

# What Is Recursion?

- The phenomenon of defining something in terms of itself
- Can be used to solve a problem and then combine the results
- In Python, a function is said to be recursive if it calls itself inside its own body

# Structure of a Recursive Function (Continued)

```
def my_func( par1, par2, .... )  
  
    if <base condition>:  
        return <base condition output>  
  
    else:  
        <do_some_operation>  
        my_func( <reduced size of input>)
```

# Structure of a Recursive Function

- Should consist of a base case or a terminating condition to know when to stop
- Should also contain a call to itself with reduced input size
- Without the base case, the function will call itself infinitely, leading to a stack overflow

# Advantages of Recursion

- Makes the code look elegant and neat
- A larger problem can be solved easily by breaking it down into smaller subproblems
- Combining recursion with memorization leads to reduction in overall time

# Disadvantages of Recursion

- Difficult to debug sometimes, as it is often not possible to unwind the recursive logic
- If the base condition is not specified, it will lead to stack overflow
- Recursive functions sometimes solve the same subproblems multiple times



# Summary

- Recursion can be used to solve a problem and then combine the results
- It consists of a terminating condition and a call to itself
- A recursive function may lead to stack overflow due to infinite function calls