PYTHON- VARIABLES ASSIGNMENT-1

Q1. Write all the conventions being followed while declaring a variable.

Ans . Start with a lowercase letter or an underscore. The first character of a variable name must be a lowercase letter or an underscore. This is because variable names are case-sensitive, so starting with a lowercase letter helps to distinguish them from keywords and other reserved words.

- Use descriptive names. The name of a variable should be descriptive of its purpose. This will make your code more readable and easier to understand for yourself and others. For example, instead of using the name x, you could use the name total number of items to describe a variable that stores the total number of items in a list.
- Use consistent naming conventions. Once you choose a naming convention, stick with it throughout your code. This will help to make your code more consistent and easier to read. Some common naming conventions include:
    - Pascal case: The first letter of each word in the variable name is capitalized. For example,
    - Camel case: The first letter of each word in the variable name is lowercase, except for the first word. For example,
    - Snake case: Each word in the variable name is separated by an underscore. For
- Avoid using keywords or reserved words. There are a number of keywords and reserved words in programming languages that have special meanings. These words should not be used as variable names, as this can cause confusion and errors.
- Avoid using abbreviations. Abbreviations can be confusing, especially if they are not well-known. It is better to use full words in variable names, as this will make your code more readable and understandable.
- Avoid using special characters. Special characters, such as spaces, punctuation marks, and mathematical symbols, should not be used in variable names. These characters can be difficult to type and can cause errors.

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

Ans If you declare a restricted keyword as a variable, the compiler will not allow it. This is because restricted keywords are reserved for special purposes in the programming language. For example, the keyword if is used to control the flow of execution in a program, and the keyword while is used to create loops. If you try to declare a restricted keyword as a variable, the compiler will give you an error message.

In some programming languages, such as C#, you can use a special character, such as an @ symbol, to prefix a restricted keyword and make it valid as a variable name. However, this is not recommended, as it can make your code more difficult to read and understand.

It is important to follow the conventions for declaring variables in your programming language. This will help to make your code more readable, understandable, and maintainable.

Here are some examples of restricted keywords in different programming languages:

- Java: abstract, continue, for, new, switch
- Python: and, del, from, not, while
- C++: break, else, if, return, this

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

Ans string_variable = "this is a string"

print(string_variable)

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

Ans _ as a variable in most programming languages. The underscore character _ is a valid character for a variable name in most programming languages. However, the meaning of the underscore character can vary depending on the programming language.

In some programming languages, the underscore character is used as a convention to denote private variables. For example, in Python, a variable name that starts with an underscore is typically considered to be a private variable. This means that the variable is not accessible outside of the class in which it is defined.

In other programming languages, the underscore character is used as a wildcard. For example, in SQL, the underscore character can be used to match any single character in a string.

Here is an example of how to declare _ as a variable in Python:

```
_ = 10  # This is a valid variable declaration

print(_)  # This will print the value 10
```

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

Ans

```
number = 10  # This is an integer

number = "Hello, world!"  # This is a string

print(number)  # This will print the string "Hello, world!"

number = 10

print(type(number))  # This will print the string "int"

number = "Hello, world!"


print(type(number))  # This will print the string "str"
```