

# ADVANCED SERVICE NOW SCRIPTING

- Duration of Training : 70 hrs.
- Batch type : Weekdays/Weekends
- Mode of Training : Classroom/Online/Corporate  
Training



# COURSE CONTENT

## INTRODUCTION TO ADVANCE SNOW SCRIPTING OBJECTIVES

- **Introduction to Advance Scripting Objectives**
- About This Learning Module
- Introduction to ServiceNow Scripting
- ServiceNow APIs
- ServiceNow Script Editor
- Background Scripts
- Fix Scripts
- Script Actions
- System Properties

- **Client-Side Scripting**

- Client-Side Scripting Objectives
- About This Learning Module
- Introduction to Client-side Scripting
- The Glide Form (g\_form) Class
- The Glide User (g\_user) Class
- UI Scripts
- UI Actions
- Mobile client Glide Form (g form) scripting
- and migration
- AJAX
- Catalog client scripts

- **Client-Side Scripting Design and Processing**
- Client Scripts: Enhance user experience by reducing
- form completion time.
- Form Load: Ensure proper client-side processing by
- loading forms first.
- Field Control: Use methods like disabling list editing,
- business rules, and data policies to manage field
- values.
- Minimize Server Lookups: Utilize g\_scratchpad for efficient
- data transfer from server to client, avoiding performance-
- heavy methods like Glide Record.
- Asynchronous Glide Ajax: Dynamically request server data,
- improving performance.
- **Client-Side Scripting Design and Processing**
- Use set Value() Correctly: Include the display
- value to prevent unnecessary server calls.
- Prefer UI Policies: They offer better control and do not
- require scripting.
- Input Validation: Use client scripts to validate user input
- before submission.
- Script Order: Control execution order to prevent conflicts
- **Client-Side Scripting Design and Processing**
- Avoid DOM Manipulation: Use Glide Form API for
- maintainability.
- Global Client Scripts: Limit their use to avoid performance
- issues.
- Encapsulate Code: Wrap code in functions to manage
- variable scope.

- Run Necessary Scripts: Optimize scripts to execute only when needed, using flags like is Loading and new Value checks.

- **Server-Side Scripting**

- Glide Server APIs
- Classic Business rules
- Script includes
- Processors
- Scripts - Background module
- Installation settings
- Using Duration Calculator to calculate a due date
- Querying tables in script

- **Server-Side Scripting**

- Running order guides automatically
- Scriptable assignment of execution plans
- Using regular expressions in server side scripts
- Scriptable service catalog variables
- Setting a Glide Record variable to null Schedule Pages
- XML Document script object
- Background Script Module
- UI pages
- UI macros
- Debugging scripts
- Script Debugger and Session Log

- Session debug
- Debugging applications
- Debugging business rules
- Debugging classifications
- Field watcher
- Writing to the debug log
- JavaScript debug window
- JS Code Coverage Debug
- Scheduled Script Executions and Events
- What is a Scheduled Script Execution?
- Creating a Scheduled Script Execution
- Scheduled Script Execution Scripts
- What is on the Schedule?
- What are Events?
- Event Registry
- Generating Events
- Checking for Events
- Responding to Events
- Using Custom Queues - Advanced Topic
- Debugging Scripts
- Script Debugger and Session Log
- Session debug
- Debugging applications
- Debugging business rules
- Debugging classifications
- Field watcher

- Writing to the debug log
- JavaScript debug window
- JS Code Coverage Debug
- Client Side Scripting Best Practices
- Server Side Scripting Best Practices

## FLOW DESIGNER

- Using Flow Designer Objectives
- About This Learning Module
- Exercise: Fork Repository and Import Application for the Using Flow Designer Module
- Exercise: Create a Branch for the Using Flow Designer Module
- What is a Flow?
- Flow Designer User Interface
- Opening Flows
- Exercise: View a Flow
- Deleting Flows
- Creating Flows
- Flow Tab Anatomy
- Triggering Flows
- Record Triggers
- Date Triggers
- Application Triggers

- Adding Actions to a Flow
- Saving Flows
- Using Flow Variables
- Using Error Handler
- Exercise: Create a Flow
- Testing Flows
- Viewing Flow Execution Details
- Activating Flows
- Running Flows with Alternate Permissions
- Exercise: Test a Flow
- Exercise: Activate a Flow
- Configuring the Ask for Approval Action
- Exercise: Add Approval
- Flow Logic
- Adding If Branches
- Adding For Each Branches
- Adding Do the Following Until Branches
- Adding Do the Following in Parallel Branches
- Adding Non-Branching Flow Logic
- Exercise: Add Flow Logic
- Exercise: Save Your Using Flow Designer Work (Optional)
- Test Your Using Flow Designer Knowledge
- Flow Designer Recap
- **Developing for Flow Designer**
- Developing for Flow Designer Objectives
- About This Learning Module
- Exercise: Fork Repository and Import Application for the Developing for Flow Designer Module

- Exercise: Create a Branch for the Developing for Flow Designer Module
- Working with Spokes
- Action Anatomy
- Creating Actions
- Adding Steps
- Configuring Script Steps
- Testing Actions
- Exercise: Create an Action
- Adding Action Inputs
- Adding Action Outputs
- Publishing Actions
- Exercise: Add Action Inputs and Outputs
- What is a Subflow?
- Subflows Versus Actions
- Creating Subflows
- Adding Subflow Inputs
- Adding Subflow Outputs
- Assigning Subflow Outputs
- Testing Subflows
- Publishing Subflows
- Using Subflows
- Exercise: Create and Test a Subflow
- Exercise: Use a Subflow
- Challenge: Create Scheduled Flow to Update Overdue NeedIt Tasks

- Scripting with Flows, Subflows, and Actions
- Exercise: Create and Use Code Snippets
- Exercise: Save Your Developing for Flow Designer Work (Optional)
- Test Your Developing for Flow Designer Knowledge
- Developing for Flow Designer Recap

## **Notifications in Flow Designer**

- Notifications in Flow Designer Objectives
- About This Learning Module
- Exercise: Fork Repository and Import Application for the Notifications in Flow Designer Module
- Exercise: Create a Branch for Notifications in Flow Designer
- Notifying Users from a Flow
- Outbound Communication Actions
- Exercise: Get a Temporary Email Address
- Exercise: Send a Notification
- Outbound Email Configuration
- Exercise: Outbound Email Configuration
- Exercise: Receive an Email
- Inbound Email and Flows
- Parsing Inbound Email Details for a Flow
- Identifying the Sender
- Creating Inbound Email Flows

- Exercise: Inbound Email Configuration
- Exercise: Create Inbound Email Flow
- Exercise: Save Your Notifications in Flow Designer Work (Optional)
- Notifications in Flow Designer Module Recap

## ADVANCED JAVASCRIPT

### Introduction to JavaScript

- JavaScript syntax editor
- Syntax editor plugin
- Context menu in the syntax editor
- Script syntax error checking
- Searching for errors by line
  
- JavaScript engine on the platform
- JavaScript modes
- JavaScript engine feature support
- JavaScript API Context-sensitive help
  
- Functions
- try/catch/finally statements
- ServiceNow GlideRecord
- Introduction to arrays
- Array forEach
- Common array methods
- ServiceNow Array Util
- Functions and arrays
- Functions with multiple parameters

- Introduction to objects
- Checking if an object has a property
- Finding all properties in an object
- Arrays of Objects
- JSON Stingily and Parse
- More string methods
- Recursion
- Classes
- Passing objects to functions
- Class Inheritance
- Scripted REST APIs
- Exercise : Build a scripted REST API
- Build a script include
- Helpful resources

## Learn MORE JavaScript on the ServiceNow Platform with Earl Duque

- Section 2 Introduction
- The for...in loop
- Exercise - The for...in loop
- Understanding "this"
- Exercise - Understanding "this"
- More on scope
- Lesson 4 - Hoisting
- Exercise - Hoisting

- Section 2 Introduction
- The for...in loop
- Exercise - The for...in loop
- Understanding "this"
- Exercise - Understanding "this"
- More on scope
- Lesson 4 - Hoisting
- Exercise - Hoisting
- Date and Time
- Exercise - Date and Time
- Regular Expressions (Reg Ex)
- Exercise - Regular Expressions (Reg Ex)
- The practice of Name spacing
- Exercise - The practice of Name Spacing
- Glide System (more than just gs.info!)
- Glide Aggregate
- Exercise - Glide Aggregate

## Learn MORE JavaScript on the ServiceNow Platform with Hack Now

- GlideDateTime
- GlideUser
- GlideAjax
- GlideForm (g\_form)
- Exercise - GlideForm (g\_form)
- GlideQuery
- Exercise - GlideQuery

## Learn ECMAScript 2021 on the ServiceNow Platform with HackNow

- Strings - Template literals
- Strings – String padding
- Strings – New string methods like startsWith(), endsWith(), trimStart(), and trimEnd()
- Strings – Even more string methods like matchAll() and includes()
- Functions - Arrow functions
- Functions – Default Parameters
- Functions – Rest and Spread Operators
- Functions – the Name property
- Functions – Trailing commas in function syntax
- Objects – Object Literal Enhancements
- Objects – More on Classes
- Objects – Symbols
- Objects – Optional chaining
- Objects – New methods like Object.entries(), Object.values(), Object.fromEntries(), and Object.getOwnPropertyDescriptors()
- Arrays – The for..of Loop
- Arrays – Maps and Sets
- Arrays – More utility functions like .from() and .of()
- Arrays – New methods like flatMap() and flat()



8055223360



[www.radicaltechnologies.co.in](http://www.radicaltechnologies.co.in)



Training Queries : [training@radicaltechnologies.co.in](mailto:training@radicaltechnologies.co.in)

---

Aundh | Kharadi | Hinjewadi | Bangalore - Electronic city |  
Kerala - Kochi | Calicut | Trivandrum | UK

