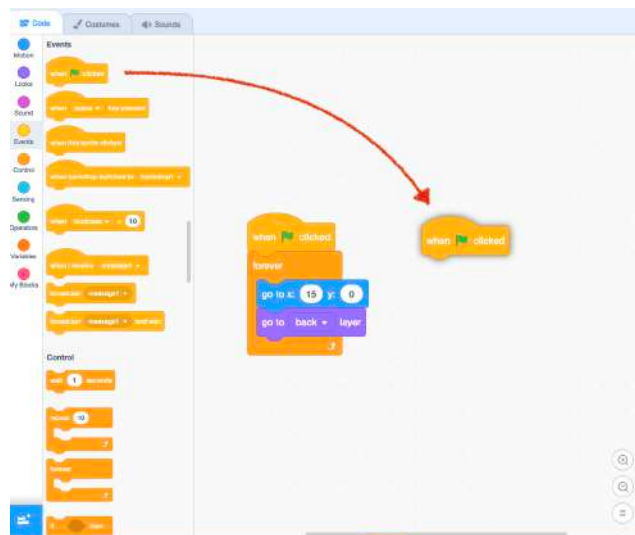
After you log into your Scratch account, follow these instructions to make your own STEM-themed dress-up game!

1. Visit the Dress Like A STEM-ist starter game. Click on **Remix** to create your own copy of the game. (It might take a few minutes to load.)

2. The **sprites** window is in the bottom right. Select the **body** sprite. You should see this block of code in your workspace.

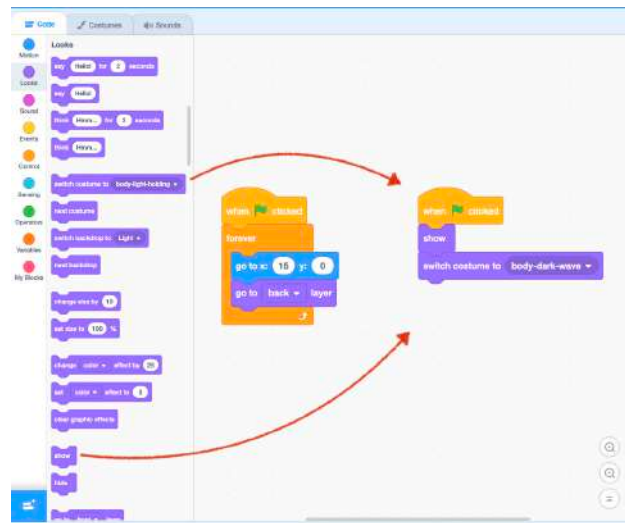


3. In the left menu, click on the **Events** tab. Drag the "when flag clicked" block into your workspace. You can place it anywhere in the white space.



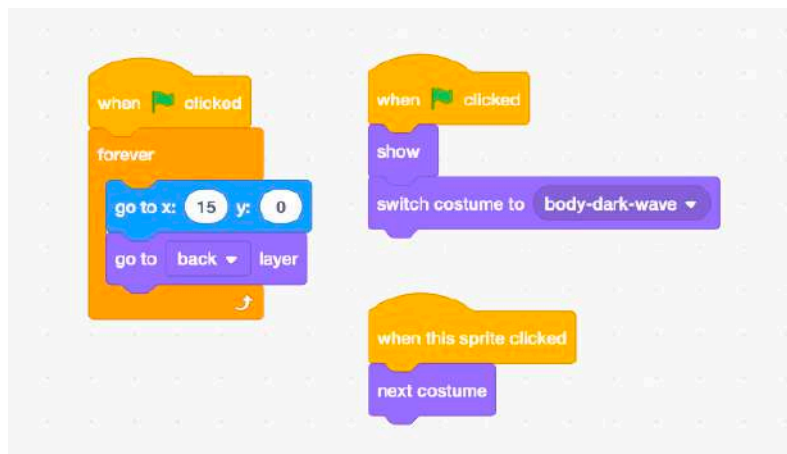
4. Click on the **Looks** tab. Drag the "show" block into your workspace and connect it to your first block.

5. Connect the "switch costume to" block to the "show" block. Choose which costume you want to see every time the game is started.

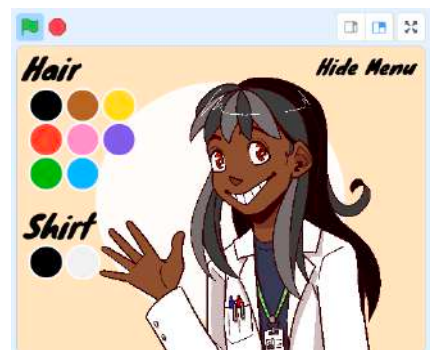


6. Go back to the **Events** tab. Drag the "when this sprite is clicked" block into your workspace.

7. From the **Looks** tab, connect the "next costume" block to the previous block. Your workspace should look like this:



**Check your work:** Click on the green flag. Then, in the player, click on the body sprite to see the hand and skin change. After you have seen all of the bodies, click the red stop sign.



8. Repeat steps 2-7 for the eyes and the mouth. Then, check your work with the green flag. There are 6 different eyes and 3 different mouths.

### How does this work?

In the **sprites** window, select the **body** sprite. Then, click on the **Costumes** tab in the top-left. Click on each costume and see what happens in the viewer! When you are done, return to the **Code** tab.

9. In the **sprites** window, select the black hair sprite. Then, drag the “when *flag* clicked” block into your workspace.

10. Go to the **Variables** tab. Connect the “set *hairColor* to 0” block to the “when *flag* clicked” block. Delete the 0, then type **black**.

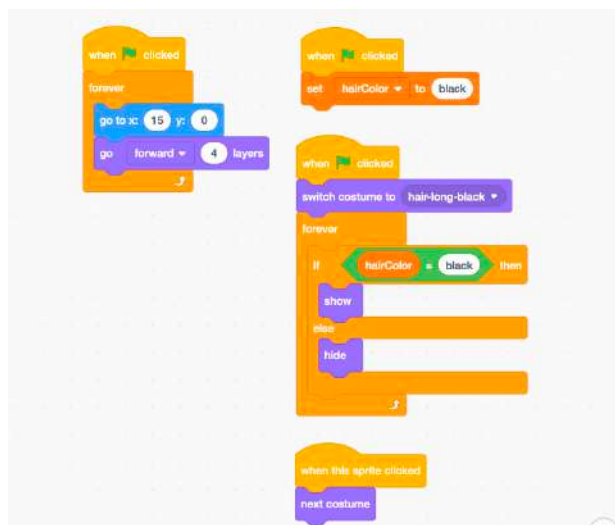
11. Drag another “when *flag* clicked” block into your workspace, then go to the Looks tab. Connect the “switch costume to” block and choose “*hair-long-black*” in the drop-down menu.

12. Click on the Control tab, then connect the “forever” block. Place an “if... else” block inside that block.

13. Under the *if* part, connect a “show” block. Under the else part, connect a “hide” block.

14. Go to the **Operators** tab. Drag a “\_=50” block into the white circle next to the *if*. Then, in the **Variables** tab, drag the “*hairColor*” variable into the operator block, and replace the “50” with the word **black**.

15. Return to the Events tab and drag in a “when *this sprite is clicked*” block. From the Looks tab, attach a “next costume” block. Your workspace should look like this:

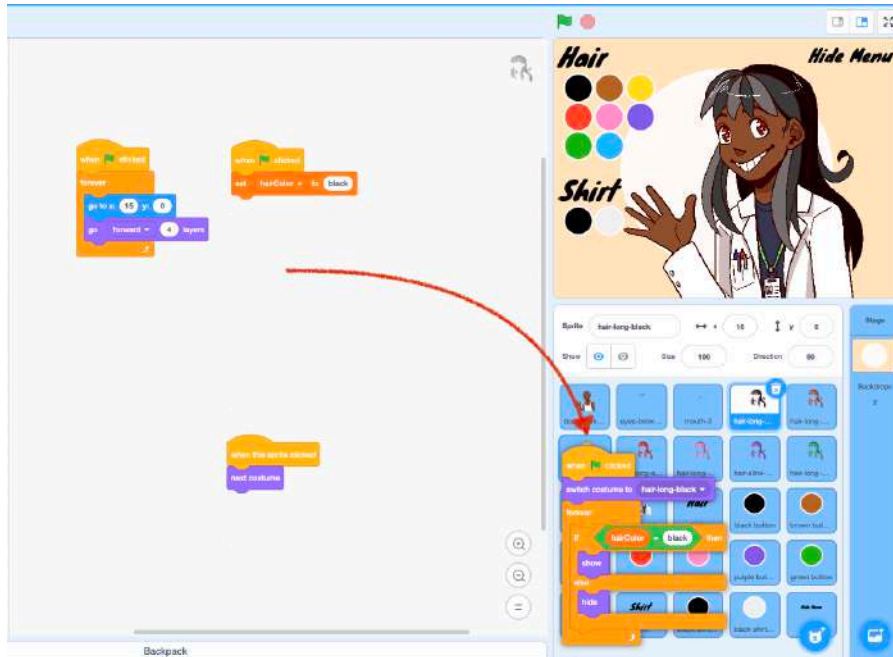


**Check your work:** Click on the green flag. Then, in the player, click on the hair sprite to see the hairstyle change. After you have seen all 6 hairstyles, click the red stop sign.



16. Repeat steps 9-15 for each hair color you want. Ask your teacher how many colors to include, then delete any extra sprites.

Tip: You can copy a code block from one sprite to another! Drag the block of code into the sprite window and drop it onto another sprite.



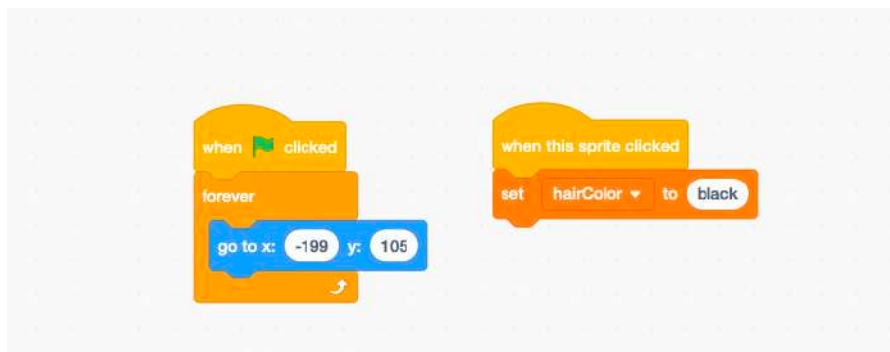
**How does this work?**

Just like before, we used costumes to change the hairstyle when the sprite is clicked on. This time, we also added a variable. **Variables** are like containers where we can save data to use again later. We'll use these variables again in the menu!

17. Next, we'll make a menu that includes all of the hair colors you chose earlier. Select the black button sprite.

18. From the **Events** tab, drag in the "when this sprite is clicked" block.

19. From the **Variables** tab, connect a "set hairColor to black" block to the previous block.



20. Copy this code block to the next color button. Then, change "black" to match the button's color.
21. Repeat step 18 for every color you included, then delete any unused sprites.

**Check your work:** Test your menu to make sure that each button changes the character's hair color. Don't forget to make sure that clicking on the hair still changes the style!

22. Repeat steps 9-20 for the shirt sprites and shirt menu, using the *shirtColor* variable instead of the *hairColor* variable.
23. Test your game. Try out every feature to make sure all the sprites appear as they should. Go back and check your code if you find any problems.

## CONGRATULATIONS!

# You've just finished a computer program!

## Extensions

### Extension 1: Hide the Menu

Goal: When the Hide Menu sprite is clicked, hide all of the menu options.

Use the `hideMenu` variable and `if... else` statements to complete the goal. Don't forget to hide the headers!

### Extension 2: Add New Sprites

Goal: Add new sprites to make the character look more like you.

Draw a new sprite that will fit on this character. Maybe you'll draw another eye shape, a new outfit, or a different hairstyle. Then, upload your sprite and code it to appear in the game!

Tip: You can draw digitally with free apps like Adobe Illustrator Draw, FireAlpaca or MediBang Paint. Or, you can take a picture of your physical drawing and remove the background with a tool like GIMP or Remove BG.

### Extension 3: Improve on the Game

Goal: Make a new dress-up game that is an improvement on the original.

Every app can be improved! First, brainstorm ways that this game could work better. What did you struggle with when playing it for the first time? What features do you wish it had? Did you find any bugs? Once you know what you want to improve, use our sprites or create your own and redesign the dress-up game from scratch! (Get it?)



## Discussion Questions:

1. A STEM-ist is someone who works in a science, technology, engineering or mathematics field. What job examples can you think of for each field? (Tip: Visit our [STEAM Career Exploration](#) blog series for examples!)
2. Computer programmers often use pseudocode to plan out their programs before they code them. Look up a definition and some examples of pseudocode. Can you write your Scratch game as pseudocode?
3. Now that you have created your first program, you're officially a computer programmer! Do some research to answer these questions:
  - What are some examples of common programming languages?
  - How do computer programmers learn new languages?
  - What are some specific examples of computer programming careers?



## Featured Product

This STEM activity was inspired by the activities found in [Scratch Camp](#).



Explore the magic of computer programming with *Scratch*, a free software developed by MIT to encourage young learners to combine creativity with logic! With a cutting-edge application and a series of exciting projects, students uncover the sensational world of coding. Every day, participants expand their skills as they build their very own interactive storyboards and games with audio, video, animated characters and more.

## Interested In More Art/STEAM Activities?



[Discover Blocksmith: 3D Coding & Design](#) takes game design to the next level! Using the Blocksmith Builder software either in the classroom or at home, learners create 3D games and experiences that can be played on desktops, laptops or VR devices.. The step-by-step curriculum quests guide students and educators of any experience level through each scaffolded lesson.



Calling all programmers and musicians! [Rockin' Robots](#) is your chance to use Sphero indi robots on a guided tour through the STEAMtastic fusion of music and robotics. If exploring songs, crafting instruments, investigating sound effects and designing a melodic coding challenge sounds fun, then this is the camp for you!



For more information, visit: <https://edventures.com/collections>  
or contact a STEM Program Specialist at (800) 429-3110

**PCS** edventures!™  
Experts in Hands-On **STEM** Education

