**Slide 2**:

You're a geologist working at NASA. You're responsible for analyzing the different types of rocks found by the Artemis Rover and producing a report. The rover will send a text document of the different types of rocks, which you'll need to parse through. Your friend tells you how the Python programming language can make the task much easier than doing it by hand, and more automated.

In this module, you'll build a simple notebook with Python to complete the task. Along the way, you'll apply a variety of introductory computer science topics. You'll have the option to use Visual Studio Code with the Python extension or the Python notebook integration right here on Learn. Either way you will run the code and use pre-built functions that will help you complete this exercise.

**Slide 6**:

To start our program, we'll want to let the user know that the program is beginning, by using the print command to output a string of text.

If you are completing this module on your local computer using Visual Studio Code, then you have a few extra steps you need to do before you get started:

First, make a folder somewhere on your computer that you can access easily. Call the folder something like SpaceRockProject. Then open the file [rocks.txt](https://docs.microsoft.com/en-us/learn/modules/resources/rocks.txt?azure-portal=true) and use file -> save as to save a copy in your folder. This file contains all of the rocks data that we're going to be reading in Python.

**Slide 7**:

Next, open Visual Studio Code and select **File** then **New File**. Once the blank file is open, make the keystrokes "Ctrl+S" to save it. Navigate to the folder you created, name the file something informative such as ArtemisRockClassifier, and change the file type to Jupyter from the drop-down menu. Now we have everything created, we can start coding!

If you are completing this module here on Microsoft Learn, you can skip all of the above instructions and skip directly to executing the following cell.

Run the following code cell to let the user know that the program is beginning:

Next, we want to create some variables that will represent the count of the different types of rocks that Artemis found on the Moon. The specific rocks that we're looking for are: Basalt: The Mare Rock, Breccia: Shocked Rock, Highland Rock: Anorthosite, and Regolith Soil/Surface Layer. These types are the four main types of rocks found on the Moon.

Let's make some variables with names of "basalt", "breccia", "highland", and "regolith". We can also set these variables to 0 since we've not yet counted any rocks. Finally, make a list called rockList that will store the names of every rock the rover found. We can make the list empty, setting it equal to brackets with nothing inside them.

**Slide 8**:

Let's start by making sure we have the code from the previous unit. If you're running this in Visual Studio Code, confirm you have this code in your notebook. If you're running this in Learn, run this code cell to make sure that we have brought this code over to this notebook:  
Let's create a sample file we can use over the course of this module. This will contain a list of rocks.

Run this code cell to:

1. Specify the text file we will write to
2. Open the file
3. Write text to the file
4. Close the file

Now that we have some variables made, we can start giving these variables values. We want to count the number of times we see a certain type of rock in the rocks.txt file and add that number to the corresponding variable. For example, if there are 12 basalt rocks that Artemis saved to the text file, then we'll want the basalt variable to be set to 12.

Let's start by telling the computer to read in the rocks.txt file. To do this, run the code below. The readline function will read the first line, which is just telling us that we are starting to read the rocks that are written in the file.

Now, we can call the readlines() function to put all of the lines into our rockList variable. We'll also print all of the rocks that are in the rockList and close the fileObject.

Although we're only printing out each rock that we see in a file at this stage, this exercise will set us up perfectly to call a function. This function will look at each rock and add to the corresponding variable if it sees a rock of the correct type.  
  
**Slide 9**:

Let's start by making sure we have the code from the previous units loaded in our notebooks. Either run this code cell in the interactive notebook experience on Microsoft Learn, or confirm that you have this in your notebook in Visual Studio Code.  
The next step in our application is to add a function that will look at a line of text, tell us which type of rock it is, and increment the corresponding variable. To start, run this code cell:  
As you recall, these lines will create a function that has a parameter of a string variable rockToID and returns nothing. We'll be calling the function and passing in one of four rock names: Basalt, Breccia, Highland, or Regolith. We'll need to make each rock type a global variable, to ensure that the variables will remember how many rocks they've counted. To increment the correct variable based on the types of rocks that Artemis found, we'll use if statements.

The code above checks if the string we've passed through to the function has the text "basalt". If it does, we'll go into the if statement and print out that we have found a Basalt rock and then increment the variable for Basalt by one.

Note: if the line of text had "basalt" in it, with other text, it would still be counted. For example, "No basalt here" would be counted as one for Basalt. This shouldn't be a problem in this case because we know where the data is coming from and what it will look like.

Before jumping into the solution code, try to add the else if statements to practice your new skills of conditional statements. Then, compare to the solution code here:

**Slide 10**:

Let's start by making sure we have the code from the previous units loaded in our notebook. Either run this code cell in the interactive notebook experience on Microsoft Learn, or confirm that you have this in your notebook in Visual Studio Code.

Now that we have the function made, we'll need to include a call to it so it will be used. We'll create a for loop to go through every value in our rock list and call the function with each rock.

This code will call the function that we've created for each line of text (each rock) that was reported by the rover. The function will add to the count for the corresponding type of rock and store this number in the respective variable.

Now that we have all of the rock counts in their variables, we can use some additional code to write summaries about them.

**Slide 11**:

Let's start by making sure we have the code from the previous units loaded in our notebook. Either run this code cell in the interactive notebook experience on Microsoft Learn, or confirm that you have this in your notebook in Visual Studio Code.  
  
Let's start by printing the number of each type of rock we found. For example, to print the number of basalt rocks, run the following code cell:  
  
As you can see, we first output a string that tells the user what they'll see. Then we concatenate on the variable name that includes the number of basalt rocks we found.

Do the same thing for the four types of rock variables we created. (Solution code is below if you get stuck).

Next, let's provide some more general data, such as the type of rock that was the most and least common. In this case, we'll use the max() and min() functions that we learned about before.

Try to figure out what you need to do, but if you get stuck read below for the answer: