**3.03 Show Me the Code!**

**Step 1**

**​**Start off your project the same way you did with 1.04 and 2.02: head to the [Scratch website](https://scratch.mit.edu/) or your [desktop Scratch app](https://scratch.mit.edu/download) and login. Then, click "Create" to start a new program.

Graphical user interface, application

Description automatically generated

**Step 2**

Since this is a quiz, it makes sense to start with the questions! Decide on three simple questions for a chosen topic.

**Example:** What is the capital of Georgia? Answer: Atlanta

**Hint:** Having questions with very short or even one-word answers will reduce the number of errors you’ll need to fix, as well as the troubleshooting you’ll need to do, later on!

**Step 3**

Armed with your questions in hand and a new program queued up, add a new sprite that will host the game.

**Hint:** Here’s a chance to show off your creativity! Try to choose a fun sprite to be a host and change up your background to match.

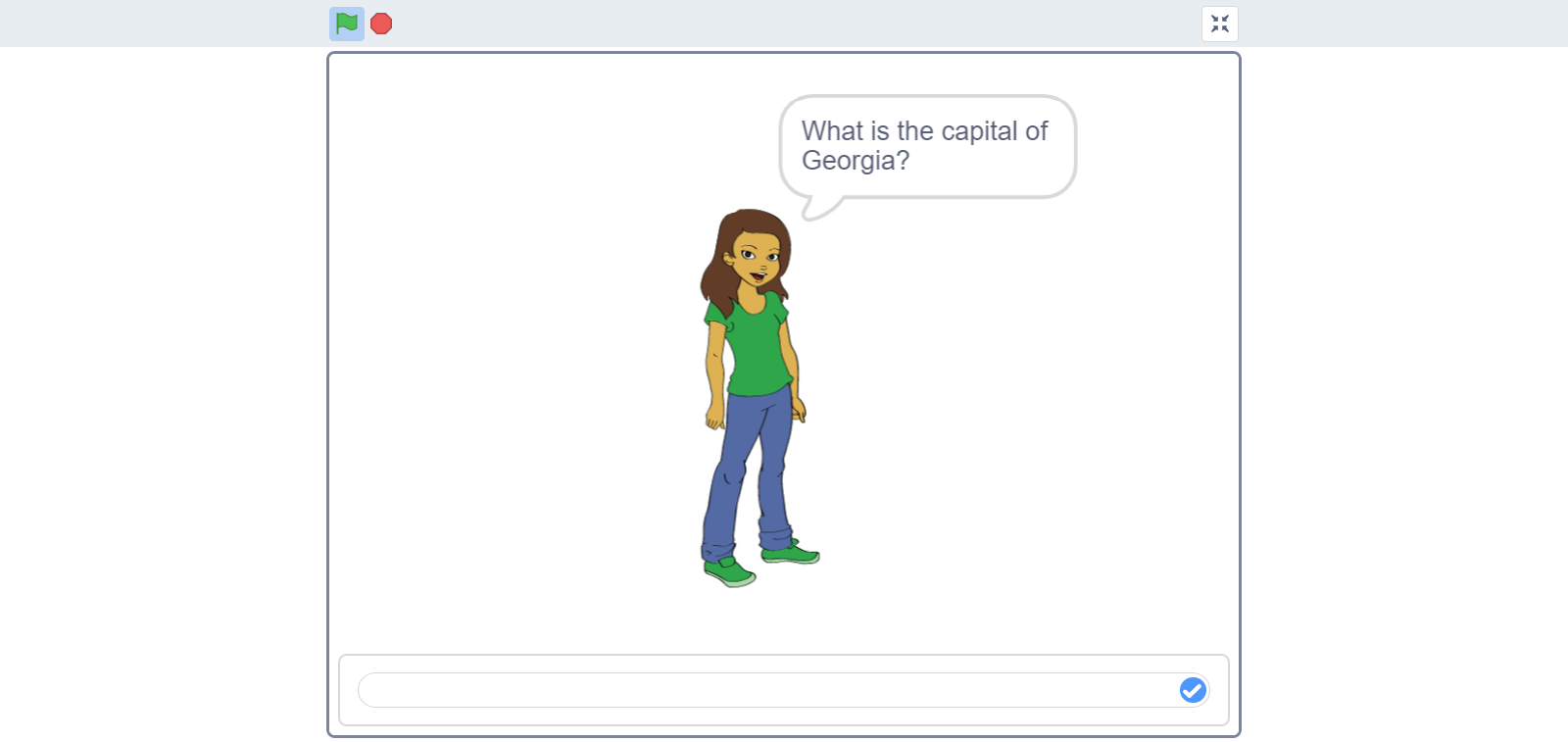
**Step 4**

Use the following steps to write the code for your game.

1. Start the program by clicking on the green flag
2. Trivia quiz host asks question number 1.

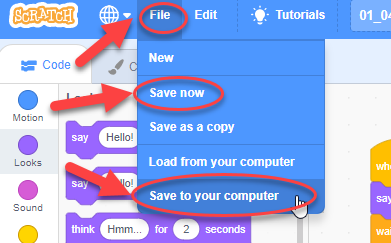
Example: What is the capital of Georgia?

1. User enters a response into the input box.
2. User input is stored in a variable called: **answer**.
3. Check to see if the user’s response is the correct answer. Note that the answer will need to be with correct capitalization.
   1. If correct, the quiz says: Good job. That’s correct.
   2. If not correct, the quiz says: That is incorrect. The correct answer is Atlanta.
4. Repeat steps 3–6 for until you have three questions or more.



**Step 5**

Save your project and download the file to submit to your instructor using the **File** menu.



**Optional Coding Challenge!**

* Can you add a scoring system?
* Would you like to remove points for incorrect scores?
* Would you like to add sound effects if the player achieves a score of 10?
* Would you like to add sound effects if the player gets the answer wrong?
* What else can you add to your quiz game project?

**Grading Rubric**

|  |  |  |  |
| --- | --- | --- | --- |
| **Categories** | **Excellent** | **Good** | **Needs Improvement** |
| **Program Design** | 11–20 pts.  Work contains**all** required elements:   * includes one sprite that is not the default sprite * includes a background | 6–10 pts.  Work contains **most** of the required elements:   * includes one sprite that is not the default sprite * includes a background | 0–5 pts.  Work contains **few** of the required elements:   * includes one sprite that is not the default sprite * includes a background |
| **Program Content** | 11–20 pts.  Work contains**all** required elements:   * includes a variable to record user input * includes conditional statements to determine what to do with the user’s answer * includes feedback if the answer is correct or incorrect | 6–10 pts.  Work contains **most** of the required elements:   * includes a variable to record user input * includes conditional statements to determine what to do with the user’s answer * includes feedback if the answer is correct or incorrect | 0–5 pts.  Work contains **few** of the required elements:   * includes a variable to record user input * includes conditional statements to determine what to do with the user’s answer * includes feedback if the answer is correct or incorrect |
| **Program Performance** | 8–10 pts.  Work contains**all** required elements:   * runs correctly * contains no syntax or runtime errors * produces the expected results | 4–7 pts.  Work contains **most** of the required elements:   * runs correctly * contains no syntax or runtime errors * produces the expected results | 0–3 pts.  Work contains **few** of the required elements:   * runs correctly * contains no syntax or runtime errors * produces the expected results |
| **Total points: 50** |  |  |  |