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

ANSWER: Keywords are the words that are already stored in the system and which cannot be used as variables.

```
In [5]: import keyword
        print(keyword.kwlist)

        ['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'de
        f', '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?

ANSWER:

- A variable name must start with a letter or the underscore character.
- A variable name cannot start with a number.
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _).
- Variable names are case-sensitive (age, Age and AGE are three different variables).

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

ANSWER: Officially, variable names in Python can be any length and can consist of uppercase and lowercase letters (A-Z , a-z), digits (0-9), and the underscore character (_). An additional restriction is that, although a variable name can contain digits, the first character of a variable name can't be a digit.

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

ANSWER: It will throw an error. Refer below code/snap.

```
In [6]: False=20
        print(False)

          File "<ipython-input-6-53acdb97e433>", line 1
            False=20
            ^
        SyntaxError: cannot assign to False
```

5. For what purpose def keyword is used?

ANSWER: def keyword is used to define a function.

```
In [7]: def myfunc():
            print("hello")

        myfunc()

        hello
```

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

ANSWER: It is used for the continuation of your code to next line so that your code is easily visible on the screen and easy to read. Refer below snap/code.

```
In [9]: a=1+2+3+4+5+6+7+\
          8+9+10+12
        print(a)

        67
```

7. Give an example of the following conditions:

(i) Homogeneous list

(ii) Heterogeneous set

(iii) Homogeneous tuple

ANSWER:

I. Homogeneous list:

```
In [10]: list1=[1,2,3] # Homogeneous list contains only data with single datatype or same data type
         print(list1)

         [1, 2, 3]
```

II. Heterogeneous set:

```
In [9]: S1={1,'a',3.3} # Heterogeneous set can store data having different data types such as string, integer, float.
        print(S1)

        {1, 3.3, 'a'}
```

III. Homogeneous tuple:

```
In [10]: t1=(1,2,3) # Homogeneous tuple can store data having same data types.
         print(t1)

         (1, 2, 3)
```

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

ANSWER: Mutable data types are those whose value can be modified once they are created.
Immutable are the ones whose value cannot be modified once they are created.

```
In [16]: # Mutable data types example
         l1=[1,2,3,4]
         l1[2]=5
         print(l1)

         [1, 2, 5, 4]
```

```
In [17]: # Immutable data types example
         t1=(1,2,3,4)
         t1[2]=5
         print(t1)

         ---------------------------------------------------------------------------
         TypeError                                 Traceback (most recent call last)
         <ipython-input-17-3b4eaef0e941> in <module>
               1 # Immutable data types example
               2 t1=(1,2,3,4)
         ----> 3 t1[2]=5
               4 print(t1)

         TypeError: 'tuple' object does not support item assignment
```

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

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

ANSWER:

```
In [3]: for i in range(0,10):
            if i%2!=0:
                print('*'*i)

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

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

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

ANSWER:

```
In [6]: for i in range(10,0,-1):
            if i%2!=0:
                print('|'*i)
```

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