**Question 1**:

Keywords in Python are reserved words that have special meaning and cannot be used as identifiers (such as variable names or function names) in the code. They are used to define the syntax and structure of the language.

import keyword  
  
keywords\_list = keyword.kwlist  
print(keywords\_list)

**Question 2**:

* Variable names can contain letters (a-z, A-Z), digits (0-9), and underscores (\_).
* They cannot start with a digit.
* cannot use Python keywords as variable names.
* Variable names can be of any length, but it's recommended to keep them reasonably short and descriptive.
* They are case-sensitive.
* Whitespaces are not allowed.

**Question 3**:

* Choose variable names that accurately describe their purpose or content.
* Use all lowercase letters with underscores (\_) between words.
* Avoid single-letter names.
* Maintain consistency in naming conventions throughout your codebase.
* Avoid using names that conflict with Python keywords.
* Avoid using ambiguous or cryptic abbreviations in variable names.

**Question 4**:

If a keyword is used as a variable name in Python, it will result in a syntax error.

**Question 5**:

To define a function.

**Question 6**:

* The special character \ in Python is known as the backslash. It is used as an escape character, which means it changes the interpretation of the subsequent character in a string literal.
* Backslash can be used as a line continuation character to split long lines of code into multiple lines.

**Question 7**:

* homogeneous\_list = [1, 2, 3, 4, 5]
* heterogeneous\_set = {1, 'hello', 3.14}
* homogeneous\_tuple = ('apple', 'banana', 'orange')

**Question 8**:

**Mutable Data Type**: Mutable data types are those whose values can be modified after they are created.

Example: List, Dictionary

Example code:

my\_list = [1, 2, 3]  
my\_list[0] = 4 # Modify the first element  
print(my\_list) # Output: [4, 2, 3]

**Immutable Data Type**: Immutable data types are those whose values cannot be modified after they are created.

Example: Tuple

Example code:

my\_tuple = (1, 2, 3)  
my\_tuple[0] = 4 # Attempting to modify the first element raises an error