In [1]:
```
def greet():
    print('good morning')
    print('hello')
    return
greet()
```

```
good morning
hello
```

In [2]:
```
def add(x,y):
    c=x+y
    return c

add(20,10)
```

Out[2]: 30

In [3]:
```
def add(x,y,z):
    c=x+y+z
    return (c)

add(29,48,36)
```

Out[3]: 113

In [4]:
```
def add(x,y,z):
    c=x+y+z
    return(c)
def greet():
    print('hello')
    print('good morning')
    return
greet()
add(10,29,32)
```

```
hello
good morning
```

Out[4]: 71

In [5]:
```
def add_sub_mul(x,y):
    c=x+y
    d=x-y
    e=x*y
    return c,d,e
result1,result2,result3=add_sub_mul(2,7)
print(result1)
print(result2)
print(result3)
```

```
9
-5
14
```

In [6]:
```
def add(x,y):
    c=x+y
    return(c)
score=add(2,3)
print(score)
```

```
5
```

In [7]:
```
def person(name,age): #position function
    print(name)
    print(age)
person('nit',22)
```

```
nit
22
```

In [8]:
```
def person(name,age): #keyword function
    print(name)
    print(age)
person(name='nit',age=22)
```

```
nit
22
```

In [9]:
```
def person(name,age=18): #default function
    print(name)
    print(age)
person('nit')
```

```
nit
18
```

In [10]:
```
def num(a,*b): #variable key length argument
    print(a)
    print(b)

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

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

In [11]:
```
def add(a,*b):
    c=a
    for i in b:
        c=a+i
        print(c)

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

```
3
4
5
6
```

In [12]:
```
def data(name,**data): #kwargs
    print('name')
    print(data)

data('akhil',age=20,location='hyd')
```

```
name
{'age': 20, 'location': 'hyd'}
```

In [13]:
```
def data(name,**data):
    print(name)
    for i,j in data.items():
        print(i,j)
```

```
data('akhil',age=20,location='hyd')
```

```
akhil
age 20
location hyd
```

In [14]:
```
a=5 #global varible
def something():
    a=9
    print(a)

something()
print(a)
```

```
9
5
```

In [15]:
```
a=5 #using globals

def add():
    a=6
    c=globals()['a']
    d=a+c
    print(d)


add()
```

```
11
```

In [16]:
```
def fib(n): #fibonacci sequence
    a,b=0,1
    if n==1:
        print(a)

    else:
        print(a)
        print(b)
        for i in range(2,n):
            c=a+b
            a=b
            b=c
            print(c)

fib(9)
```

```
0
1
1
2
3
5
8
13
21
```

In [17]:
```
def fact(n): #factorial
    f=1
    for i in range(1,n+1):
        f=f*i
```

```
    return f
fact(5)
```

Out[17]: 120

In [18]:
```
import sys #recursion function
sys.setrecursionlimit(100)

i=0

def wish():
    global i
    i +=1
    print('hello',i)
    wish()

wish()
```

```
hello 1
hello 2
hello 3
hello 4
hello 5
hello 6
hello 7
hello 8
hello 9
hello 10
hello 11
hello 12
hello 13
hello 14
hello 15
hello 16
hello 17
hello 18
hello 19
hello 20
hello 21
hello 22
hello 23
hello 24
hello 25
hello 26
hello 27
hello 28
hello 29
hello 30
hello 31
hello 32
hello 33
hello 34
hello 35
hello 36
hello 37
hello 38
hello 39
hello 40
hello 41
hello 42
hello 43
hello 44
hello 45
hello 46
hello 47
hello 48
hello 49
hello 50
hello 51
hello 52
hello 53
hello 54
hello 55
hello 56
hello 57
hello 58
hello 59
hello 60
```

```
hello 61
hello 62
hello 63
hello 64
hello 65
hello 66
hello 67
hello 68
hello 69
hello 70
hello 71
hello 72
hello 73
hello 74
hello 75
```

```
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
Cell In[18], line 12
      9     print('hello',i)
     10     wish()
---> 12 wish()

Cell In[18], line 10, in wish()
      8 i +=1
      9 print('hello',i)
---> 10 wish()

Cell In[18], line 10, in wish()
      8 i +=1
      9 print('hello',i)
---> 10 wish()

    [... skipping similar frames: wish at line 10 (72 times)]

Cell In[18], line 10, in wish()
      8 i +=1
      9 print('hello',i)
---> 10 wish()

Cell In[18], line 9, in wish()
      7 global i
      8 i +=1
----> 9 print('hello',i)
     10 wish()

File ~\AppData\Roaming\Python\Python312\site-packages\ipykernel\iostream.py:664,
in OutStream.write(self, string)
    655 def write(self, string: str) -> Optional[int]:  # type:ignore[override]
    656     """Write to current stream after encoding if necessary
    657
    658     Returns
   (...)
    662
    663     """
--> 664     parent = self.parent_header
    666     if not isinstance(string, str):
    667         msg = f"write() argument must be str, not {type(string)}"  # typ
e:ignore[unreachable]

RecursionError: maximum recursion depth exceeded
```

In [19]:
```
f=lambda a,b :a+b #lambda function
f(1,2)
```

Out[19]: 3

In [20]:
```
f=lambda a,b:a*b
f(20,3)
```

Out[20]: 60

In [21]:
```
from functools import reduce #filter,map,reduce features
def is_even(n):
    return n % 2==0

def is_odd(n):
    return n % 2!=0

def update(n):
    return n*2

def add_all(a,b):
    return a+b

num=[1,4,3,6,7,1,9,2,6]
even=list(filter(is_even,num))
odd=list(filter(is_odd,num))
double=list(map(update,num))
sums=reduce(add_all,double)
print(odd)
print(even)
print(double)
print(sums)
```

```
[1, 3, 7, 1, 9]
[4, 6, 2, 6]
[2, 8, 6, 12, 14, 2, 18, 4, 12]
78
```

In [22]:
```
from functools import reduce
num=[1,4,3,6,7,1,9,2,6]
even=list(filter(lambda n:n % 2==0,num))
odd=list(filter(lambda n:n%2!=0,num))
double=list(map(lambda n:n*2,num))
sums=reduce(lambda a,b:a+b,double)
print(odd)
print(even)
print(double)
print(sums)
```

```
[1, 3, 7, 1, 9]
[4, 6, 2, 6]
[2, 8, 6, 12, 14, 2, 18, 4, 12]
78
```

In [23]:
```
def my_decorator(func): #decorator
    def wrapper():
        print('good evening')
        func()
        print('bye')
```

```
    return wrapper

@my_decorator
def greet():
    print('hello')

greet()
```

```
good evening
hello
bye
```