What is an Identifier ? Purpose of an Identifier ?

In programming, an identifier is a name given to entities such as **variables**, **functions**, **classes**, **modules**, or any other user-defined **objects**. Identifiers are used to uniquely identify these entities within a program.

In Python, identifiers follow certain rules:

1. Character Set: Identifiers can consist of letters (both lowercase and uppercase), digits, and underscores (_). However, they cannot start with a digit.

2. Case Sensitivity: Python is case-sensitive, meaning **myVar**, **MyVar**, and **myvar** are all considered different identifiers.

3. Reserved Keywords: Identifiers cannot be the same as Python keywords (also known as reserved words), which have special meanings in the language. Examples of keywords include **if**, **else**, **for**, **while**, **def**, **class**, **import**, etc.

4. Length: There is no limit on the length of an identifier, but it's recommended to keep them reasonably short for readability.

Convention: While not enforced by the Python interpreter, there are conventions for naming identifiers to enhance code readability.
For example:

- Use descriptive names that convey the purpose or meaning of the entity.
- Use lowercase letters for variable names and function names, and separate words with underscores for readability (snake_case).
- Use uppercase letters for constants.
- Use CamelCase for class names.
- Avoid using single-character names except for loop variables (**i**, **j**, **k**).

Examples of valid identifiers:

my_var
variable_1
some_function
PI
MyClass

Examples of invalid identifiers:

```
123var       # Cannot start with a digit
if           # Cannot use reserved keywords as identifiers
some-var     # Hyphens are not allowed
my_var!      # Special characters like '!' are not allowed
```

Following these rules ensures that your identifiers are valid and adhere to Python's naming conventions, leading to more readable and maintainable code.