**Download Your Daily Notes**

**My own views on Introduction to Variables**

 Variables are temporary storage spaces that hold different values or data items in a computer's memory, and they are important elements in programming languages. They are created with an initial value and can be in three states: declaration, initialization, and execution. In Python, variables are defined using the assignment character (=), and naming conventions specify how variables should be named to make code more readable. Variable names may contain upper or lower case letters, numbers, or underscores, but they may not start with a number or contain spaces. Identifiers are case sensitive, and variables cannot have the same name as Python's keywords or built-in attributes.

**Daily Notes - Introduction to Variables**

 Variables are temporary storage spaces that hold different values or data items in a computer's memory, and they are important elements in programming languages. They are created with an initial value and can be in three states: declaration, initialization, and execution. In Python, variables are defined using the assignment character (=), and naming conventions specify how variables should be named to make code more readable. Variable names may contain upper or lower case letters, numbers, or underscores, but they may not start with a number or contain spaces. Identifiers are case sensitive, and variables cannot have the same name as Python's keywords or built-in attributes.

**Daily Notes - Using variables**

 Variables must be assigned to a data type, such as a string or an integer. Python automatically assigns a variable to an appropriate data type based on its input or value, and values of the same type can be manipulated together. Sometimes, Python casts values automatically to manipulate them into a common type, but there are cases where values need to be cast explicitly. An example of automatic assignment of variables in Python is shown, where the variables are assigned as integer values and then added together. If a value of a different data type were to be added, such as an integer and a string, an error would occur.

**Daily Notes - Casting**

 The text explains the two ways of casting in Python: implicitly and explicitly. The first one happens when the compiler automatically casts a value from one data type to another when assured that there will be no data loss. The second one occurs when there is data loss risk, and extra code has to be written to ensure that the value stays the same, and only the data type changes. To clarify this, the text presents four examples of casting, which includes casting values between data types such as integers, floats, and strings. The examples illustrate how the casting can be done and what to expect in terms of output. The text also explains the difference between using ‘+’ or ‘,’ when printing variables. Finally, it highlights that an integer can always be cast to a float data type implicitly, but the reverse is not true.

**Daily Notes - Activity 1 - Consolidating Learnings**

 This activity has been completed and has been uploaded.

**My Views on the Day**

 1. Learning how to cast to different variable types.  
  
2. Activity 1  
  
3. Activity 1  
  
4. None

**Daily Notes - Day 1 Reflections**

 1. Learning how to cast to different variable types.  
  
2. Activity 1  
  
3. Activity 1  
  
4. None