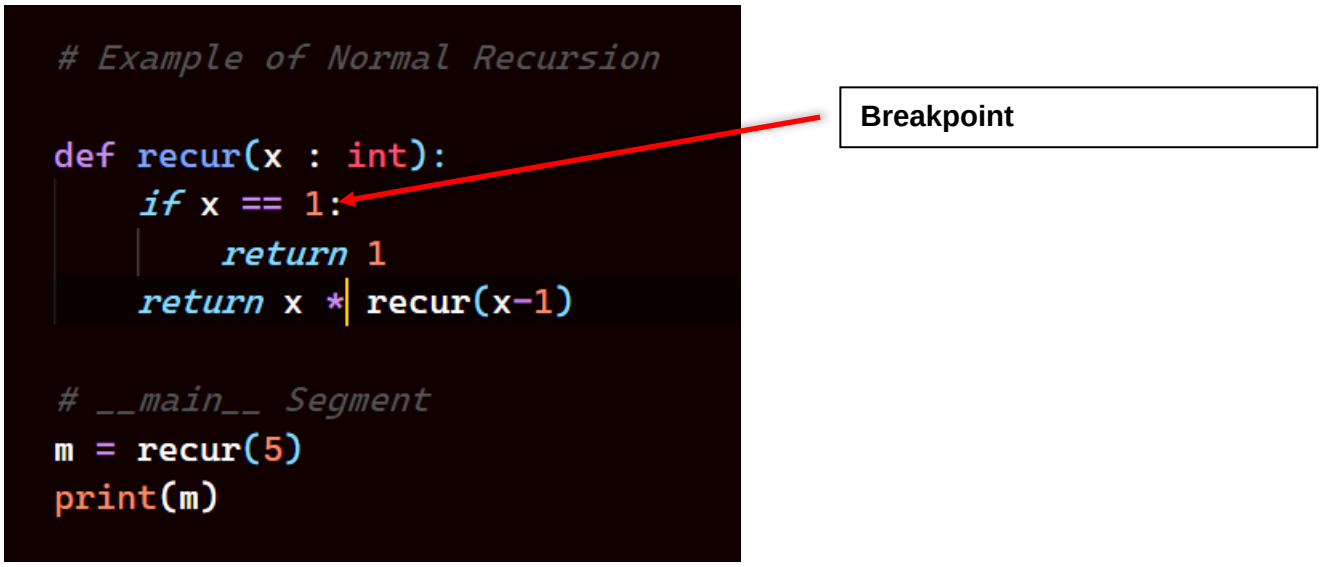
Recursion Limit:

Recursion is a process where we call a function directly or indirectly multiple number of times. But then, if we do not set a breakpoint then it should run infinite ... number of times, which will get the computer memory hanged.

But, to solve that problem Python has introduced, what we call as, Global Interpreter Lock. This is a feature, which enables a maximum limit for the depth of the Recursion in the Interpreter Stack that we perform in Python. It prevents the Recursion from causing an overflow of the C stack and crashing Python.

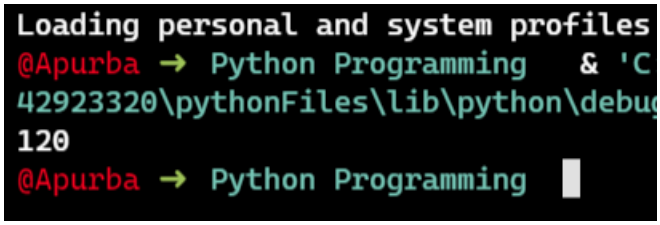
Let's see an example of a normal recursion in Python:




```
# Example of Normal Recursion

def recur(x : int):
    if x == 1:
        return 1
    return x * recur(x-1)

# __main__ Segment
m = recur(5)
print(m)
```



Now, if we remove the **Breakpoint** then the **Output** would show an Error… RecursionError: maximum recursion depth exceeded. Since, we have not given a limit where would the Recursion stop so it goes on … and due to the Global Interpreter Lock 🔒 it shows **RecursionError**.

Let's See the **Code** and the **Output**

```
# Example of Recursion Error

def recur(x : int):
    return x * recur(x-1)

# __main__ Segment
m = recur(5)
print(m)
```

O/P:

```
@Apurba → Python Programming   c:; cd 'c:\Users\Apurba\Desktop
xe' 'c:\Users\Apurba\.vscode\extensions\ms-python.python-2021.5
r\Python Programming\School2.py'
Traceback (most recent call last):
  File "c:\Users\Apurba\Desktop\Text Folder\Python Programming\
    m = recur(5)
  File "c:\Users\Apurba\Desktop\Text Folder\Python Programming\
    return x * recur(x-1)
  File "c:\Users\Apurba\Desktop\Text Folder\Python Programming\
    return x * recur(x-1)
  File "c:\Users\Apurba\Desktop\Text Folder\Python Programming\
    return x * recur(x-1)
  [Previous line repeated 996 more times]
RecursionError: maximum recursion depth exceeded
```

Note: Here, it is said that the line i.e the recursive function has been Executed 996 times …

Now, you can also get to know what is the Maximum Recursion Limit in your System.

To know that do the following …

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

```
() → int

Return the current value of the recursion limit.

The recursion limit is the maximum depth of the Python
interpreter stack. This limit prevents infinite recursion from
causing an overflow of the C stack and crashing Python.
```

O/P

```
@Apurba → Python Programming
xe' 'c:\Users\Apurba\.vscode\ext
r\Python Programming\School2.py'
1000
```

And for instance you can change the Recursion limit of your System too..

Here is how you do it,

```
import sys
sys.setrecursionlimit(2000)
print(sys.getrecursionlimit())
```

Note: It sets the **maximum depth** of the **Python Interpreter Stack** to **n**. This limit prevents infinite **recursion** from causing an overflow of the **C stack** and crashing **Python.** The highest possible limit is platform- dependent.

O/P

```
@Apurba → Python Programming
xe' 'c:\Users\Apurba\.vscode\ext
r\Python Programming\School2.py'
2000
```