

## D - Transport the Nuclear Waste

We need to transport some very unstable nuclear waste across the planet, so we must program the robot to move as smoothly as we can.



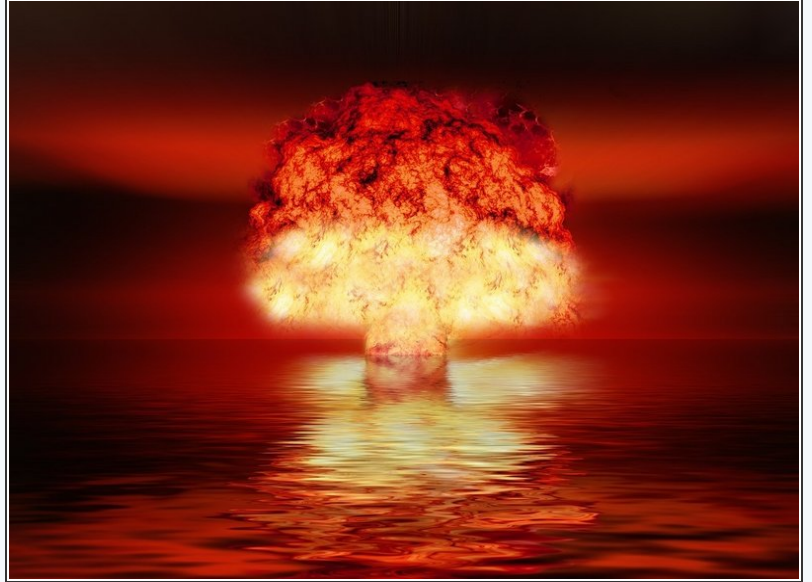
# INTRODUCTION

We need to transport some very unstable nuclear waste across the planet, so we must program the robot to move as smoothly as we can.

## Step 1

### Nuclear Waste

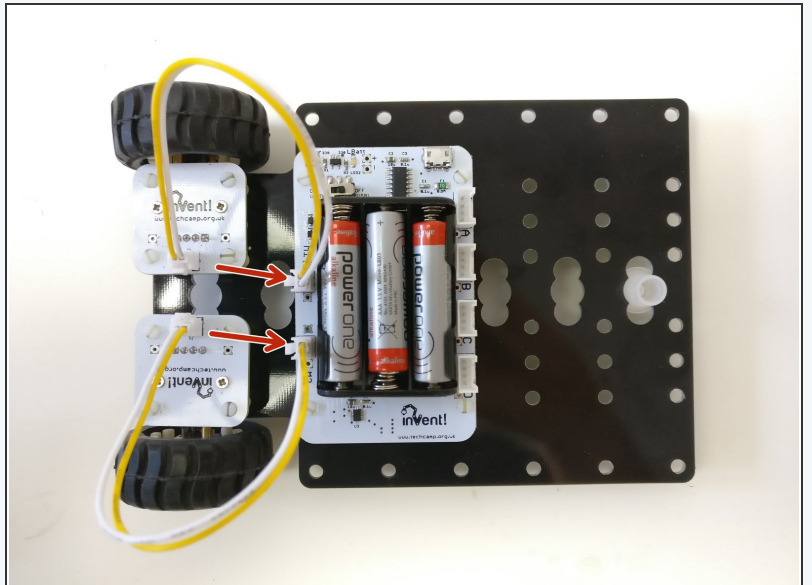
- Some **nuclear waste** has been found near the base, and we need to move it to the other side of the planet as it is **very dangerous**.
- The nuclear waste is **extremely unstable**, so we need to make our robot **accelerate** and **decelerate smoothly** so it doesn't explode!
- To do this, we are going to learn about **variables**.



## Step 2

### Assemble the Robot

- Put your robot together just like the picture! The connections should be:
- Left Motor > **M1**
- Right Motor > **M2**

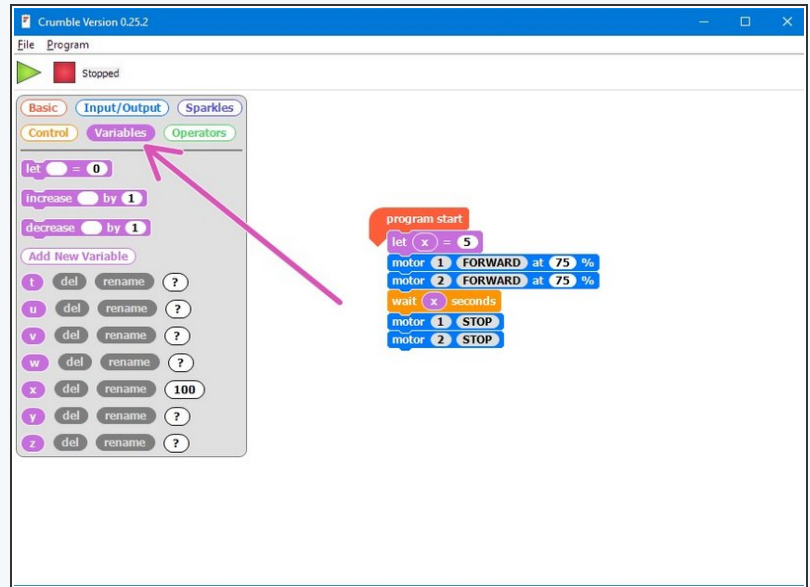




## Step 3

### Test Program

- **Build** the program in the picture!
- You can find all the blocks you need in the **variables** menu.
- **Before** you try programming your robot, **what do you think this program will do?**



## Step 4

### What are Variables?

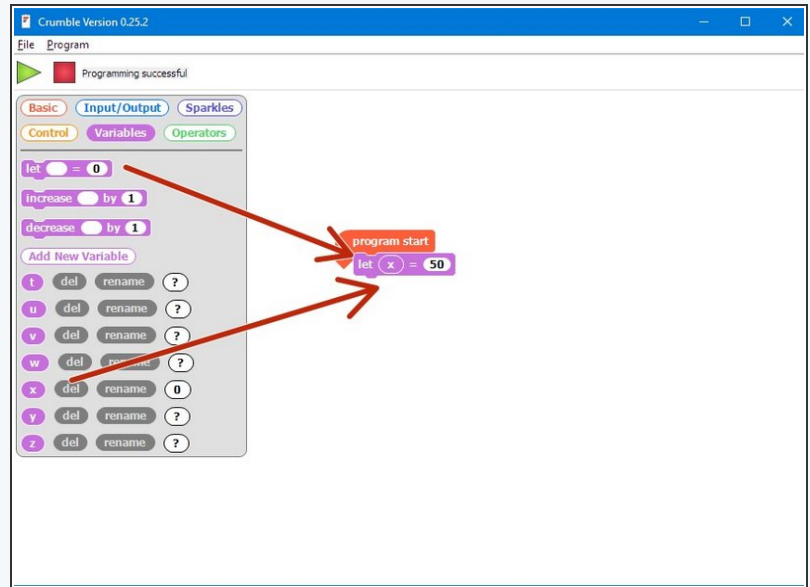
- A good way to understand variables is to think of them like your **lockers** at school.
- To use a locker, you need to put your **name** on it - we do the same with a variable, and you can call it **anything you want!**
- We can then put **whatever we like** inside the locker - books, bags, clothes, anything! We can do the same with variables, but for now we'll just put **numbers** inside them.
- We can **add, remove and change things** in the locker whenever we like - its the same for the number in the variable!
- Most usefully, we can go back to the locker or variable at any time and **see what's in it** (so long as we know the **name** of the locker or variable!).



## Step 5

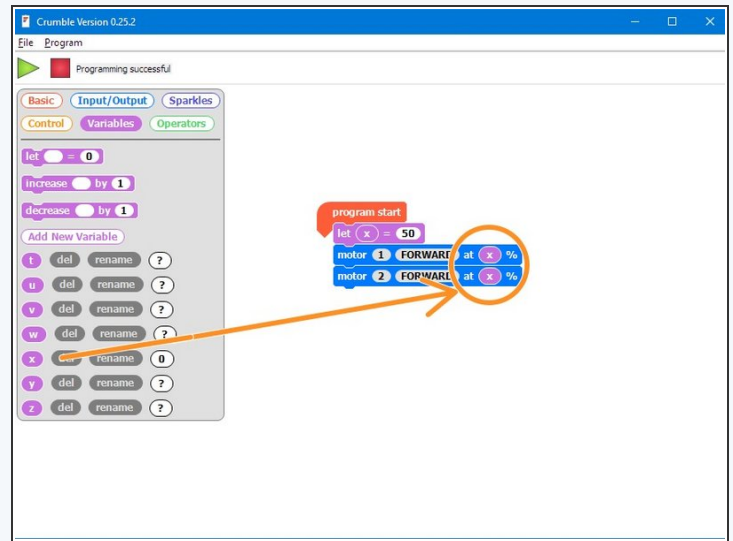
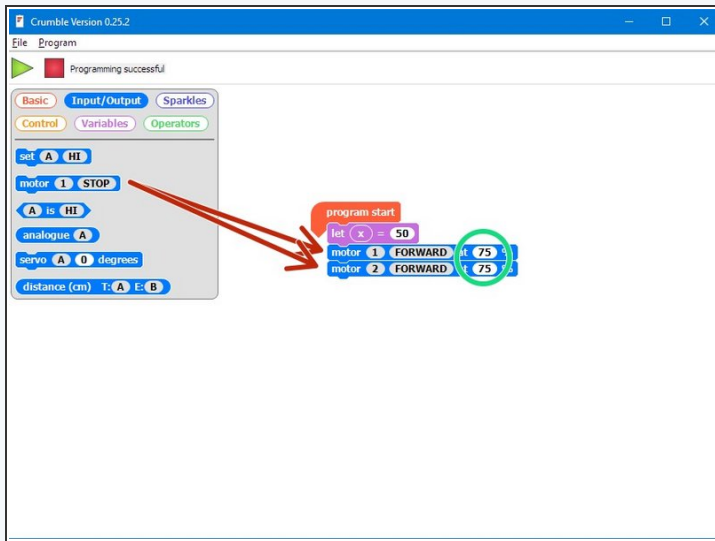
# Using Variables

- If you don't quite understand, don't worry - for now, just remember we can do these things with variables:
- **Call** them anything we like (variable **name**)
- **Store** any number we like inside them (variable **contents**)
- **Change** the contents at any time (add, subtract, multiply, divide and so on)
- **Access** the contents at any time, so long as we know the **name** of the variable.
- Let's start a new program by **calling** a new variable **x**, and **setting** it to **50**.



## Step 6

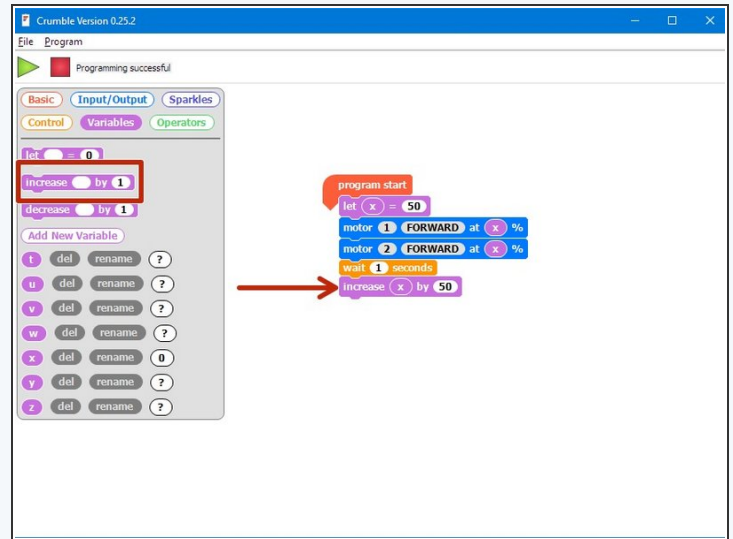
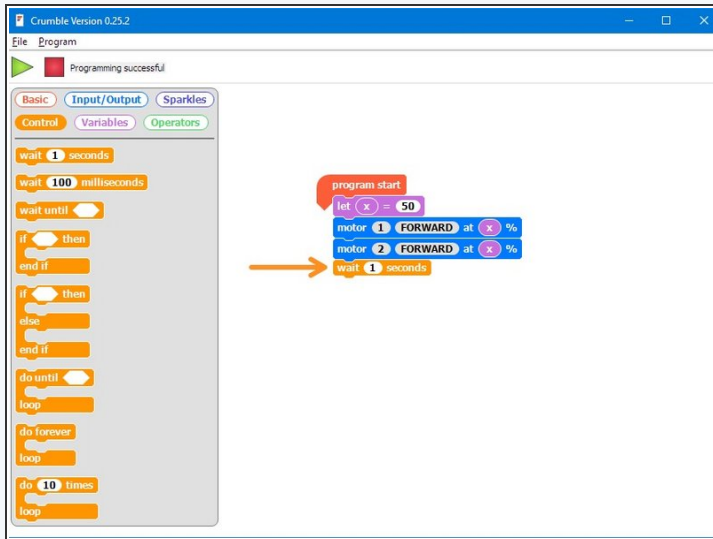
# Accessing Variables



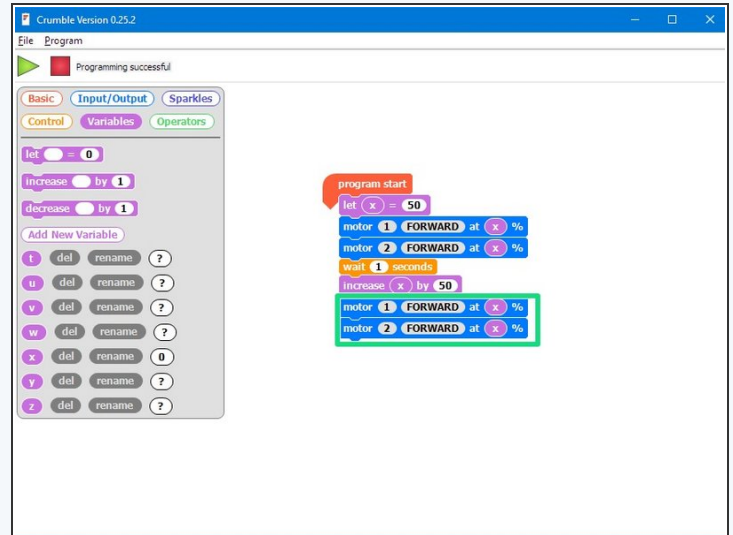
- Let's **access** the number inside our variable, **x**, and use it to **turn on the motors**.
- Drag in **two motor blocks** and make them set motor 1 and motor 2 going **forwards**.
- Wherever there is a **number** with a **white background** on a block, we can use a **variable** in its place if we want to.
- Replace each motor speed with **one x block**, so the motor speed is set by the **number** in the **variable x**

## Step 7

# Changing Variables

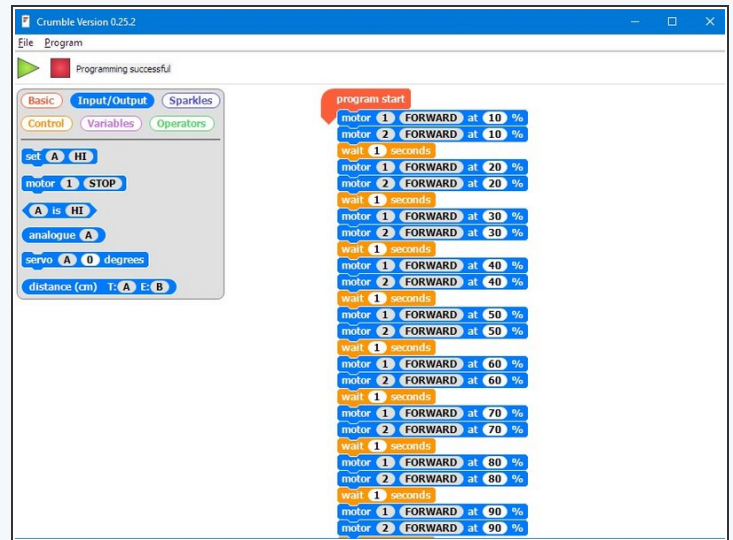
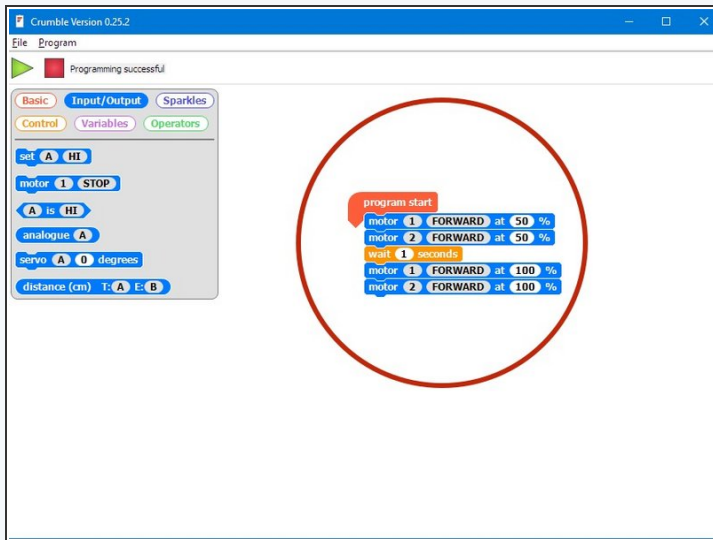


- Now let's try **changing** the motor speeds by **changing the variable**!
- First, **add a wait block** so the robot moves forward at the first speed (50) for 1 second.
- Now change the variable by using an **increase block** to increase x by 50.
- Finally, add **two more motor blocks** to set each motor to speed x again - try it out, the robot should **change speed** this time!



## Step 8

# Why do we need variables anyway?

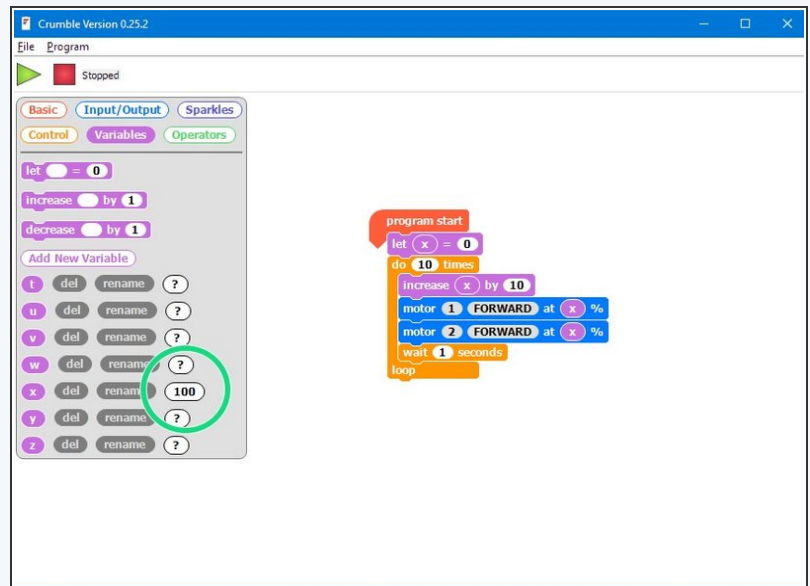


- You might be thinking - **why bother** using a variable to do this? We could have just used the **simple** program in the picture!
- Well, what if we wanted to increase the speed of our robot (**accelerate**) slowly?
- Even if we started the speed at **0** and increased the speed by just **10 every second** (0,10,20,30,40.....), the program would require **20 motor blocks**!
- Have a look at the second picture to see this program - it is **far too long**.

## Step 9

### Variables and Loops

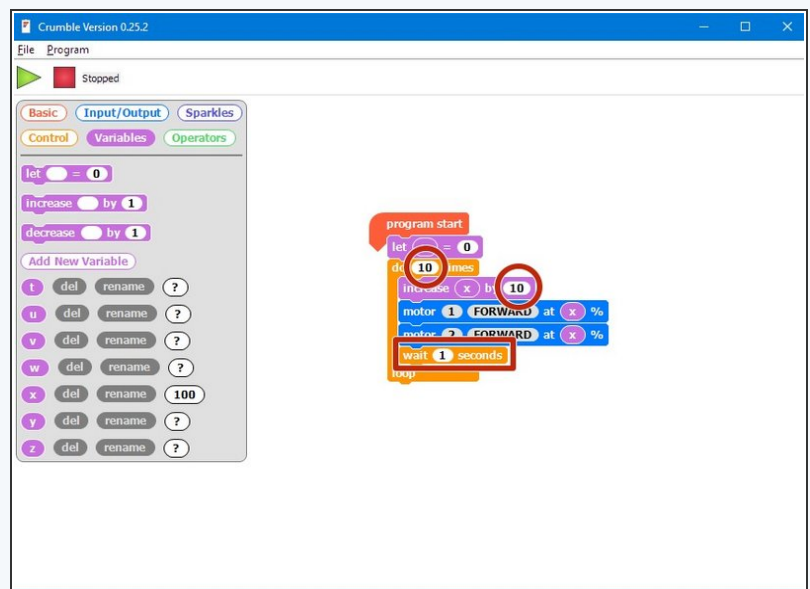
- By combining **variables** and **loops**, we can program things like acceleration **very easily**.
- **Have a look** at the program in the picture - can you **work out** what is going on?
- The loop runs **10 times** - each time it runs, **x is increased by 10**, and the speed of the motors is **set to x**!
- See how much **shorter** this program is?
- Special tip - when you are using variables, you can see what they are equal to when the program is running in the **variables menu**. **Test** this program in your robot and what happens to **x**! **Your robot must be plugged in for this to work.**



## Step 10

### Smooth Acceleration

- We're nearly there! Let's **change some things** in our program to make the robot accelerate **really smoothly**, so we don't set off the nuclear waste.
- Change the program so that:
  - The loop repeats every **0.1 seconds** (100 milliseconds) instead of every second
  - Each time the loop repeats, x is **increased by 1**
  - The loop runs enough times for the motors to change speed from **0** all the way to **100**.
- If you need some **hints**, we've marked the parts of the program you will need to **change**!
- **Test your program** - your robot should now speed up really smoothly.

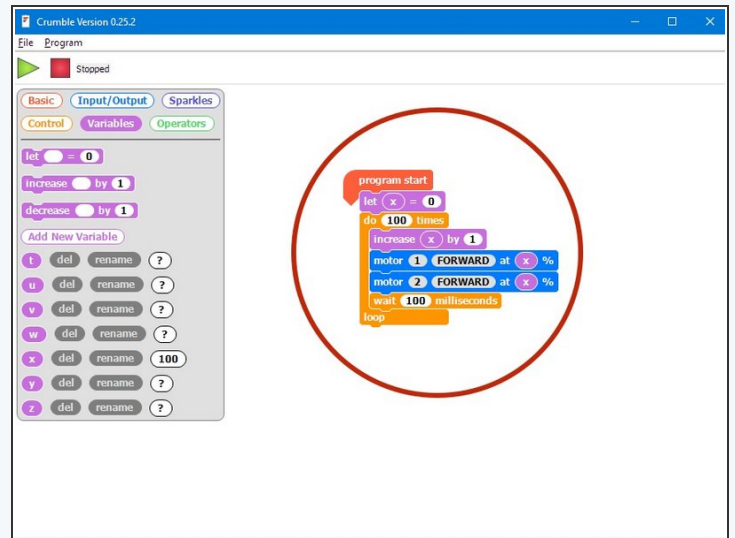




## Step 11

### Challenge - Smooth Deceleration

# Challenge!



- We need to be able to **decelerate** (slow down) smoothly as well to stop on the other side of the planet.
- **Change your code** so that the motors start at 100%, and **decelerate smoothly to 0**.
- If you need it, the second picture has the correct program for smooth **acceleration**.

## Step 12

### Transport the Waste

- Now you have learnt everything you need to **safely** move the waste!
- Write a program that:
  - **Starts** at your base
  - **Accelerates smoothly** towards the other side of the planet
  - **Decelerates smoothly** and **stops** at the other side
  - **Waits** for **5 seconds** so the waste can be unloaded
  - **Spins** on the spot **180** degrees
  - Drives back to base at **full speed**, and stops in the right place.

