20 May

Python Basic - 1

* 1. 1: What are keywords in python? Using the keyword library, print all the python keywords.
  2. Ans: In Python, keywords are reserved words that have a specific meaning and purpose in the language. These keywords cannot be used as variable names or identifiers because they are already predefined by Python.
  3. import keyword
  4. all\_keywords = keyword.kwlist
  5. print(all\_keywords)

output:

['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']

2: What are the rules to create variables in python?

Ans: In Python, variables are created using a simple assignment statement. The general rules for creating variables in Python are as follows:

A: Variable names must start with a letter or an underscore (\_).

B: Variable names can contain letters (both uppercase and lowercase), digits, and underscores.

C: Variable names are case-sensitive, meaning "myVariable" and "myvariable" would be considered two different variables.

D: Python keywords (reserved words) cannot be used as variable names. For example, you cannot use words like "if," "for," "while," etc., as variable names.

E: Variable names should be descriptive and meaningful, helping to convey the purpose or content of the variable.

F: Avoid using special characters, such as !, @, #, $, %, etc., in variable names.

G: It is recommended to use lowercase letters and underscores to separate words in a variable name for better readability. For example, "my\_variable" instead of "myVariable" or "MyVariable."

3: What are the standards and conventions followed for the nomenclature of variables in python to improve code readability and maintainability?

Ans: In Python, there are several standards and conventions for naming variables that help improve code readability and maintainability. The most widely used standards are defined in the official Python style guide, also known as PEP 8. Here are some key guidelines:

A: Use descriptive names: Variable names should be meaningful and indicate the purpose or content of the variable. Avoid single-letter variable names (except for counters or iterators in loops) or cryptic abbreviations.

B: Use lowercase with underscores: Variable names should be written in lowercase letters, and multiple words should be separated by underscores. This convention is known as snake\_case. For example: user\_name, total\_count.

C: Avoid reserved words: Do not use reserved words or built-in function names as variable names. For example, avoid using names like print, str, or list.

D: Be consistent: Maintain consistent naming conventions throughout your codebase. If you use certain naming patterns or abbreviations for variables, follow them consistently.

E:Use uppercase for constants: If you have constants in your code, they should be written in uppercase letters. For example: MAX\_SIZE, DEFAULT\_TIMEOUT.

F: Avoid single leading underscores: By convention, a single leading underscore indicates that a variable or method is intended for internal use within a class or module. It is a hint to other developers that they should not access or modify it directly unless necessary.

G: Use meaningful variable names for loops: When using loops, choose variable names that describe the purpose of the loop. For example: for item in items, for character in string.

H: Use noun or noun phrases for object-oriented programming: For classes and objects, use noun or noun phrases to describe the entity they represent. For example: Customer, FileParser.

* 1. 4: What will happen if a keyword is used as a variable name?
  2. Ans: If a keyword is used as a variable name in a programming language, it will likely result in a syntax error or a compilation error.
  3. 5: For what purpose def keyword is used?
  4. Ans: The def keyword in Python is used to define a function. It is followed by the name of the function and a set of parentheses that can contain parameters (inputs) to the function.
  5. 6: What is the operation of this special character ‘\’?
  6. Ans: In Python, the backslash ('') character is primarily used as an escape character.

Escape Sequences: Backslashes are used to create escape sequences within strings. Some common escape sequences include:

'\n' represents a newline character.

'\t' represents a tab character.

'\"' represents a double quote character.

'\'' represents a single quote character.

'\\' represents a literal backslash.

Raw Strings: By prefixing a string literal with the letter 'r', you can create a raw string. In a raw string, backslashes are treated as literal characters and are not interpreted as escape characters.

Unicode Escapes: Backslashes can be used to represent Unicode characters using their hexadecimal or octal value. The escape sequence '\u' is followed by four hexadecimal digits, while '\U' is followed by eight hexadecimal digits.

* 1. 7: Give an example of the following conditions:

1. Homogeneous list
2. Heterogeneous set
3. Homogeneous tuple

(i) Homogeneous list:

An example of a homogeneous list would be a list of integers, where all the elements in the list are of the same data type (in this case, integers).

Example: [1, 2, 3, 4, 5]

(ii) Heterogeneous set:

A heterogeneous set contains elements of different data types. It can include integers, strings, booleans, or any other data type.

Example: {1, "apple", True, 3.14}

(iii) Homogeneous tuple:

A homogeneous tuple is a tuple where all the elements have the same data type.

Example: (10, 20, 30, 40, 50)

* 1. 8: Explain the mutable and immutable data types with proper explanation & examples.

Ans: In programming, mutable and immutable are terms used to describe the behavior of data types. Mutable data types can be changed or modified after they are created, while immutable data types cannot be modified once they are created. Let's explore these concepts further with proper explanations and examples.

Mutable Data Types: Mutable data types are those whose values can be altered after they are created. This means you can modify their internal state without creating a new instance. Examples of mutable data types in many programming languages include lists, sets, and dictionaries.

Let's take an example in Python using a list:

my\_list = [1, 2, 3]

print(my\_list) # Output: [1, 2, 3]

# Modifying the list

my\_list.append(4)

print(my\_list) # Output: [1, 2, 3, 4]

my\_list[0] = 5

print(my\_list) # Output: [5, 2, 3, 4]

As shown in the example, the list my\_list can be modified by appending new elements or changing the value of existing elements.

Immutable Data Types: Immutable data types, on the other hand, cannot be changed once they are created. Any operation that appears to modify an immutable data type actually creates a new instance with the modified value. Examples of immutable data types in many programming languages include integers, floats, strings, and tuples.

my\_string = "Hello"

print(my\_string) # Output: Hello

# Modifying the string

new\_string = my\_string + " World"

print(new\_string) # Output: Hello World

print(my\_string) # Output: Hello (original string remains unchanged)

In the example, when we concatenate the original string with " World," a new string is created and assigned to new\_string. The original string, my\_string, remains unmodified.

* 1. 9: Write a code to create the given structure using only for loop.

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

Ans: n=4

for I in range(1,n+1):

print("\*"\*i)

* 1. 10: Write a code to create the given structure using while loop.

|||||||||

|||||||

|||||

|||

|

Ans: n=5

For I in range(5,0,-1):

Print(“|”\*i)