

In [ ]: 20TH may

In [ ]: Q.1. What are keywords **in** python? Using the keyword library, print all the python keywords.  
ANS: In Python, keywords are reserved words that have a special meaning to the interpreter. They cannot be used **as** variable names, function names, **or** any other identifier.  
*# using this we can import all keyword*  
**import** keyword  
print(keyword.kwlist)

In [ ]: Q.2. What are the rules to create variables **in** python?  
ANS: The rules are **for** creating the variable **in** python **is**  
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).  
Reserved words(keywords) **in** Python cannot be used to name the variable **in** Python.  
Here are some examples of valid variable names:  
my\_name, age, student\_roll\_number, total\_marks, average\_score.

In [ ]: Q.3: What are the standards **and** conventions followed **for** the nomenclature of variables **in** python to improve code readability **and** maintainability?  
ANS: Use grammatically correct variable names, the **class** name should start **with** an uppercase **and** must follow camelCase convention If more than two words are to be used.

In [ ]: Q.4: What will happen **if** a keyword **is** used **as** a variable name?  
ANS: Using keywords **as** variable names can make it difficult **for** compilers **and** developers to distinguish between variables **and** keywords  
In Python, using a keyword **as** a variable name will cause a syntax error

In [ ]: Q.5. For what purpose **def** keyword **is** used?  
ANS: In Python, the **def** keyword **is** used to define a function.

In [ ]: Q.6. What **is** the operation of this special character **'\'**?  
ANS: In Python strings, the backslash **"\"** is a special character, also called the "escape" character. It is used in representing certain whitespace characters: **"\t"** is a tab, **"\n"** :

In [ ]: Q.7. Give an example of the following conditions:  
(i) Homogeneous list  
(ii) Heterogeneous set  
(iii) Homogeneous tuple  
  
ANS **1:** Homogeneous\_list=['red','green','blue','orange','pink']  
**2:** Heterogeneous\_set = {1, "hello", **True**}  
**3:** Homogeneous\_tuple=(2,3,4,5,6)

In [ ]: Q.8. Explain the mutable **and** immutable data types **with** proper explanation **&** examples.  
ANS: Mutable data types can be changed **or** have new values assigned to them. Immutable data types cannot be modified after they are created.  
  
In Python, mutable data types include: Lists, Sets, Dictionaries.  
Immutable data types include: Int, Float, Strings, Tuples, Numbers, Unicode, Bool.

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

In [3]: **def** print\_pattern(rows):  
 **for** i **in** range(1, rows + 1):  
 print(" " \* (rows - i) + "\*" \* (2 \* i - 1))  
  
 *# Change the value of rows as needed*  
 rows = 5  
 print\_pattern(rows)

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

In [11]: **def** print\_pattern(rows):  
 **for** i **in** range(rows, 0, -1):  
 print(" " \* (rows - i) + "|" \* (2 \* i - 1))  
  
 *# Change the value of rows as needed*  
 rows = 5  
 print\_pattern(rows)

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

In [ ]:

In [ ]: