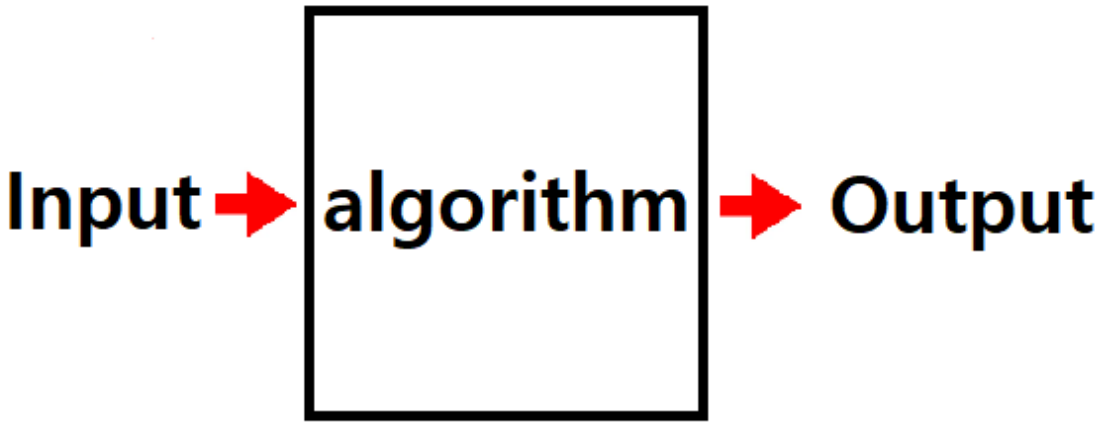
CS50 - Introduction to Computer Science, Harvard University

Lecture 00 - Scratch



Algorithm

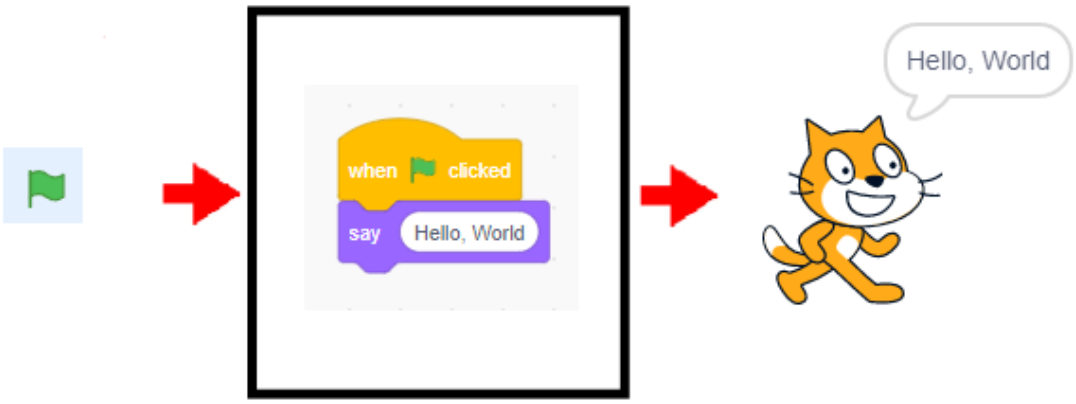
Key concept before starting to write an algorithm is learning how to express intuitions and ideas in code. We can first approach this method by writing a pseudocode. Pseudocode is not a specific language- it is written in English or any other human language to succinctly correctly express an idea of desired algorithm.

While writing pseudocode it is important to keep in mind **all corner cases**. If left unhandled they will cause the program to act unpredictably.

Commonalities in Coding Languages

- Functions - Actions or verbs that solve specific or precise problem
- Conditionals - Conditional set of code based on answer to specific question
- Boolean expression - questions that have simply yes or no answers- in other words, true or false. (0 or 1)
- Loops - directives of cycle to avoid writing repetitive codes

How Scratch mimics programming model



Figuring out corner cases

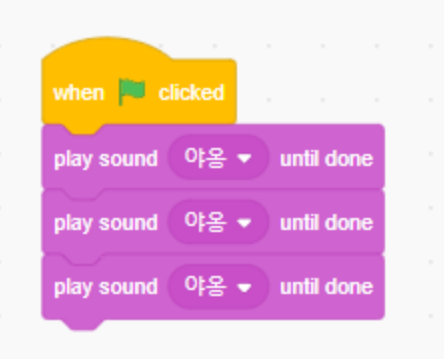
When executed, after typing in input, human eyes cannot detect “Hello!” that is rendered too fast before the rendering of “answer” (user input). This is known as a bug, a corner case that has not been handled by the developer.

Possible solutions include:

- Wait certain amount of time before rendering next output
- Join two outputs to render at the same time

Why some designs are better than others

Take a look at following code



If the object of the code was to repeat “meow” 3 times, it does accomplish its goal. However it is quite repetitive and does not support changing variables well.

By creating separate methods and defining its values to repeat the codes however many times desired, we can improve its reusability and readability.