In [ ]:

```

```

Functions in Python

FUNCTION --> - Inbuild function -- print(), type(), id(), sqrt() etc - User defined function - function is collection of statement - 2 main property of the function is -- define the function & calling the function - function always define with def & function always declares as () difference between variable & function || a - variable || b() - function

In [134…

```
# when we run the code we havent got any output

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

In [136…

```
#if you need call multiple times

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

```
hello
good morning
```

In [138…

```
#if you need call multiple times

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

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

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

In [140…

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

greet()
```

```
hello
good afternoon
hello
good afternoon
```

In [142…

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

print()# gives space b/w them
```

```
greet()
```

```
hello
good afternoon

hello
good afternoon
```

In [144…

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

```
11
```

In [145…

```
def add(x,y,z):
    c=x+y+z
    print(c)
add(1,4,5)
```

```
10
```

In [146…

```
# you can create mutliple function and call them as many time as you want

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

def add(x,y):
    c=x+y
    print(c)
add(5,4)
```

```
hello
good noon
9
```

In [147…

```
# you can create mutliple function and call them as many time as you want

def greet():
    print('hello')
    print('good noon')
greet()
print()
def add(x,y):
    c=x+y
    print(c)
add(5,4)
```

```
hello
good noon

9
```

In [148…

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

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

def sub(x,y,z):
    d = x-y-z
    print(d)

greet()
add(5,4)
sub(10,2,4)
```

```
hello
good noon
9
4
```

In [149…

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

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

result = add(5,4)
print(result)
```

```
hello
good noon
9
```

In [150…

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

def add(x,y,z):
    c=x+y+z
    return c

greet()

result = add(5,4,3)
print(result)
```

```
hello
good noon
12
```

In [151…

```
def add_sub(x,y): # what if i want to return 2 values add_sub & i want to return
    c= x+y
    d= x-y
    return c, d

add_sub(4,5)
```

Out[151…

```
(9, -1)
```

In [152…

```
def add_sub(x,y): # what if i want to return 2 values add_sub & i want to return
    c= x+y
    d= x-y
    return c, d
```

```
result = add_sub(4,5)

print(result)
print(type(result))
```

```
(9, -1)
<class 'tuple'>
```

In [153…

```
def add_sub(x,y):
    c= x+y
    d= x-y
    return c, d

result1,result2 = add_sub(5,4)

print(result1,result2)

print(type(result1))
print(type(result2))
```

```
9 1
<class 'int'>
<class 'int'>
```

In [154…

```
a,b = 6,7
```

Function Arguments

In [155…

```
def update():
    x = 8
    print(x)
update()
```

```
8
```

In [156…

```
#update function take the value from the user

def update():
    x = 8
    print(x)
update(8)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[156], line 6
      4     x = 8
      5     print(x)
----> 6 update(8)

TypeError: update() takes 0 positional arguments but 1 was given
```

In [ ]:

```
def update(x): #update function take the value from the user
    x = 8
    print(x)

update(100)
```

In [157…

```
 # user want to update the value from 8 to 10

def update(x):
    x = 8
    print(x)

update(10)
```

```
8
```

In [158…

```
def update(x):
    x = 8
    print(x)

a = 10
update(a)
print(a)
```

```
8
10
```

In [159…

```
def update(x):
    x = 8
    #print(x)

a = 10
update(a)
print(a) # this print will update 8 to 5
```

```
10
```

Pass By Value Pass By Referencee

In [208…

```
def change(a):
    a = a+ 10
    print('inside the fun a =',a)

x = 10
print('x before calling:', x)
change(x)
print('x after calling:', x)
```

```
x before calling: 10
inside the fun a = 20
x after calling: 10
```

In [210…

```
def change(a):
    a = a+ 10
    print('inside the fun a =',a)

a = 10
print('a before calling:', a)
change(a)
print('a after calling:', a)
```

```
a before calling: 10
inside the fun a = 20
a after calling: 10
```

In [212…

```
def change(a):
    print('This is original a',id(a))
    a = a+ 10
    print('This is the new a =',a)
    print('inside the fun a =',a)

a = 10
print('a before calling:', a)
```

```python
print('This is main a:',id(a))
change(a)
print('a after calling:', a)
```

```
a before calling: 10
This is main a: 140716705331928
This is original a 140716705331928
This is the new a = 20
inside the fun a = 20
a after calling: 10
```

In [214…

```python
def change(a):
    print('This is original a',id(a))
    a = a+ 10
    print('This is the new a =',id(a))
    print('inside the fun a =',a)

a = 10
print('a before calling:', a)
print('This is main a:',id(a))
change(a)
print('a after calling:', a)
```

```
a before calling: 10
This is main a: 140716705331928
This is original a 140716705331928
This is the new a = 140716705332248
inside the fun a = 20
a after calling: 10
```

In [216…

```python
def change(a):
    #print('This is original a',id(a))
    a = a+ 10
    print('This is the new a =',id(a))
    print('inside the fun a =',a)

a = 10
print('a before calling:', a)
print('This is main a:',id(a))
change(a)
print('a after calling:', a)
print('This is original a',id(a))
```

```
a before calling: 10
This is main a: 140716705331928
This is the new a = 140716705332248
inside the fun a = 20
a after calling: 10
This is original a 140716705331928
```

In [218…

```python
def change(lst):
    lst[0] = lst[0]+10
    print('inside fun =', lst)

lst = [10]
print('Before calling:', lst)
change(lst)
print('After calling:',lst)
```

```
Before calling: [10]
inside fun = [20]
After calling: [20]
```

In [220…

```
def update(x):
    x = 8
    print('x : ', x)

a = 10
update(a)
print('a : ',a)
```

```
x :  8
a :  10
```

In [222…

```
def update(x):
    print(id(x))
    x = 8
    #print(id(x))
    print('x', x)

a = 10
print(id(a))
update(a)
print('a',a)
```

```
140716705331928
140716705331928
x 8
a 10
```

In [224…

```
def update(x):
    #print(id(x))
    x = 8
    print(id(x))
    print('x', x)

a = 10
print(id(a))
update(a)
print('a',a)
```

```
140716705331928
140716705331864
x 8
a 10
```

Expectation & Reality

In [226…

```
# we will understand more when we learn more

def update(x):
    x = 8

    print(id(x))
    print('x', x)

a = 10
print(id(a))

update(a)
print('a',a)
```

```
140716705331928
140716705331864
x 8
a 10
```

In [228…

```python
def update(x):
    print(id(x))
    x = 8
    print(id(x))
    print('x', x)

a = 10
print(id(a))
update(a)
print('a',a)
```

```
140716705331928
140716705331928
140716705331864
x 8
a 10
```

In [230…

```python
def update(lst):
    print(id(lst))

    lst[1] = 25
    print(id(lst))
    print('x', lst)

lst = [10,20,30] #lets pass list hear
print(id(lst))
update(lst)
print('lst',lst)
```

```
2800017456192
2800017456192
2800017456192
x [10, 25, 30]
lst [10, 25, 30]
```

NO concept for pass by value in python (please refer the code below)

In [232…

```python
def modify_integer(x):
    x = 10
    print("Inside function:", x)

my_integer = 5
modify_integer(my_integer)
print("Outside function:", my_integer)
```

```
Inside function: 10
Outside function: 5
```

In [234…

```python
def modify_integer(x):
    x = 10
    print("Inside function:", x)
    print('Inside function:',id(x))

my_integer = 5
modify_integer(my_integer)
print("Outside function:", my_integer)
print('Outside function:',id(x))
```

```
Inside function: 10
Inside function: 140716705331928
Outside function: 5
Outside function: 140716705331928
```

In [236…

```python
def modify_integer(x):
    print('original Inside function:',id(x))
    x = 10
    print("Inside function:", x)
    print('Inside function:',id(x))

my_integer = 5
modify_integer(my_integer)
print("Outside function:", my_integer)
print('Outside function:',id(x))
```

```
original Inside function: 140716705331768
Inside function: 10
Inside function: 140716705331928
Outside function: 5
Outside function: 140716705331928
```

Formal Arguments & Actual Arguments

In [238…

```python
def add(a,b): # a & b called formal argument
    c = a+b
    print(c)

add(5,6) #5 and 6 we called as actual argument
```

```
11
```

In [240…

```python
def add(a,b,d): # a & b called formal argument
    c = a+b+d
    print(c)

add(5,6,7) #5 and 6 we called as actual argument
```

```
18
```

Postional Arguments

In [242…

```python
def person(name,age):
    print(name)
    print(age)

person('nit', 22)
```

```
nit
22
```

In [244…

```python
def person(name,age):
    print(name)
    print(age)

person(22,'nit')
```

```
22
nit
```

In [246…

```python
def person(name,age):
    print(name)
    print(age-5)
```

```
person(20,'nit')
```

```
20
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[246], line 5
      2     print(name)
      3     print(age-5)
----> 5 person(20,'nit')

Cell In[246], line 3, in person(name, age)
      1 def person(name,age):
      2     print(name)
----> 3     print(age-5)

TypeError: unsupported operand type(s) for -: 'str' and 'int'
```

In [248…

```
def person(name,age):
    print(name)
    print(age-5)

person('nit',20)
```

```
nit
15
```

Keyword Arguments

In [250…

```
def person(name,age):
    print(name)
    print(age-5)

person(20,'nit')
```

```
20
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[250], line 5
      2     print(name)
      3     print(age-5)
----> 5 person(20,'nit')

Cell In[250], line 3, in person(name, age)
      1 def person(name,age):
      2     print(name)
----> 3     print(age-5)

TypeError: unsupported operand type(s) for -: 'str' and 'int'
```

In [252…

```
def person(name,age):
    print(name)
    print(age-5)

person(age = 20, name = 'nit')
```

```
nit
15
```

In [254…

```
def person(name, age):
    print(name)
    print(age)
```

```
person(age = 20,  name = 'nit')
```

```
nit
20
```

In [256…

```
def person(name, age, age2):
    print(name)
    print(age)
    print(age2)

person(age = 20,  name = 'nit', age2 = 1)
```

```
nit
20
1
```

Default Argument

In [258…

```
def person(name): #In this code we expected to print 2 but we got bydefault
    print(name)
    print(age)

person('nit')
```

```
nit
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[258], line 5
      2     print(name)
      3     print(age)
----> 5 person('nit')

Cell In[258], line 3, in person(name)
      1 def person(name): #In this code we expected to print 2 but we got bydefau
lt
      2     print(name)
----> 3     print(age)

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

In [260…

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

person('nit')
```

```
nit
18
```

In [262…

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

person('nit',24)  #in hear bydefault override the existing default value
```

```
nit
24
```

Variable Argument

In [264…

```
def sum(a, b):
    c = a+b
    print(c)
```

```
sum(5,6)
```

```
11
```

In [266…

```
def sum(a, b):
    c = a+b
    print(c)

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

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[266], line 5
      2     c = a+b
      3     print(c)
----> 5 sum(5,6,7,8)

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

In [268…

```
def sum(a, *b):
    c = a+b
    print(c)

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

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[268], line 5
      2     c = a+b
      3     print(c)
----> 5 sum(5,6,7,8)

Cell In[268], line 2, in sum(a, *b)
      1 def sum(a, *b):
----> 2     c = a+b
      3     print(c)

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

In [270…

```
def sum(a, *b):
    #c = a+b
    print(type(a))
    print(type(b))

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

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

In [272…

```
def sum(a, *b):
    #c = a+b
    print(a)
    print(b)

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

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

In [274…

```
def sum(a, *b):
    c = a
    for i in b:
        c = c + i
    print(c)

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

```
26
```

In [276…

```
def sum(a, *b):

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

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

```
26
```

In [278…

```
def sum(a, *b):

    c = 0

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

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

```
21
```

In [280…

```
def sum(a, *b):

    c = 0
    for i in b:
        c = c + i
    print(c)

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

```
26
```

Keyworded Variable Length Arguments

In [282…

```
def person():
    person('ALEX', 36, 'JOHN', 987767)
```

In [284…

```
def person(name,*data):
    print(name)
    print(data)

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

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

In [286…

```
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[286], 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 [288…

```
def person(name, **data):
    print(name)
    print(data)

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

```
mark
{'age': 36, 'home_place': 'southcity', 'mob': 987767}
```

In [290…

```
def person(name,**data):
    print('name')
    print(data)

person('mark', age = 36, home_place ='southcity', mob =987767, edu='phd', actor
```

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

In [292…

```
def person(name, **data):
    print(name)

    for i, j in data.items():
        print(i, j)

person('john', age = 36, home_place ='southcity', mob =987767, place = 'USA')
```

```
john
age 36
home_place southcity
mob 987767
place USA
```

Global variable s Local Variable

In [294…

```
a = 10
print(a)
```

```
10
```

In [296…

```
a = 10  #-- globla variable

def something():
    b = 15 #local variable
    print('in function',b)
    print('out function',a)
```

In [298…

```
a = 10

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

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

```
out function 10
```

In [300…

```
a = 10

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

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

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

In [302…

```
a = 10

def something():
    a = 15

print('in function',a)

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

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

In [304…

```
a = 10

def something():
    a = 15  #hear a is local variable
    print(a)

print(a)
```

```
10
```

In [306…

```
a = 10

def something():
    a = 15
    print('in function',a)  # local variable

print('out function',a) #gloabl variable

# In this code we ddint call the function
```

```
out function 10
```

In [308…

```
a = 10

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

something()

print('out function',a) #gloabl variable
```

```
# 1st preference is always local variable
```

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

In [310…

```
a = 10

def something():
    #if we remove this variable then can befault it consider as global variable
    print('in function',a)

something()
print('out function',a)
# if we dont assign any variabel inside the functin bydefault both considerd as
```

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

In [312…

```
a = 10

def something():
    b= 25
    #if we remove this variable then can befault it consider as global variable
    print('in function',b)

something()
print('out function',a)
# if we dont assign any variabel inside the functin bydefault both considerd as
```

```
in function 25
out function 10
```

In [314…

```
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 [316…

```
a = 20

def something():
    global a
    a = 15      # we refered local to global
    print('in function',a)

    a = 9
print('out function',a)
```

```
out function 20
```

In [318…

```
import keyword
keyword.kwlist
```

Out[318…

```
['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 [320…

```python
a = 10
print(id(a))

def something():
    a = 9
    x = globals()['a'] #gloabls can give you all the gloabls

    print(id(x))
    print('in function',a)


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

```
140716705331928
140716705331928
in function 9
out function 10
```

In [322…

```python
a = 10
print(id(a))

def something():
    a = 9
    x = globals() # if i dont mention a then it will creat new memory

    print(id(x))
    print('in function',a)

    globals()['a'] = 15


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

```
140716705331928
2799765024384
in function 9
out function 15
```

Global Variable

In [324…

```
x = 10  # 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
```

```
15
```

Globals

In [326…

```
x = 10  # Global variable

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

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

```
15
```

Pass List to Function

In [328…

```
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, 20]
even, odd = count(lst)

print(even)
print(odd)
```

```
7
6
```

In [330…

```
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]
even,odd = count(lst)

print("Even Number: {} and odd Number : {}".format(even,odd))
#format is function belongs to string& bydefault you need to pass 2 parameter
```

```
Even Number: 5 and odd Number : 6
```

In [ ]:

```
Fibonacci Sequence
```

def fib(n): print(0) print(1) fib(0)

In [332…

```
def fib(n):
    print(0)
    print(1)
    print(1)
    print(2)
    print(3)
    print(5)

fib(0)
```

```
0
1
1
2
3
5
```

In [334…

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

    print(a)
    print(b)

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

        print(c)

fib(5)
```

```
0
1
1
2
3
```

In [336…

```
def fib(n):
    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(2)
```

```
0
1
```

Factorial Of a Number in Python

In [338…

```
def fact(n):
    f = 1
    for i in range(1, n+1):
        f = f*i

    return f

x = 5
result = fact(x)
print(result)
```

```
120
```

Recurrsion Function

In [340…

```
def wish():
    print('hello')

wish()
```

```
hello
```

In [342…

```
def wish():
    print('hello')
wish()

wish()
```

```
hello
hello
```

In [344…

```
import sys
sys.getrecursionlimit()
```

Out[344…

```
3000
```

In [346…

```
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
hello 126
hello 127
hello 128
hello 129
hello 130
hello 131
hello 132
hello 133
hello 134
hello 135
hello 136
hello 137
hello 138
hello 139
hello 140
hello 141
hello 142
hello 143
hello 144
hello 145
hello 146
hello 147
hello 148
hello 149
hello 150
hello 151
hello 152
hello 153
hello 154
hello 155
hello 156
hello 157
hello 158
hello 159
hello 160
hello 161
hello 162
hello 163
hello 164
hello 165
hello 166
hello 167
hello 168
hello 169
hello 170
hello 171
hello 172
hello 173
hello 174
hello 175
hello 176
hello 177
hello 178
hello 179
hello 180
hello 181
hello 182
hello 183
hello 184
hello 185
hello 186
hello 187
hello 188
hello 189
hello 190
hello 191
hello 192
hello 193
hello 194
hello 195
hello 196
hello 197
hello 198
hello 199
hello 200
hello 201
hello 202
hello 203
hello 204
hello 205
hello 206
hello 207
hello 208
hello 209
hello 210
hello 211
hello 212
hello 213
hello 214
hello 215
hello 216
hello 217
hello 218
hello 219
hello 220
hello 221
hello 222
hello 223
hello 224
hello 225
hello 226
hello 227
hello 228
hello 229
hello 230
hello 231
hello 232
hello 233
hello 234
hello 235
hello 236
hello 237
hello 238
hello 239
hello 240
hello 241
hello 242
hello 243
hello 244
hello 245
hello 246
hello 247
hello 248
hello 249
hello 250
hello 251
hello 252
hello 253
hello 254
hello 255
hello 256
hello 257
hello 258
hello 259
hello 260
hello 261
hello 262
hello 263
hello 264
hello 265
hello 266
hello 267
hello 268
hello 269
hello 270
hello 271
hello 272
hello 273
hello 274
hello 275
hello 276
hello 277
hello 278
hello 279
hello 280
hello 281
hello 282
hello 283
hello 284
hello 285
hello 286
hello 287
hello 288
hello 289
hello 290
hello 291
hello 292
hello 293
hello 294
hello 295
hello 296
hello 297
hello 298
hello 299
hello 300
hello 301
hello 302
hello 303
hello 304
hello 305
hello 306
hello 307
hello 308
hello 309
hello 310
hello 311
hello 312
hello 313
hello 314
hello 315
hello 316
hello 317
hello 318
hello 319
hello 320
hello 321
hello 322
hello 323
hello 324
hello 325
hello 326
hello 327
hello 328
hello 329
hello 330
hello 331
hello 332
hello 333
hello 334
hello 335
hello 336
hello 337
hello 338
hello 339
hello 340
hello 341
hello 342
hello 343
hello 344
hello 345
hello 346
hello 347
hello 348
hello 349
hello 350
hello 351
hello 352
hello 353
hello 354
hello 355
hello 356
hello 357
hello 358
hello 359
hello 360
hello 361
hello 362
hello 363
hello 364
hello 365
hello 366
hello 367
hello 368
hello 369
hello 370
hello 371
hello 372
hello 373
hello 374
hello 375
hello 376
hello 377
hello 378
hello 379
hello 380
hello 381
hello 382
hello 383
hello 384
hello 385
hello 386
hello 387
hello 388
hello 389
hello 390
hello 391
hello 392
hello 393
hello 394
hello 395
hello 396
hello 397
hello 398
hello 399
hello 400
hello 401
hello 402
hello 403
hello 404
hello 405
hello 406
hello 407
hello 408
hello 409
hello 410
hello 411
hello 412
hello 413
hello 414
hello 415
hello 416
hello 417
hello 418
hello 419
hello 420
hello 421
hello 422
hello 423
hello 424
hello 425
hello 426
hello 427
hello 428
hello 429
hello 430
hello 431
hello 432
hello 433
hello 434
hello 435
hello 436
hello 437
hello 438
hello 439
hello 440
hello 441
hello 442
hello 443
hello 444
hello 445
hello 446
hello 447
hello 448
hello 449
hello 450
hello 451
hello 452
hello 453
hello 454
hello 455
hello 456
hello 457
hello 458
hello 459
hello 460
hello 461
hello 462
hello 463
hello 464
hello 465
hello 466
hello 467
hello 468
hello 469
hello 470
hello 471
hello 472
hello 473
hello 474
hello 475
hello 476
hello 477
hello 478
hello 479
hello 480
hello 481
hello 482
hello 483
hello 484
hello 485
hello 486
hello 487
hello 488
hello 489
hello 490
hello 491
hello 492
hello 493
hello 494
hello 495
hello 496
hello 497
hello 498
hello 499
hello 500
hello 501
hello 502
hello 503
hello 504
hello 505
hello 506
hello 507
hello 508
hello 509
hello 510
hello 511
hello 512
hello 513
hello 514
hello 515
hello 516
hello 517
hello 518
hello 519
hello 520
hello 521
hello 522
hello 523
hello 524
hello 525
hello 526
hello 527
hello 528
hello 529
hello 530
hello 531
hello 532
hello 533
hello 534
hello 535
hello 536
hello 537
hello 538
hello 539
hello 540
hello 541
hello 542
hello 543
hello 544
hello 545
hello 546
hello 547
hello 548
hello 549
hello 550
hello 551
hello 552
hello 553
hello 554
hello 555
hello 556
hello 557
hello 558
hello 559
hello 560
hello 561
hello 562
hello 563
hello 564
hello 565
hello 566
hello 567
hello 568
hello 569
hello 570
hello 571
hello 572
hello 573
hello 574
hello 575
hello 576
hello 577
hello 578
hello 579
hello 580
hello 581
hello 582
hello 583
hello 584
hello 585
hello 586
hello 587
hello 588
hello 589
hello 590
hello 591
hello 592
hello 593
hello 594
hello 595
hello 596
hello 597
hello 598
hello 599
hello 600
hello 601
hello 602
hello 603
hello 604
hello 605
hello 606
hello 607
hello 608
hello 609
hello 610
hello 611
hello 612
hello 613
hello 614
hello 615
hello 616
hello 617
hello 618
hello 619
hello 620
hello 621
hello 622
hello 623
hello 624
hello 625
hello 626
hello 627
hello 628
hello 629
hello 630
hello 631
hello 632
hello 633
hello 634
hello 635
hello 636
hello 637
hello 638
hello 639
hello 640
hello 641
hello 642
hello 643
hello 644
hello 645
hello 646
hello 647
hello 648
hello 649
hello 650
hello 651
hello 652
hello 653
hello 654
hello 655
hello 656
hello 657
hello 658
hello 659
hello 660
hello 661
hello 662
hello 663
hello 664
hello 665
hello 666
hello 667
hello 668
hello 669
hello 670
hello 671
hello 672
hello 673
hello 674
hello 675
hello 676
hello 677
hello 678
hello 679
hello 680
hello 681
hello 682
hello 683
hello 684
hello 685
hello 686
hello 687
hello 688
hello 689
hello 690
hello 691
hello 692
hello 693
hello 694
hello 695
hello 696
hello 697
hello 698
hello 699
hello 700
hello 701
hello 702
hello 703
hello 704
hello 705
hello 706
hello 707
hello 708
hello 709
hello 710
hello 711
hello 712
hello 713
hello 714
hello 715
hello 716
hello 717
hello 718
hello 719
hello 720
hello 721
hello 722
hello 723
hello 724
hello 725
hello 726
hello 727
hello 728
hello 729
hello 730
hello 731
hello 732
hello 733
hello 734
hello 735
hello 736
hello 737
hello 738
hello 739
hello 740
hello 741
hello 742
hello 743
hello 744
hello 745
hello 746
hello 747
hello 748
hello 749
hello 750
hello 751
hello 752
hello 753
hello 754
hello 755
hello 756
hello 757
hello 758
hello 759
hello 760
hello 761
hello 762
hello 763
hello 764
hello 765
hello 766
hello 767
hello 768
hello 769
hello 770
hello 771
hello 772
hello 773
hello 774
hello 775
hello 776
hello 777
hello 778
hello 779
hello 780
hello 781
hello 782
hello 783
hello 784
hello 785
hello 786
hello 787
hello 788
hello 789
hello 790
hello 791
hello 792
hello 793
hello 794
hello 795
hello 796
hello 797
hello 798
hello 799
hello 800
hello 801
hello 802
hello 803
hello 804
hello 805
hello 806
hello 807
hello 808
hello 809
hello 810
hello 811
hello 812
hello 813
hello 814
hello 815
hello 816
hello 817
hello 818
hello 819
hello 820
hello 821
hello 822
hello 823
hello 824
hello 825
hello 826
hello 827
hello 828
hello 829
hello 830
hello 831
hello 832
hello 833
hello 834
hello 835
hello 836
hello 837
hello 838
hello 839
hello 840
hello 841
hello 842
hello 843
hello 844
hello 845
hello 846
hello 847
hello 848
hello 849
hello 850
hello 851
hello 852
hello 853
hello 854
hello 855
hello 856
hello 857
hello 858
hello 859
hello 860
hello 861
hello 862
hello 863
hello 864
hello 865
hello 866
hello 867
hello 868
hello 869
hello 870
hello 871
hello 872
hello 873
hello 874
hello 875
hello 876
hello 877
hello 878
hello 879
hello 880
hello 881
hello 882
hello 883
hello 884
hello 885
hello 886
hello 887
hello 888
hello 889
hello 890
hello 891
hello 892
hello 893
hello 894
hello 895
hello 896
hello 897
hello 898
hello 899
hello 900
hello 901
hello 902
hello 903
hello 904
hello 905
hello 906
hello 907
hello 908
hello 909
hello 910
hello 911
hello 912
hello 913
hello 914
hello 915
hello 916
hello 917
hello 918
hello 919
hello 920
hello 921
hello 922
hello 923
hello 924
hello 925
hello 926
hello 927
hello 928
hello 929
hello 930
hello 931
hello 932
hello 933
hello 934
hello 935
hello 936
hello 937
hello 938
hello 939
hello 940
hello 941
hello 942
hello 943
hello 944
hello 945
hello 946
hello 947
hello 948
hello 949
hello 950
hello 951
hello 952
hello 953
hello 954
hello 955
hello 956
hello 957
hello 958
hello 959
hello 960
hello 961
hello 962
hello 963
hello 964
hello 965
hello 966
hello 967
hello 968
hello 969
hello 970
hello 971
hello 972
hello 973
hello 974
hello 975
hello 976
hello 977
hello 978
hello 979
hello 980
hello 981
hello 982
hello 983
hello 984
hello 985
hello 986
hello 987
hello 988
hello 989
hello 990
hello 991
hello 992
hello 993
hello 994
hello 995
hello 996
hello 997
hello 998
hello 999
hello 1000
hello 1001
hello 1002
hello 1003
hello 1004
hello 1005
hello 1006
hello 1007
hello 1008
hello 1009
hello 1010
hello 1011
hello 1012
hello 1013
hello 1014
hello 1015
hello 1016
hello 1017
hello 1018
hello 1019
hello 1020
hello 1021
hello 1022
hello 1023
hello 1024
hello 1025
hello 1026
hello 1027
hello 1028
hello 1029
hello 1030
hello 1031
hello 1032
hello 1033
hello 1034
hello 1035
hello 1036
hello 1037
hello 1038
hello 1039
hello 1040
hello 1041
hello 1042
hello 1043
hello 1044
hello 1045
hello 1046
hello 1047
hello 1048
hello 1049
hello 1050
hello 1051
hello 1052
hello 1053
hello 1054
hello 1055
hello 1056
hello 1057
hello 1058
hello 1059
hello 1060
hello 1061
hello 1062
hello 1063
hello 1064
hello 1065
hello 1066
hello 1067
hello 1068
hello 1069
hello 1070
hello 1071
hello 1072
hello 1073
hello 1074
hello 1075
hello 1076
hello 1077
hello 1078
hello 1079
hello 1080
hello 1081
hello 1082
hello 1083
hello 1084
hello 1085
hello 1086
hello 1087
hello 1088
hello 1089
hello 1090
hello 1091
hello 1092
hello 1093
hello 1094
hello 1095
hello 1096
hello 1097
hello 1098
hello 1099
hello 1100
hello 1101
hello 1102
hello 1103
hello 1104
hello 1105
hello 1106
hello 1107
hello 1108
hello 1109
hello 1110
hello 1111
hello 1112
hello 1113
hello 1114
hello 1115
hello 1116
hello 1117
hello 1118
hello 1119
hello 1120
hello 1121
hello 1122
hello 1123
hello 1124
hello 1125
hello 1126
hello 1127
hello 1128
hello 1129
hello 1130
hello 1131
hello 1132
hello 1133
hello 1134
hello 1135
hello 1136
hello 1137
hello 1138
hello 1139
hello 1140
hello 1141
hello 1142
hello 1143
hello 1144
hello 1145
hello 1146
hello 1147
hello 1148
hello 1149
hello 1150
hello 1151
hello 1152
hello 1153
hello 1154
hello 1155
hello 1156
hello 1157
hello 1158
hello 1159
hello 1160
hello 1161
hello 1162
hello 1163
hello 1164
hello 1165
hello 1166
hello 1167
hello 1168
hello 1169
hello 1170
hello 1171
hello 1172
hello 1173
hello 1174
hello 1175
hello 1176
hello 1177
hello 1178
hello 1179
hello 1180
hello 1181
hello 1182
hello 1183
hello 1184
hello 1185
hello 1186
hello 1187
hello 1188
hello 1189
hello 1190
hello 1191
hello 1192
hello 1193
hello 1194
hello 1195
hello 1196
hello 1197
hello 1198
hello 1199
hello 1200
hello 1201
hello 1202
hello 1203
hello 1204
hello 1205
hello 1206
hello 1207
hello 1208
hello 1209
hello 1210
hello 1211
hello 1212
hello 1213
hello 1214
hello 1215
hello 1216
hello 1217
hello 1218
hello 1219
hello 1220
hello 1221
hello 1222
hello 1223
hello 1224
hello 1225
hello 1226
hello 1227
hello 1228
hello 1229
hello 1230
hello 1231
hello 1232
hello 1233
hello 1234
hello 1235
hello 1236
hello 1237
hello 1238
hello 1239
hello 1240
hello 1241
hello 1242
hello 1243
hello 1244
hello 1245
hello 1246
hello 1247
hello 1248
hello 1249
hello 1250
hello 1251
hello 1252
hello 1253
hello 1254
hello 1255
hello 1256
hello 1257
hello 1258
hello 1259
hello 1260
hello 1261
hello 1262
hello 1263
hello 1264
hello 1265
hello 1266
hello 1267
hello 1268
hello 1269
hello 1270
hello 1271
hello 1272
hello 1273
hello 1274
hello 1275
hello 1276
hello 1277
hello 1278
hello 1279
hello 1280
hello 1281
hello 1282
hello 1283
hello 1284
hello 1285
hello 1286
hello 1287
hello 1288
hello 1289
hello 1290
hello 1291
hello 1292
hello 1293
hello 1294
hello 1295
hello 1296
hello 1297
hello 1298
hello 1299
hello 1300
hello 1301
hello 1302
hello 1303
hello 1304
hello 1305
hello 1306
hello 1307
hello 1308
hello 1309
hello 1310
hello 1311
hello 1312
hello 1313
hello 1314
hello 1315
hello 1316
hello 1317
hello 1318
hello 1319
hello 1320
hello 1321
hello 1322
hello 1323
hello 1324
hello 1325
hello 1326
hello 1327
hello 1328
hello 1329
hello 1330
hello 1331
hello 1332
hello 1333
hello 1334
hello 1335
hello 1336
hello 1337
hello 1338
hello 1339
hello 1340
hello 1341
hello 1342
hello 1343
hello 1344
hello 1345
hello 1346
hello 1347
hello 1348
hello 1349
hello 1350
hello 1351
hello 1352
hello 1353
hello 1354
hello 1355
hello 1356
hello 1357
hello 1358
hello 1359
hello 1360
hello 1361
hello 1362
hello 1363
hello 1364
hello 1365
hello 1366
hello 1367
hello 1368
hello 1369
hello 1370
hello 1371
hello 1372
hello 1373
hello 1374
hello 1375
hello 1376
hello 1377
hello 1378
hello 1379
hello 1380
hello 1381
hello 1382
hello 1383
hello 1384
hello 1385
hello 1386
hello 1387
hello 1388
hello 1389
hello 1390
hello 1391
hello 1392
hello 1393
hello 1394
hello 1395
hello 1396
hello 1397
hello 1398
hello 1399
hello 1400
hello 1401
hello 1402
hello 1403
hello 1404
hello 1405
hello 1406
hello 1407
hello 1408
hello 1409
hello 1410
hello 1411
hello 1412
hello 1413
hello 1414
hello 1415
hello 1416
hello 1417
hello 1418
hello 1419
hello 1420
hello 1421
hello 1422
hello 1423
hello 1424
hello 1425
hello 1426
hello 1427
hello 1428
hello 1429
hello 1430
hello 1431
hello 1432
hello 1433
hello 1434
hello 1435
hello 1436
hello 1437
hello 1438
hello 1439
hello 1440
hello 1441
hello 1442
hello 1443
hello 1444
hello 1445
hello 1446
hello 1447
hello 1448
hello 1449
hello 1450
hello 1451
hello 1452
hello 1453
hello 1454
hello 1455
hello 1456
hello 1457
hello 1458
hello 1459
hello 1460
hello 1461
hello 1462
hello 1463
hello 1464
hello 1465
hello 1466
hello 1467
hello 1468
hello 1469
hello 1470
hello 1471
hello 1472
hello 1473
hello 1474
hello 1475
hello 1476
hello 1477
hello 1478
hello 1479
hello 1480
hello 1481
hello 1482
hello 1483
hello 1484
hello 1485
hello 1486
hello 1487
hello 1488
hello 1489
hello 1490
hello 1491
hello 1492
hello 1493
hello 1494
hello 1495
hello 1496
hello 1497
hello 1498
hello 1499
hello 1500
hello 1501
hello 1502
hello 1503
hello 1504
hello 1505
hello 1506
hello 1507
hello 1508
hello 1509
hello 1510
hello 1511
hello 1512
hello 1513
hello 1514
hello 1515
hello 1516
hello 1517
hello 1518
hello 1519
hello 1520
hello 1521
hello 1522
hello 1523
hello 1524
hello 1525
hello 1526
hello 1527
hello 1528
hello 1529
hello 1530
hello 1531
hello 1532
hello 1533
hello 1534
hello 1535
hello 1536
hello 1537
hello 1538
hello 1539
hello 1540
hello 1541
hello 1542
hello 1543
hello 1544
hello 1545
hello 1546
hello 1547
hello 1548
hello 1549
hello 1550
hello 1551
hello 1552
hello 1553
hello 1554
hello 1555
hello 1556
hello 1557
hello 1558
hello 1559
hello 1560
hello 1561
hello 1562
hello 1563
hello 1564
hello 1565
hello 1566
hello 1567
hello 1568
hello 1569
hello 1570
hello 1571
hello 1572
hello 1573
hello 1574
hello 1575
hello 1576
hello 1577
hello 1578
hello 1579
hello 1580
hello 1581
hello 1582
hello 1583
hello 1584
hello 1585
hello 1586
hello 1587
hello 1588
hello 1589
hello 1590
hello 1591
hello 1592
hello 1593
hello 1594
hello 1595
hello 1596
hello 1597
hello 1598
hello 1599
hello 1600
hello 1601
hello 1602
hello 1603
hello 1604
hello 1605
hello 1606
hello 1607
hello 1608
hello 1609
hello 1610
hello 1611
hello 1612
hello 1613
hello 1614
hello 1615
hello 1616
hello 1617
hello 1618
hello 1619
hello 1620
hello 1621
hello 1622
hello 1623
hello 1624
hello 1625
hello 1626
hello 1627
hello 1628
hello 1629
hello 1630
hello 1631
hello 1632
hello 1633
hello 1634
hello 1635
hello 1636
hello 1637
hello 1638
hello 1639
hello 1640
hello 1641
hello 1642
hello 1643
hello 1644
hello 1645
hello 1646
hello 1647
hello 1648
hello 1649
hello 1650
hello 1651
hello 1652
hello 1653
hello 1654
hello 1655
hello 1656
hello 1657
hello 1658
hello 1659
hello 1660
hello 1661
hello 1662
hello 1663
hello 1664
hello 1665
hello 1666
hello 1667
hello 1668
hello 1669
hello 1670
hello 1671
hello 1672
hello 1673
hello 1674
hello 1675
hello 1676
hello 1677
hello 1678
hello 1679
hello 1680
hello 1681
hello 1682
hello 1683
hello 1684
hello 1685
hello 1686
hello 1687
hello 1688
hello 1689
hello 1690
hello 1691
hello 1692
hello 1693
hello 1694
hello 1695
hello 1696
hello 1697
hello 1698
hello 1699
hello 1700
hello 1701
hello 1702
hello 1703
hello 1704
hello 1705
hello 1706
hello 1707
hello 1708
hello 1709
hello 1710
hello 1711
hello 1712
hello 1713
hello 1714
hello 1715
hello 1716
hello 1717
hello 1718
hello 1719
hello 1720
hello 1721
hello 1722
hello 1723
hello 1724
hello 1725
hello 1726
hello 1727
hello 1728
hello 1729
hello 1730
hello 1731
hello 1732
hello 1733
hello 1734
hello 1735
hello 1736
hello 1737
hello 1738
hello 1739
hello 1740
hello 1741
hello 1742
hello 1743
hello 1744
hello 1745
hello 1746
hello 1747
hello 1748
hello 1749
hello 1750
hello 1751
hello 1752
hello 1753
hello 1754
hello 1755
hello 1756
hello 1757
hello 1758
hello 1759
hello 1760
hello 1761
hello 1762
hello 1763
hello 1764
hello 1765
hello 1766
hello 1767
hello 1768
hello 1769
hello 1770
hello 1771
hello 1772
hello 1773
hello 1774
hello 1775
hello 1776
hello 1777
hello 1778
hello 1779
hello 1780
hello 1781
hello 1782
hello 1783
hello 1784
hello 1785
hello 1786
hello 1787
hello 1788
hello 1789
hello 1790
hello 1791
hello 1792
hello 1793
hello 1794
hello 1795
hello 1796
hello 1797
hello 1798
hello 1799
hello 1800
hello 1801
hello 1802
hello 1803
hello 1804
hello 1805
hello 1806
hello 1807
hello 1808
hello 1809
hello 1810
hello 1811
hello 1812
hello 1813
hello 1814
hello 1815
hello 1816
hello 1817
hello 1818
hello 1819
hello 1820
hello 1821
hello 1822
hello 1823
hello 1824
hello 1825
hello 1826
hello 1827
hello 1828
hello 1829
hello 1830
hello 1831
hello 1832
hello 1833
hello 1834
hello 1835
hello 1836
hello 1837
hello 1838
hello 1839
hello 1840
hello 1841
hello 1842
hello 1843
hello 1844
hello 1845
hello 1846
hello 1847
hello 1848
hello 1849
hello 1850
hello 1851
hello 1852
hello 1853
hello 1854
hello 1855
hello 1856
hello 1857
hello 1858
hello 1859
hello 1860
hello 1861
hello 1862
hello 1863
hello 1864
hello 1865
hello 1866
hello 1867
hello 1868
hello 1869
hello 1870
hello 1871
hello 1872
hello 1873
hello 1874
hello 1875
hello 1876
hello 1877
hello 1878
hello 1879
hello 1880
hello 1881
hello 1882
hello 1883
hello 1884
hello 1885
hello 1886
hello 1887
hello 1888
hello 1889
hello 1890
hello 1891
hello 1892
hello 1893
hello 1894
hello 1895
hello 1896
hello 1897
hello 1898
hello 1899
hello 1900
hello 1901
hello 1902
hello 1903
hello 1904
hello 1905
hello 1906
hello 1907
hello 1908
hello 1909
hello 1910
hello 1911
hello 1912
hello 1913
hello 1914
hello 1915
hello 1916
hello 1917
hello 1918
hello 1919
hello 1920
hello 1921
hello 1922
hello 1923
hello 1924
hello 1925
hello 1926
hello 1927
hello 1928
hello 1929
hello 1930
hello 1931
hello 1932
hello 1933
hello 1934
hello 1935
hello 1936
hello 1937
hello 1938
hello 1939
hello 1940
hello 1941
hello 1942
hello 1943
hello 1944
hello 1945
hello 1946
hello 1947
hello 1948
hello 1949
hello 1950
hello 1951
hello 1952
hello 1953
hello 1954
hello 1955
hello 1956
hello 1957
hello 1958
hello 1959
hello 1960
hello 1961
hello 1962
hello 1963
hello 1964
hello 1965
hello 1966
hello 1967
hello 1968
hello 1969
hello 1970
hello 1971
hello 1972
hello 1973
hello 1974
hello 1975
hello 1976
hello 1977
hello 1978
hello 1979
hello 1980
hello 1981
hello 1982
hello 1983
hello 1984
hello 1985
hello 1986
hello 1987
hello 1988
hello 1989
hello 1990
hello 1991
hello 1992
hello 1993
hello 1994
hello 1995
hello 1996
hello 1997
hello 1998
hello 1999
hello 2000
hello 2001
hello 2002
hello 2003
hello 2004
hello 2005
hello 2006
hello 2007
hello 2008
hello 2009
hello 2010
hello 2011
hello 2012
hello 2013
hello 2014
hello 2015
hello 2016
hello 2017
hello 2018
hello 2019
hello 2020
hello 2021
hello 2022
hello 2023
hello 2024
hello 2025
hello 2026
hello 2027
hello 2028
hello 2029
hello 2030
hello 2031
hello 2032
hello 2033
hello 2034
hello 2035
hello 2036
hello 2037
hello 2038
hello 2039
hello 2040
hello 2041
hello 2042
hello 2043
hello 2044
hello 2045
hello 2046
hello 2047
hello 2048
hello 2049
hello 2050
hello 2051
hello 2052
hello 2053
hello 2054
hello 2055
hello 2056
hello 2057
hello 2058
hello 2059
hello 2060
hello 2061
hello 2062
hello 2063
hello 2064
hello 2065
hello 2066
hello 2067
hello 2068
hello 2069
hello 2070
hello 2071
hello 2072
hello 2073
hello 2074
hello 2075
hello 2076
hello 2077
hello 2078
hello 2079
hello 2080
hello 2081
hello 2082
hello 2083
hello 2084
hello 2085
hello 2086
hello 2087
hello 2088
hello 2089
hello 2090
hello 2091
hello 2092
hello 2093
hello 2094
hello 2095
hello 2096
hello 2097
hello 2098
hello 2099
hello 2100
hello 2101
hello 2102
hello 2103
hello 2104
hello 2105
hello 2106
hello 2107
hello 2108
hello 2109
hello 2110
hello 2111
hello 2112
hello 2113
hello 2114
hello 2115
hello 2116
hello 2117
hello 2118
hello 2119
hello 2120
hello 2121
hello 2122
hello 2123
hello 2124
hello 2125
hello 2126
hello 2127
hello 2128
hello 2129
hello 2130
hello 2131
hello 2132
hello 2133
hello 2134
hello 2135
hello 2136
hello 2137
hello 2138
hello 2139
hello 2140
hello 2141
hello 2142
hello 2143
hello 2144
hello 2145
hello 2146
hello 2147
hello 2148
hello 2149
hello 2150
hello 2151
hello 2152
hello 2153
hello 2154
hello 2155
hello 2156
hello 2157
hello 2158
hello 2159
hello 2160
hello 2161
hello 2162
hello 2163
hello 2164
hello 2165
hello 2166
hello 2167
hello 2168
hello 2169
hello 2170
hello 2171
hello 2172
hello 2173
hello 2174
hello 2175
hello 2176
hello 2177
hello 2178
hello 2179
hello 2180
hello 2181
hello 2182
hello 2183
hello 2184
hello 2185
hello 2186
hello 2187
hello 2188
hello 2189
hello 2190
hello 2191
hello 2192
hello 2193
hello 2194
hello 2195
hello 2196
hello 2197
hello 2198
hello 2199
hello 2200
hello 2201
hello 2202
hello 2203
hello 2204
hello 2205
hello 2206
hello 2207
hello 2208
hello 2209
hello 2210
hello 2211
hello 2212
hello 2213
hello 2214
hello 2215
hello 2216
hello 2217
hello 2218
hello 2219
hello 2220
hello 2221
hello 2222
hello 2223
hello 2224
hello 2225
hello 2226
hello 2227
hello 2228
hello 2229
hello 2230
hello 2231
hello 2232
hello 2233
hello 2234
hello 2235
hello 2236
hello 2237
hello 2238
hello 2239
hello 2240
hello 2241
hello 2242
hello 2243
hello 2244
hello 2245
hello 2246
hello 2247
hello 2248
hello 2249
hello 2250
hello 2251
hello 2252
hello 2253
hello 2254
hello 2255
hello 2256
hello 2257
hello 2258
hello 2259
hello 2260
hello 2261
hello 2262
hello 2263
hello 2264
hello 2265
hello 2266
hello 2267
hello 2268
hello 2269
hello 2270
hello 2271
hello 2272
hello 2273
hello 2274
hello 2275
hello 2276
hello 2277
hello 2278
hello 2279
hello 2280
hello 2281
hello 2282
hello 2283
hello 2284
hello 2285
hello 2286
hello 2287
hello 2288
hello 2289
hello 2290
hello 2291
hello 2292
hello 2293
hello 2294
hello 2295
hello 2296
hello 2297
hello 2298
hello 2299
hello 2300
hello 2301
hello 2302
hello 2303
hello 2304
hello 2305
hello 2306
hello 2307
hello 2308
hello 2309
hello 2310
hello 2311
hello 2312
hello 2313
hello 2314
hello 2315
hello 2316
hello 2317
hello 2318
hello 2319
hello 2320
hello 2321
hello 2322
hello 2323
hello 2324
hello 2325
hello 2326
hello 2327
hello 2328
hello 2329
hello 2330
hello 2331
hello 2332
hello 2333
hello 2334
hello 2335
hello 2336
hello 2337
hello 2338
hello 2339
hello 2340
hello 2341
hello 2342
hello 2343
hello 2344
hello 2345
hello 2346
hello 2347
hello 2348
hello 2349
hello 2350
hello 2351
hello 2352
hello 2353
hello 2354
hello 2355
hello 2356
hello 2357
hello 2358
hello 2359
hello 2360
hello 2361
hello 2362
hello 2363
hello 2364
hello 2365
hello 2366
hello 2367
hello 2368
hello 2369
hello 2370
hello 2371
hello 2372
hello 2373
hello 2374
hello 2375
hello 2376
hello 2377
hello 2378
hello 2379
hello 2380
hello 2381
hello 2382
hello 2383
hello 2384
hello 2385
hello 2386
hello 2387
hello 2388
hello 2389
hello 2390
hello 2391
hello 2392
hello 2393
hello 2394
hello 2395
hello 2396
hello 2397
hello 2398
hello 2399
hello 2400
hello 2401
hello 2402
hello 2403
hello 2404
hello 2405
hello 2406
hello 2407
hello 2408
hello 2409
hello 2410
hello 2411
hello 2412
hello 2413
hello 2414
hello 2415
hello 2416
hello 2417
hello 2418
hello 2419
hello 2420
hello 2421
hello 2422
hello 2423
hello 2424
hello 2425
hello 2426
hello 2427
hello 2428
hello 2429
hello 2430
hello 2431
hello 2432
hello 2433
hello 2434
hello 2435
hello 2436
hello 2437
hello 2438
hello 2439
hello 2440
hello 2441
hello 2442
hello 2443
hello 2444
hello 2445
hello 2446
hello 2447
hello 2448
hello 2449
hello 2450
hello 2451
hello 2452
hello 2453
hello 2454
hello 2455
hello 2456
hello 2457
hello 2458
hello 2459
hello 2460
hello 2461
hello 2462
hello 2463
hello 2464
hello 2465
hello 2466
hello 2467
hello 2468
hello 2469
hello 2470
hello 2471
hello 2472
hello 2473
hello 2474
hello 2475
hello 2476
hello 2477
hello 2478
hello 2479
hello 2480
hello 2481
hello 2482
hello 2483
hello 2484
hello 2485
hello 2486
hello 2487
hello 2488
hello 2489
hello 2490
hello 2491
hello 2492
hello 2493
hello 2494
hello 2495
hello 2496
hello 2497
hello 2498
hello 2499
hello 2500
hello 2501
hello 2502
hello 2503
hello 2504
hello 2505
hello 2506
hello 2507
hello 2508
hello 2509
hello 2510
hello 2511
hello 2512
hello 2513
hello 2514
hello 2515
hello 2516
hello 2517
hello 2518
hello 2519
hello 2520
hello 2521
hello 2522
hello 2523
hello 2524
hello 2525
hello 2526
hello 2527
hello 2528
hello 2529
hello 2530
hello 2531
hello 2532
hello 2533
hello 2534
hello 2535
hello 2536
hello 2537
hello 2538
hello 2539
hello 2540
hello 2541
hello 2542
hello 2543
hello 2544
hello 2545
hello 2546
hello 2547
hello 2548
hello 2549
hello 2550
hello 2551
hello 2552
hello 2553
hello 2554
hello 2555
hello 2556
hello 2557
hello 2558
hello 2559
hello 2560
hello 2561
hello 2562
hello 2563
hello 2564
hello 2565
hello 2566
hello 2567
hello 2568
hello 2569
hello 2570
hello 2571
hello 2572
hello 2573
hello 2574
hello 2575
hello 2576
hello 2577
hello 2578
hello 2579
hello 2580
hello 2581
hello 2582
hello 2583
hello 2584
hello 2585
hello 2586
hello 2587
hello 2588
hello 2589
hello 2590
hello 2591
hello 2592
hello 2593
hello 2594
hello 2595
hello 2596
hello 2597
hello 2598
hello 2599
hello 2600
hello 2601
hello 2602
hello 2603
hello 2604
hello 2605
hello 2606
hello 2607
hello 2608
hello 2609
hello 2610
hello 2611
hello 2612
hello 2613
hello 2614
hello 2615
hello 2616
hello 2617
hello 2618
hello 2619
hello 2620
hello 2621
hello 2622
hello 2623
hello 2624
hello 2625
hello 2626
hello 2627
hello 2628
hello 2629
hello 2630
hello 2631
hello 2632
hello 2633
hello 2634
hello 2635
hello 2636
hello 2637
hello 2638
hello 2639
hello 2640
hello 2641
hello 2642
hello 2643
hello 2644
hello 2645
hello 2646
hello 2647
hello 2648
hello 2649
hello 2650
hello 2651
hello 2652
hello 2653
hello 2654
hello 2655
hello 2656
hello 2657
hello 2658
hello 2659
hello 2660
hello 2661
hello 2662
hello 2663
hello 2664
hello 2665
hello 2666
hello 2667
hello 2668
hello 2669
hello 2670
hello 2671
hello 2672
hello 2673
hello 2674
hello 2675
hello 2676
hello 2677
hello 2678
hello 2679
hello 2680
hello 2681
hello 2682
hello 2683
hello 2684
hello 2685
hello 2686
hello 2687
hello 2688
hello 2689
hello 2690
hello 2691
hello 2692
hello 2693
hello 2694
hello 2695
hello 2696
hello 2697
hello 2698
hello 2699
hello 2700
hello 2701
hello 2702
hello 2703
hello 2704
hello 2705
hello 2706
hello 2707
hello 2708
hello 2709
hello 2710
hello 2711
hello 2712
hello 2713
hello 2714
hello 2715
hello 2716
hello 2717
hello 2718
hello 2719
hello 2720
hello 2721
hello 2722
hello 2723
hello 2724
hello 2725
hello 2726
hello 2727
hello 2728
hello 2729
hello 2730
hello 2731
hello 2732
hello 2733
hello 2734
hello 2735
hello 2736
hello 2737
hello 2738
hello 2739
hello 2740
hello 2741
hello 2742
hello 2743
hello 2744
hello 2745
hello 2746
hello 2747
hello 2748
hello 2749
hello 2750
hello 2751
hello 2752
hello 2753
hello 2754
hello 2755
hello 2756
hello 2757
hello 2758
hello 2759
hello 2760
hello 2761
hello 2762
hello 2763
hello 2764
hello 2765
hello 2766
hello 2767
hello 2768
hello 2769
hello 2770
hello 2771
hello 2772
hello 2773
hello 2774
hello 2775
hello 2776
hello 2777
hello 2778
hello 2779
hello 2780
hello 2781
hello 2782
hello 2783
hello 2784
hello 2785
hello 2786
hello 2787
hello 2788
hello 2789
hello 2790
hello 2791
hello 2792
hello 2793
hello 2794
hello 2795
hello 2796
hello 2797
hello 2798
hello 2799
hello 2800
hello 2801
hello 2802
hello 2803
hello 2804
hello 2805
hello 2806
hello 2807
hello 2808
hello 2809
hello 2810
hello 2811
hello 2812
hello 2813
hello 2814
hello 2815
hello 2816
hello 2817
hello 2818
hello 2819
hello 2820
hello 2821
hello 2822
hello 2823
hello 2824
hello 2825
hello 2826
hello 2827
hello 2828
hello 2829
hello 2830
hello 2831
hello 2832
hello 2833
hello 2834
hello 2835
hello 2836
hello 2837
hello 2838
hello 2839
hello 2840
hello 2841
hello 2842
hello 2843
hello 2844
hello 2845
hello 2846
hello 2847
hello 2848
hello 2849
hello 2850
hello 2851
hello 2852
hello 2853
hello 2854
hello 2855
hello 2856
hello 2857
hello 2858
hello 2859
hello 2860
hello 2861
hello 2862
hello 2863
hello 2864
hello 2865
hello 2866
hello 2867
hello 2868
hello 2869
hello 2870
hello 2871
hello 2872
hello 2873
hello 2874
hello 2875
hello 2876
hello 2877
hello 2878
hello 2879
hello 2880
hello 2881
hello 2882
hello 2883
hello 2884
hello 2885
hello 2886
hello 2887
hello 2888
hello 2889
hello 2890
hello 2891
hello 2892
hello 2893
hello 2894
hello 2895
hello 2896
hello 2897
hello 2898
hello 2899
hello 2900
hello 2901
hello 2902
hello 2903
hello 2904
hello 2905
hello 2906
hello 2907
hello 2908
hello 2909
hello 2910
hello 2911
hello 2912
hello 2913
hello 2914
hello 2915
hello 2916
hello 2917
hello 2918
hello 2919
hello 2920
hello 2921
hello 2922
hello 2923
hello 2924
hello 2925
hello 2926
hello 2927
hello 2928
hello 2929
hello 2930
hello 2931
hello 2932
hello 2933
hello 2934
hello 2935
hello 2936
hello 2937
hello 2938
hello 2939
hello 2940
hello 2941
hello 2942
hello 2943
hello 2944
hello 2945
hello 2946
hello 2947
hello 2948
hello 2949
hello 2950
hello 2951
hello 2952
hello 2953
hello 2954
hello 2955
hello 2956
hello 2957
hello 2958
hello 2959
hello 2960
hello 2961
hello 2962
hello 2963
hello 2964
hello 2965
hello 2966
hello 2967
hello 2968
hello 2969
hello 2970
hello 2971
hello 2972
hello 2973
hello 2974
hello 2975
```

```
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
Cell In[346], line 8
      6     print('hello', i)
      7     wish()
----> 8 wish()

Cell In[346], line 7, in wish()
      5 i += 1
      6 print('hello', i)
----> 7 wish()

Cell In[346], line 7, in wish()
      5 i += 1
      6 print('hello', i)
----> 7 wish()

    [... skipping similar frames: wish at line 7 (2972 times)]

Cell In[346], line 7, in wish()
      5 i += 1
      6 print('hello', i)
----> 7 wish()

Cell In[346], line 6, in wish()
      4 global  i
      5 i += 1
----> 6 print('hello', i)
      7 wish()

File ~\anaconda3\Lib\site-packages\ipykernel\iostream.py:649, in OutStream.write
(self, string)
    646     msg = "I/O operation on closed file"
    647     raise ValueError(msg)
--> 649 is_child = not self._is_master_process()
    650 # only touch the buffer in the IO thread to avoid races
    651 with self._buffer_lock:

RecursionError: maximum recursion depth exceeded
```

In [350…

```python
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[350], line 12
     10     print('hello', i)
     11     wish()
---> 12 wish()

Cell In[350], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

Cell In[350], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

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

Cell In[350], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

Cell In[350], line 10, in wish()
      8 global  i
      9 i += 1
---> 10 print('hello', i)
     11 wish()

File ~\anaconda3\Lib\site-packages\ipykernel\iostream.py:649, in OutStream.write
(self, string)
    646     msg = "I/O operation on closed file"
    647     raise ValueError(msg)
--> 649 is_child = not self._is_master_process()
    650 # only touch the buffer in the IO thread to avoid races
    651 with self._buffer_lock:

RecursionError: maximum recursion depth exceeded
```

In [352…

```python
import sys
sys.getrecursionlimit()
```

Out[352…

```
150
```

In [354…

```python
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
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

```
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
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[354], line 4
      2     print('hello')
      3     wish()
----> 4 wish()

Cell In[354], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

Cell In[354], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

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

Cell In[354], line 3, in wish()
      1 def wish():
      2     print('hello')
----> 3     wish()

Cell In[354], line 2, in wish()
      1 def wish():
----> 2     print('hello')
      3     wish()

File ~\anaconda3\Lib\site-packages\ipykernel\iostream.py:649, in OutStream.write
(self, string)
    646     msg = "I/O operation on closed file"
    647     raise ValueError(msg)
--> 649 is_child = not self._is_master_process()
    650 # only touch the buffer in the IO thread to avoid races
    651 with self._buffer_lock:

RecursionError: maximum recursion depth exceeded
```

In [356…

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

```
150
```

In [358…

```
sys.setrecursionlimit(1000)
```

In [360…

```
print(sys.getrecursionlimit())
```

```
1000
```

In [362…

```
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[362], line 12
     10     print('hello', i)
     11     wish()
---> 12 wish()

Cell In[362], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

Cell In[362], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

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

Cell In[362], line 11, in wish()
      9 i += 1
     10 print('hello', i)
---> 11 wish()

Cell In[362], line 10, in wish()
      8 global  i
      9 i += 1
---> 10 print('hello', i)
     11 wish()

File ~\anaconda3\Lib\site-packages\ipykernel\iostream.py:649, in OutStream.write
(self, string)
    646     msg = "I/O operation on closed file"
    647     raise ValueError(msg)
--> 649 is_child = not self._is_master_process()
    650 # only touch the buffer in the IO thread to avoid races
    651 with self._buffer_lock:

RecursionError: maximum recursion depth exceeded
```

Factorial Using Recurssion

In [364…

```
def fact(n):
    if n==0:
        return 1
    return n * fact(n-1)

result = fact(4)

result
```

Out[364…

```
24
```

Anonymous Function /Lambda

In [366…

```
def square(a):
    return a * a

result = square(5)
print(result)
```

```
25
```

In [368…

```
f = lambda a : a * a
result = f(5)
result
```

Out[368…

```
25
```

In [370…

```
f = lambda a, b : a + b
result = f(1,4)
print(result)
```

```
5
```

In [372…

```
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
```

filter() map() reduce()

In [376…

```
#lets take one list & i want to find the list of even numbers

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

evens = list(filter(is_even, nums)) #is_even is not an inbuild function
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[376], line 5
      1 #lets take one list & i want to find the list of even numbers
      3 nums = [3,2,6,8,4,6,2,9]
----> 5 evens = list(filter(is_even, nums))

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

In [378…

```
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 [380…

```
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 [382…

```
# lets write above function using help of lambda & lambda helps to reduce the li
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 [384…

```
nums = [3,2,6,8,4,6,2,9]
odd = list(filter(lambda n : n%2 !=0, nums))
print(odd)
```

```
[3, 9]
```

In [386…

```
# lets write above function using help of lambda & lambda helps to reduce the li
nums = [3,2,6,8,4,6,2,9]

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, 4, 6, 2]
[3, 9]
```

In [388…

```
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(double)
```

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

In [390…

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

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

print(double)
```

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

In [392…

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

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

print(double)
print(double_)
```

```
[4, 12, 16, 8, 12, 4]
[4, 8, 10, 6, 8, 4]
```

In [394…

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

evens = list(filter(is_even, nums))

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

print(double)
print(double_)
```

```
[4, 12, 16, 8, 12, 4]
[0, 4, 6, 2, 4, 0]
```

In [396…

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

```
[4, 12, 16, 8, 12, 4]
[0, 4, 6, 2, 4, 0]
[4, 8, 10, 6, 8, 4]
```

In [398…

```
from functools import reduce

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

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

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

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

```
40
```

In [400…

```
from functools import reduce

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

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)
```

```
[2, 6, 8, 4, 6, 2]
[4, 12, 16, 8, 12, 4]
56
```

Python Decorators

In [402…

```
def div(a,b):
    print(a / b)
div(4,2)
# but what if we pass the value 2, 4
```

```
2.0
```

In [404…

```
def div(a, b):
    print(a / b)
div(2,4)
```

```
0.5
```

In [406…

```
def div(a,b):

    if a<b:
        a,b = b,a
    print(a / b)

div(2,4)
```

```
2.0
```

In [408…

```
# using help of the decorator you can add the extra feature in the exicting func

def div(a,b):
    print(a / b)

def div_decorator(func): # hear div_dectorator will accept the div function
    def inner(a,b):
        if a<b:
            a,b = b,a
        return func(a,b)
    return inner

div = div_decorator(div)

div(2,4)
```

```
2.0
```

In [410…

```
def my_decorator(func):
    def wrapper():
        print("Something is happening before the function is called.")
        #func()
        print("Something is happening after the function is called.")
    return wrapper

@my_decorator
def say_hello():
    print("Hello!")

say_hello()
```

```
Something is happening before the function is called.
Something is happening after the function is called.
```

In [416…

```python
def my_decorator(func):
    def wrapper():
        print("Something is happening before the function is called.")
        func()
        print("Something is happening after the function is called.")
    return wrapper

@my_decorator
def say_hello():
    print("Hello!")

say_hello()
```

```
Something is happening before the function is called.
Hello!
Something is happening after the function is called.
```

Special Variable_name_

In [418…

```python
__name__
```

Out[418…

```
'__main__'
```

In [420…

```python
print(__name__)
```

```
__main__
```

In [422…

```python
__main__
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[422], line 1
----> 1 __main__

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

In [424…

```python
__name__
```

Out[424…

```
'__main__'
```