# ServiceNow Debugger Example: Step Over, Step Into, Step Out

This example demonstrates how Step Over, Step Into, and Step Out behave in the ServiceNow Script Debugger.

## Business Rule Script Example

(function executeRule(current, previous /\*null when async\*/) {  
  
 function logMessage(msg) {  
 gs.info("Message: " + msg);  
 }  
  
 function calculateSum(a, b) {  
 var sum = a + b;  
 gs.info("Sum is: " + sum);  
 return sum;  
 }  
  
 function mainLogic() {  
 gs.info("Step 1: Starting process");  
  
 logMessage("Hello Debugger"); // LINE A  
  
 var result = calculateSum(5, 10); // LINE B  
  
 gs.info("Step 3: Final Result = " + result); // LINE C  
 }  
  
 mainLogic();  
  
})(current, previous);

## Debugger Behavior

### At LINE A: logMessage("Hello Debugger");

• Step Over → Executes the whole logMessage() function in one go, debugger moves to LINE B.  
• Step Into → Moves inside the logMessage() function and stops at gs.info("Message: " + msg).  
• Step Out → If you are inside logMessage(), it will run the rest and return you back to LINE B.

### At LINE B: calculateSum(5, 10);

• Step Over → Executes calculateSum() fully, returns 15, then moves to LINE C.  
• Step Into → Goes inside calculateSum() and lets you debug line by line (var sum..., gs.info...).  
• Step Out → If you are inside calculateSum(), it exits and returns control back to LINE C.

## Summary

• Step Over → Run function completely without going inside.  
• Step Into → Enter inside the function line by line.  
• Step Out → Exit the function and return to caller.