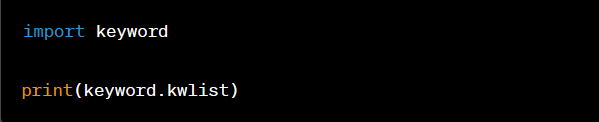
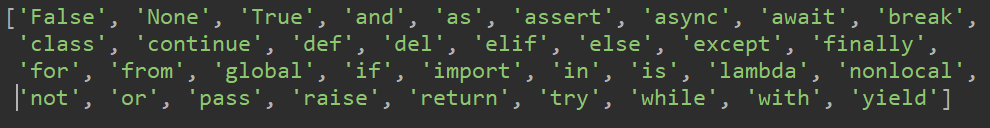
**Q.1. What are keywords in python? Using the keyword library, print all the python keywords.**

**Ans:** *Keywords in Python are reserved words that have special meanings and purposes within the Python programming language. These keywords cannot be used as identifiers (variable names, function names, etc.) because they are reserved for specific language features. They play a fundamental role in defining the structure and logic of Python code.*

**

*You can access a list of Python keywords using the `keyword` module and the `kwlist` attribute. Here's how to print all the Python keywords:*

*When you run this code, it will print the following list of Python keywords:*

**

*These keywords are an integral part of Python's syntax and have specific meanings and usages in the language. You cannot use them as variable names or redefine their functionality***.**

**Q.2. What are the rules to create variables in python?**

**Ans:** *In Python, variable names (identifiers) have to follow certain rules and conventions:*

*1. Valid Characters: Variable names can only contain letters (a-z, A-Z), digits (0-9), and underscores (\_). They must start with a letter or an underscore. Variable names are case-sensitive, which means `myVar` and `myvar` are considered different variables.*

*2. Reserved Keywords: You cannot use Python's reserved keywords (like `if`, `else`, `while`, `for`, `def`, `class`, etc.) as variable names. These words have special meanings in the language.*

*3. No Spaces: Variable names cannot contain spaces. Use underscores or CamelCase to represent multi-word variable names. For example, `my\_variable` or `myVariable`.*

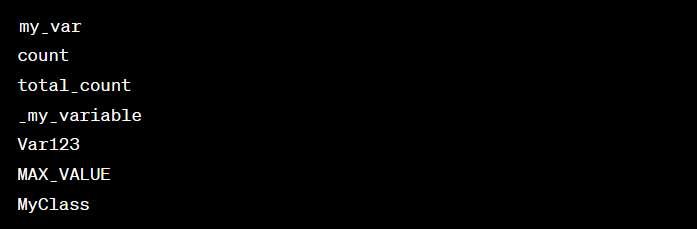
*4. Cannot Start with a Digit: Variable names cannot start with a digit. They must begin with a letter or an underscore.*

*5. Case-Sensitivity: Variable names are case-sensitive. This means that `myVar`, `myvar`, and `MyVar` would be considered different variables.*

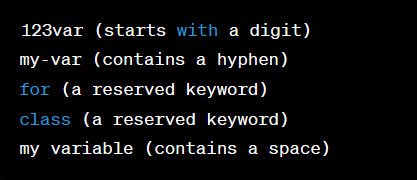
*6. Meaningful Names: It's a good practice to use descriptive and meaningful names for your variables to improve code readability.*

*Here are some examples of valid and invalid variable names:*

*Valid Variable Names:*

****

*Invalid Variable Names:*

****

*Following these rules and conventions for variable names will help you write clean, readable, and error-free Python code.*

**Q.3. What are the standards and conventions followed for the nomenclature of variables in**

**python to improve code readability and maintainability?**

**Ans** :

1. **Limited use of globals:**  
   These rules tell about which types of data that can be declared global and the data that can’t be.
2. **Standard headers for different modules:**  
   For better understanding and maintenance of the code, the header of different modules should follow some standard format and information. The header format must contain below things that is being used in various companies:
   * Name of the module
   * Date of module creation
   * Author of the module
   * Modification history
   * Synopsis of the module about what the module does
   * Different functions supported in the module along with their input output parameters
   * Global variables accessed or modified by the module
3. **Naming conventions for local variables, global variables, constants and functions:**  
   Some of the naming conventions are given below:
   * Meaningful and understandable variables name helps anyone to understand the reason of using it.
   * Local variables should be named using camel case lettering starting with small letter (e.g. **localData**) whereas Global variables names should start with a capital letter (e.g. **GlobalData**). Constant names should be formed using capital letters only (e.g. **CONSDATA**).
   * It is better to avoid the use of digits in variable names.
   * The names of the function should be written in camel case starting with small letters.
   * The name of the function must describe the reason of using the function clearly and briefly.
4. **Indentation:**  
   Proper indentation is very important to increase the readability of the code. For making the code readable, programmers should use White spaces properly. Some of the spacing conventions are given below:
   * There must be a space after giving a comma between two function arguments.
   * Each nested block should be properly indented and spaced.
   * Proper Indentation should be there at the beginning and at the end of each block in the program.
   * All braces should start from a new line and the code following the end of braces also start from a new line.
5. **Error return values and exception handling conventions:**  
   All functions that encountering an error condition should either return a 0 or 1 for simplifying the debugging.

**Q.4. What will happen if a keyword is used as a variable name?**

**Ans***: If a keyword is used as a variable name in Python, it will result in a `SyntaxError`. Keywords in Python are reserved for specific language features and have predefined meanings and behaviors. Attempting to use a keyword as a variable name is not allowed, as it would lead to confusion and conflicts with the intended use of those keywords.*

*For example, if you try to use a keyword like "if" as a variable name:*

**

*You will immediately get a `SyntaxError`:*

**

*To avoid this error, always use valid and meaningful variable names that do not clash with Python's reserved keywords.*

**Q.5. For what purpose def keyword is used?**

**Ans***: The def keyword in Python is used to define functions. Functions are blocks of reusable code that perform a specific task or set of tasks.*

**Q.6. What is the operation of this special character ‘\’?**

**Ans:** *the special character '\' is known as the escape character. It is used to escape or modify the interpretation of characters in a string. When you use the escape character before a character, it changes the character's interpretation, making it different from its literal value.*

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

**(i)**

**Homogeneous list**

**(ii)**

**Heterogeneous set**

**(iii)**

**Homogeneous tuple**

**Ans:**

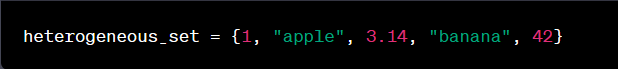
**(i) Homogeneous List:**

*A homogeneous list contains elements of the same data type.*

****

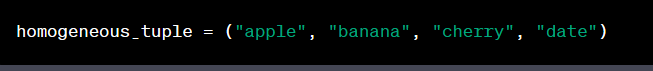
**(ii) Heterogeneous Set:**

*A heterogeneous set can contain elements of different data types.*

****

**(iii) Homogeneous Tuple:**

*A homogeneous tuple contains elements of the same data type.*

****

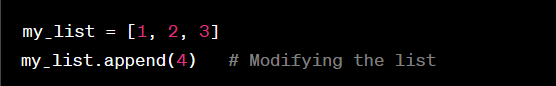
**Q.8. Explain the mutable and immutable data types with proper explanation & examples.**

**Ans:** *In Python, data types can be categorized as either mutable or immutable. This classification is based on whether the data can be changed or modified after it is created. Here's an explanation of both mutable and immutable data types with examples:*

**Mutable Data Types:**

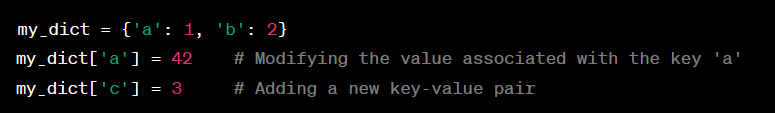
**1. Lists:**

* *Lists are mutable, which means you can modify their content after creation. You can add, remove, or change elements in a list.*
* *Example:*

****

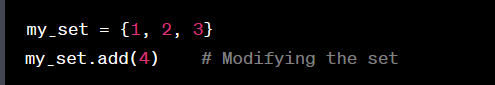
**2. Dictionaries:**

* *Dictionaries are mutable and allow you to change the values associated with keys or add new key-value pairs.*
* *Example:*

****

**3. Sets:**

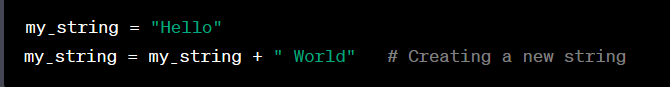
* *Sets are mutable, and you can add or remove elements from a set.*
* *Example:*

****

**Immutable Data Types:**

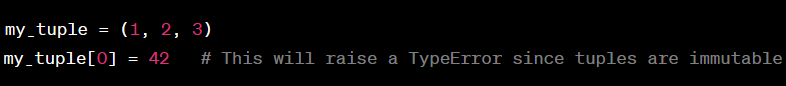
**1. Strings:**

* *Strings are immutable, which means you cannot change their individual characters once they are created. You can, however, create new strings based on the original string.*
* *Example:*

****

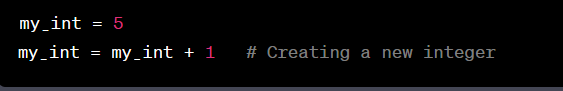
**2. Tuples:**

* *Tuples are immutable, so you cannot change their elements after creation. However, you can create new tuples or use slicing to extract elements.*
* *Example:*

****

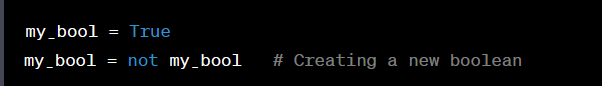
**3. Numbers (int, float):**

* *Numeric data types like integers and floats are also immutable. You cannot change the value of a numeric variable; you can only create new variables.*
* *Example:*

****

**4.Booleans:**

* *Booleans are immutable, and you cannot change `True` to `False` or vice versa. You can create new boolean variables based on existing ones.*
* *Example:*

****

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

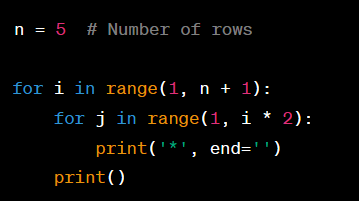
**\***

**\*\*\***

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

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

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

**Ans:** ****

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

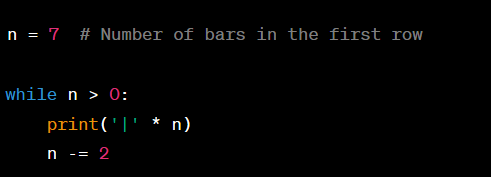
**|||||||||**

**|||||||**

**|||||**

**|||**

**|**

**Ans:** ****