Title: Heads or tails

Please refer to the online lesson plan on the DT Hub to access all website links and additional resources.

## Summary

In this lesson we show how to transition from a visual based programming language to using a text-based programming language using the example of a heads or tails coin toss application.

Students at the visual programming stage can build the application using Scratch and then with support and guidance, build the application in JavaScript. Students will learn how to create variables, get user input, generate random numbers, compare user input to random numbers to get different outputs.

This lesson was designed in collaboration with Jason Vearing QSITE (Gold Coast Chapter).

## Description

In this example we are using JavaScript for the text based programming language but the same task can be completed in Python. While considered simple, the program draws on fundamental programming concepts such as generating random numbers, declaring and assigning a variable, creating a string, using if-else statements and adding a loop.

**Australian Curriculum**

Years 5-6

* Design algorithms involving multiple alternatives (branching) and iteration (AC9TDI6P02).

Years 7-8

* Design algorithms involving nested control structures and represent them using flowcharts and pseudocode (AC9TDI8P05).
* Trace algorithms to predict output for a given input and to identify errors (AC9TDI8P06).

Keywords

Algorithms, Branching, User input, JavaScript, Text based programming, General purpose programming, Code, Code editor, W3Schools, Random, Variables, String, Transition, Scratch, Visual programming, Repetition

## Learning hook

Can you guess what the computer will ‘toss’ - heads or tails?

How would we create a computer program to do this?

We want to build a simple program that compares your guess (heads or tails) to the computer pick of heads or tails.

You can build first in Scratch or build in JavaScript depending on your familiarity with a text based programming language.

## Learning map and outcomes

In this task students will use and develop some important programming skills.

To build the end program a useful approach is to build in stages and encourage students to test as they go. They check for errors as they run the program and trace any errors and modify the code as necessary. Ask them to keep a record of how they have used this approach to build their program.

Before starting to build their program, ask them to create an algorithm in Pseudocode. This will help them think about the programming requirements and the logical flow.

## Learning input

Students could create a flow chart of their algorithm using relevant software for example the Gliffy add-on for Chrome is easy for students to use. Alternatively, model how to create Pseudocode with the class. Discuss the types of code they expect to use in their JavaScript program when analysing the elements of the Pseudocode.

Pseudocode (example)

START

PROMPT: Guess heads or tails?

INPUT: Answer “heads” or “tails”

OUTPUT: (variable ANSWER)

GENERATE RANDOM NUMBER: Computer pick random number (between 0 and1)

Assign 0 = Heads, 1 = Tails

IF ANSWER = Computer pick

Then OUTPUT: “You guessed it”

ELSE OUPUT: “Try again!”

REPEAT

END

For students creating the program in a visual based programming language provide Scratch or similar coding environment.

For students creating a text-based program, provide students with the link to W3Schools code editor to create and run their JavaScript program. For any students unfamiliar with this code editor demonstrate or ask a student with expertise how to create a simple string, add a button, change text onscreen etc.

## Learning construction

Provide a list of steps in building the program to which students can refer, for example:

This can be provided in an online space where students add to this document as required.

|  |  |  |
| --- | --- | --- |
| **Stages**  | JavaScript (JS) code examples | Comments  |
| **Prompt user for input**  | var answer = prompt();var answer = prompt("Guess heads or tails?"); | // empty prompt box// prompt for heads/tail game |
| **Output** | alert(“test”);prompt(“test”); | // use to test function and show alert box// use to test function and show prompt box |
| **Create random number** | var randTest = Math.random();document.write(randTest)var randTest = Math.floor(Math.random());document.write(randTest)var randTest = Math.floor(Math.random()\*2);document.write(randTest)var computerPick = Math.floor(Math.random()\*2); | // run a test to see how random works (decimal numbers between 0 and 1)//round the numbers to a whole number (between 0 and 1)// Multiply by 2 to get either a 0 or 1// this is the code used when creating the variable for computer to pick a random number |
| **Use if/else**  | if(computerPick ==0) {computerPick = "heads";}else {computerPick = "tails";} | // logical condition// executed if condition is true// else is executed if condition is false |

## **Learning input**

For students that require support

Refer to the *Scratch coding example* to assist students that require support to get the visual programming blocks to function as intended.

Refer to this completed code to assist students that require support to get the JavaScript code to function as intended.

<h2>Heads or tails game</h2>

<script>

// User input guess heads or tails

var answer = prompt("Guess heads or tails?");

// computerpick random number 0 or 1

var computerPick = Math.floor(Math.random()\*2);

//if computerPick is 0 that equals heads

//else computerPick is 1 and that equals tails

if(computerPick ==0) {

computerPick = "heads";

}

else {

computerPick = "tails";

}

// compare answer to computerPick

if (answer == computerPick) {

document.write("YOU GUESSED IT");

} else {

document.write("TRY AGAIN!");

}

</script>

Compare their code to this code and provide guidance as to what they have entered incorrectly.

To trace errors, check for:

* incorrect use of curly brackets { and }
* including opening a script and closing the script <script> </script>
* import random number and round up to 1
* include an answer = prompt
* use of semi colons at the end of statements (used to separate statements)
* use of double quotes for all text to show on screen “text”

###  Challenge

1. Continue to build the program and add a way of looping the program
2. Continue to build the program and add a way of counting and displaying your correct guesses.
3. Reuse and modify the code for a dice roll instead of a coin toss.

## Resources

Please refer to the online lesson plan on the DT Hub to access all website links and additional resources.

Teacher background

* View the YouTube video tutorial created by Jason Vearing showing how to build the Heads and tails application using Scratch first then showing how to build in stages in JavaScript.

Code editor and programming resources

* For visual programming use the online *Scratch editor*
* Scratch completed game: *Heads or tails game*
* Use this website *W3Schools My First JavaScript* to use the code editor to create and run your program built in JavaScript:
* This *JS CheatSheet* provides a comprehensive list of example code that can be used to implement a JavaScript program.