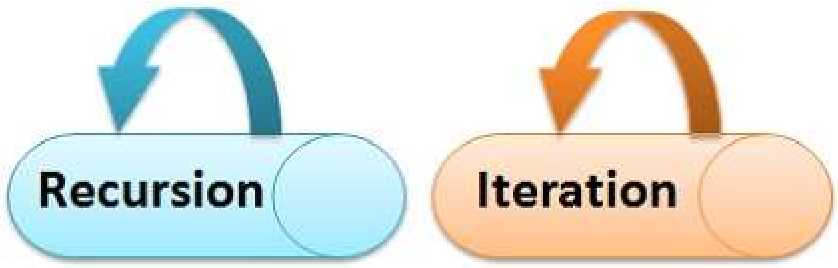
Article taken from: https://techdifferences.com/difference-between-recursion-and-iteration-2.html

Difference Between Recursion and Iteration



Recursion and iteration both repeatedly executes the set of instructions. Recursion is when a statement in a function calls itself repeatedly. The iteration is when a loop repeatedly executes until the controlling condition becomes false. The primary difference between recursion and iteration is that is a **recursion** is a process, always applied to a function. The **iteration** is applied to the set of instructions which we want to get repeatedly executed.

Content: Recursion Vs Iteration

Comparison Chart

BASIS FOR COMPARISON	RECURSION	ITERATION
Basic	The statement in a body of function calls the function itself.	Allows the set of instructions to be repeatedly executed.
Format	In recursive function, only termination condition (base case) is specified.	Iteration includes initialization, condition, execution of statement within loop and update (increments and decrements) the control variable.

BASIS FOR COMPARISON	RECURSION	ITERATION
Termination	A conditional statement is included in the body of the function to force the function to return without recursion call being executed.	The iteration statement is repeatedly executed until a certain condition is reached.
Condition	If the function does not converge to some condition called (base case), it leads to infinite recursion.	If the control condition in the iteration statement never become false, it leads to infinite iteration.
Infinite Repetition	Infinite recursion can crash the system.	Infinite loop uses CPU cycles repeatedly.
Applied	Recursion is always applied to functions.	Iteration is applied to iteration statements or "loops".
Stack	The stack is used to store the set of new local variables and parameters each time the function is called.	Does not uses stack.
Overhead	Recursion possesses the overhead of repeated function calls.	No overhead of repeated function call.
Speed	Slow in execution.	Fast in execution.
Size of Code	Recursion reduces the size of the code.	Iteration makes the code longer.

Definition of Recursion

Python allows a function to call itself within its code. That means the definition of the function possesses a function call to itself. Sometimes it is also called "**circular definition**". The set of local variables and parameters used by the function are newly created each time the function calls itself and are stored at the top of the stack. But, each time when a function calls itself, it does not create a new copy of that function. The recursive function does not significantly reduce the size of the code and does not even improve the memory utilization, but it does some when compared to the iteration.

To terminate the recursion, you must include a select statement in the definition of the function to force the function to return without giving a recursive call to itself. The absence of the select statement in the definition of a recursive function will let the function in infinite recursion once called.

Definition of Iteration

Iteration is a process of executing the set of instructions repeatedly till the condition in iteration statement becomes false. The iteration statement includes the initialization, comparison, execution of the statements inside the iteration statement and finally the updating of the control variable. After the control variable is updated it is compared again, and the process repeats itself, till the condition in iteration statement turns out to be false. The iteration statements are "for" loop, "while" loop, "do-while" loop.

The iteration statement does not use a stack to store the variables. Hence, the execution of the iteration statement is faster as compared to recursive function. Even the iteration function do not have the overhead of repeated function calling which also make its execution faster than recursive function. The iteration is terminated when the control condition becomes false. The absence of control condition in iteration statement may result in an infinite loop, or it may cause a compilation error.

Key Differences Between Recursion and Iteration

1. Recursion is when a method in a program repeatedly calls itself whereas, iteration is when a set of instructions in a program are repeatedly executed.
2. A recursive method contains set of instructions, statement calling itself, and a termination condition whereas iteration statements contain initialization, increment, condition, set of instruction within a loop and a control variable.
3. A conditional statement decides the termination of recursion and control variable's value decide the termination of the iteration statement.

4. If the method does not lead to the termination condition it enters to infinite recursion. On the other hand, if the control variable never leads to the termination value the iteration statement iterates infinitely.
5. Infinite recursion can lead to system crash whereas, infinite iteration consumes CPU cycles.
6. Recursion is always applied to method whereas, iteration is applied to set of instruction.
7. Variables created during recursion are stored on stack whereas, iteration doesn't require a stack.
8. Recursion causes the overhead of repeated function calling whereas, iteration does not have a function calling overhead.
9. Due to function calling overhead execution of recursion is slower whereas, execution of iteration is faster.
10. Recursion reduces the size of code whereas, iterations make a code longer.

Conclusion:

The recursive function is easy to write, but they do not perform well as compared to iteration whereas, the iteration is hard to write but their performance is good as compared to recursion.