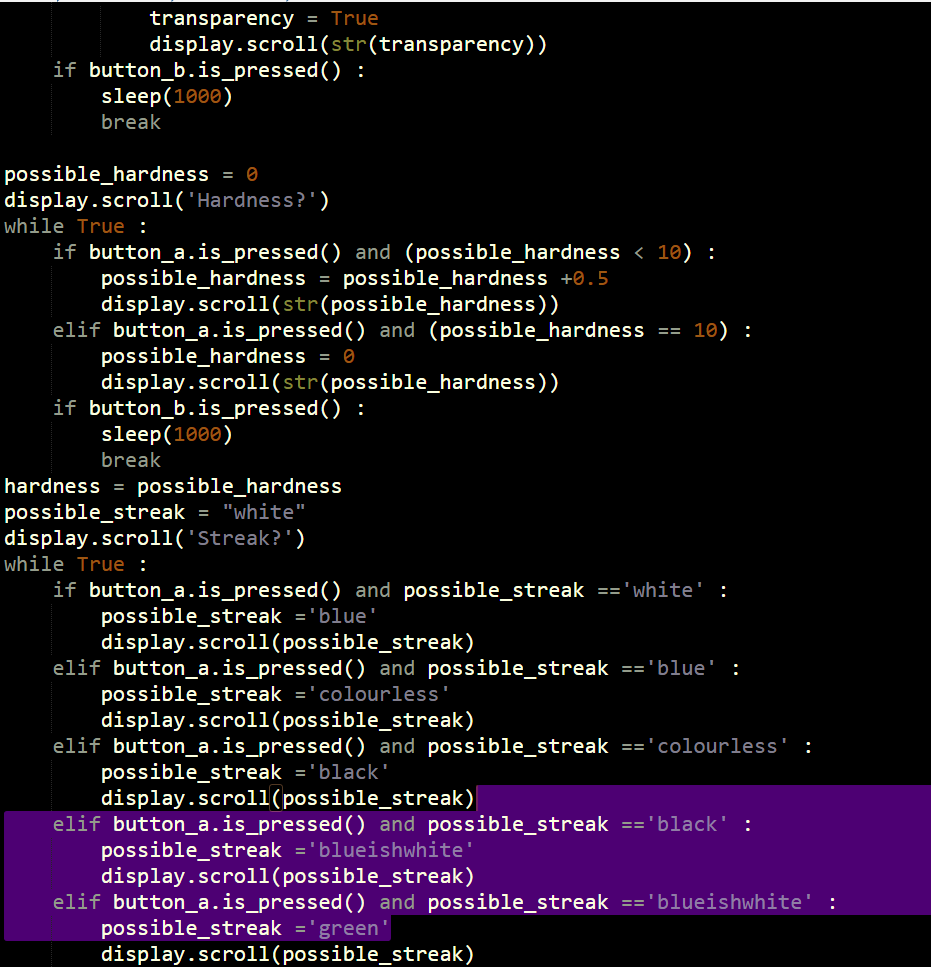


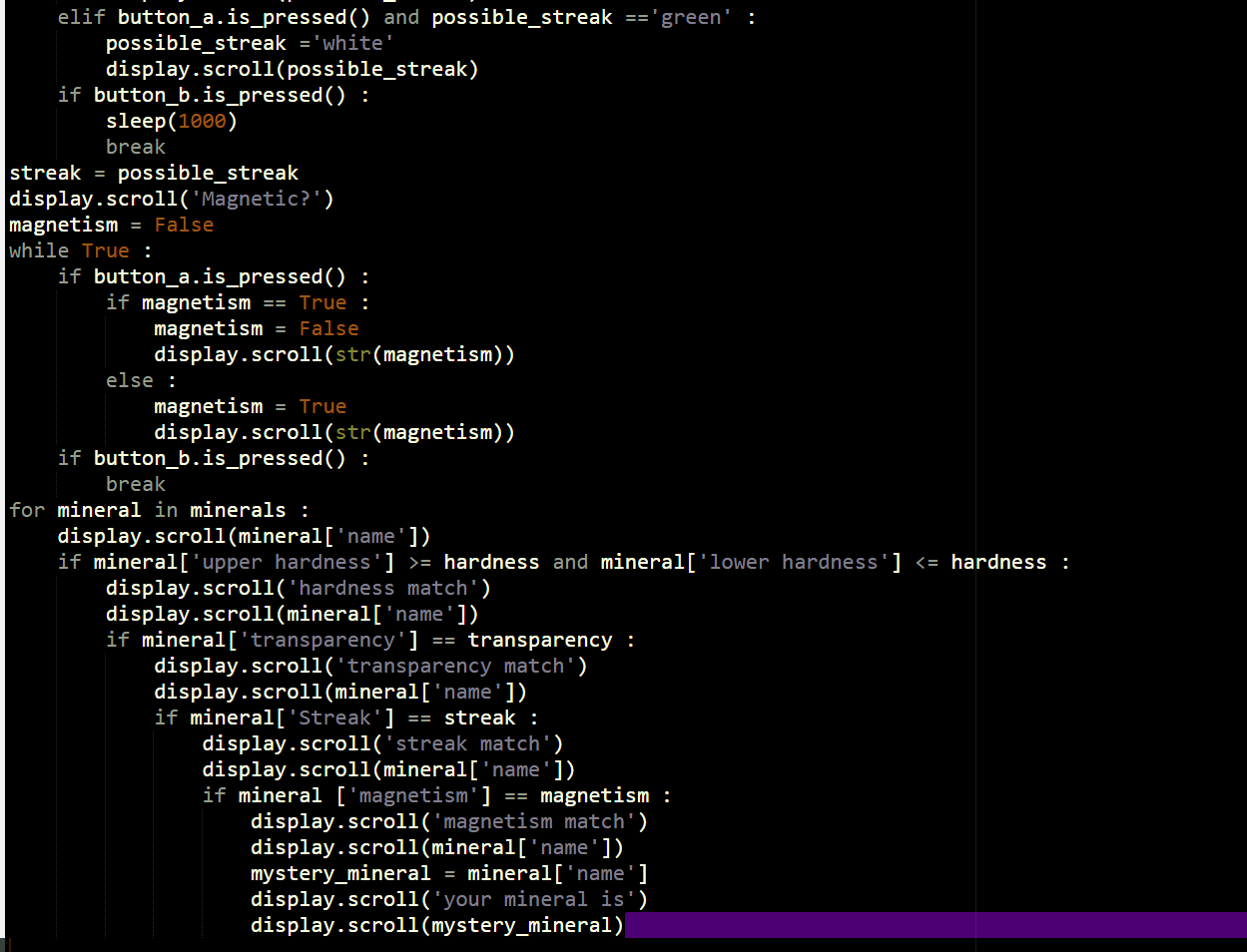
*‘’WHAT DOES ALL THIS CODE DO?’’*

*‘’AND WHAT DOES IT HAVE TO DO WITH GEOLOGY?’’*

‘’WELL LET ME EXPLAIN!’’







[Grab your reader’s attention with a great quote from the document or use this space to emphasize a key point. To place this text box anywhere on the page, just drag it.]

[Cite your source here.]



‘’WE’RE NOT SKIPPING IT I NEED TO COVER A FEW THINGS FIRST. DO ANY OF YOU KNOW WHAT THIS CODE IS FOR: DO YOU KNOW WHAT IT DOES?’’



*‘’WHY ARE WE SKIPPING ALL THIS CODE? ISN’T THIS WHAT YOU’RE HERE TO TALK ABOUT?’’*

*‘’NO, I DON’T THINK WE DO!’’*



‘’WELL THIS CODE IS FOR THE BBC:MICROBIT WHICH IS 70 TIMES SMALLER AND 18 TIMES FASTER THAN THE ORIGINAL MICRO COMPUTERS IN THE LATE ‘80s! ON THE BBC:MICROBIT WEBSITE YOU CAN CREATE A CODE, COMPILE IT AND THEN DOWNLOAD IT TO USE ON YOUR OWN MICROBIT. THE CODE USED HERE IS MICRO PYTHON A TYPE OF PYTHON CREATED FOR ‘MICRO DEVICES’ SUCH AS THE RASPBERRY PI (ALTHOUGH THERE ARE SOME CHANGES TO MICROPYTHON SUCH AS THE OBJECTS BUTTON A AND BUTTON B ADDED IN SO THAT IT IS POSSIBLE TO USE INPUTS).’’





Object button a .is\_pressed() detects whether button is pressed or not the same is for button b (the colons are from the end of an if statement)

BBC:microbit

C:\Users\family\Documents\jake's homework\not homework\images\button b object.PNGC:\Users\family\Documents\jake's homework\not homework\images\button a object.PNGC:\Users\family\Documents\jake's homework\not homework\images\display.scroll.PNG



‘’AS FOR WHAT THE PROGRAM DOES; IT ASKS THE USER FOUR QUESTIONS: WHAT’S THE STREAK, WHAT’S THE HARDNESS, IS IT TRANSPARENT AND IS IT MAGNETIC. AFTER THIS, IT WORKS OUT WHICH MINERAL YOU HAVE. BECAUSE IT RUNS ON A MICROBIT, YOU CAN TAKE THIS ANYWHERE YOU LIKE PROVIDED YOU HAVE A BATTERY PACK AND ENOUGH BATTERYS.’’



*‘’wow! This could be really usefull!’’*

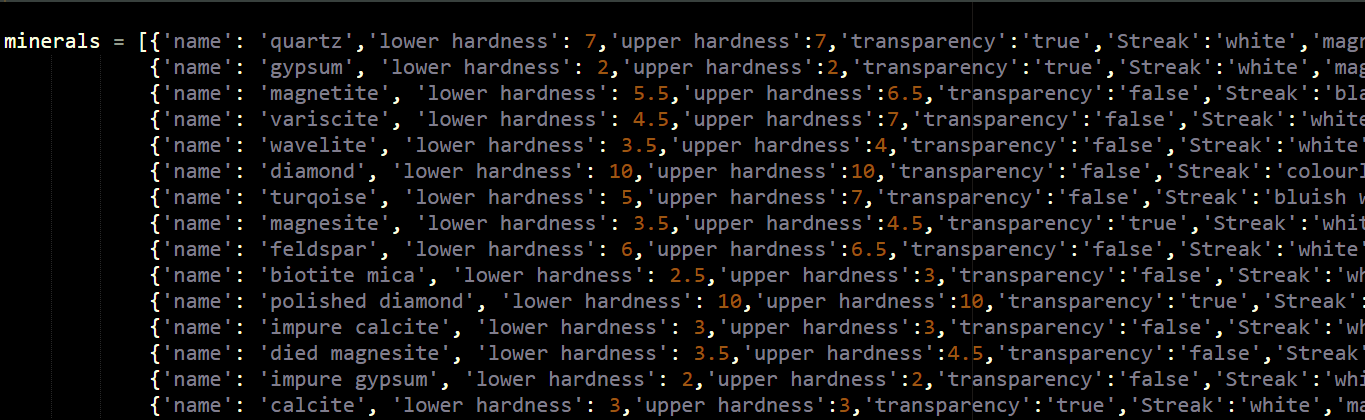
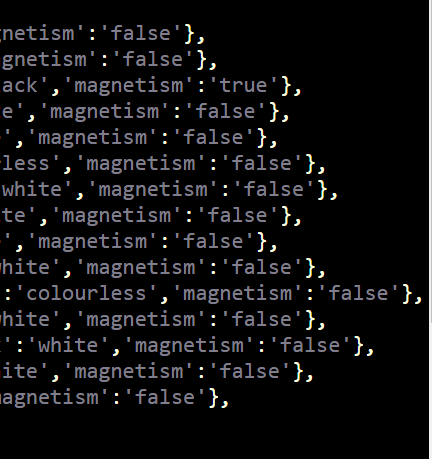
Normally in python ‘print’ is used to display text. However, a microbit does not have the same display as a computer monitor so the object display has to be used. .scroll tells the microbit that it needs to display something (‘magnetic’) tells the microbit that it needs to display the string ‘magnetic’ and that magnetic isn’t a variable. This is important because python allows a microbit to work out the value of a variable and then display it.

*‘’All right, but we still have to look at this code’’*

Of course, let’s start with the data bank. The microbit needs to access a databank so that it can work out what your mineral is.



*‘’is this the data base?’’*





‘’ Yes! The data base assigns every mineral six different variables. Every mineral has its own line in which variables are assigned values and strings’’



*‘’Why are there two different hardness’s for each mineral?’’*



‘’Some minerals have a range of possible hardness’s like biotite mica (the green arrows are pointing to the lower and upper hardness’s of biotite mica). When the program comes to working out what mineral you have it can take in to account the possible variation of hardness from one specimen to a different specimen of the same mineral by working out whether the hardness variable that you have given it is equal to or in between the upper hardness and the lower hardness.’’



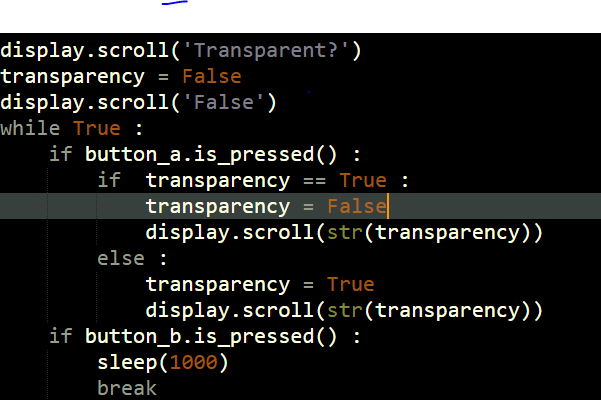
*‘’How does this program work out what mineral you have? i mean it’s not like it can read your mind or anything ’’*



‘’One of the biggest problems I had to face was the fact that a microbit doesn’t have a keyboard; it only has two buttons. In short, there is no way to type anything into a microbit and the only inputs it has are two buttons. We shall start with the transparency.’’



*‘’This looks like it! I can see a couple of if statements. Although I can’t tell how it works!’’*





‘’Well, the microbit uses ‘display.scroll’ to ask what the transparency is, blue arrow, and to tell the user what they have set the value to (green and yellow arrows). Python allows the microbit to display the value of a variable without having to be told what the variable is in the ‘display.scroll’ line of text (yellow arrows). (str( turns the value of the variables into a string so that python can display this. However, the variable does need to be set beforehand (it can be set with a calculation with different variables making python better than a calculator in some ways)”



*‘’But what does the rest of the chunk of code do?’’*



“The chunk of code allows whoever is running the program to change what the variable transparency is set to by pressing the A button and to move on to the next bit of code by pressing the B button.”



*‘’Yeah, and why is there some times one equal sign used and sometimes two equal signs used’’*



‘’in python equal signs have two jobs, to set a variable, orange arrows, and to compare a variable (grey arrows). To make it easy for the program to distinguish between the two you have to use one equal sign to set a variable and two to compare a variable”

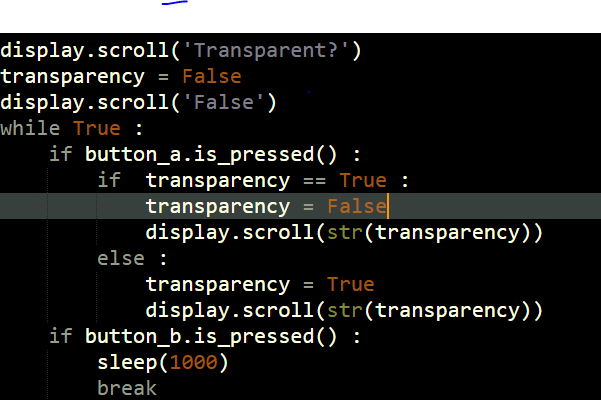


*‘’Ok, but what does ‘while true’ mean? What does break mean and why would you want the microbit to go to sleep?’’*



‘’While true makes an infinite loop because true is always true (green arrow). The colon is needed to complete the line so that the microbit knows that this is the end of the line that is creating a loop. Anything indented four lines from the left is included in this while true loop, the same is for ‘if’ loops. In this case there are ‘if’ loops inside another loop, while true loop, so anything inside those loops is indented eight spaces from the left; the code from an if loop inside other loops inside other loops are indented twelve spaces from the left (yellow arrows). Break is used to break out of this infinite loop so one second after button b is pressed the microbit will move on to the next bit of code (orange arrow).”





*‘’But what about sleep? Why would a microbit need to sleep?’’*



‘’Microbits don’t need to sleep ‘sleep’ is used to tell the microbit it needs to wait (blue arrow). The number inside the brackets tells the microbit the amount of mini seconds it needs to wait before moving to the next line of code. Early I mentioned that the microbit will break out of the infinite loop one second after button b was pressed, this why it takes one second.”



*“Surely this one second wait isn’t necessary?”*



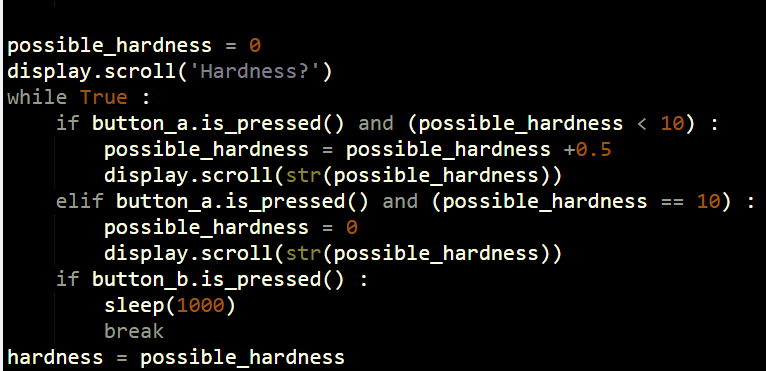
“When I was testing this program I found that the microbit would go through the code so fast that when it came to the next loop it would break out as button B would still be pressed. By adding a wait button B isn’t pressed when the microbit comes to the next loop.”





“Let’s move onto the hardness.”





*“why have you set hardness without waiting for an imput or asking what the hardness is?”*



“If you see later on in the script I have written ‘possible\_hardness =possible\_hardness +0.5’, yellow arrow, which is just me telling it to add 0.5 to the variable possible hardness. The microbit wouldn’t be able to do this if I hadn’t set the variable beforehand.”





“You may also have noticed I used possible hardness instead of hardness and then set hardness as possible hardness (orange arrow). This isn’t necessary for the code as you could just replace possible hardness with hardness and not have hardness = possible hardness. However, I left it there because it’s easier to see what’s going on.”



*“What does ‘elif’ mean? Is it some kind of if statement.”*



“Yes it is a type of if statement, it’s short for else if. This means that when the microbit goes through this infinite loop it thinks ‘first if statement: button A is pressed possible hardness is not under ten, let’s move on. Else if statement: button a is pressed and possible hardness is ten so I need to set possible hardness to 0 and display 0. This is also a bit of coding that means that if you accidently go over your intended number you can easily cycle through and get to your number again (dark blue arrow)’’



*“But why doesn’t the program allow you to set the hardness above ten and what do the numbers mean anyway?”*

*“The hardness NUMBERS ARE from the moh’s scale of hardness in which ten different minerals define different numbers on the scale. For example, gypsum defines number 2 on the mohs scale and calcite defines number 3 on the mohs scale. a mineral can scratch any other mineral with a lower hardness than it but it can’t scratch any mineral with a higher hardness than it.”*



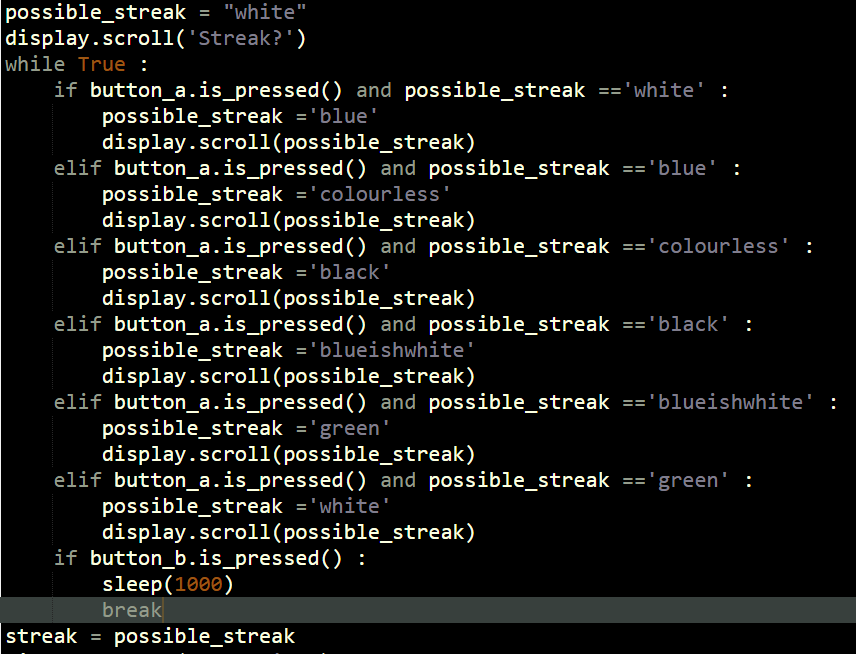
“Yes that’s right! Diamond is the hardest mineral in the world and it also defines number ten on the Moh’s scale of hardness so therefore you can’t find a mineral with a hardness higher than ten!’’





“Now we should move on to the streak.’’





*“this looks really complicated!”*



“It’s not as complicated as it looks. Believe it or not everything in this chunk of code has been covered before, just with different variables and more lines of code than the others.’’





“What the program does is it asks what the streak is. After that, it lets you cycle through the different possible inputs before you press b to break out of it and set the hardness.’’

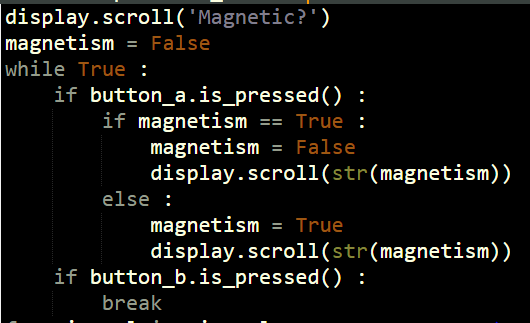


*“WOW is that all?”*



“Yes it is! Now we need to move on to the magnetism input.”





*“Why does this look familiar?”*



“It looks familiar because it is the same code as the transparency input just with different variables and it asks whether the mineral is magnetic or not.”

