1. **Write all the conventions being followed while declaring a variable.**

**Ans:-** 1. Use Descriptive Names

* Variable names should be meaningful and descriptive, reflecting the purpose of the variable.
* Example: age, student\_name, total\_sales

### 2. Use Lowercase Letters

* Variables are typically written in all lowercase letters.
* Example: first\_name, total\_price

### 3. Separate Words with Underscores (Snake Case)

* Use underscores (\_) to separate words in multi-word variable names (also known as "snake\_case").
* Example: user\_name, employee\_id

### 4. Avoid Starting with Numbers

* Variable names must start with a letter or an underscore, not a number.
* Example: age is valid, but 1age is invalid.

### 5. No Special Characters Except Underscore

* Apart from underscores (\_), no special characters (like @, !, #) are allowed in variable names.
* Example: user\_id is valid, but user@id is invalid.

### 6. Avoid Python Reserved Keywords

* Variable names must not be any of Python’s reserved keywords, like if, else, for, while, etc.
* Example: class is invalid, but class\_name is valid.
* Reserved keywords include: if, else, for, while, try, except, return, etc.

### 7. Case Sensitivity

* Variable names are case-sensitive, so Name, name, and NAME are treated as different variables.
* Example: age and Age would be different variables.

### 8. Avoid Single-character Variable Names

* Except for temporary or loop variables (like i, j), avoid single-character names as they are not descriptive.
* Example: Use index instead of i, unless it's for simple loop control.

### 9. Constants in All Uppercase

* Although variables can change, if we create a constant (a variable that shouldn't change), it is conventionally named in all uppercase letters.
* Example: PI = 3.14, MAX\_SIZE = 100

### 10. Avoid Using Dashes and Spaces

* Variable names cannot contain dashes (-) or spaces.
* Example: total-sales or total sales is invalid; use total\_sales instead.

### 11. Short, Yet Descriptive

* Variable names should not be overly long, but still descriptive enough to make code understandable.
* Example: x is not descriptive, but num\_students is clearer.

**Q2. What will happen if we declare a restricted keyword as a variable?**

**Ans:** If we declare a restricted (or reserved) keyword as a variable name, it will result in a syntax error. This is because reserved keywords have special meaning in the programming language and are used to define the structure and behaviour of the code. Therefore, they cannot be used as variable names.

**Q3. Can we actually declare a string as a variable name?**

**Ans:-** No, we cannot declare a string as a variable name in most programming languages, including Python. Variable names must follow specific rules, and strings (i.e., text enclosed in quotes) are not valid as variable names.

**Q4. Is it possible for us to declare “\_” as a variable? If so, then write an example of it.**

**Ans:** Yes, you can declare \_ (underscore) as a variable in Python, and it is perfectly valid. The underscore is often used in Python for special purposes, but it can still function as a regular variable name.

Ex:-

for \_ in range(5):

print("Hello")

Here, \_ is used as a placeholder because we don't need the loop variable.

Thus, we can declare \_ as a variable, though it is often used for special purposes in Python.

**Q5. Using an example, explain how the variables in python are dynamic in nature.**

**Ans:-** In Python, variables are dynamic in nature, meaning that the type of a variable can change based on the value assigned to it at runtime. Unlike statically-typed languages (e.g., Java, C++), where the data type must be declared and cannot change, Python allows variables to hold values of different types at different points in the program.

Example:-

x = 10

print(x, type(x))

x = "Hello"

print(x, type(x))

x = [1, 2, 3]

print(x, type(x))

x = 10 assigns the integer value 10 to the variable x. At this point, x is of type int.

x = "Hello" reassigns the string "Hello" to x. Now, the type of x has changed to str (string)

x = [1, 2, 3] assigns a list to x. Now, the type of x has changed to list.