Programming Tasks in Ketchup

Let’s learn the basics of the Ketchup programming language. We will start with some simple programs and move onto more interesting problems.

# Outputting information

The first thing you should learn in any programming language is how to print out information for the user. In Ketchup it is as simple as this:



Now it’s your turn to try this. Write the same program and check what the output is in the console, at the bottom of the screen.

Copy and paste the exact text that is displayed at the bottom of the screen in the console.

[Paste Here]

# Storing information

Another important thing to know is how to store information that we have. This is done using variables, let’s see how a variable is created in Ketchup.



The letter a can be replaced with any other suitable name, as long as it begins with an English letter.

Delete the previous code that you wrote.

So, let’s try creating a variable called score and **set** it to 0. If we run the program now, we will notice nothing happens. This is because we have only stored the information without printing it to the screen. Let’s print the score out.

In the navigation bar at the top of the screen open the Console menu and select Clear, this will clear the console at the bottom of the screen.

Now let’s print out our score:



# Changing information

Our current score does not look very impressive, right now. Let’s **change** that by writing this extra line in.

A close up of words

Description automatically generated

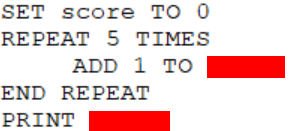
Run this program and see what it outputs now. Write what the output was below.

Output: [Write the output your console showed here]

# Repetition

We changed the score to a specific number, what if we want to change it multiple times?

Well, we can use **repeat** loops to do this:



Here’s the correction for the program we wrote earlier.

Both the red splodges contain the same word, which refers to the variable we are changing and printing.

Once you have written out the program and filled in the red splodges, write what the output for the program is.

Output: [Write the output your console showed here]

# More repetition

So, we have used **repeat** loops, but you may have noticed it just does something 5 times, what if we wanted a variable which ticked up by some amount every time. This could be useful for a program which shows you your times tables.

Let’s write a **count** loop, this will allow us to have a specific variable which changes with every loop that we do.

A close up of a text

Description automatically generated with medium confidence

Check the output for this program. It should be the numbers 1 to 10 printed out in a column.

You can also control the step count between each loop by adding this extension

A close up of a word

Description automatically generated

Write a program which counts from 10 down to 1 counting **down** in increments of 1 (Hint: Make the loop change **by** a negative number). With the output looking like this:

10  
9  
8  
7  
6  
5  
4  
3  
2  
1

Paste your program below:

Your program:

[Paste your entire program here]

# Taking user inputs

Taking user inputs in Ketchup is done like this:

INPUT WITH PROMPT [Appropriate prompt]

Where the prompt is a string, using “”, which directs the user to enter certain information.

However, we cannot simply use this block of code anywhere, it must be in context. So, let’s assign a variable to it.

Clear your code window.



The first splodge is part of setting a variable to a certain value. You can look back through the exercises if you don’t remember how to do this.

The second splodge is part of the input, you can use the snippet of code provided above to help.

Fill in the red splodges and copy and paste your program in the area below.

Your program:

[Paste your entire program here]

Now try the same trick, but this time print the result without setting the user input to a variable. Here’s another code snippet to help you with this:



In this program you need to request the user’s name and then use that to say **“Hello, [Inputted Name]”**.

Your program:

[Paste your entire program here]

# Using user inputted information

Now we can take in a user’s input we can use it to do something. Let’s take in 2 numbers from the user and output their sum.

A red rectangle with black text

Description automatically generated

The last line may look complicated, but it is literally printing the + and the = and then substituting in the values that we set for a, b and sum.

Try the following inputs and copy the result in the console:

Program 1:

a: 1

b: 2

Output: [Paste the console output for this program]

Program 2:

a: 3

b: 1

Output: [Paste the console output for this program]

# Times Tables

Now you will write a program by yourself. The program will ask the user for a times table, similar to how we asked the user for a number in the last exercise. Then we will output that number being multiplied by numbers 1 to 12.

The program in the console should look something like this:

INPUT: 4

1 x 4 = 4  
2 x 4 = 8  
3 x 4 = 12  
4 x 4 = 16  
5 x 4 = 20  
6 x 4 = 24  
7 x 4 = 28  
8 x 4 = 32  
9 x 4 = 36  
10 x 4 = 40  
11 x 4 = 44  
12 x 4 = 48

Hint: Your print statement should take this format:

**PRINT variable “ x ” variable “ = ” variable**

Where the variables are the variable names from your program.

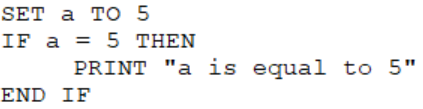
Your program:

[Paste your entire program here]

# Selection

Selection is when you **select** to go down a specific code path, depending on some condition.

Here is an example of a simple **if** statement, which prints **“a is equal to 5”** in the case that a is 5.



Below is an expanded version of this handling the case that a is not equal to 5.

A close up of words

Description automatically generated

And the final form and full form looks like this:

A white background with black text

Description automatically generated

Now that you are familiar with the if statement let’s write a program with it.

We are going to write a program to give out the result of an examination for a student. The exam can be failed, passed with a merit or passed with a distinction.

The program will work as such:

* Ask the user what their test result is (set this as the variable **mark**).
* Check that their test result is not between 0 and 30, if so then output **“Invalid score”.**
* Check that their test result is less than 15, if so then output **“Failed”**.
* Check that their test result is less than 25, if so then output **“Merit”**.
* In all other cases output **“Distinction”**.

Hint 1: Use the full form of the if statement using 1 IF, at least 2 ELSE IFs and 1 ELSE statement.

Hint 2: When checking the score is not between 0 and 30 you can use this condition statement:

**mark > 30 OR mark < 0**

Your program:

[Paste your entire program here]

# Error Checking

Now that you have written some complicated programs, let’s go back to the basics and fix someone else’s errors.

You can copy and paste the code into the Ketchup IDE and use the built-in error detection system, if you need help.

## It just keeps going

PRINT "Hello, World

Correct version: [Paste your working version of the program]

## If this … what?

SET a TO 4

IF a > 2

PRINT "a is greater than 2"

END IF

Correct version: [Paste your working version of the program]

## You need to ask for it!

SET name TO INPUT WITH PROMPT

Correct version: [Paste your working version of the program]

## Something weird.

This is your first time encountering the **CREATE** keyword, but effectively it makes the variable exist, but have no value. A little weird, but it seems like we cannot set an already existing variable.

CREATE x

SET x TO 7

PRINT x

Correct version: [Paste your working version of the program]