Recursion and Iteration are both looping mechanisms.  Recursion utilizes concepts within mathematical induction and allocates memory on the Stack each succeeding recursive function call.  Unlike an iterative loop, a recursive loop will examine an exact subset of the original problem and usually has a basis case of 1; therefore, starting at some known value of n running down to 1. As an elementary explanation, algorithmic work can be done prior to recursive calls being called and added to the stack or after.

Iterative loops typically run from either 0 or 1 to some value of n; however, they can have negative or positive starting values and ending values – provided there is distance between the initial value and the nth value. These loops repeat the same steps within the boundaries of the definition of the loop. Runtime can be effected by adjusting the iteration; for example, for an iterative loop to be expressed logarithmically the incremented value must use some type of multiplicative – multiplication or division – resulting in a runtime of O(log(n)), provided the inner algorithm runs at O(1).