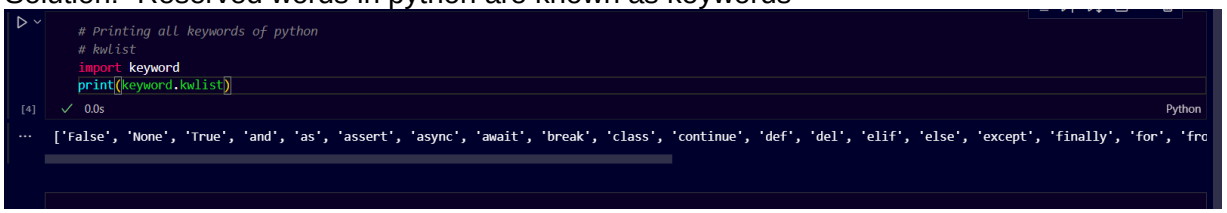
20 May

Python Basic - 1

Q.1. What are keywords in python? Using the keyword library, print all the python keywords.
Solution:- Reserved words in python are known as keywords

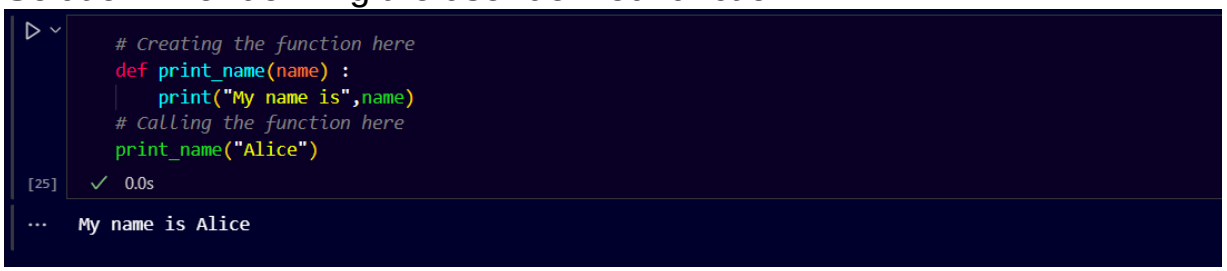


Q.2. What are the rules to create variables in python?
Solutuion:- 1-> Variable should not start with number
2-> In special character only _ is valid all other special character are not valid
3-> Variable name should not be any keyword
4-> Spaces are not allowed in the name of the variable

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

Q.4. What will happen if a keyword is used as a variable name?
Solution:- It will give the invalid syntax error

Q.5. For what purpose def keyword is used?
Solution:- For defining the user defined function



Q.6. What is the operation of this special character '\'?
Solution:- Also known as the escape character and it is used for representing the certain whitespace

Q.7. Give an example of the following conditions:
(i) Homogeneous list
Solution :- Lis containing same type of elements which have same type of data type

(ii) Heterogeneous set

Solution:- Set containing different type of element which have different type of data type

```
Hetero_set={1,"Hello",2,"Byee"}
print(Hetero_set)
```

[32] ✓ 0.0s

```
{'Hello', 1, 2, 'Byee'}
```

(iii) Homogeneous tuple

Solution:- Elements have same data type

```
Homo_tuple=(1,2,3,4,5,6)
print(Homo_tuple)
```

[35] ✓ 0.0s

```
(1, 2, 3, 4, 5, 6)
```

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

Mutable-> List are the mutable data table which means you can add,delete,remove and modify the elmement of the list

```
list=[1,2,3,4,5,6]
list.append(10)
print(list)
list.insert(2,20)
print(list)
list.remove(2)
print(list)
list.pop(4)
print(list)
list[5] = 50
print(list)
```

[41] ✓ 0.0s

```
[1, 2, 3, 4, 5, 6, 10]
[1, 2, 20, 3, 4, 5, 6, 10]
[1, 20, 3, 4, 5, 6, 10]
[1, 20, 3, 4, 6, 10]
[1, 20, 3, 4, 6, 50]
```

Set-> In set you can add and remove particular item but you cannot modify the item so sets are also mutable in nature

```
set={10,20,30,40,50}
set.add(40)
print(set)
set.remove(10)
print(set)
# set[2] = 100 These is not possible you cannot modify the items in the set
```

[48] ✓ 0.0s

```
{50, 20, 40, 10, 30}
{50, 20, 40, 30}
```

Tuple-> Tuple are immutable in nature you cannot modify add , delete the items from the tuple

```
tuple=(100,200,300,400)
tuple[1]=600 # These is not possible tuples are immutable
print(tuple)
```

[50] ⊗ 0.0s

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[50], line 2
      1 tuple=(100,200,300,400)
----> 2 tuple[1]=600
      3 print(tuple)

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

Dictionary-> Dictionary are mutable in nature

```
dict={"one":1,"two":2,"three":3,"four":4}
# Adding the item in the dictionary
dict["five"] = 5
print(dict)
```

[54] ✓ 0.0s

```
{'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5}
```

```
dict={"one":1,"two":2,"three":3,"four":4}
# Deleting the item from the dictionary
dict.pop("two")
print(dict)
```

[55] ✓ 0.0s

```
{'one': 1, 'three': 3, 'four': 4}
```

```
dict={"one":1,"two":2,"three":3,"four":4}
# Deleting the whole ditionary
del dict
print(dict)
```

[56] ✓ 0.0s

```
<class 'dict'>
```

```
dict={"one":1,"two":2,"three":3,"four":4}
# Deleting the entries of the dictonary not the structure of the dictionary
dict.clear()
print(dict)
```

[57] ✓ 0.0s

Strings->String are immutable in nature

```
String = "Hello"
# Strings are immutable in nature
String[1] = "A"
print(String)
```

[58] ⊗ 0.0s

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[58], line 2
      1 String = "Hello"
----> 2 String[1] = "A"
      3 print(String)

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

You can replace the particular character in the string with the help of the replace fucntion

```
String = "Hello"
print(String.replace("e","a"))
```

[60] ✓ 0.0s

```
Hallo
```

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

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

Solution

```
def print_pattern(rows):
    for i in range(rows):

        for j in range(rows - i - 1):
            print(" ", end="")


        for k in range(2*i + 1):
            print("*", end="")


        print()


print_pattern(5)
```

[80] ✓ 0.0s

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

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

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

|||||
|||
|

Sol:-

```
def print_pattern(rows):
    for i in range(rows, 0, -1):
        for j in range(i):
            print("|", end="")
        print()
print_pattern(5)
```

[81] ✓ 0.0s

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