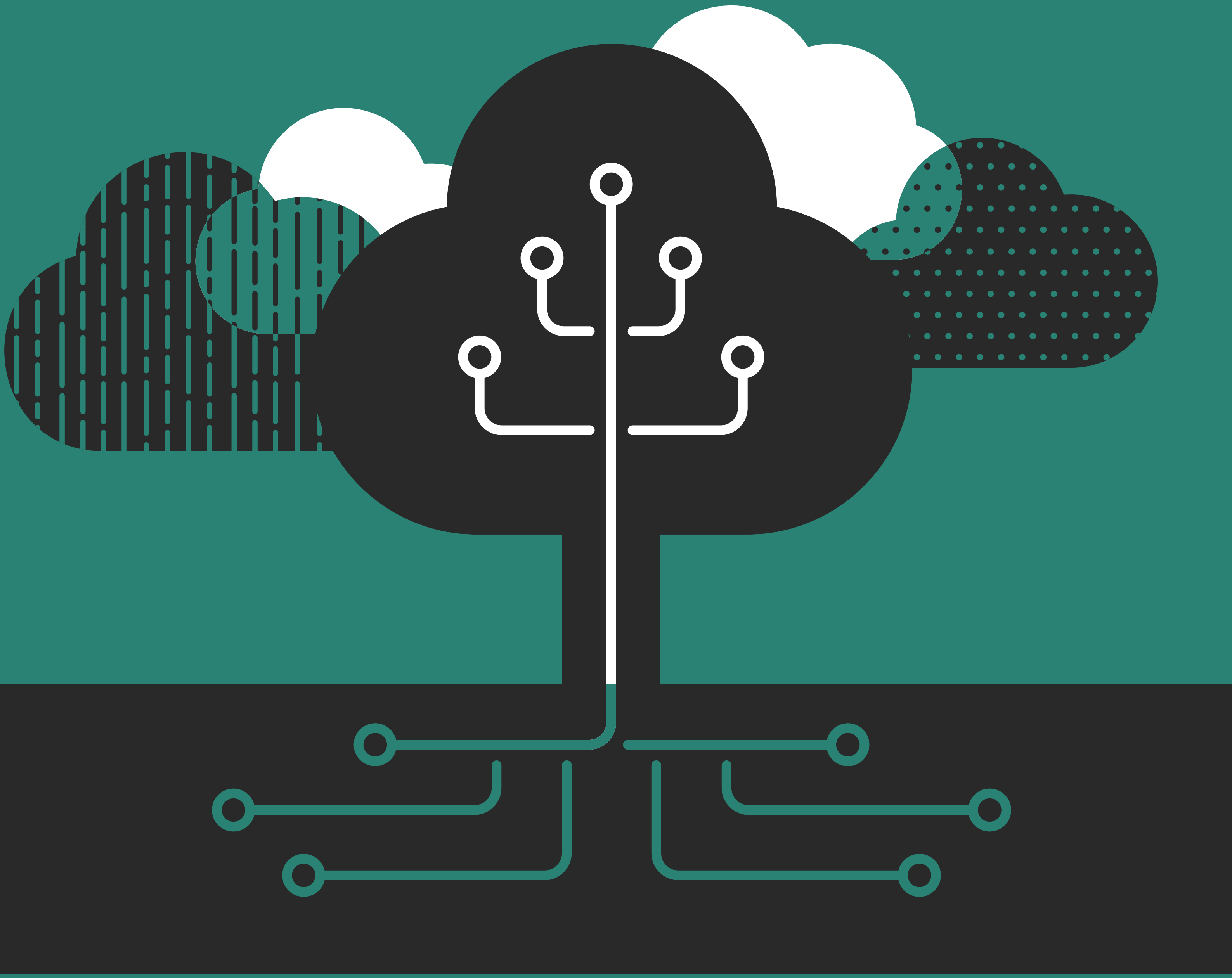
Met Office

Deeper Discovery

These are instructions for learners who are familiar with coding.



www.metoffice.gov.uk/schools

MakeCode Arcade activity

You will be working towards the following technology/computing objectives:

- I can explore features within the game to help solve problems
- I can design, write and debug a program
- I can explain my code and how it works
- I notice errors and correct them in the program

Your Quest:

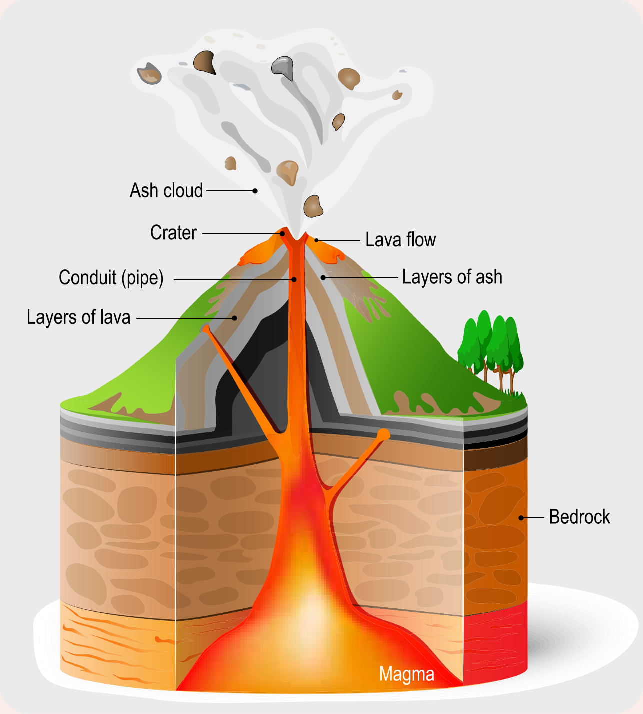
We have an emergency! A volcano has erupted, and we need you to gather observations of volcanic ash levels and send it back to the Met Office supercomputer.

The Science Behind the Game:

There are many active volcanoes all over the world. When volcanoes erupt, they release magma, hot gases and ash. Eruptions are dangerous to people living near the volcano, but also to people living far away, because ash in the atmosphere can travel all around the globe. This ash can also be dangerous for aeroplanes travelling through it.

Volcanic ash is made up of tiny pieces of rock that have been heated to very high temperatures. When ash particles get sucked into the jet engines of planes they cause a lot of damage. It can be invisible to our eyes; we use special equipment to find it.

To help pilots stay safe, the Met Office provides forecasts of volcanic ash, using a combination of observations and computer models.



To create a forecast, you need to know where the ash is now. This is called an observation. Volcanic ash observations are made in a number of different ways including from satellite data, by radar and by LiDAR (Light Detection and Ranging).

The Met Office uses an instrument called LiDAR to measure ash clouds. LiDAR works by sending pulses of laser light through the atmosphere, which is scattered by volcanic ash particles. Some of this light reflects back to the instrument and can be used to work out the height and concentrations of ash.

As well as a ground-based LiDAR network, and LiDAR fitted to aircraft, the Met Office has a van fitted with a LiDAR, which can be deployed around the UK to locations where observations are needed.

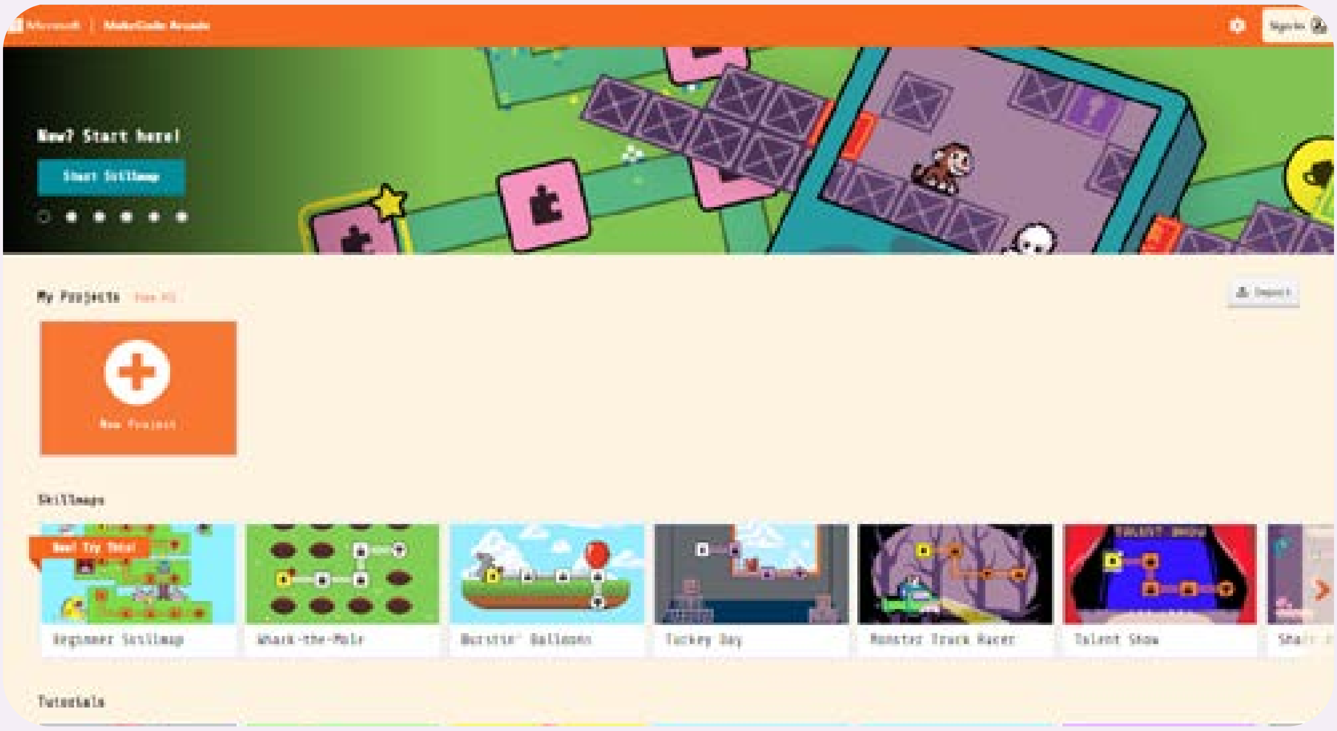
The Game - Met Office Volcanic Ash Emergency

In the game you will drive the LiDAR van around the map to collect volcanic ash observations, and then return the data to the Met Office supercomputer.
The game has three main aspects to it:

- Collecting observations
- Sending the data to the Met Office
- Deciding whether to shut the airport, based on the data collected

Section A – Opening the game

Open a web browser and go to: https://arcade.makecode.com/



Navigate to Met Office Volcanic Ash Emergency game.

There are three Met Office Volcanic Ash Emergency games. The first two require players to make changes to the code to get it to work properly and are what we will focus on in the recorded session. The third one is the finished game.

1. A Deeper Discovery game, suitable for beginner coders (recommended for KS2 or equivalent) and is what we cover in this document. **Click here** for direct link.
2. A Deeper Discovery, suitable for those who like a greater challenge (recommended for KS3 or equivalent). **Click here** for direct link.
3. A full solution game. **Click here** for direct link.

Section B – Playing the Game

Task 1 - Learn how the full solution game currently works

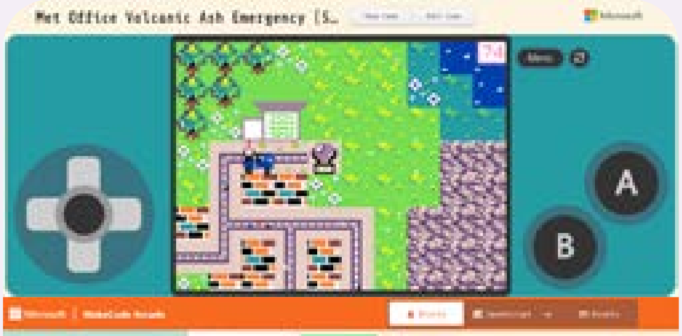
Our presenters will play the full solution game. This is how we would like the finished game to look. Have a go yourself and figure out how the game works?

Section C – Let's Get Coding

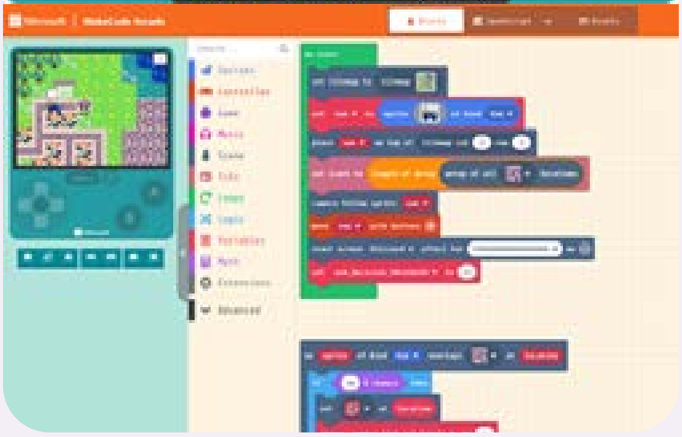
Now, load and play the First Exploration game to familiarise yourself with the controls and what you can currently do. This version is unfinished.

- We are missing information needed to make a decision.
- The messages are incorrect at the end of the game.

You should now be familiar with how the game currently works. Let's warm up and get familiar with block coding.



Click the "Edit Code"button at the top of the screen.



This will display a screen like this

This is where we write our code to get the game to work.

We will be coding with blocks. Find the blocks you want using the search or browse the menus. We can drag the blocks into the coding grid. The blocks slot together like jigsaw pieces. This helps us see where the code belongs.

Task 2 - How to change/delete existing code:

The `On Start` event is what sets the scene of our game when we open it.
In this `On Start` section of the code. Change the blizzard. Have a play changing it to display something different. Make sure you can see your changes by checking the game on the left. See if you can turn the blizzard effect off altogether.

Section C – Let’s Get Coding

Pro-tip: If you make a mistake, don’t panic. You can always undo any changes by pressing `CTRL + Z`.

Pro-tip: you can use `CTRL + C` to copy code from one event to another – for example, you can copy the `show long text` and just change the message.

Pro-tip: You can select just one piece of code by holding down the CTRL button.

Section D – Bug Hunting

Part of the job of a Software Engineer is finding and fixing bugs in computer programs. Bugs are mistakes in the code.

Task 3 – What’s broken?

Select game either the First Exploration (for beginners) or Deeper Discovery (for those who like a challenge). Have a play and see if you can find the bugs by comparing it to the full solution game from Task 1.

Section E - Fixing the game [10 mins or longer]

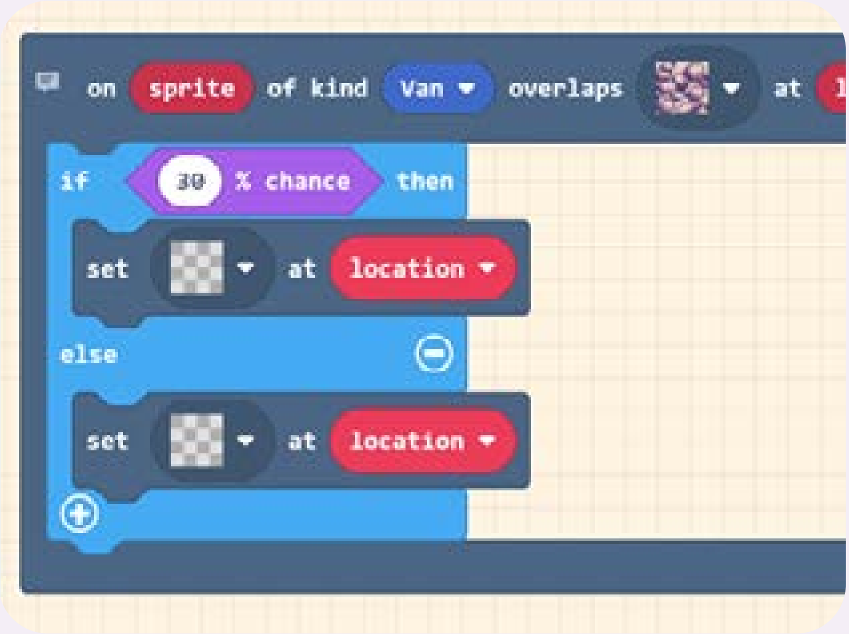
Let's now look at fixing some of the problems we found in Section D.

The following tasks are available for both **First Exploration** and **Deeper Discovery** games, be careful to follow the correct section for your chosen game.

Task 4: Counting the ash level measurements

Deeper Discovery

Have a look at the following event block:



```
on sprite of kind Van ▾ overlaps ▾ at
  if 30 % chance then
    set ▾ at location ▾
  else
    set ▾ at location ▾
```

This event happens when the LiDAR van sprite moves over each of the ash squares that we haven't yet measured.

We need to update the if statement:

- Change the square from being the current ash type to an ash square with a red or blue border, to represent high or low levels of ash in the cells. Whether it is low or high depends on the random chance in the if statement.
- Depending on whether it is high or low, we need to increase our count of high or low measurements. There are two variables which act as counters for the two types of ash concentration: `number_high_ash_levels` and `number_low_ash_levels`. Select which one should be updated in each part.

After the if statement, we also need to change the overall count of how many measurements are left to make, which is what we are calling the score in this game. Change the event to decrease the score by one.

Section E - Fixing the game [10 mins or longer]

Task 5: Reporting the result of the game

Deeper Discovery

Zooming into this section of code:

We can see that this is incomplete.

Let's fix the logic.

We have four scenarios:

The user presses `A` to close the airport:

- The ` high_ash_levels_fraction` is greater than the `ash_decision_threshold`. The airport is correctly closed.

or

- The airport is closed unnecessarily. This will cost the airline money.

The user presses `B` to keep the airport open

- The `high_ash_levels_fraction` is greater than the `ash_decision_threshold`. The airport is open when it should be closed!

or

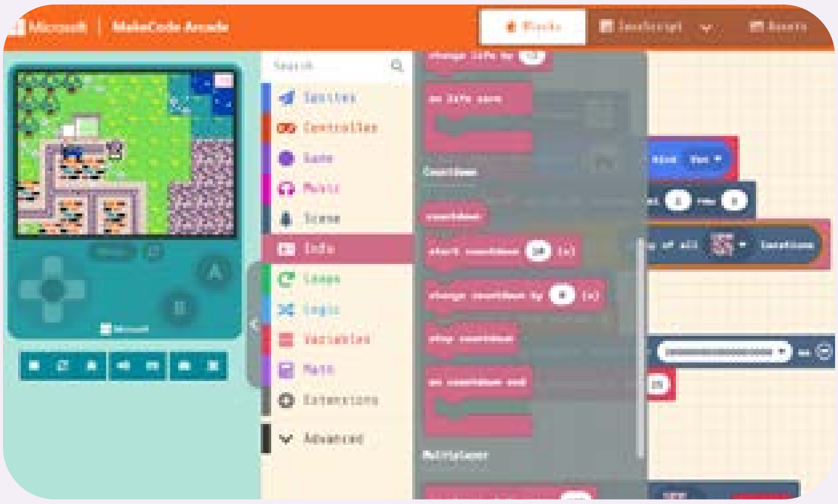
- It is safe to keep the airport open.

Update the code to make sure these four scenarios are covered. For completeness, we would also like to reset the game after the message has been shown. You can find `reset game` by searching in the search box.

Section E - Fixing the game [10 mins or longer]

Task 6 – How to add new code

Add a new event – a countdown timer to our game, our game should have a 40 second limit.



Hint – The countdown code can be found in the info section.

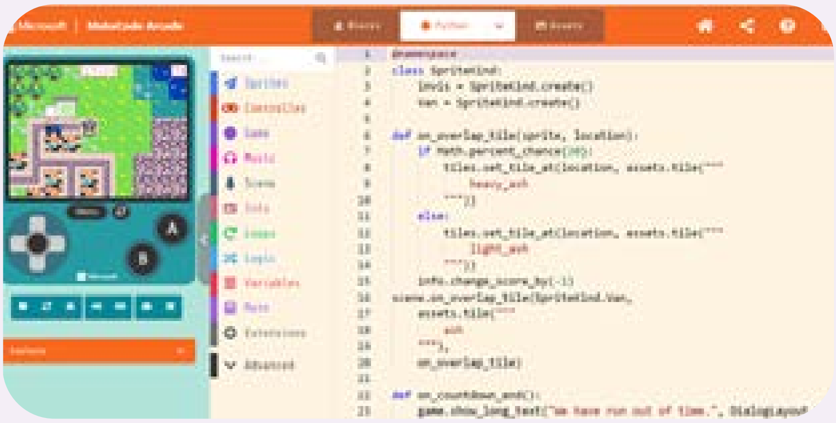
Task 7 – Add walls by editing the `tilemap`

Currently our LiDAR van can drive into the volcano – that is not safe!

Add a wall around the volcano to make sure our van cannot drive there.

Task 8 - Investigation

At the top of the screen you can switch from `block` code to `Python`. There are many programming languages, Python is a great one to learn if you want to get into coding as it is freely available and easy to learn. Python is one of the languages used at the Met Office to write our code.



```
@namespace
class SpriteKind:
    Invis = SpriteKind.create()
    Van = SpriteKind.create()

def on_overlap_tile(sprite, location):
    if Math.percent_chance(30):
        tiles.set_tile_at(location, assets.tile("""
            heavy_ash
        """))
    else:
        tiles.set_tile_at(location, assets.tile("""
            light_ash
        """))
    info.change_score_by(-1)
scene.on_overlap_tile(SpriteKind.Van,
    assets.tile("""
        ash
    """),
    on_overlap_tile)

def on_countdown_end():
    game.show_long_text("We have run out of time.", DialogLayout
```

Have a look at the code and see how the block events translate into Python. Have a go at, for example changing the messages that are displayed to the screen. Change the `percent_chance` from 30 to 60 and re-play the game. You should have more `heavy ash` (red-border) ash squares.

Glossary

Algorithm - series of step-by-step instructions.

Data Point – In science, we collect information, or data. One of these pieces of information is known as a data point.

Grid boxes - In our game, a grid box represents a small section of the map. We drive over these to collect our observations. We collect one observation for each grid box.

LiDAR – Light Detection And Ranging – this is a piece of equipment used at the Met Office to shoot a beam of light into the air. This reflects off the cloud and the time taken is measured.

Observation - Observations are measurements of weather variables such as temperature, rainfall and wind.

Plume - a long cloud of smoke, ash or vapour sometimes resembling a feather as it spreads from the source of the cloud.

Precipitation – Water that falls from the clouds as rain, snow, hail, sleet
Variable - In computer programming we use variables to store information that might change and can be used later in our program.

Variable - In computer programming we use variables to store information that might change and can be used later in our program