Factorial of Number using Python

In [3]:

```python
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
```

Recurssion

In [13]:

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


wish()
```

```
hello
hi
```

In [14]:

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

Out[14]:

```
3000
```

In [15]:

```python
import sys
sys.setrecursionlimit(200)
print(sys.getrecursionlimit())
```

```
200
```

In [16]:

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

Out[16]:

```
200
```

In [17]:

```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
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
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File _pydevd_sys_monitoring\\_pydevd_sys_monitoring_cython.pyx:580, in _pydevd_sys_m
onitoring_cython._get_func_code_info()

KeyError: 'C:\\Users\\admin\\AppData\\Local\\Temp\\ipykernel_19140\\2680176686.py'

During handling of the above exception, another exception occurred:

KeyError                                  Traceback (most recent call last)
File ~\AppData\Roaming\Python\Python313\site-packages\debugpy\_vendored\pydevd\pydev
d_file_utils.py:901, in get_abs_path_real_path_and_base_from_file(filename, NORM_PAT
HS_AND_BASE_CONTAINER)
    900 try:
--> 901     return NORM_PATHS_AND_BASE_CONTAINER[filename]
    902 except:

KeyError: 'C:\\Users\\admin\\AppData\\Local\\Temp\\ipykernel_19140\\2680176686.py'

During handling of the above exception, another exception occurred:

KeyError                                  Traceback (most recent call last)
File ~\AppData\Roaming\Python\Python313\site-packages\debugpy\_vendored\pydevd\pydev
d_file_utils.py:433, in _abs_and_canonical_path(filename, NORM_PATHS_CONTAINER)
    432 try:
--> 433     return NORM_PATHS_CONTAINER[filename]
    434 except:

KeyError: 'C:\\Users\\admin\\AppData\\Local\\Temp\\ipykernel_19140\\2680176686.py'

During handling of the above exception, another exception occurred:

FileNotFoundError                         Traceback (most recent call last)
File <frozen ntpath>:738, in realpath(path, strict)

FileNotFoundError: [WinError 2] The system cannot find the file specified: 'C:\\User
s\\admin\\AppData\\Local\\Temp\\ipykernel_19140\\2680176686.py'

During handling of the above exception, another exception occurred:

FileNotFoundError                         Traceback (most recent call last)
File <frozen ntpath>:675, in _getfinalpathname_nonstrict(path, ignored_error)

FileNotFoundError: [WinError 2] The system cannot find the file specified: 'C:\\User
s\\admin\\AppData\\Local\\Temp\\ipykernel_19140\\2680176686.py'

During handling of the above exception, another exception occurred:

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 (167 times)]

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

File <stringsource>:69, in cfunc.to_py.__Pyx_CFunc_4904d5__29_pydevd_sys_monitoring_
cython_object__lParen__etc_to_py_4code_11instruction_3exc.wrap()

File _pydevd_sys_monitoring\\_pydevd_sys_monitoring_cython.pyx:898, in _pydevd_sys_m
onitoring_cython._unwind_event()

File _pydevd_sys_monitoring\\_pydevd_sys_monitoring_cython.pyx:582, in _pydevd_sys_m
onitoring_cython._get_func_code_info()

File ~\AppData\Roaming\Python\Python313\site-packages\debugpy\_vendored\pydevd\pydev
d_file_utils.py:922, in get_abs_path_real_path_and_base_from_file(filename, NORM_PAT
HS_AND_BASE_CONTAINER)
    919     elif f.endswith("$py.class"):
    920         f = f[: -len("$py.class")] + ".py"
--> 922 abs_path, canonical_normalized_filename = _abs_and_canonical_path(f)
    924 try:
    925     base = os_path_basename(canonical_normalized_filename)

File ~\AppData\Roaming\Python\Python313\site-packages\debugpy\_vendored\pydevd\pydev
d_file_utils.py:459, in _abs_and_canonical_path(filename, NORM_PATHS_CONTAINER)
    456 abs_path = _apply_func_and_normalize_case(filename, os_path_abspath, isabs,
normalize)
    458 normalize = True
--> 459 real_path = _apply_func_and_normalize_case(filename, os_path_real_path, isab
s, normalize)
    461 # cache it for fast access later
    462 NORM_PATHS_CONTAINER[filename] = abs_path, real_path

File ~\AppData\Roaming\Python\Python313\site-packages\debugpy\_vendored\pydevd\pydev
d_file_utils.py:486, in _apply_func_and_normalize_case(filename, func, isabs, normal
ize_case, os_path_exists, join)
    481 if filename.startswith("<"):
    482     # Not really a file, rather a synthetic name like <string> or <ipython
-...>;
    483     # shouldn't be normalized.
    484     return filename
--> 486 r = func(filename)
    488 if not isabs:
    489     if not os_path_exists(r):

File <frozen ntpath>:750, in realpath(path, strict)
File <frozen ntpath>:684, in _getfinalpathname_nonstrict(path, ignored_error)

File <frozen ntpath>:625, in _readlink_deep(path, ignored_error)

RecursionError: maximum recursion depth exceeded
```

Factorianl using Recurssion

In [22]:

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

result = fact(5)

result
```

Out[22]:

```
120
```

Anonymous Function|Lambada

In [23]:

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

square(5)
```

Out[23]:

```
25
```

In [24]:

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

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

```
25
```

In [25]:

```
#lambda expresion or lambda function
f = lambda a : a * a # hear a is an argument & operation in the argument is a * a
result = f(5)
result
```

Out[25]:

```
25
```

In [26]:

```
f = lambda a, b : a + b
f1 = lambda a, b : a - b

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

print(result)
print(result1)
```

```
5
3
```

In [27]:

```
f = lambda a, b : a + b
f1 = lambda a, b : a - b
f2 = lambda a,b : a * b

result = f(1,4)
result1 = f1(4,1)
result2 = f2(4,1)

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

```
5
3
4
```

In [28]:

```
#lets define function which will add 2 number
# we are defining a function whcih doesnt have name
#lambda expresion or lambda function

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 [29]:

```
import keyword
keyword.kwlist
```

Out[29]:

```
['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']
```

How we can use Lambda in other Function like Filter,Map,Reduce

In [30]:

```python
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 [31]:

```python
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 [32]:

```
# lets write above function using help of lambda & lambda helps to reduce the line
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 [33]:

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

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

print(odd)
```

```
[3, 9]
```

In [34]:

```
# lets write above function using help of lambda & lambda helps to reduce the line
nums = [3,2,6,8,4,6,2,9, 34,77, 120]

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, 34, 120]
[3, 9, 77]
```

In [35]:

```
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 [36]:

```
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 [37]:

```
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))
doubble_1 = list(map(lambda n : n-2, evens))

print(evens)
print(double)
print(double_)
print(doubble_1)
```

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

In [38]:

```
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 [39]:

```
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 [40]:

```
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 [41]:

```
a = [7,8]
print(type(a))
```

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

In [42]:

```
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 [43]:

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

```
2.0
```

In [44]:

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

```
0.5
```

In [45]:

```
def div(a,b):

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

div(2,4)
```

```
2.0
```

In [46]:

```
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 [47]:

```
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 [48]:

```
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.
```