ServiceNow Scripting Fundamentals and Functions

(MODULE#4)

SCRIPTING IN SERVICENOW :

Scripting on the ServiceNow platform is used for customizing and automating workflows in the ServiceNow environment. ServiceNow uses JavaScript for scripting, and there are several types of scripts.

1. Client Scripts

- Run on the user's browser and are used to enhance the user interface and experience.
- Form validation, dynamic field updates, and UI feedback.
- onLoad(): Executes when the form loads.
- onChange(): Executes when a specified field value changes.
- onSubmit(): Executes before the form is submitted.
- onCellEdit(): Executes when a cell in a list is edited.

2. Business Rules

- Run on the server side and are used to perform operations or enforce rules when records are inserted, updated, or deleted.
- Data validation, automated notifications, and record updates.
- Before: Executes before the database operation.
- After: Executes after the database operation.
- Async: Executes asynchronously after the database operation.
- Display: Executes before the form is rendered (rarely used).

3. Script Includes

- Define reusable server-side JavaScript functions and classes that can be called from other server-side scripts or from client scripts via AJAX.
- Common Uses: Modular code, utility functions, and complex business logic.
- Global: Available across the entire instance.
- Scoped: Available only within a specific application scope.

4. Scheduled Jobs

- Run server-side scripts on a scheduled basis.
- Data cleanup, periodic notifications, and batch processing.
- Scheduled Script Executions: Scripts that run on a defined schedule.
- Scheduled Jobs: More complex scheduling with additional options.

5. Workflow Scripts

- Execute custom logic within ServiceNow workflows.
- Custom workflow steps, complex logic within workflows.
- Script Tasks: Custom script execution within a workflow.
- Flow Designer: Low-code environment for creating workflows with scriptable actions.

6. REST API Scripts

- Integrate ServiceNow with external systems via REST APIs.
- Data exchange with external systems, third-party integrations.
- Inbound Web Services: Handle incoming API requests.
- Outbound Web Services: Send data to external systems.

7. UI Actions

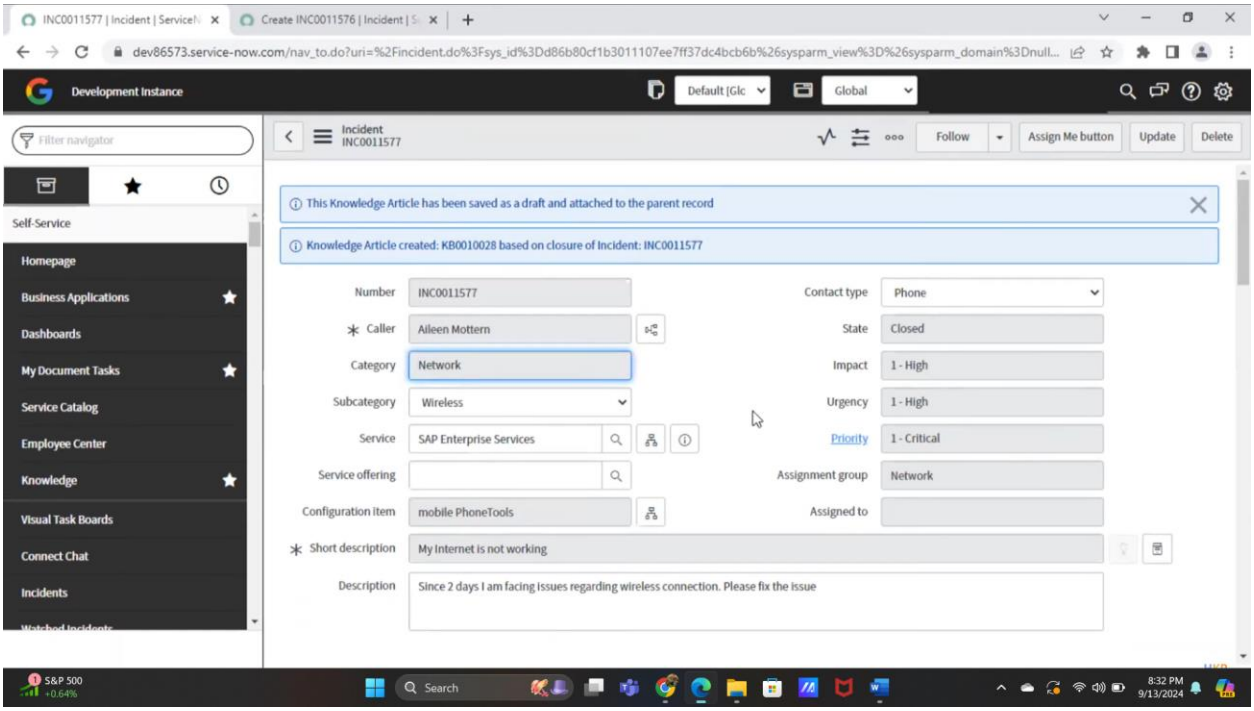
- Create custom buttons or links on forms and lists that execute scripts.
- Custom actions on forms or list views, like custom buttons or context menus.
- Form Buttons: Buttons on forms.
- List Buttons: Buttons in list views.
- Context Menus: Actions in the right-click context menu.

SERVICENOW CORE MODULES :

1.Incident Module:

- Manage and resolve incidents efficiently to restore normal service operation quickly.
- **Incident Form**: Capture details about the incident.
- **Incident List**: View and manage incidents.
- **SLAs**: Service Level Agreements define the expected response and resolution times.
- **Categories and Subcategories**: Define types of incidents for better categorization.

- **Incident States**: Customize the lifecycle of an incident (e.g., New, In Progress, Resolved).
- **Notifications**: Set up automatic email or SMS notifications for incident updates.



The above figure shows the freshly created incident module form.

2. Problem Module:

- Identify and resolve the root causes of incidents to prevent future occurrences.
- **Problem Form**: Document the problem, associated incidents, and known errors.
- **Problem List**: View and manage problems.
- **Known Errors**: Document workarounds and solutions for known issues.
- **Problem States**: Customize the stages (e.g., New, In Investigation, Closed).
- **Root Cause Analysis**: Set up fields and processes for root cause identification.
- **Relationships with Incidents**: Link problems to related incidents to track impact.

3. Change Module:

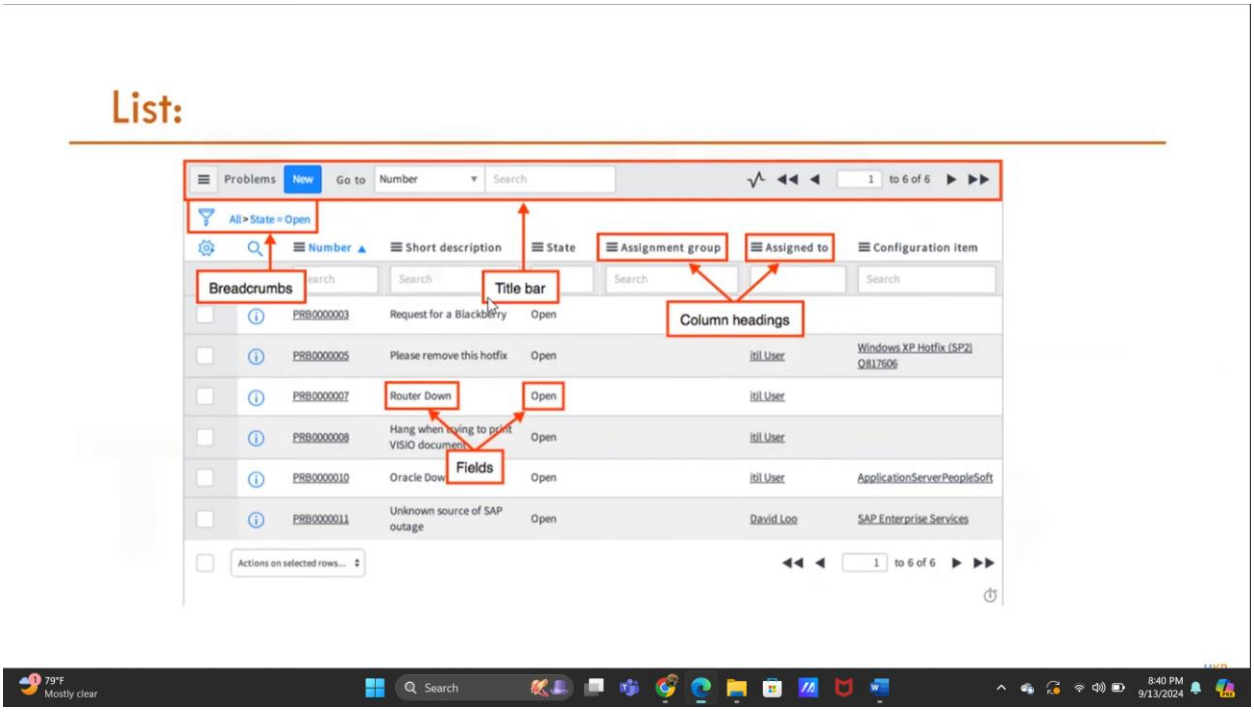
- Manage changes to the IT environment to minimize impact and ensure compliance.
- **Change Form**: Document the details of the change, including impact, risk, and approval.
- **Change List**: View and manage change requests.
- **Change Calendar**: Visualize scheduled changes to avoid conflicts.

- **Change Types**: Define different types of changes (e.g., Standard, Emergency).
- **Change States**: Customize the lifecycle (e.g., New, Approved, Implemented).
- **Approval Process**: Set up workflows for change approvals.

Lists and Forms

1. Lists:

- Display collections of records (e.g., incidents, changes) in a tabular format.
- **Columns**: Select which columns are displayed and their order.
- **Filters**: Create and apply filters to show relevant records.
- **Sort Order**: Define how records are sorted (e.g., by priority, date).
- **List Layouts**: Customize the appearance of lists to fit user needs.

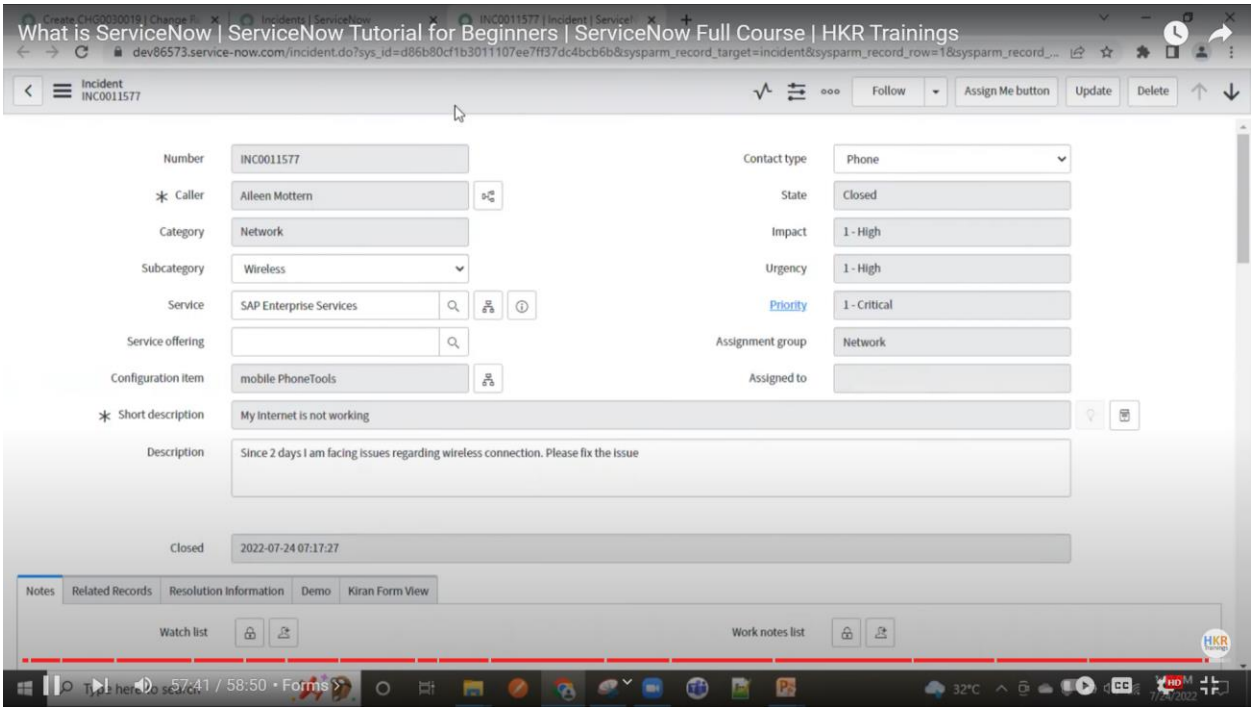


The above figure shows different fields available in the List.

2. Forms:

- Collect and display data for individual records.
- **Fields**: Add, remove, or reorder fields to capture necessary information.
- **Form Layout**: Arrange fields into sections and tabs for better organization.
- **UI Policies**: Define rules for field visibility, mandatory status, or read-only status based on conditions.

- **Client Scripts**: Add custom JavaScript to enhance form behavior, such as field validations or dynamic updates.



The above figure shows the typical view of a form.