**AP CSP CodeX**

|  |  |  |
| --- | --- | --- |
| **LESSON: RGB Colors** | | **Time: 45 minutes** |
| **Project Goal:** Students will learn about RGB lights in the NeoPixels and how to control the amount of each light.  **Learning Targets**   * I can identify the Red, Green and Blue values of a color using color-picking software. * I can define a color triplet for a specific color. * I can define a function that sets the NeoPixels to a specific color using RGB values. * I can use random numbers to generate a random color. | **Key Concepts**   * NeoPixels are made up of 3 lights: red, green and blue. Each light has a brightness level that combines together to create color. * A color can be defined by its RGB values, and those values can be used in code. * Python includes a random library with functions that generate random numbers. * The randrange() function in the random library generates a random integer between 0 and one less than the range in parenthesis. | |
| **Assessment Opportunities**   * RGB Colors Activity Guide * Pixels1\_RGB program | **Success Criteria**   * Create a function that sets the NeoPixels to a color using an RGB triplet. * Create a function that sets the NeoPixels to a random color using the randrange() function. | |
| **AP CSP Framework**  **AAP-3.A** Write statements to call procedures.  **AAP-3.E** Write expressions to generate possible values, and evaluate expressions to determine the possible results.  **Computational Thinking Practices 1.D** Evaluate solution options.  **Computational Thinking Practices 4.C** Identify and correct errors in algorithms and programs, including error discovery through testing. | **Materials**   * RGB Colors Activity Guide / Answers * Color-picking software: <https://htmlcolorcodes.com/color-picker/> * Solution code for Pixels\_RGB * AP CSP CodeX Vocabulary List * AP CSP CodeX Python Code List * Unit 1 Review Links and Test Questions | |
| **Teacher Notes**   * This lesson will have students modify the program created in Defining Functions by defining more functions using RGB triplets. * This lesson will be completed on the computer, using CodeSpace for programming. Use the Sandbox in CodeSpace for programming. This lesson is not part of a mission. * Students should do a “Save As” with each program to keep the original code. * The assignment can be distributed digitally. Space is provided for students to take notes while programming. * Students will modify a previous program. The best experience will come from them modifying their own code. However, we want all students to be engaged, so you can give them the original code to modify if needed. * If you are giving code to students, use the solution code found there: Pixels1\_functions (Defining Functions) * Students will need to use software to pick a color and identify its RGB values. A website is recommended, but you can use any website that lets students pick a color and see the RGB values. * Follow the slides for instructions and guidance. * Slide #3 includes a short video on RGB colors. It is embedded in the PowerPoint. However, if you need to access the video separately, a link to the YouTube video is also in the speaker notes. * Solution code for the final program is provided. * At the end of the lesson, remind students to clear the CodeX before turning it off. * Another suggestion for assessment is for students to keep a daily journal, or use a reflection form for students to process information they learned and reflect on questions they may still have. * You can also add vocabulary to a word wall and keep a document or chart of the Python code learned during each mission. * The teaching guide (below) gives the narration for one way to present the lesson. | | |

**Teaching Guide**

**Warm-up (5 minutes)**

Review the last two lessons with students: Mission 3 Pixels and Defining Functions.

Then pass out the activity guide to students. It is best distributed and completed digitally, but it can also be printed.

**Learn about RGB colors (10 minutes)**

💡 **Teaching tip – warm-up**

* Go through slides #1-4
* Slide 3 has a short video from Code.org on RGB colors. You can show the video embedded on the slide, or use the YouTube link. If you cannot show YouTube videos in your classroom, the video can also be found on other digital platforms. Or you may need to download it to show it. You can also find a different short video on RGB colors. There are many good ones easily available.
* Students fill out notes on their activity guide.

💡 **Teaching tip – warm-up**

* Go through slides #5-7
* Students take notes as needed on their activity guide.

**Programming Activity (15 minutes)**

💻 Students will work in pairs or individually at their computers.

💡 **Teaching tip:**

Part 1. You can go over the instructions with the students, or you can use the slides (slides 8-11).

* Students are asked to modify their code from “Defining Functions”. They will add another function for an RGB color. Students should use their own code. However, if you want to give them code to use, use the solution code for Pixels1\_functions (Defining Functions).
* Students should run the code and make sure it is error-free before continuing.
* Students can take notes in their activity guide as needed.

💡 **Teaching tip:**

Part 2. You can go over the instructions with the students, or you can use the slides (slides 12-13).

* Students will create another function for another RGB color. Little instruction is given here.
* Students should run the code and make sure it is error-free before continuing.
* Students can take notes in their activity guide as needed.

💡 **Teaching tip:**

Part 3. You can go over the instructions with the students, or you can use the slides (slides 14-17).

* Students will create another function for a random color. Students must import the randrange() function from the random library.
* Students should run the code and make sure it is error-free before continuing.
* Students can take notes in their activity guide as needed.

**Programming Challenges (15 minutes)**

💻 After the programming exercises, students can try more variations to the program by completing the challenges. These are optional, so if you don’t have time, that is okay. But if you do have time, this gives the students additional experience.

💡 **Teaching tip:**

The challenges can be completed in any order, and they don’t all have to be completed. The solution code for this activity includes a function for each challenge as an example. Each student program will be different.

✅ You decide what want you want students to turn in for a grade, and how they turn it in: activity guide and/or pixels1\_RGB program. You can look at the code on student computers, or have them submit code. The assignment document can also be turned in.

**Wrap-Up (5 minutes)**

Students answer two questions in their activity guide. You can also have a class discussion on the questions, or have students share their answers.

Formative Assessment:

* Daily reflection journal
* Class discussion on what they learned about functions and RGB colors
* Activity Guide completion
* Pixels1\_RGB program
* Exit ticket

**SUCCESS CRITERIA:**

* Use RGB triplets to control LEDs
* Use random integers in a program
* Correctly define a function
* Understand how and when to call a function
* Define and call one or more functions in a program