PYTHON PROGRAMMING

UNIT 2

Python Program Flow Control Conditional blocks

LEC-3

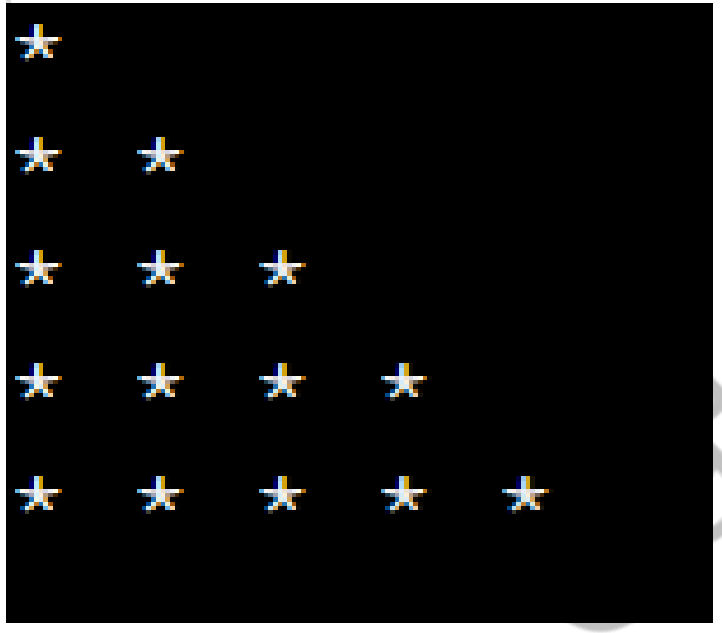
By

Pragya rajvanshi

Gateway classes

Pattern printing-1

```
rows = 5
for i in range(1, rows+1):
    print('* ' * i)
```



Pattern printing-2

```
rows = 5
for i in range(rows,0,-1):
    print('* ' * i)
```



Gateway classes

Pattern printing-3

```
rows = 5
for i in range(1, rows + 1):
    print(' '.join(str(x) for x in range(1, i + 1)))
```

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Pattern printing-4

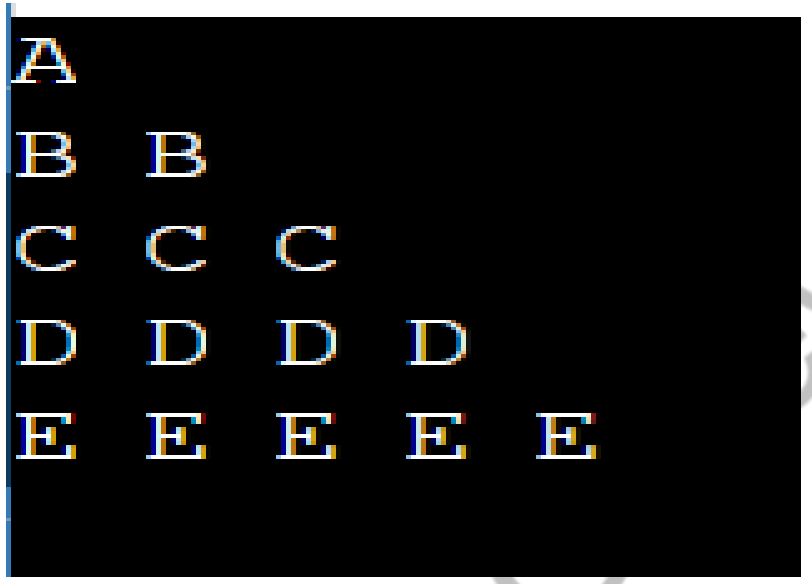
```
rows = 5
for i in range(rows, 0, -1):
    print(' '.join(str(x) for x in range(1, i + 1)))
```

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

Gateway classes

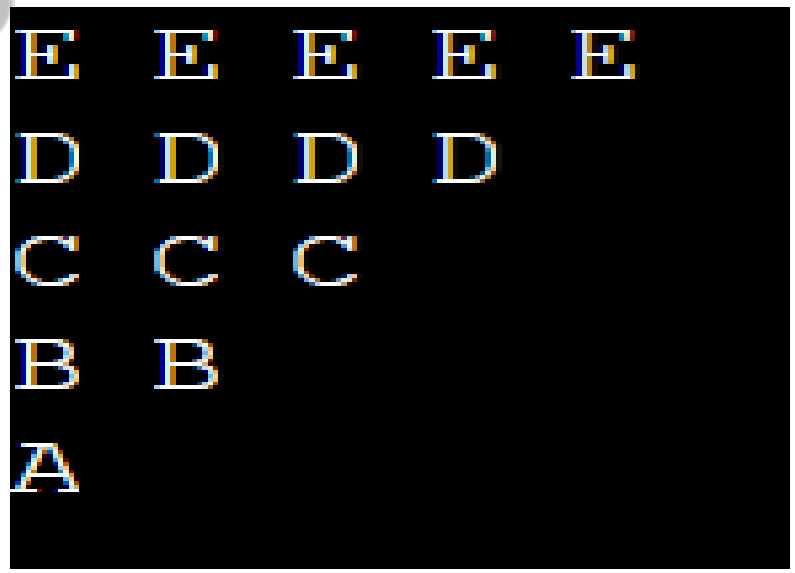
Pattern printing-5

```
rows = 5
for i in range(1, rows + 1):
    print(' '.join(chr(64 + i) for _ in range(i)))
```



Pattern printing-6

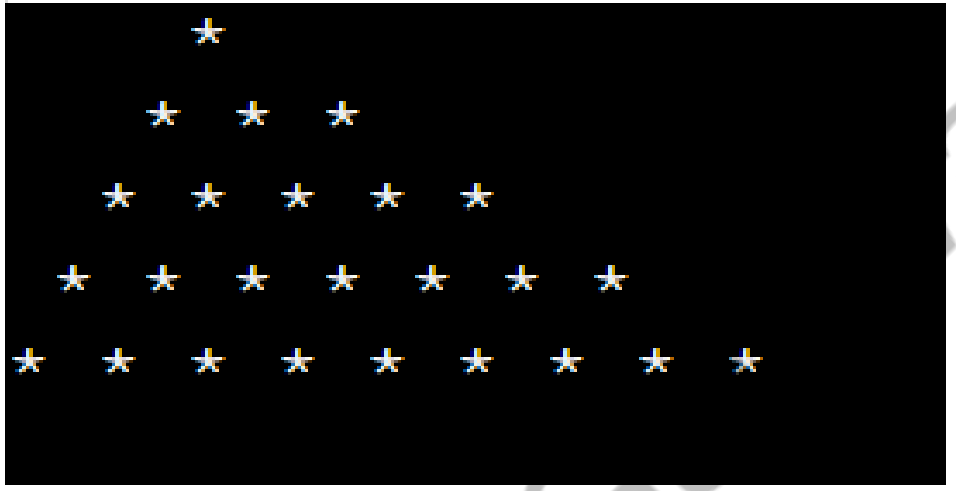
```
rows = 5
for i in range(rows, 0, -1):
    print(' '.join(chr(64 + i) for _ in range(i)))
```



Oneway classes

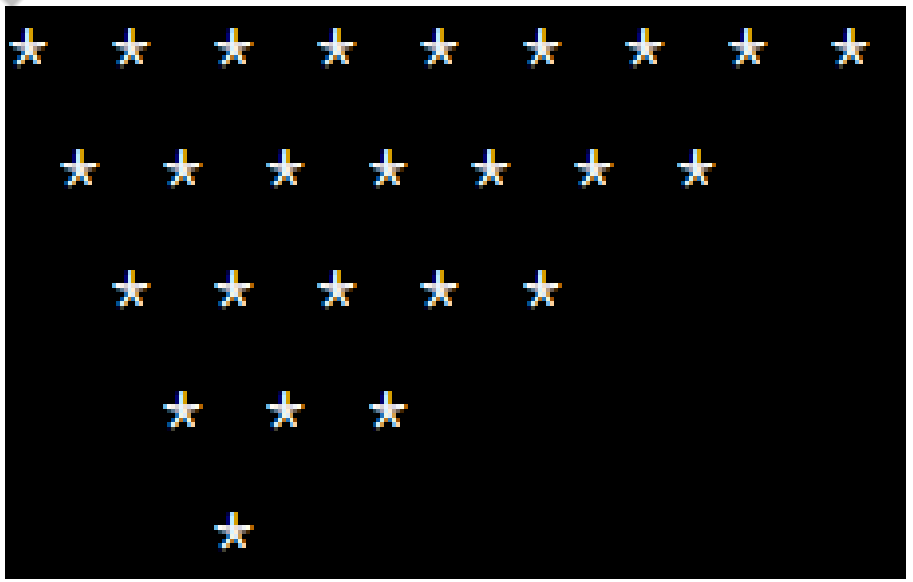
Pattern printing-7

```
rows = 5
for i in range(1, rows + 1):
    print(' ' * (rows - i) + '* ' * (2 * i - 1))
```



Pattern printing-8

```
rows = 5
for i in range(rows,0,-1):
    print(' ' * (rows - i) + '* ' * (2 * i - 1))
```



Gurukay classes

1. Write a Python script to add a key to a dictionary

```
my_dict = {0: 10, 1: 20}
my_dict[2] = 30
print("Updated dictionary:", my_dict)
```

```
Updated dictionary: {0: 10, 1: 20, 2: 30}
```

2. Write a Python script to concatenate/merge the following dictionaries to create a new one.

```
dict1 = {1: 10, 2: 20}
dict2 = {3: 30, 4: 40}
dict3 = {5: 50, 6: 60}
new_dict = {}
for d in (dict1, dict2, dict3):
    new_dict.update(d)
print("Concatenated dictionary:", new_dict)
```

```
Concatenated dictionary: {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

Gateway classes

3. Write a Python script to check whether a given key already exists in a dictionary.

```
my_dict = {1: 10, 2: 20, 3: 30}
key_to_check = 2
if key_to_check in my_dict:
    print("yes")
else:
    print("no")
```

```
yes
```

4.Write a Python program to iterate over dictionaries using for loops

```
dict1 = {'apple': 10, 'banana': 20, 'cherry': 30}
print("Keys:")
for key in dict1:
    print(key)
print("\nValues:")
for value in dict1.values():
    print(value)
print("\nKey-Value Pairs:")
for key, value in dict1.items():
    print(f"{key}: {value}")
```

Gateway Classes

```
Keys:
apple
banana
cherry

Values:
10
20
30

Key-Value Pairs:
apple: 10
banana: 20
cherry: 30
```

5. Write a Python script to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x*x).

```
n = int(input("Enter a number: "))
squared_dict = {x: x*x for x in range(1, n+1)}
print("Generated dictionary:", squared_dict)
```

```
Enter a number: 5
Generated dictionary: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```

6.Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are the square of the keys.

```
squared_dict = {x: x*x for x in range(1, 16)}
print("Generated dictionary:", squared_dict)
```

```
Generated dictionary: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81
, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 15: 225}
```

Gateway Classes

7. Write a Python program to sum all the items in a dictionary

```
my_dict = {'a': 10, 'b': 20, 'c': 30}
total_sum = sum(my_dict.values())
print("Total sum of values:", total_sum)
```

```
Total sum of values: 60
```

8.Write a Python program to Multiply all the items in a dictionary

```
my_dict = {'a': 2, 'b': 3, 'c': 4}
result = 1
for value in my_dict.values():
    result *= value
print("Product of all values:", result)
```

```
Product of all values: 24
```

9. Write a Python program to remove a key from a dictionary.

```
my_dict = {'a': 1, 'b': 2, 'c': 3}
key_to_remove = 'b'
if key_to_remove in my_dict:
    del my_dict[key_to_remove]
print("Updated dictionary:", my_dict)
```

```
Updated dictionary: {'a': 1, 'c': 3}
```

Gateway Classes

10.Write a Python program to map two lists into a dictionary.

```
keys = ['a', 'b', 'c']
values = [1, 2, 3]
mapped_dict = dict(zip(keys, values))
print("Mapped dictionary:", mapped_dict)
```

11 Write a Python program to sort a given dictionary by key.

```
my_dict = {'banana': 3, 'grapes': 1, 'cherry': 2}
sorted_dict = dict(sorted(my_dict.items()))
print("Dictionary sorted by key:", sorted_dict)
```

```
Dictionary sorted by key: {'banana': 3, 'cherry': 2, 'grapes': 1}
```

12. Write a Python program to get the maximum and minimum values of a dictionary.

```
my_dict = {'a': 10, 'b': 20, 'c': 30, 'd': 5}
max_value = max(my_dict.values())
min_value = min(my_dict.values())
print("Maximum value:", max_value)
print("Minimum value:", min_value)
```

```
Maximum value: 30
Minimum value: 5
```

Gateway classes

13. Write a Python program to remove duplicates from the dictionary.

```
my_dict = {'a': 10, 'b': 20, 'c': 10, 'd': 30, 'e': 20}
unique_dict = {}
for key, value in my_dict.items():
    if value not in unique_dict.values():
        unique_dict[key] = value
print("Dictionary without duplicates:", unique_dict)
```

```
Dictionary without duplicates: {'a': 10, 'b': 20, 'd': 30}
```

14 Write a Python program to check if a dictionary is empty or not.

```
my_dict = {}
if my_dict:
    print("The dictionary is  not empty.")
else:
    print("The dictionary is  empty.")
```

```
The dictionary is  empty.
```

15. Write a Python program to combine two dictionary by adding values for common keys.

d1 = {'a': 100, 'b': 200, 'c':300}

d2 = {'a': 300, 'b': 200, 'd':400}

Sample output: Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})

```
from collections import Counter
d1 = {'a': 100, 'b': 200, 'c': 300}
d2 = {'a': 300, 'b': 200, 'd': 400}
combined_dict = Counter(d1) + Counter(d2)
print("Combined dictionary:", combined_dict)
```

Gateway classes

16. Write a Python program to print all distinct values in a dictionary.

Sample Data : [{"V":"S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII":"S005"}, {"V":"S009"},{"VIII":"S007"}]

Expected Output : Unique Values: {'S005', 'S002', 'S007', 'S001', 'S009'}

```
data = [{"V": "S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"},
        {"VII": "S005"}, {"V": "S009"}, {"VIII": "S007"}]
distinct_values = set()
for dictionary in data:
    distinct_values.update(dictionary.values())
print("Distinct values:", distinct_values)
```

```
Distinct values: {'S005', 'S009', 'S007', 'S001', 'S002'}
```

Gateway classes

17. Write a Python program to create and display all combinations of letters, selecting each letter from a different key in a dictionary.
Sample data : {'1':['a','b'], '2':['c','d']}

```
from itertools import product
data = {'1': ['a', 'b'], '2': ['c', 'd']}
combinations = product(*data.values())
for combo in combinations:
    print(''.join(combo))
```

```
ac
ad
bc
bd
```

18. Write a Python program to create a dictionary from a string.
Note: Track the count of the letters from the string

```
sample_string = 'pythonpro'
char_count = {}
for char in sample_string:
    if char in char_count:
        char_count[char] += 1
    else:
        char_count[char] = 1
print(char_count)
```

```
{'p': 2, 'y': 1, 't': 1, 'h': 1, 'o': 2, 'n': 1, 'r': 1}
```

Gateway classes

19. Write a Python program to remove spaces from dictionary keys

```
original_dict = {"first name": "John", "last name": "Doe", "age": 25}
new_dict = {key.replace(" ", ""): value for key, value in original_dict.items()}
print("Dictionary with no spaces in keys:", new_dict)

```

input

```
Dictionary with no spaces in keys: {'firstname': 'John', 'lastname': 'Doe', 'age': 25}
```

20. Write a Python program to check if multiple keys exist in a dictionary.

```
my_dict = {'a': 1, 'b': 2, 'c': 3}
keys_to_check = ['a', 'b']
all_keys_exist = all(key in my_dict for key in keys_to_check)
if all_keys_exist:
    print("All keys exist in the dictionary.")
else:
    print("Not all keys exist in the dictionary.")
```

```
All keys exist in the dictionary.
```

Gateway classes

21. Write a Python program to get the key, value and item in a dictionary

```
my_dict = {'a': 100, 'b': 200, 'c': 300}
for key, value in my_dict.items():
    print(f"Key: {key}, Value: {value}, Item: ({key}, {value})")
```

```
Key: a, Value: 100, Item: (a, 100
Key: b, Value: 200, Item: (b, 200
Key: c, Value: 300, Item: (c, 300
```

22 Write a Python program to print a dictionary line by line.

```
my_dict = {'a': 100, 'b': 200, 'c': 300}
for key, value in my_dict.items():
    print(f"{key}: {value}")
```

```
a: 100
b: 200
c: 300
```

23. Write a Python program to replace dictionary values with their sums.

```
my_dict = {'a': [1, 2, 3], 'b': [4, 5], 'c': [6, 7, 8]}
for key in my_dict:
    my_dict[key] = sum(my_dict[key])
print(my_dict)
```

```
{'a': 6, 'b': 9, 'c': 21}
```

Gateway classes

Dictionary

24.Write a Python program to create a dictionary of keys x, y, and z where each key has as value a list from 11-20, 21-30, and 31-40 respectively. Access the fifth value of each key from the dictionary.

{'x': [11, 12, 13, 14, 15, 16, 17, 18, 19],

'y': [21, 22, 23, 24, 25, 26, 27, 28, 29],

'z': [31, 32, 33, 34, 35, 36, 37, 38, 39]}

15

25

35

x has value [11, 12, 13, 14, 15, 16, 17, 18, 19]

y has value [21, 22, 23, 24, 25, 26, 27, 28, 29]

z has value [31, 32, 33, 34, 35, 36, 37, 38, 39]

```
my_dict = {
    'x': list(range(11, 20)),
    'y': list(range(21, 30)),
    'z': list(range(31, 40))
}
# Accessing the fifth value from each key
print(my_dict['x'][4])  # 15
print(my_dict['y'][4])  # 25
print(my_dict['z'][4])  # 35
# Printing the entire dictionary with the lists
for key, value in my_dict.items():
    print(f"{key} has value {value}")
```

Gateway classes

25. Write a Python program to match key values in two dictionaries.

Sample dictionary: {'key1': 1, 'key2': 3, 'key3': 2}, {'key1': 1, 'key2': 2}

Expected output: key1: 1 is present in both x and y

```
# Sample dictionaries
dict1 = {'key1': 1, 'key2': 3, 'key3': 2}
dict2 = {'key1': 1, 'key2': 2}

# Compare and find matching keys with the same values
for key in dict1:
    if key in dict2 and dict1[key] == dict2[key]:
        print(f"{key}: {dict1[key]} is present in both x and y")
```

```
key1: 1 is present in both x and y
```

Gateway classes

26.Write a Python program to find the key of the maximum value in a dictionary.

Sample Output:

Original dictionary elements:

{'Theodore': 19, 'Roxanne': 22, 'Mathew': 21, 'Betty': 20}

Finds the key of the maximum and minimum value of the said dictionary:

('Roxanne', 'Theodore')

```
my_dict = {'Theodore': 19, 'Roxanne': 22, 'Mathew': 21, 'Betty': 20}
max_key = max(my_dict, key=my_dict.get)
min_key = min(my_dict, key=my_dict.get)
print("Original dictionary elements:")
print(my_dict)
print((max_key, min_key))
```

```
Original dictionary elements:
{'Theodore': 19, 'Roxanne': 22, 'Mathew': 21, 'Betty': 20}
('Roxanne', 'Theodore')
```

Gateway Classes