**Distinguish between variables and constants**

A variable is a named location used to store data in the memory. It is helpful to think of variables as a container that holds data which can be changed later throughout programming.

Eg: Number= 10

A constant is a type of variable whose value cannot be changed. It is helpful to think of constants as containers that hold information which cannot be changed later.

Non technically, you can think of constant as a bag to store some books and those books cannot be replaced once placed inside the bag.

**Assigning value to a constant in Python**

In Python, constants are usually declared and assigned on a module. Here, the module means a new file containing variables, functions etc which is imported to main file. Inside the module, constants are written in all capital letters and underscores separating the words.

Demonstrate an understanding of the rules to be followed when naming identifiers.

What is an identifier ?

A Python identifier is a name used to identify a variable, function, class, module or other object. An identifier starts with a letter A to Z or a to z or an underscore (\_) followed by zero or more letters, underscores and digits (0 to 9).

Python does not allow punctuation characters such as @, $, and % within identifiers. Python is a case sensitive programming language. Thus, **Manpower** and **manpower** are two different identifiers in Python.

## Rules for writing Identifiers in Python

1. The identifier is a combination of character digits and underscore and the character includes letters in lowercase (a-z), letters in uppercase (A-Z), digits (0-9), and an underscore (\_).
2. An identifier cannot begin with a digit. If an identifier starts with a digit, it will give a Syntax error.
3. In Python, keywords are the reserved names that are built-in to Python, so a keyword cannot be used as an identifier - they have a special meaning and we cannot use them as identifier names.
4. Special symbols like !, @, #, $, %, etc. are not allowed in identifiers.
5. Python identifiers cannot only contain digits.
6. There is no restriction on the length of identifiers.
7. Identifier names are case-sensitive.

**Python Keywords**

Here is the list of some reserved keywords in Python that cannot be used as identifiers.

|  |  |  |  |
| --- | --- | --- | --- |
| False | def | if | raise |
| None | del | import | return |
| True | elif | in | try |
| and | else | is | while |
| as | except | lambda | with |
| assert | finally | nonlocal | yield |
| break | for | not | await |
| class | form | or | async |
| continue | global | pass |  |

As you can see here all the keywords except 'True', 'False', and 'None' are in lowercase, therefore they must be written as they are.

Recommend appropriate data types that should be used to store given data.

In Python, there are several built-in data types that you can use to store values in your programs. Here is a list of the most common data types in Python:

1. **Integers**: These are whole numbers that can be positive, negative, or zero. For example: **42**, **-23**, **0**.
2. **Floating-point numbers**: These are numbers with decimal points. For example: **3.14**, **-9.81**, **0.0**.
3. **Strings**: These are sequences of characters, represented with quotes. You can use single quotes or double quotes to define a string. For example: **'hello'**, **"world"**, **"123"**.
4. **Booleans**: These are True/False values. They are commonly used in control statements to check if a condition is True or False. For example: **True**, **False**.
5. **Lists**: These are ordered sequences of values that can be of any data type. Lists are defined using square brackets and commas. For example: **[1, 2, 3]**, **['apple', 'banana', 'cherry']**, **[True, False, True]**.
6. **Tuples**: These are also ordered sequences of values, but they are immutable (i.e., they cannot be modified). Tuples are defined using parentheses and commas. For example: **(1, 2, 3)**, **('apple', 'banana', 'cherry')**, **(True, False, True)**.
7. **Sets**: These are unordered collections of unique values. Sets are defined using curly braces and commas. For example: **{1, 2, 3}**, **{'apple', 'banana', 'cherry'}**, **{True, False}**.
8. **Dictionaries**: These are unordered collections of key-value pairs. Dictionaries are defined using curly braces and colons. For example: **{'name': 'Alice', 'age': 25}**, **{1: 'apple', 2: 'banana', 3: 'cherry'}**, **{True: 'Yes', False: 'No'}**.

