**Recursion**

* A function call itself again and again is called recursion
* A base condition for termination must be present

**Types of Recursion**

* There are five types of recursion

1. Tail recursion
2. Head recursion
3. Tree recursion
4. Indirect recursion
5. Nested recursion

* There are two phases in recursion

1. Calling phase--->Execution is done before recursion function call
2. Returning phase-->Execution is done after calling recursion function call

### Tail Recursion

* + Recursive call is the last statement of the function
  + Everything is performed at calling time
  + Tail Recursion can converted into a loop

### Head Recursion

* + Recursive call is the first statement in the function
  + Everything is performed at returning time
  + Head Recursion can be converted into a loop

### Tree Recursion

* + Function call is present more than one time in the function

### Indirect Recursion

* + More than one function is present that both function is called indirectly
  + A function call is placed in the another function
  + So the function present is called in a circular manner

### Nested Recursion

* + A recursive function will pass a parameter as a recursive function call
  + Recursion inside the recursion
* If we clearly know what to be happen at that time loops can used because it is efficient than recursion
* So recursion is suggested for doing certain mathematical functions
* But during compiling compiler will converts recursion into loops for efficiency