Build a simple quiz game in Scratch

A digital solutions project for Year 3-4 students

Let's build a quiz game!

Let's build a simple quiz game using drag-and-drop coding blocks in Scratch! This is a great digital solutions project that suits Year 3-4 students, and teaches core computational concepts such as branching, user input and variables.
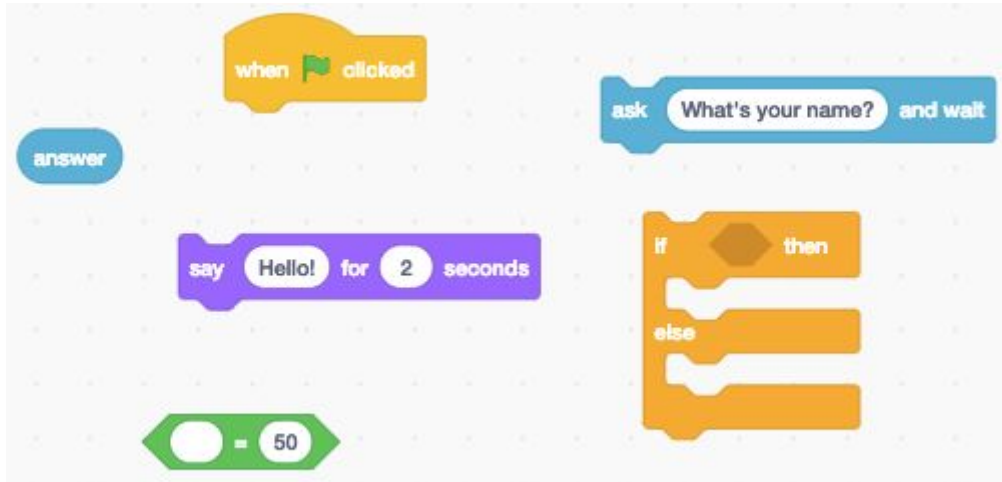

Digital Technologies Year 3-4
Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input (ACTDIP011)

Hello. Welcome to the capital cities quiz!
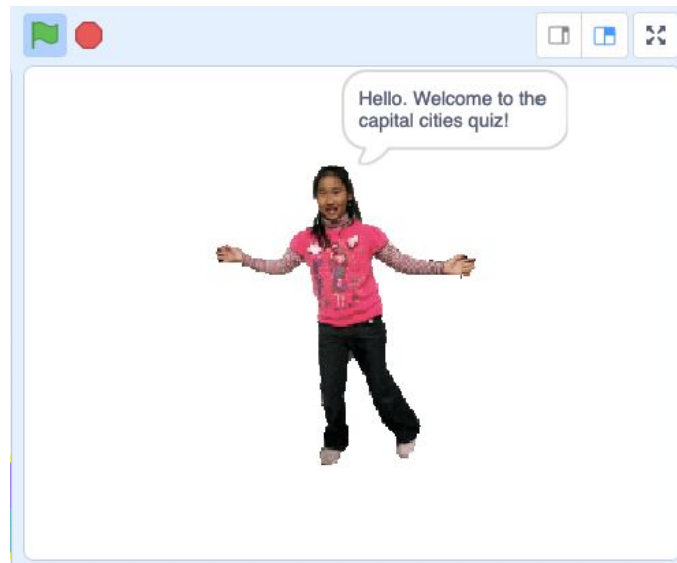
PLAY THE GAME >

What do I need to build a quiz game?

This is quite a simple game to build. You only need the following 6 coding blocks to build it:

*The six blocks you'll need to make a quiz game!*

You will also need to know a few things before getting started. Make sure to check out our blog post for a comprehensive guide on:
- The concepts that underlie a quiz game
- How to write pseudocode for your game

Making the quiz game

Step 1 & 2

When we start our code by clicking the green flag, we want our Quizmaster to say, "Hello. Welcome to the capital cities quiz!" We can do this by using the following blocks in the following way:
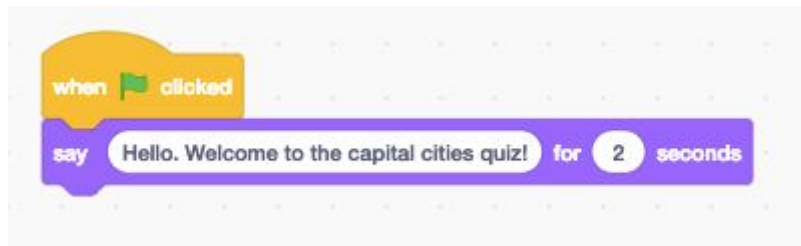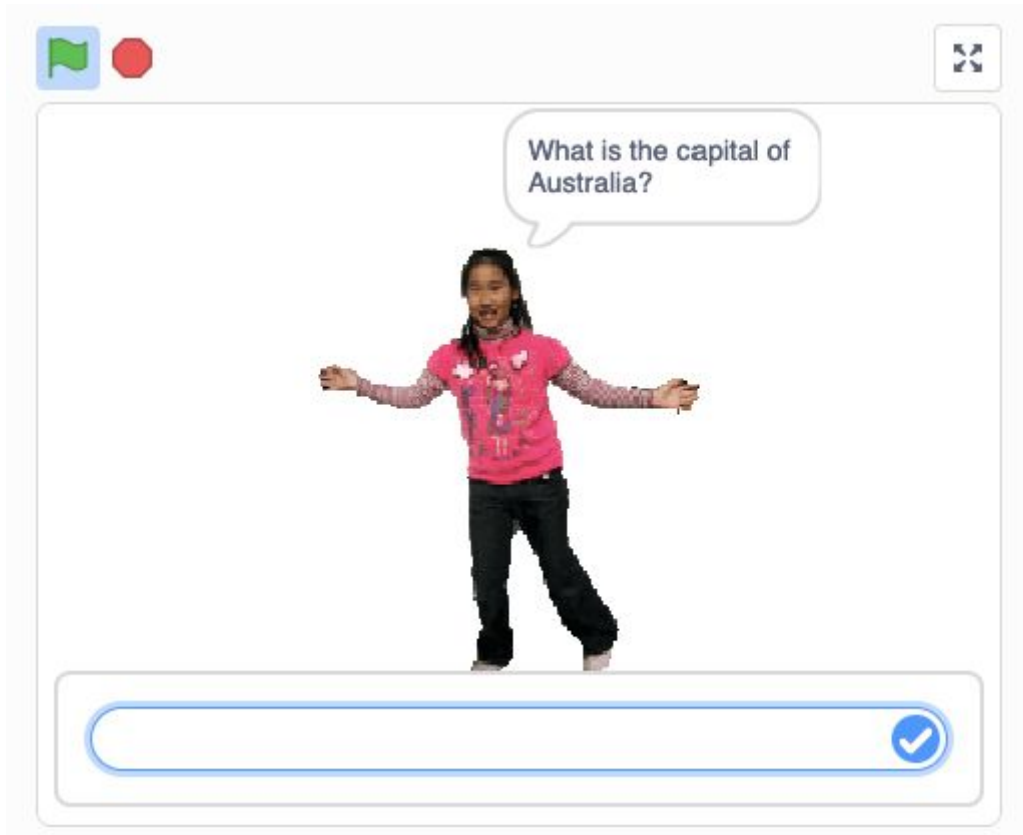

Our Quizmaster's introduction.

Step 3

Next, we want our Quizmaster to ask the first quiz question. We can use the "ask" block (and not the "say" block) to do this. The "ask" block creates an input box into which the user can type their answer to a question, like so:

Our Quizmaster's first question.

The aforementioned code will produce the following result:

Users can type their answer into the input box.

Step 5

The user's answer to a question is automatically stored in a variable called "answer", which looks like this:

The answer variable.

Step 6

Now we want to check to see if the user's input (i.e. the information contained within our "answer" block) is the same as the actual correct answer. In other words, we want to make a **conditional statement** to see whether the user's answer is correct or not.

Creating a conditional statement

Let's say our Quizmaster asks, "What is the capital city of Australia?"

We want to check that the user's input is indeed "Canberra". We do this by using the "equals" block, like so:

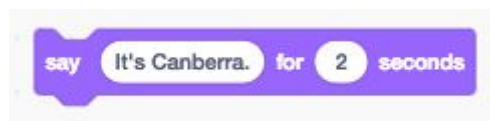*We can use the equals block to check that the user's answer matches the correct answer.*

Using branching

If the user's answer is correct, we want our Quizmaster to say, "That's correct. Well done."


If the user's answer is correct, we can let them know using the "say" block.

If not, we want our Quizmaster to tell us the correct answer.

If the user's answer is incorrect, we can tell them the correct answer.
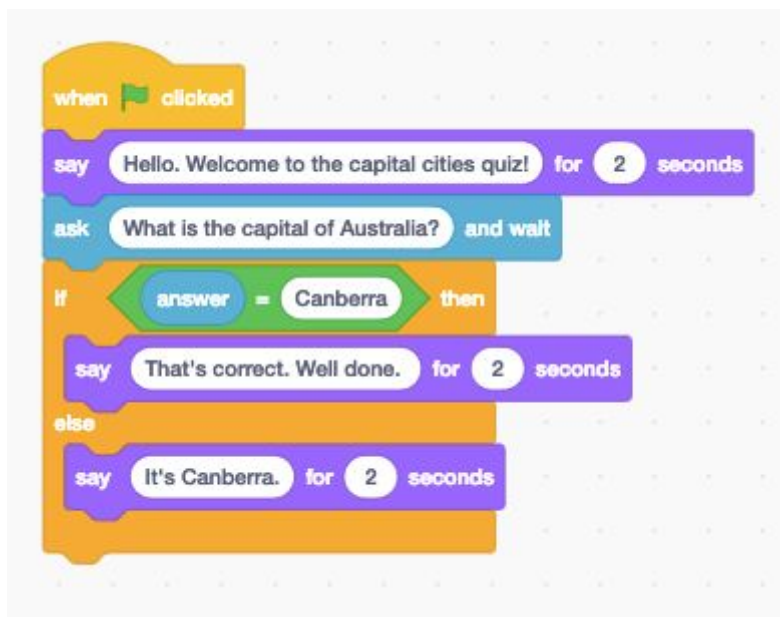
Combining conditional statements and branching

We can use an **if-then-else** block to combine our conditional statement with our two branching options to create a fully-functioning block of code:
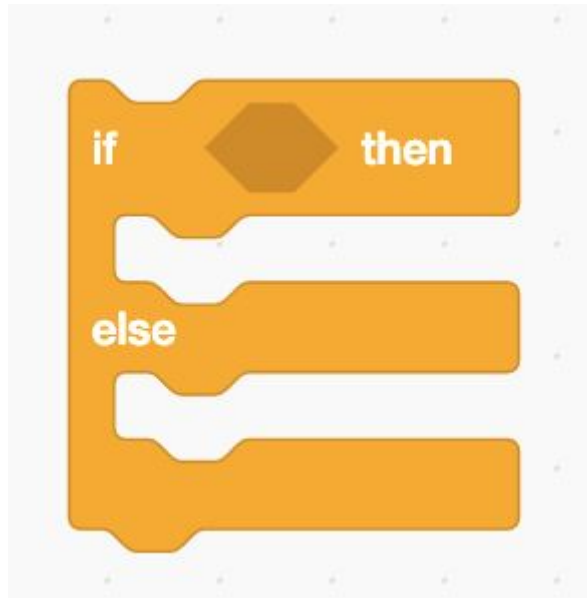
The if-then-else block.

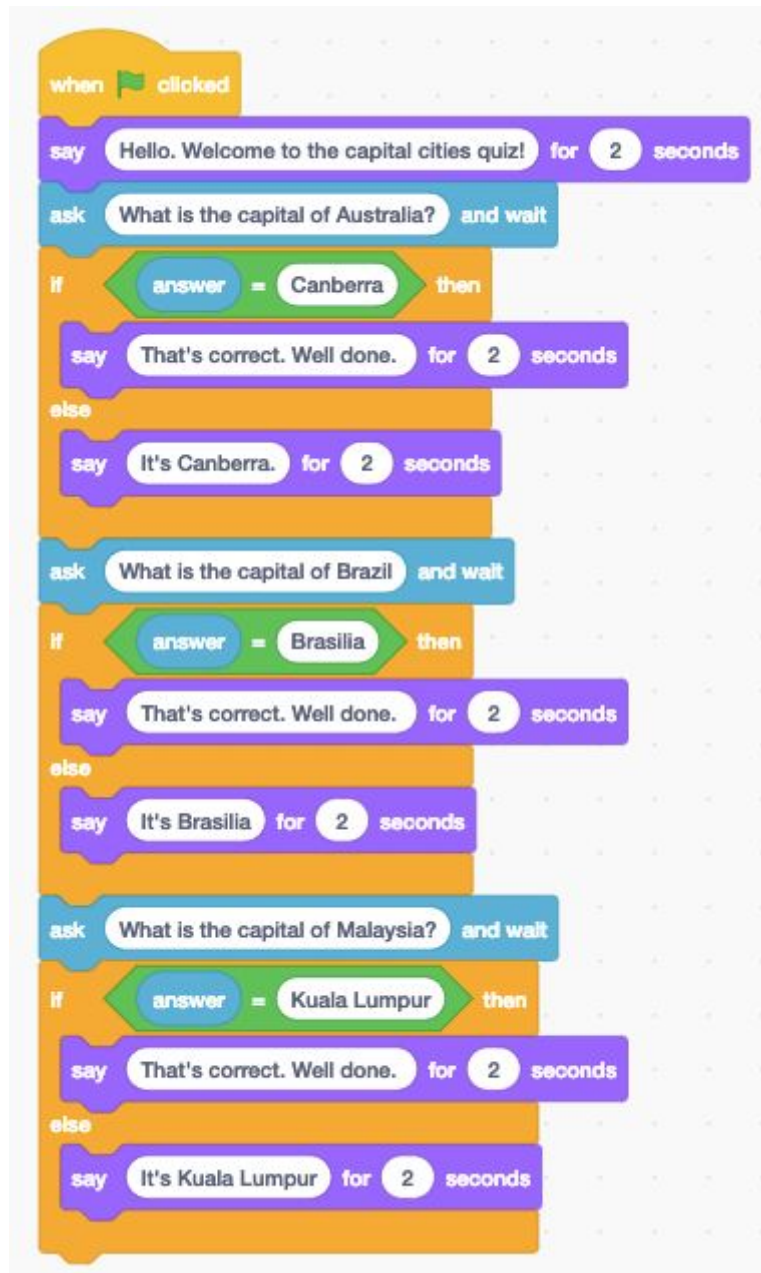When we put all of the aforementioned elements together, your code should look like this:

A breakdown of how to program the question-and-answer segment of your code.

You can repeat this structure as many times as you like to create more questions to be answered!

Your final code

The final version of your code should look something like this:

Your finished code!

Adding extra challenge

You might want to make your quiz game more difficult by adding the following things:

- Can you add a scoring system?
- Would you like to deduct points for incorrect scores?
- Would you like to add sounds effects, e.g. an applause, if the player achieves a score of 10?
- Would you like to add sounds effects, e.g. a gong sound, if the player gets the answer wrong?

What else can you add to your quiz game project?

Happy Scratching!