list , dict , strings,loops

patterns

In [20]:

```python
n=int(input("enter the number"))
for i in range(0,n):
    for j in range(0,i+1):
        print("*",end=" ")
    print(" ")
```

```
enter the number4
*
* *
* * *
* * * *
```

In [3]:

```python
n=int(input("enter the number"))
for i in range(n+1,0,-1):
    for j in range(0,i-1):
        print("*",end=" ")
    print(" ")
```

```
enter the number4
* * * *
* * *
* *
*
```

In [14]:

```python
n=int(input("enter the number"))
m=2 * n - 2
for i in range(0,n):
    for j in range(0,m):
        print(end=" ")
    m=m-2
    for j in range(0,i+1):
        print("* ",end="")
    print("")
```

```
enter the number4
      *
    * *
  * * *
* * * *
```

In [126]:

```python
n=int(input("enter the number"))
m=2 * n - 2
for i in range(0,n):
    for j in range(0,m):
        print(end=" ")
    m=m-1
    for j in range(0,i+1):
        print("* ",end="")
    print("")
```

```
enter the number4
      *
     * *
    * * *
   * * * *
```

In [132]:

```python
n=int(input("enter the number"))
m=2 * n - 2
for i in range(n,0,-1):
    for j in range(m,0,-1):
        print(end=" ")
    m=m+1
    for j in range(0,i-1):
        print("* ",end=" ")
    print("")
```

```
enter the number5
        *   *   *   *
          *   *   *
            *   *
              *
```

lists

creations of lists

In [21]:

```python
list=[]
print("empty list:",list)
```

```
empty list: []
```

In [22]:

```python
list1=[1,2,3,4,5]
print("elements of list with same type:",list1)
```

```
elements of list with same type: [1, 2, 3, 4, 5]
```

In [23]:

```python
list2=[1,"hello",[1,2,"str"],(1,6)]
print("the list with mixed elements:",list2)
```

the list with mixed elements: [1, 'hello', [1, 2, 'str'], (1, 6)]

lists are ordered

In [85]:

```python
lt=[1,2,3,4]
lt1=[1,2,4,3]
list4=[1,2,3,4]
lt==list4
```

Out[85]:

True

In [86]:

```python
lt==lt1
```

Out[86]:

False

updating the list

In [81]:

```python
list3=[1,4,7,3]
print(list3)
list3[0]=7
print(list3)
list3[1:3]=[6,"hi!"]
print(list3)
```

[1, 4, 7, 3]
[7, 4, 7, 3]
[7, 6, 'hi!', 3]

indexing in lists

In [31]:

```python
print(list1[0])
print(list2[3])
print(list2[2][0])
print(list2[3][1])
```

```
1
(1, 6)
1
6
```

negative indexing

In [35]:

```python
print(list1[-1])
print(list2[-3])
print(list2[-1][0])
print(list2[-2][-1])
```

```
5
hello
1
str
```

slicing

In [77]:

```python
print(list1[:])
print(list2[1:3])
print(list2[-2:-1])
print(list2[3][:])
print(list2[2][0:2])
print(list2[-1::-3])
```

```
[1, 2, 3, 4, 5]
['hello', [1, 2, 'str']]
[[1, 2, 'str']]
(1, 6)
[1, 2]
[(1, 6), 1]
```

size of the list

In [46]:

```python
print(len(list))
print(len(list1))
print(len(list2))
```

```
0
5
4
```

inserting elements into the lists

In [47]:

```python
# using append() func
```

In [52]:

```python
l4=[]
l4.append(3)
print(l4)
l4.append(8)
print(l4)
l4.append([1,6])
print(l4)
l5=[4,6,(0,9)]
l4.append(l5)
print(l4)
```

```
[3]
[3, 8]
[3, 8, [1, 6]]
[3, 8, [1, 6], [4, 6, (0, 9)]]
```

In [53]:

```python
# using insert() func
```

In [55]:

```python
l6=[1,5,"hello"]
l6.insert(0,"hi")
print(l6)
l6.insert(3,[3,6])
print(l6)
```

```
['hi', 1, 5, 'hello']
['hi', 1, 5, [3, 6], 'hello']
```

In [56]:

```python
# using extend() func
```

In [58]:

```python
l7=[1,8,4,5,"python"]
l7.extend([10,11,12])
print(l7)
```

```
[1, 8, 4, 5, 'python', 10, 11, 12]
```

reversing the list

In [60]:

```python
l8=[1,4,45,57,89]
l8.reverse()
print(l8)
```

```
[89, 57, 45, 4, 1]
```

removing elements from the list

In [61]:

```python
# using remove() func
```

In [67]:

```python
l1=[1,2,3,5,[1,2,3,4,5],"latin"]
l1.remove(3)
print(l1)
l1.remove("latin")
print(l1)
l1.remove(1)
print(l1)
```

```
[1, 2, 5, [1, 2, 3, 4, 5], 'latin']
[1, 2, 5, [1, 2, 3, 4, 5]]
[2, 5, [1, 2, 3, 4, 5]]
```

In [68]:

```python
# using pop() func
```

In [71]:

```python
l2=[1,2,3,4,[8,9],(0,10)]
l2.pop()
print(l2)
l2.pop(2)
print(l2)
```

```
[1, 2, 3, 4, [8, 9]]
[1, 2, 4, [8, 9]]
```

list operations

In [91]:

```python
print(l1*2)
print(l1+l2)
print(2 in l5)
print("hello" in l6)
```

```
[2, 5, [1, 2, 3, 4, 5], 2, 5, [1, 2, 3, 4, 5]]
[2, 5, [1, 2, 3, 4, 5], 1, 2, 4, [8, 9]]
False
True
```

iteraring through the list

In [99]:

```python
for i  in l7:
 print(i)
```

```
1
8
4
5
python
10
11
12
```

some func in list

In [114]:

```python
l3=[1,2,3,4,55,76,87]
print(min(l3))
print(max(l3))
print(type(l3))
```

```
1
87
<class 'list'>
```

dict

creating a dict

In [119]:

```python
dict1={}
print(dict1)
dict2={"a":1}
print(dict2)
print(type(dict1))
dict3=dict([(1,"a"),(2,"b")])
print(dict3)
```

```
{}
{'a': 1}
<class 'dict'>
{1: 'a', 2: 'b'}
```

In [ ]:

```python
# accessing and adding the values to dict
```

In [144]:

```python
print(dict2['a'])
print(dict3[1])
print(dict3.values())
dict3["g"]=[1,6,7]
print(dict3)
```

```
1
a
dict_values(['a', 'b'])
{1: 'a', 2: 'b', 'g': [1, 6, 7]}
```

In [134]:

```python
# accessing the items
print(dict2.items())
print(dict3.items())
```

```
dict_items([('a', 1)])
dict_items([(1, 'a'), (2, 'b')])
```

In [148]:

```python
# accessing keys
print(dict2.keys())
print(dict3.keys())
```

```
dict_keys(['a'])
dict_keys([1, 2, 'g', 3])
```

size of dict

In [141]:

```python
print(len(dict2))
print(len(dict3))
```

1
2

duplicates in dict

In [143]:

```python
dict4={
    "a":"harika",
    "b":"kushal",
    "c":"manasa",
    "c":"manasa"
}
print(dict4)
```

{'a': 'harika', 'b': 'kushal', 'c': 'manasa'}

iterating through the dict

In [151]:

```python
for i in dict3:
    print(i)
```

1
2
g
3

In [155]:

```python
for i in dict3.values():
 print(i)
```

a
employee
[1, 6, 7]
employee

In [162]:

```python
for i,j in dict3.items():
    print(i,j)
```

1 a
2 employee
g [1, 6, 7]
3 employee

updating the values of the dict

In [161]:

```python
dict4.update({"emp":"harika"})
print(dict4)
```

```
{'a': 'harika', 'b': 'kushal', 'c': 'manasa', 'emp': 'harika'}
```

deleting the values of the dict

In [171]:

```python
del dict4["b"]
print(dict4)
del dict3["g"]
print(dict3)
```

```
{'c': 'manasa', 'emp': 'harika'}
{1: 'a', 2: 'employee', 3: 'employee'}
```

In [189]:

```python
dict3={"a":"b"}
print(dict3.pop("a"))
print(dict3)
```

```
b
{}
```

In [191]:

```python
dict2.clear()
print(dict2)
```

```
{}
```

some other func in dict

In [200]:

```python
dict7={
    "a":1,
    "b":2,
    "c":3
}
print(len(dict7))
print(type(dict7))
dict7.popitem()
print(dict7)
dict8=dict7.copy()
print(dict8)
```

```
3
<class 'dict'>
{'a': 1, 'b': 2}
{'a': 1, 'b': 2}
```

strings

creation of strings

In [204]:

```python
str="harika"
print(str)
print(type(str))
str1='hari'
print(str1)
str2="""hi hello
welcome to the world of python"""
print(str2)
```

```
harika
<class 'str'>
hari
hi hello
welcome to the world of python
```

indexing

In [211]:

```python
print(str1[0])
print(str2[11])
print(str2[-4])
print(str1[-3])
```

```
h
l
t
a
```

slicing

In [215]:

```python
print(str1[:])
print(str1[2:5])
print(str2[-3:-1])
print(str2[::-2])
print(str1[::-1])
```

```
hari
ri
ho
nhy odrweto mce
le h
irah
```

reassign the values of the string

In [217]:

```
str1[2]="g"
print(str1)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [217], in <cell line: 1>()
----> 1 str1[2]="g"
      2 print(str1)

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

string operations

In [218]:

```
s="harika"
y=" bagadhi"
print(s+y)
```

```
harika bagadhi
```

In [219]:

```
print(s*2)
```

```
harikaharika
```

In [222]:

```
print("z" in s)
```

```
False
```

In [223]:

```
print("j" not in s)
```

```
True
```

In [229]:

```
print("i am %s"%(s))
```

```
i am harika
```

In [234]:

```
print("i am \"harika\"")
```

```
i am "haika"
```

some func in strings

In [236]:

```python
len(str1)
```

Out[236]:

4

In [237]:

```python
str1.isupper()
```

Out[237]:

False

In [238]:

```python
str1.islower()
```

Out[238]:

True

In [239]:

```python
str2.isalpha()
```

Out[239]:

False

In [241]:

```python
str2.isdigit()
```

Out[241]:

False

In [242]:

```python
str2.isspace()
```

Out[242]:

False

In [243]:

```python
str1.isidentifier()
```

Out[243]:

True

In [244]:

```python
str2.isprintable()
```

Out[244]:

```
False
```

In [245]:

```python
str1.isalnum()
```

Out[245]:

```
True
```

In [247]:

```python
st=str1.capitalize()
print(st)
```

```
Hari
```

In [248]:

```python
st=str1.casefold()
print(st)
```

```
hari
```

In [250]:

```python
print(str1.count("a"))
```

```
1
```

In [253]:

```python
print(str1.encode(encoding="UTF8",errors='strict'))
```

```
b'hari'
```

In [255]:

```python
st4=str2.split(" ")
print(st4)
```

```
['hi', 'hello\nwelcome', 'to', 'the', 'world', 'of', 'python']
```

In [257]:

```python
st6=str2.join(str1)
print(st6)
```

```
hhi hello
welcome to the world of pythonahi hello
welcome to the world of pythonrhi hello
welcome to the world of pythoni
```

In [ ]: