In [9]:
```
def am9():
    print('good morning team')
```

In [10]:
```
def am9():
    print('good morning team')
am9()
```

```
good morning team
```

In [11]:
```
def greet():
    print('hello')
    print('good morning')
```

In [12]:
```
def greet():
    print('hello')
    print('good morning')
greet()
```

```
hello
good morning
```

In [13]:
```
def greet():
    print('hello')
    print('good morning')
greet()

def greet():
    print('hello')
    print('good morning')
greet()

def greet():
    print('hello')
    print('good morning')
greet()
```

```
hello
good morning
hello
good morning
hello
good morning
```

In [14]:
```
def greet():

    print('hello good morning boss')

greet()
```

```
hello good morning boss
```

In [15]:
```
def greet():

    print('hello good morning boss')
```

```
greet()

greet()

greet()

greet()
```

```
hello good morning boss
hello good morning boss
hello good morning boss
hello good morning boss
```

In [16]:
```
def add(x,y):
    c = x+y
    print(c)

add(5,6,7,8)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[16], line 5
      2     c = x+y
      3     print(c)
----> 5 add(5,6,7,8)

TypeError: add() takes 2 positional arguments but 4 were given
```

In [17]:
```
def add(x,y):
    c = x+y
    print(c)

add(5,6)
```

```
11
```

In [18]:
```
def add(x,y,z):
    c=x+y+z+m
    print(c)
add(1,4,5)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[18], line 4
      2     c=x+y+z+m
      3     print(c)
----> 4 add(1,4,5)

Cell In[18], line 2, in add(x, y, z)
      1 def add(x,y,z):
----> 2     c=x+y+z+m
      3     print(c)

NameError: name 'm' is not defined
```

In [19]:
```
def add(x,y,z,m):
    c=x+y+z+m
```

```
    print(c)
add(1,4,5,6)
```

```
16
```

In [20]:
```
def greet():
    print('hello')
    print('good morning')
greet()
```

```
hello
good morning
```

In [21]:
```
def add(x,y):
    c = x+y
    print(c)
add(5,6)
```

```
11
```

In [22]:
```
def greet():
    print('hello')
    print('good morning')
greet()

def add(x,y):
    c = x+y
    print(c)
add(5,6)
```

```
hello
good morning
11
```

In [23]:
```
def greet():
    print('hello')
    print('good morning')
def add(x,y):
    c = x+y
    print(c)

add(5,6)
greet()
```

```
11
hello
good morning
```

In [24]:
```
def greet():
    print('hello')
    print('good noon')

def add(x,y):
    c=x+y
    print(c)

def sub(x,y):
    d = x-y
```

```
    print(d)

greet()
add(5,5)
sub(10,5)
```

```
hello
good noon
10
5
```

In [25]:
```
def add_sub(x,y):
    c = x+y
    d = x-y
    print(c)
    print(d)

add_sub(10,5)
```

```
15
5
```

In [26]:
```
def add_sub(x,y):
    c = x+y
    d = x-y
    return c,d

add_sub(10,5)
```

Out[26]: (15, 5)

In [27]:
```
def add_sub(x,y):
    c= x+y
    d= x-y
    return c, d

result = add_sub(5,4)

print(result)
```

```
(9, 1)
```

In [28]:
```
def add_sub(x,y):
    c= x+y
    d= x-y
    return c, d

result1,result2 = add_sub(5,4)

print(result1,result2)
```

```
9 1
```

In [29]:
```
def add(x,y):
    c = x+y
    print(c)
add(5,6)
```

```
11
```

FORMAL ARGUMENT & ACTUAL ARGUMENT

In [30]:
```
def person(name,age):
    print(name)
    print(age)

person('nit',23,34)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[30], line 5
      2     print(name)
      3     print(age)
----> 5 person('nit',23,34)

TypeError: person() takes 2 positional arguments but 3 were given
```

In [31]:
```
def person(name,age):
    print(name)
    print(age)

person('nit',23)
```

```
nit
23
```

In [32]:
```
def person(name,age):
    print(name)
    print(age)

person(23,'nit')
```

```
23
nit
```

In [33]:
```
def person(name,age):
    print(name)
    print(age+1)

person(23,'nit')
```

```
23
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[33], line 5
      2     print(name)
      3     print(age+1)
----> 5 person(23,'nit')

Cell In[33], line 3, in person(name, age)
      1 def person(name,age):
      2     print(name)
----> 3     print(age+1)

TypeError: can only concatenate str (not "int") to str
```

keyword argument

In [34]:
```
def person(name,age):
    print(name)
    print(age+1)

person(age=23, name='nit')
```

```
nit
24
```

In [35]:
```
def person(name,age):
    print(name)
    print(age+1)

person(age1=23, name='nit')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[35], line 5
      2     print(name)
      3     print(age+1)
----> 5 person(age1=23, name='nit')

TypeError: person() got an unexpected keyword argument 'age1'. Did you mean 'age'?
```

In [36]:
```
def person(name,age1):
    print(name)
    print(age1+1)

person(age1=23, name='nit')
```

```
nit
24
```

In [37]:
```
def person(name,age,city):
    print(name)
    print(age+1)
    print(city)
```

```
person(age=23, name='nit', city = 'hyd')
```

```
nit
24
hyd
```

In [ ]:
```
def person(name, age=18):
    print(name)
    print(age)

person('nit', 24)
```

variable length argument

In [39]:
```
def sum(a, b):
    c = a+b
    return c

sum(5,6)
```

Out[39]: 11

In [42]:
```
def sum(a, b):
    c = a+b
    return c

sum(5,6,7,8,9,10)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[42], line 5
      2     c = a+b
      3     return c
----> 5 sum(5,6,7,8,9,10)

TypeError: sum() takes 2 positional arguments but 6 were given
```

In [43]:
```
def sum(a, *b):
    c = a+b
    return c

sum(5,6,7,8,9,10)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[43], line 5
      2     c = a+b
      3     return c
----> 5 sum(5,6,7,8,9,10)

Cell In[43], line 2, in sum(a, *b)
      1 def sum(a, *b):
----> 2     c = a+b
      3     return c

TypeError: unsupported operand type(s) for +: 'int' and 'tuple'
```

In [44]:
```
def sum(a, *b): # 1st argument is fixed but for 2nd argument
    #c = a+b
    print(type(a))
    print(type(b))

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

```
<class 'int'>
<class 'tuple'>
```

In [45]:
```
def sum(a, *b): # 1st argument is fixed & we fetch each value from the tuple & we c
    c = a

    for i in b:
        c = c + i
    print(c)

sum(5,6,7,8,9,20,200,300,400)
```

```
955
```

In [46]:
```
def sum(a, *b): # 1st argument is fixed & we fetch each value from the tuple & we c
    c = a

    for i in b:
        c = c + i
    print(c)

sum(5,6,7,8)
```

```
26
```

- positional argument
- keyword argument
- default
- variable lenght (* at last arg) | (args)
- keyword + variable lenght(kwargs)

In [48]:
```
def person():
    person('ALEX', 36, 'JOHN', 987767)
```

In [49]:
```
def person(name, *data):
    print(name)
    print(data)
person('ALEX',36,'JOHN',987767)
```

```
ALEX
(36, 'JOHN', 987767)
```

In [50]:
```
def person(name,*data):
    print('name')
    print(data)

person('ALEX', age = 36, home_place ='southcity', mob =987767)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[50], line 5
      2     print('name')
      3     print(data)
----> 5 person('ALEX', age = 36, home_place ='southcity', mob =987767)

TypeError: person() got an unexpected keyword argument 'age'
```

In [51]:
```
def person(name,**data):
    print('name')
    print(data)

person('ALEX', age = 36, home_place ='southcity', mob =987767, edu = 'phd')
```

```
name
{'age': 36, 'home_place': 'southcity', 'mob': 987767, 'edu': 'phd'}
```

function arguments we are completed

In [ ]:

global varaibe vs local variable

In [52]:
```
a = 10
print(a)
```

```
10
```

In [53]:
```
a = 10
def something():
    b = 15
    print('in function',b)
    print('out function',a)
```

In [54]:
```
a = 10
def something():
    b = 15
```

```
print('in function',b)
print('out function',a)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[54], line 5
      2 def something():
      3     b = 15
----> 5 print('in function',b)
      6 print('out function',a)

NameError: name 'b' is not defined
```

In [55]:
```
a = 10

def something():
    b = 15
    print('in function',b)

print('out function',a)
```

```
out function 10
```

In [56]:
```
a = 10

def something():
    a = 15

print('in function',a)

print('out function',a)
```

```
in function 10
out function 10
```

In [57]:
```
a = 10

def something():
    b = 15
    print('in function',b)

something()

print('out function',a)
```

```
in function 15
out function 10
```

In [59]:
```
a = 20  #globla var

def something():
    b = 75 # local var
    print('in function',b)
something()

print('out function',a)
```

```
in function 75
out function 20
```

In [60]:
```
# if i want to define global variabel inside the function
a = 10

def something():
    global a
    b = 15 # 15 is converted to local when user assigned global a
    print('in function',b)
    print('gloabl variable', a)
something()
print('out function',a)
```

```
in function 15
gloabl variable 10
out function 10
```

In [63]:
```
x = 20  # Global variable

def update_x():
    global x  # Declare that we are using the global variable x
    x += 5    # Modify the global variable

update_x()
print(x)  # Output: 15
```

```
25
```

In [64]:
```
x = 20  # Global variable

def update_x():
    globals()['x'] += 5  # Access and modify the global variable

update_x()
print(x)  # Output: 15
```

```
25
```

In [65]:
```
import keyword
keyword.kwlist
```

Out[65]:
```
['False',
 'None',
 'True',
 'and',
 'as',
 'assert',
 'async',
 'await',
 'break',
 'class',
 'continue',
 'def',
 'del',
 'elif',
 'else',
 'except',
 'finally',
 'for',
 'from',
 'global',
 'if',
 'import',
 'in',
 'is',
 'lambda',
 'nonlocal',
 'not',
 'or',
 'pass',
 'raise',
 'return',
 'try',
 'while',
 'with',
 'yield']
```

In [74]:
```
def count(lst):
    lst = [1, 2, 3, 4, 8, 9, 10]
    lst
```

In [75]:
```
def count(lst):

    even = 0
    odd = 0

    for i in lst:
        if i%2 == 0:
            even += 1
        else:
            odd +=1
    return even,odd

lst = [1,2,3,4,8,9,10]
even, odd = count(lst)
```

```
print(even)
print(odd)
```

```
4
3
```

In [76]:
```
def count(lst):

    even = 0
    odd = 0

    for i in lst:
        if i%2 == 0:
            even += 1
        else:
            odd +=1
    return even,odd

lst = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13]
even,odd = count(lst)

print("Even Number: {} and odd Number : {}".format(even,odd))
#format is function belongs to string & bydefault you need to pass any parameter
```

```
Even Number: 6 and odd Number : 7
```

In [77]:
```
# in progammin we need to continue these process thats why we need to use loop hear

def fib(n):
    a = 0
    b = 1

    print(a)
    print(b)

    for i in range(0, n):
        c = a + b
        a = b
        b = c

        print(c)
```

In [78]:
```
fib(10)
```

```
0
1
1
2
3
5
8
13
21
34
55
89
```

In [3]:
```
def fib(n):
    print(0)
    print(1)

fib(0)
```

```
0
1
```

Factorical of a Number in Python

In [6]:
```
def fact(n):
    f =1
    for i in range(1, n+1):
        f = f*i
    return f

x = 7
result = fact(x)
print(result)
```

```
5040
```

In [7]:
```
def wish():
    print('hello')
    print('hi')
wish()
```

```
hello
hi
```

In [8]:
```
def wish():
    print('hello')
    print('hi')
    wish()
wish()
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
```

```
hello
hi
hello
hi
hello
hi
hello
hi
hello
hi
hello
hi
hello
hi
```

```
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
Cell In[8], line 5
      3     print('hi')
      4     wish()
----> 5 wish()

Cell In[8], line 4, in wish()
      2 print('hello')
      3 print('hi')
----> 4 wish()

Cell In[8], line 4, in wish()
      2 print('hello')
      3 print('hi')
----> 4 wish()

    [... skipping similar frames: wish at line 4 (2972 times)]

Cell In[8], line 4, in wish()
      2 print('hello')
      3 print('hi')
----> 4 wish()

Cell In[8], line 2, in wish()
      1 def wish():
----> 2     print('hello')
      3     print('hi')
      4     wish()

File ~\anaconda3\Lib\site-packages\ipykernel\iostream.py:664, in OutStream.write(sel
f, 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)}"  # type:ig
nore[unreachable]

RecursionError: maximum recursion depth exceeded
```

In [9]:
```
import sys
sys.getrecursionlimit()
```

Out[9]: 3000

In [11]:
```
import sys
sys.setrecursionlimit(200)
print(sys.getrecursionlimit())
```

```
200
```

In [12]:
```
import sys
sys.getrecursionlimit()
```

Out[12]:
```
200
```

In [13]:
```
def wish():
    print('hello')
    wish()
wish()
```

```
hello
hello
hello
```

```
hello
hello
hello
```

```
hello
hello
hello
```

```
hello
hello
hello
hello
hello
hello
hello
```

```
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
Cell In[13], line 4
      2     print('hello')
      3     wish()
----> 4 wish()

Cell In[13], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

Cell In[13], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

    [... skipping similar frames: wish at line 3 (172 times)]

Cell In[13], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

Cell In[13], line 2, in wish()
      1 def wish():
----> 2     print('hello')
      3     wish()

File ~\anaconda3\Lib\site-packages\ipykernel\iostream.py:664, in OutStream.write(sel
f, 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)}"  # type:ig
nore[unreachable]

RecursionError: maximum recursion depth exceeded
```

In [14]:

```
import sys
sys.setrecursionlimit(150)
print(sys.getrecursionlimit())
```

```
i = 0

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

```
150
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
hello 76
hello 77
hello 78
hello 79
hello 80
hello 81
hello 82
hello 83
hello 84
hello 85
hello 86
hello 87
hello 88
hello 89
hello 90
hello 91
hello 92
hello 93
hello 94
hello 95
hello 96
hello 97
hello 98
hello 99
hello 100
hello 101
hello 102
hello 103
hello 104
hello 105
hello 106
hello 107
hello 108
hello 109
hello 110
hello 111
```

```
hello 112
hello 113
hello 114
hello 115
hello 116
hello 117
hello 118
hello 119
hello 120
hello 121
hello 122
hello 123
hello 124
hello 125
```

```
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
Cell In[14], line 12
     10     print('hello', i)
     11     wish()
---> 12 wish()

Cell In[14], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

Cell In[14], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

    [... skipping similar frames: wish at line 11 (122 times)]

Cell In[14], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

Cell In[14], line 10, in wish()
      8 global  i
      9 i += 1
---> 10 print('hello', i)
     11 wish()

File ~\anaconda3\Lib\site-packages\ipykernel\iostream.py:664, in OutStream.write(sel
f, 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)}"  # type:ig
nore[unreachable]

RecursionError: maximum recursion depth exceeded
```

In [15]:

```
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
hello 76
hello 77
hello 78
hello 79
hello 80
hello 81
hello 82
hello 83
hello 84
hello 85
hello 86
hello 87
hello 88
hello 89
hello 90
hello 91
hello 92
hello 93
hello 94
hello 95
hello 96
hello 97
hello 98
hello 99
hello 100
hello 101
hello 102
hello 103
hello 104
hello 105
hello 106
hello 107
hello 108
hello 109
hello 110
hello 111
hello 112
```

```
hello 113
hello 114
hello 115
hello 116
hello 117
hello 118
hello 119
hello 120
hello 121
hello 122
hello 123
hello 124
hello 125
```

```
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
Cell In[15], line 8
      6     print('hello', i)
      7     wish()
----> 8 wish()

Cell In[15], line 7, in wish()
      5 i += 1
      6 print('hello', i)
----> 7 wish()

Cell In[15], line 7, in wish()
      5 i += 1
      6 print('hello', i)
----> 7 wish()

    [... skipping similar frames: wish at line 7 (122 times)]

Cell In[15], line 7, in wish()
      5 i += 1
      6 print('hello', i)
----> 7 wish()

Cell In[15], line 6, in wish()
      4 global  i
      5 i += 1
----> 6 print('hello', i)
      7 wish()

File ~\anaconda3\Lib\site-packages\ipykernel\iostream.py:664, in OutStream.write(sel
f, 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)}"  # type:ig
nore[unreachable]

RecursionError: maximum recursion depth exceeded
```

In [16]:
```
import sys
sys.getrecursionlimit()
```

Out[16]: 150

In [17]:
```
def wish():
    print('hello')
    wish()
wish()
```

```
hello
hello
hello
```

```
hello
hello
hello
```

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

```
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
Cell In[17], line 4
      2     print('hello')
      3     wish()
----> 4 wish()

Cell In[17], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

Cell In[17], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

    [... skipping similar frames: wish at line 3 (122 times)]

Cell In[17], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

Cell In[17], line 2, in wish()
      1 def wish():
----> 2     print('hello')
      3     wish()

File ~\anaconda3\Lib\site-packages\ipykernel\iostream.py:664, in OutStream.write(sel
f, 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)}"  # type:ig
nore[unreachable]

RecursionError: maximum recursion depth exceeded
```

In [18]:

```
import sys
sys.setrecursionlimit(150)
print(sys.getrecursionlimit())

i = 0

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

```
    wish()
wish()
```

```
150
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
hello 76
hello 77
hello 78
hello 79
hello 80
hello 81
hello 82
hello 83
hello 84
hello 85
hello 86
hello 87
hello 88
hello 89
hello 90
hello 91
hello 92
hello 93
hello 94
hello 95
hello 96
hello 97
hello 98
hello 99
hello 100
hello 101
hello 102
hello 103
hello 104
hello 105
hello 106
hello 107
hello 108
hello 109
hello 110
hello 111
```

```
hello 112
hello 113
hello 114
hello 115
hello 116
hello 117
hello 118
hello 119
hello 120
hello 121
hello 122
hello 123
hello 124
hello 125
```

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

Cell In[18], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

Cell In[18], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

    [... skipping similar frames: wish at line 11 (122 times)]

Cell In[18], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

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

File ~\anaconda3\Lib\site-packages\ipykernel\iostream.py:664, in OutStream.write(sel
f, 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)}"  # type:ig
nore[unreachable]

RecursionError: maximum recursion depth exceeded
```

FACTORICAL USING RECURSSION

In [24]:
```
def fact(n):
    if n==0:
        return 1
    return n * fact(n-1)

result = fact(5)
```

```
result
```

Out[24]: 120

Anonymous Function | Lambda

Function without name is called -Anonymous Function Or Lambda

In [25]:
```
def square(a):
    return a * a

square(5)
```

Out[25]: 25

In [26]:
```
def square(a):
    return a * a

result = square(9)
print(result)
```

```
81
```

In [27]:
```
f = lambda a : a * a
result = f(5)
result
```

Out[27]: 25

In [28]:
```
f = lambda a : a * a
result = f(9)
result
```

Out[28]: 81

In [29]:
```
f = lambda a, b : a + b
f1 = lambda a, b : a - b

result = f(354,254)
result1 = f1(180,76)

print(result)
print(result1)
```

```
608
104
```

In [30]:
```
f = lambda a, b : a + b
f1 = lambda a, b : a - b
f2 = lambda a,b : a * b

result = f(50,30)
result1 = f1(50,10)
```

```
result2 = f2(78,7)

print(result)
print(result1)
print(result2)
```

```
80
40
546
```

In [31]:
```
f = lambda a, b : a + b
f1 = lambda a, b : a - b

result = f(1,4)
result1 = f1(2, 3)

print(result)
print(result1)
```

```
5
-1
```

In [32]:
```
import keyword
keyword.kwlist
```

Out[32]:
```
['False',
 'None',
 'True',
 'and',
 'as',
 'assert',
 'async',
 'await',
 'break',
 'class',
 'continue',
 'def',
 'del',
 'elif',
 'else',
 'except',
 'finally',
 'for',
 'from',
 'global',
 'if',
 'import',
 'in',
 'is',
 'lambda',
 'nonlocal',
 'not',
 'or',
 'pass',
 'raise',
 'return',
 'try',
 'while',
 'with',
 'yield']
```

In [33]:

```
  Cell In[33], line 1
    [image.png](attachment:image.png)
                                     ^
SyntaxError: invalid syntax
```

Lambda in other function like -filter,map & reduce

In [34]:
```
nums = [3,2,6,8,4,6,2,9]
evens = list(filter(is_even, nums))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[34], line 2
      1 nums = [3,2,6,8,4,6,2,9]
----> 2 evens = list(filter(is_even, nums))

NameError: name 'is_even' is not defined
```

In [35]:
```
def is_even(n):
    return n % 2 == 0

nums = [3,2,6,8,4,6,2,9]

evens = list(filter(is_even, nums))
print(evens)
```

```
[2, 6, 8, 4, 6, 2]
```

In [39]:
```
def is_odd(n):
    return n % 2 != 0

nums = [3,2,6,8,4,6,2,9]

odd = list(filter(is_odd, nums))
print(odd)
```

```
[3, 9]
```

In [40]:
```
nums = [3,2,6,8,4,6,2,9]

evens = list(filter(lambda n : n%2 == 0, nums))

print(evens)
```

```
[2, 6, 8, 4, 6, 2]
```

In [41]:
```
nums = [3,2,6,8,4,6,2,9]

odd = list(filter(lambda n : n%2 != 0, nums))

print(odd)
```

```
[3, 9]
```

In [42]:
```
nums = [3,2,6,8,5,23,45,11,180]

evens = list(filter(lambda n : n%2 ==0, nums))
odd = list(filter(lambda n : n%2 !=0, nums))

print(evens)
print(odd)
```

```
[2, 6, 8, 180]
[3, 5, 23, 45, 11]
```

In [43]:
```
def update(n):
    return n+2

nums = [3,2,6,8,4,6,2,9]

evens = list(filter(is_even, nums))
double = list(map(update, evens))

print(evens)
print(double)
```

```
[2, 6, 8, 4, 6, 2]
[4, 8, 10, 6, 8, 4]
```

In [44]:
```
nums = [3,2,6,8,4,6,2,9]

evens = list(filter(is_even, nums))
double = list(map(lambda n : n*2, evens))

print(evens)
print(double)
```

```
[2, 6, 8, 4, 6, 2]
[4, 12, 16, 8, 12, 4]
```

In [45]:
```
nums = [13,22,36,48,54,66,72,89]
evens = list(filter(is_even, nums))

double = list(map(lambda n : n*2, evens))
double_ = list(map(lambda n : n-2, evens))
double1 = list(map(lambda n : n+2, evens))

print(double)
print(double_)
print(double1)
```

```
[44, 72, 96, 108, 132, 144]
[20, 34, 46, 52, 64, 70]
[24, 38, 50, 56, 68, 74]
```

In [46]:
```
from functools import reduce

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

nums = [3,2,6,8,4,6,2]

evens = list(filter(is_even, nums))
double = list(map(lambda n : n*2, evens))

sums = reduce(add_all, double)
sums
print(sums)
```

```
56
```

In [47]:
```
a = [6,8]
print(type(a))
```

```
<class 'list'>
```

In [48]:
```
from functools import reduce

nums = [23,12,46,38,54,76,22,59]

evens = list(filter(is_even, nums))
double = list(map(lambda n : n*2, evens))
sums = (reduce(lambda a,b : a + b, double))
```

```
print(evens)
print(double)
print(sums)
```

```
[12, 46, 38, 54, 76, 22]
[24, 92, 76, 108, 152, 44]
496
```

In [ ]: