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

Answer –

Keywords are certain words that have special meaning in python. These keywords are used to define the syntax and structure. These keywords cannot be used as identifiers, which means we can't use them as variable names, function names, etc. because they are predefined and have specific purposes in python.

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

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

Answer –

1. Variable names can contain letters (a-z, A-Z), digits (0-9), and underscores (_). They must start with a letter or an underscore.
2. We cannot use Python reserved keywords as variable names because they have special meanings. For example, if, for, while, def, etc., are keywords and cannot be used as variable names.
3. Variable names are case sensitive. This means that *testvariable* and *TestVariable* will act as two different variables.
4. Variable names cannot start with a number.
5. Variable names cannot contain spaces.

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

Answer –

1. Variable names are usually written in small cases
2. Variable names should be descriptive. This means that the variable name should give an idea about what it's going to store. Example, ***first_name , last_name***
3. Avoid reserved keywords such as if, def, while etc.
4. The variable names should maintain consistency. This means that while working on a particular functionality, the variable names should revolve around the context of the functionality.

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

Answer –

We will get invalid syntax error while execution.

Q 5. For what purpose def keyword is used?

Answer –

def keyword is used to define/declare a user defined function.

Q 6. What is the operation of this special character '\'?

Answer –

It is called an escape character. It is used to represent certain special characters or sequences in strings and other literals. For example, \' represents a quote \" represents double quote.

Q 7. Give an example of the following conditions:

(i) Homogeneous list
(ii) Heterogeneous set
(iii) Homogeneous tuple

Answer –

(i) Homogeneous list - A homogenous list has elements of similar data type.

Example -
```
 test_list = [1,2,3,4,5]
str_list=['a','b','c','d']
```

(ii) Heterogeneous set - A heterogeneous set is a set with elements of different data type.

Example
```
test_set = {1, 'a', 2, 'c', 'apple'}
```

(iii) Homogeneous tuple - A homogeneous tuple contains elements of similar data type.

Example

```
Test_tuple =(1,2,3,4,5)
```

<u>Q 7.</u> Explain the mutable and immutable data types with proper explanation & examples.

<u>Answer</u> –

Mutable data types - Mutable data types are the ones that we can change after they are defined/ declared. When a mutable object is modified, the changes reflect in the same object and no new object is created.

Lists, dictionaries and sets are mutable data types.

Example -

We declare *list_a* and *list_b* and assign *list_a* to *list_b*. Now, if we make changes to *list_a* ,it will reflect in *list_b* as well because the original object got modified.

```
list_a = [1, 2, 3]
list_b = list_a
list_a.append(4)
print(list_a)  # Output: [1, 2, 3, 4]
print(list_b)  # Output: [1, 2, 3, 4]
```

Immutable data types - The values of these data types cannot be changed once they are assigned. When an immutable object is modified, the changes are not made to the original object, rather a new object is created.

String, tuple, int are immutable data types.

Example - Just like above example we have *str1* string and we assign *str1* to *str2*. Now if we modify *str1*, the change will not be reflected in *str2*.

```
str1 = "Hello"
str2 = str1
str1 += " World"
print(str1)  # Output: Hello World
print(str2)  # Output: Hello
```

<u>Q 8.</u> Write a code to create the given structure using only for loop.

```
    *
   ***
  *****
 *******
*********
```

<u>Answer</u> –

```
for a in range(5):
    for b in range(5-i-1):
        print(" ", end="")
    for c in range(2*i+1):
        print("*",end=" ")
    print()
```

<u>Q 9.</u> Write a code to create the given structure using a while loop.

```
|||||||||
 |||||||
  |||||
   |||
    |
```

<u>Answer</u> –

```
rows = 5
space = 0
pipe = rows * 2 - 1

while rows > 0:
    a = 0
    while a < space:
        print(" ", end="")
        a += 1
```

```
b = 0
while b < pipe:
    print("|", end="")
    b += 1



print()



space += 1
pipe -= 2
rows -= 1
```