Day_26_171123

January 23, 2024

1 Functions

- Arguments can be given using positional and keyword
- While passing arguments first we have to pass positional first and then keyworded arguments

[8]:
```
model = 'Q8'
name = 'Audi'
def car_spec(name, model, year):
    name = 'Auto'
    print(f"{name}")
    print(f"{model}")
    print(f"{year}")
car_spec(name,model,year=2000)
```

```
Auto
Q8
2000
```

[6]:
```
car_spec(year=2019, name='Benz',model = '800')
```

```
Benz
800
2019
```

[12]:
```
def multi(a,b):
    res = a*b
    print("Multiplication of two numbers:",res)
    return 1
ret = multi(3,10)
```

```
Multiplication of two numbers: 30
```

2 Star Patterns

Right Angle triangle

[34]:
```
n = int(input("No of rows want:"))
for i in range(1,n+1):
    for k in range(i):
```

```
        print("*",end=" ")
    print()
```

```
No of rows want: 4
```

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

Reverse Mirror right angled triangle

[43]:
```
n = int(input("No of rows want:"))
for i in range(1,n+1):
    print(" "*(i-1),"*"*n,end=" ")
    n -= 1
    print()
```

```
No of rows want: 9
```

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

Reversed Equilateral Triangle

[70]:
```
n = int(input("No of rows want:"))
for i in range(1,n+1):
    print(" "*(i-1),"*"*n,end=" ")
    n -= 2
    print()
    if n==0:
        print(" "*(i),"*",end=" ")
```

```
No of rows want: 10
```

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

Equilateral Triangle

[76]:
```
n = int(input("No of rows want:"))
for i in range(1,n+1):
    print(" "*n,end=" ")
    n -= 1
    for k in range(i):
        print("*",end=" ")
    print()
```

```
No of rows want: 10
          *
         * *
        * * *
       * * * *
      * * * * *
     * * * * * *
    * * * * * * *
   * * * * * * * *
  * * * * * * * * *
 * * * * * * * * * *
```

Names in Stars

[91]:
```
letter_patterns = {
    'A': ["  *  ", " * * ", "*   *", "*****", "*   *", "*   *"],
    'B': ["**** ", "*   *", "**** ", "*   *", "**** "],
    'C': [" *** ", "*   *", "*    ", "*   *", " *** "],
    'D': ["**** ", "*   *", "*   *", "*   *", "**** "],
    'E': ["*****", "*    ", "**** ", "*    ", "*****"],
    'F': ["*****", "*    ", "**** ", "*    ", "*    "],
    'G': [" ****", "*    ", "*  **", "*   *", " *** "],
    'H': ["*   *", "*   *", "*****", "*   *", "*   *"],
    'I': ["*****", "  *  ", "  *  ", "  *  ", "*****"],
    'J': ["  ***", "   * ", "   * ", "*  * ", " **  "],
    'K': ["*   *", "*  * ", "***  ", "*  * ", "*   *"],
    'L': ["*    ", "*    ", "*    ", "*    ", "*****"],
    'M': ["*   *", "** **", "* * *", "*   *", "*   *"],
    'N': ["*   *", "**  *", "* * *", "*  **", "*   *"],
    'O': [" *** ", "*   *", "*   *", "*   *", " *** "],
    'P': ["**** ", "*   *", "**** ", "*    ", "*    "],
    'Q': [" *** ", "*   *", "* * *", "*  **", " ** *"],
    'R': ["**** ", "*   *", "**** ", "*  * ", "*   *"],
    'S': [" ****", "*    ", " *** ", "    *", "**** "],
    'T': ["*****", "  *  ", "  *  ", "  *  ", "  *  "],
    'U': ["*   *", "*   *", "*   *", "*   *", " *** "],
```

```
    'V': ["*   *", "*   *", " * * ", " * * ", "  *  "],
    'W': ["*   *", "*   *", "* * *", "** **", "*   *"],
    'X': ["*   *", " * * ", "  *  ", " * * ", "*   *"],
    'Y': ["*   *", " * * ", "  *  ", "  *  ", "  *  "],
    'Z': ["*****", "   * ", "  *  ", " *   ", "*****"],
    ' ': ["     ", "     ", "     ", "     ", "     "]
}

name = input("Enter your name: ")
def star_name(name):
    for row in range(5):
        for letter in name.upper():
            if letter in letter_patterns:
                print(letter_patterns[letter][row], end="  ")
            else:
                print(letter_patterns[' '][row], end="  ")
        print()
star_name(name)
```

```
Enter your name:  KGF
*   *   ****  *****
*  *   *      *
***    *  **  ****
*  *   *   *  *
*   *   ***   *
```

[ ]: