

20 May

Python Basic - 1

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

Ans:- Keywords are reserved words that cannot be used as ordinary identifiers. They are used to define the syntax and structure of the Python language. They are case sensitive.

```
import keyword

# display all keywords
print(keyword.kwlist)
```

Output:-

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

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

Ans:-

- 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)

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

Ans:-

- Variable names should be descriptive: Choose meaningful names that reflect the purpose or content of the variable.
- Use lowercase letters and underscores: Variable names should be written in lowercase, with words separated by underscores. For example, user_name or total_count.
- Avoid using single-letter names, except for simple loop variables.

- Avoid reserved words or built-in function names: Don't use names that are already reserved by Python or built-in functions, such as print or list.

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

Ans:- If a keyword is used as a variable name, it will result in a syntax error because keywords are reserved words in the programming language and cannot be used as identifiers.

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

Ans:-

The def keyword is used to define a user-defined function.

Ex:

```
def greet():  
    print("Hello, world!")  
greet()
```

Output:

```
Hello, world!
```

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

Ans:- The '\ ' character is used for continuing the lines of code.

Ex:

```
def greet():  
    print("Hello \  
        world!")  
greet()
```

Output:-

```
Hello world!
```

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

(i) **Homogeneous list**

Ans:-

```
fruits = ["apple", "mango", "banana"]  
print(fruits)
```

Output:-

```
['apple', 'mango', 'banana']
```

(ii) **Heterogeneous set**

Ans:-

```
fruit = {1, "apple", True, 3.14}  
print(fruit)
```

Output:-

```
{1, 3.14, 'apple'}
```

(iii) **Homogeneous tuple**

Ans:-

```
color = ("red", "green", "blue")  
print(color)
```

Output:-

```
('red', 'green', 'blue')
```

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

Ans:-

1. Mutable data types:

It means their elements can be changed (added, modified, or deleted) after they are created without creating a new instance of the data type. Ex. List , Set , Dictionary, etc.

Example:

```
my_list = [1, 2, 3]
my_list.append(4)    # Output [1, 2, 3, 4]
my_list[1] = 5       # Output [1, 5, 3, 4]
print(my_list)
```

In the above example , first we added 4 to the list using append function. Later , we added 5 to index 1 . By this ex we can observe that modification is done easily without creating new instance of datatype.

2. Immutable data types:

It means their elements cannot be changed (no addition, modification, or deletion) after they are created. If you need to modify an immutable object, you create a new object with the desired changes. Ex. String, Tuple, int, etc.

Example:

```
my_tuple = (1, 2, 3)
new_tuple = my_tuple + (4,)
print(new_tuple)    Output (1, 2, 3, 4)
```

In the above example , first my_tuple was declared , but we want to change it , so we created new_tuple and added it with my_tuple. By this ex we can observe that modification is hectic in immutable data types.

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

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

Ans:-

```
n = 5
for i in range(n):
    for j in range(i,n):
        print(" ",end=' ')
    for j in range(i):
        print("*",end=' ')
    for j in range(i + 1):
        print("*",end=' ')
    print()
```

Output:-

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

Q.8. Write a code to create the given structure using while loop.

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

Ans:

```
n = 5
for i in range(n):
    for j in range(i + 1):
        print(" " , end=' ')
    for j in range(i,n-1):
        print("|" , end = ' ')
    for j in range(i,n):
        print("|", end = ' ')
    print()
```

Output:-

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