

#Recursion

1. If a function called itself is called recursive function
2. Recursive function keeps on executing until its termination

#If we want the beyond the limit, need to set

import sys

x = 1500

sys.setrecursionlimit(x)

#Direct Function

```
def d1():
```

```
    print("D1 Function")
```

```
    d1()
```

#Works Infinity Times, so we want stop this using base case

```
d1()
```

Output

D1 Function

D1 Function

D1 Function.....

RecursionError: maximum recursion depth exceeded while calling a Python object

#Find the recursion limit

i = 0

def d1():

 global i

 i=i+1

 print("D1 Function", i)

 d1()

#Calling within the scope gain, recursion limit is 1000/ or platform dependent

d1()

Output

D1 Function 1

D1 Function 2

D1 Function 3.....

D1 Function 993

D1 Function 994

D1 Function 995

RecursionError: maximum recursion depth exceeded while calling a Python object

Recursion limit

```
import sys
print(sys.getrecursionlimit()) # 1000
```

```
sys.setrecursionlimit(1000)
print(sys.getrecursionlimit()) #1000
```

```
i = 0
def d1():
    global i
    i = i+1
    print("D1 Function", i)
    d1()
d1()
```

Output

1000

1000

D1 Function 1

D1 Function 2

D1 Function 3

D1 Function 4

.....

D1 Function 990

D1 Function 991

D1 Function 992

D1 Function 993

D1 Function 994

D1 Function 995

D1 Function 996

RecursionError: maximum recursion depth exceeded while calling a Python object

#Indirect Recursion

```
def d1():  
    print("d1 function")  
    d2()
```

```
def d2():  
    print("d2 function")
```

```
d1()
```

output

d1 function

d2 function

Recursion using factorial

```
def factorial(n):  
    if n==1:  
        return 1  
    else:  
        return n*factorial(n-1)
```

```
n = int(input("Enter a number: "))
```

```
r = factorial(n)
```

```
print('Factorial of ', n, 'is', r)
```

Output

Enter a number: 5

Factorial of 5 is 120