Python

Keywords and Variables:

Keywords:

Keywords are reserved words in Python that have a predefined meaning and cannot be used for anything other than their intended purpose, such as naming variables or functions. They are part of the language syntax and serve specific roles like defining functions, control flow, or handling exceptions.

For example:

- **Control flow:** if, else, elif, for, while
- **Function definition:** def, return
- **Boolean values:** True, False
- **Exception handling:** try, except, finally, raise

We can see all the list of available keywords using below simple python program.

```
import keyword
print(keyword.kwlist)
```

Results:

['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is',

'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

Variables:

Symbolic names that reference values stored in memory. In Python, variables do not need explicit declaration and are created when you assign a value to them. Variable names can contain letters, numbers, and underscores, but they **cannot start with a number** and **cannot use keywords** as names.

Rules for Variables:

- Must start with a letter or underscore (_).
- Can contain letters, numbers, and underscores (_).
- Case-sensitive, so myVar and myvar are different variables.
- Cannot use Python keywords as variable names.

Example:

```
# Creating variables name = "Alice"          # String variable age = 30
# Integer variable is_active = True        # Boolean variable height = 5.6
# Float variable
```