**WEEK 4**

**Module 4 – ServiceNow Scripting Fundamentals and Functions**

### **1. Scripting in ServiceNow**

Scripting in ServiceNow allows developers and administrators to customize the platform’s behavior, create automated workflows, and manage processes. It involves using **JavaScript** to extend the functionality of ServiceNow, enabling advanced configurations and automations.

#### **ServiceNow Scripting Overview:**

ServiceNow supports **two main types of scripting**:

1. **Client-Side Scripting:**
   * Executes in the user's web browser.
   * Typically used to control form fields, implement form validation, or enhance the user interface.
   * Common types of client-side scripts include:
     + **UI Policies:** Manage form behavior such as hiding fields or making fields mandatory.
     + **Client Scripts:** Can run at different times: when the form loads, when it changes, or when it's submitted.
     + **Example Use Case:** Automatically filling in the “Department” field when a user selects their name from a list.
2. **Server-Side Scripting:**
   * Runs on the ServiceNow server and is responsible for database interactions, data manipulation, and background tasks.
   * Includes:
     + **Business Rules:** Run when records are created, updated, or deleted.
     + **Script Actions:** Triggered by specific events in the system.
     + **Scheduled Jobs:** Automate tasks like cleanup or recurring operations.
     + **Example Use Case:** Running a scheduled job to automatically close inactive incidents after 30 days.

#### **Key Scripting Concepts:**

1. **Client Scripts:**
   * These run in the browser and are typically used for **validations**, **data auto-filling**, and **form manipulations**.
   * **Types:**
     + **onLoad:** Triggered when a form is loaded.
     + **onChange:** Triggered when a field value changes.
     + **onSubmit:** Validates the form when the user submits it.
     + **onCellEdit:** Triggered when a cell in a list is edited.
   * **Example Use Case:** Prevent a form from being submitted if mandatory fields are not filled.
2. **Business Rules:**
   * Server-side scripts that run in response to record changes (insert, update, delete).
   * Useful for automating actions like setting field values, sending notifications, or modifying records when a condition is met.
   * **Example Use Case:** Automatically change the status of an incident to "Resolved" when all tasks related to the incident are closed.
3. **Script Includes:**
   * Server-side reusable code that can be invoked by other scripts (like business rules, workflows, or client scripts).
   * Ideal for storing functions or logic that need to be used across multiple parts of the platform.
   * **Example Use Case:** A function that calculates the working days between two dates, used across multiple business rules to determine SLA compliance.
4. **Glide API:**
   * The **Glide API** provides a set of methods and objects to interact with ServiceNow's database.
   * **GlideRecord:** Used to query, insert, update, and delete records in the database.
   * **Example Use Case:** Using GlideRecord to query all open incidents assigned to a specific user.
5. **UI Actions:**
   * Allow developers to add custom buttons or links on forms and lists to perform specific actions.
   * **Example Use Case:** Adding a button on the incident form to allow users to escalate incidents directly.

### **2. ServiceNow Full Course for Beginners**

This course focuses on configuring and personalizing the **ServiceNow platform** to suit an organization’s needs. It explains the structure of key ServiceNow modules, such as **Incident Management**, **Problem Management**, and **Change Management**, along with the use of **Lists** and **Forms**.

#### **Configuring & Personalizing ServiceNow**

1. **Incident Module:**
   * **Purpose:** The Incident Module helps restore normal service operations quickly and efficiently by tracking and resolving disruptions.
   * **Key Elements:**
     + **Creating Incidents:** Users can log incidents when they experience service disruptions.
     + **Assigning Incidents:** Automatically or manually assign incidents to relevant technicians or teams.
     + **Resolving Incidents:** Technicians update the incident as they work on resolving the issue and change the status once it’s resolved.
   * **Example:** An IT user reports a broken printer, logs an incident, and assigns it to the IT support team.
2. **Problem Module:**
   * **Purpose:** Problem Management is designed to find and address the root cause of incidents, preventing recurring issues.
   * **Key Elements:**
     + **Logging Problems:** Problems are created when recurring incidents are noticed.
     + **Analyzing Problems:** IT teams investigate the root causes.
     + **Resolving Problems:** Implement solutions that prevent future incidents.
   * **Example:** After multiple incidents of server outages, a problem is logged to investigate the server's performance.
3. **Change Module:**
   * **Purpose:** The Change Module ensures that any changes made to IT services (such as software updates or infrastructure changes) are managed in a controlled way to reduce risks and service disruptions.
   * **Key Elements:**
     + **Creating Changes:** Changes can be proposed based on incidents or problems.
     + **Assessing Changes:** Each change request is reviewed to evaluate risks and impacts.
     + **Implementing Changes:** Once approved, changes are scheduled and implemented.
   * **Example:** A major software update is scheduled, and a change request is created to manage the deployment.

#### **Lists and Forms:**

1. **Lists:**
   * Lists display **records** from a specific database in a table format. They allow users to view, filter, and sort records.
   * **Example:** The incident list shows all open incidents, allowing IT teams to filter incidents based on priority, status, or assignee.
2. **Forms:**
   * Forms are used to view and update individual records. Each form is tailored to show the relevant fields for a specific type of record.
   * **Example:** When a user clicks on an incident from the list, the incident form opens, showing fields like the description of the incident, assigned technician, status, and resolution notes.

### **Summary:**

**Scripting in ServiceNow** offers extensive customization possibilities, whether through **client-side scripting** for handling form behaviors or **server-side scripting** for automating backend processes. Key concepts like **client scripts, business rules, Glide API, and UI actions** provide flexibility for creating tailored workflows.

The **ServiceNow Full Course for Beginners** helps understand the platform's modules, such as **Incident, Problem, and Change Management**, and teaches how to configure **lists and forms** to streamline service delivery. These modules are essential in managing IT services and ensuring minimal disruption to organizational processes.

These foundational elements allow ServiceNow administrators and developers to efficiently manage and automate business processes.