Title: Understanding Python Keywords and Identifiers

In programming languages, keywords play a crucial role in defining the syntax and structure of the language. Identifiers, on the other hand, are used to name variables, classes, methods, and more. It's important to understand the differences and rules associated with keywords and identifiers in Python.

**Python Keywords:**

Keywords in Python are predefined, reserved words with special meanings to the compiler. They cannot be used as variable names or any other identifier.

Examples of keywords in Python include False, None, class, and while. It's essential to note that keywords are all in lowercase and must be written as they are.

All the keywords **except True, False, and None** are **in lowercase** and they must be written as they are. The list of all the keywords is given below.

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

Some Python keywords **that define the fundamental structure** are False, await, else, import, pass, None, break, except, in, raise, def, class, and more..

**Python Identifiers:**

Identifiers are names given to variables, classes, methods, and other entities in Python.

**For example,**

**programming\_language = 'Python'",**

'programming\_language' is an identifier that holds the value 'Python'.

To write a variable name follow specific rules when naming identifiers in Python.

**Rules for Naming an Identifier:**

## **Rules for Naming an Identifier**

* Identifiers cannot be a keyword.
* Identifiers are case-sensitive.
* It can have a sequence of letters and digits. However, it must begin with a letter or \_. The first letter of an identifier cannot be a digit.
* It's a convention to start an identifier with a letter rather.
* Whitespaces are not allowed.
* We cannot use special symbols like !, @, #, $, and so on.

### **Some Valid and Invalid Identifiers in Python**

| Valid Identifiers | Invalid Identifiers |
| --- | --- |
| score | @core |
| return\_value | return |
| highest\_score | highest score |
| name1 | 1name |
| convert\_to\_string | convert to\_string |

**Best Practices:**

* Use meaningful names for identifiers in Python to enhance code readability.
* Avoid using keywords as identifiers.
* Ensure that identifiers follow the naming conventions.
* Multiple words in an identifier can be separated by underscores for clarity.

**Conclusion:**

Python keywords and identifiers are essential for writing clean and efficient code. By understanding the significance of keywords and following the rules for identifiers, you can elevate your Python programming skills to the next level.

Remember to choose identifiers wisely and adhere to the best practices to enhance the readability and maintainability of your code.