

Client Script

Presentated by Priya Talan

Content

- **Introduction**
- **Types**
- **On Load Use Cases**
- **On Change Use Cases**
- **On Cell Edit Use Cases**
- **On Submit Use Cases**
- **Catalog Client Script**
- **UI Policy Vs Client Script**

Introduction

Client scripts allow the system to run JavaScript on the client (web browser) when client-based events occur, such as when a form loads, after form submission, or when a field changes value.

Client Scripts can:

- make fields hidden or visible
- make fields read only or writable
- make fields optional or mandatory based on the user's role
- set the value in one field based on the value in other fields
- modify the options in a choice list based on a user's role
- display messages based on a value in a field

Name Auto Populate User Application Global

Table Incident [incident] UI Type All

Type onLoad Active Inherited

Description Auto populating the Logged In User in the Caller ID Field

Messages

Script

Isolate script

It Indicates whether the client script applies to extended tables.

If true, the client script runs on all views of the table.

This button is used to disable strict mode and direct DOM access for a client script

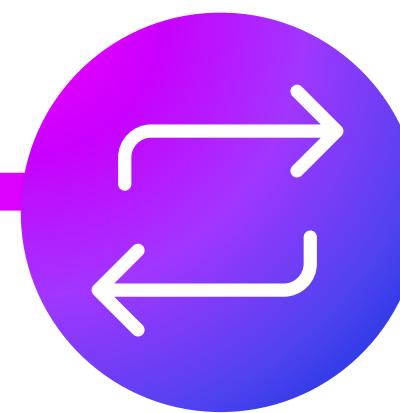
```
function onLoad() {  
    //Type appropriate comment here, and begin script below  
    g_form.setValue('caller_id',g_user.userID);  
}
```

Types



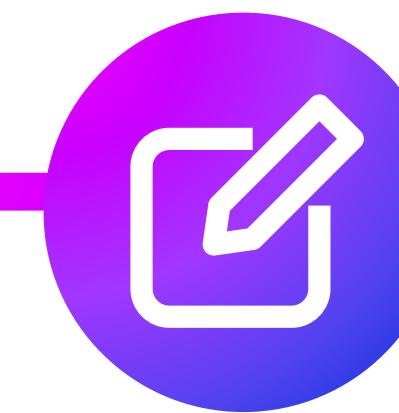
OnLoad

runs when the system first renders the form and before users can enter data.



OnChange

runs when a particular field value changes on the form.



OnCellEdit

runs when the list editor changes a cell value.



OnSubmit

runs when a form is submitted. Typically, onSubmit() scripts validate things on the form. An onSubmit() client script can cancel form submission by returning a value of false.

USE CASES

OnLoad

- Auto Populate the Logged In Username when User opens the Form.
- If the Current Incident is a Child Incident, add a Info message with a clickable Parent Incident number.
- Show an error message if the incident priority is critical.
- Show all the Child Incidents of the Incident in the Description Field using Display BR.

USE CASES

OnChange

- If the incident's short description starts with the string 'Database' set the category as a Database.
- When the Caller is VIP set the impact and urgency to High and show the alert message.
- In a problem record, hide Notes section when problem state is equal to '102'.

USE CASES

OnCellEdit

- Prevents users from changing the State to closed in a list.
- When editing the Priority field on a Change Request list, the Risk field needs to be updated to reflect the new priority.
- When editing a task list, make sure the Due Date isn't set earlier than the Start Date.

USE CASES

OnSubmit

- If the same user and its delegate user record exist in the table, then do not submit the record using Script Include.
- If the Caller is VIP then do not submit the record.
- Put Validation to make sure users on a Incident form do not enter a due date that is earlier than the start date.

Catalog Client Script

Other than Client Script we have Catalog Client Script also in ServiceNow.

Catalog client scripts are written for catalog items or variables sets and Client Scripts are written for backend tables e.g. incident, problem etc.

Catalog client script have 3 types only -

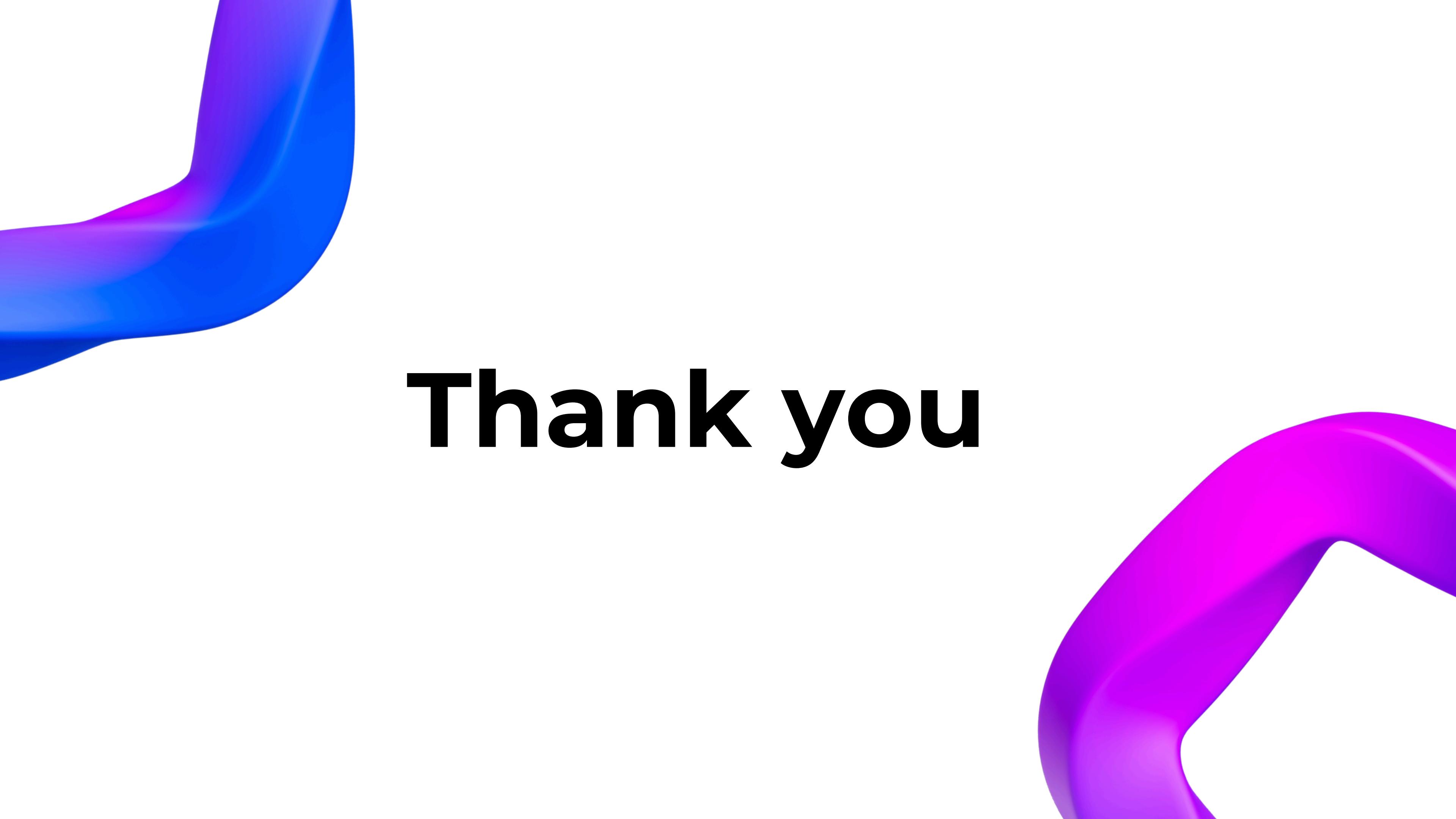
- OnLoad
- OnChange
- OnSubmit

Note: In OnChange we don't have isTemplate parameter there.

UI Policy vs Client Script

- Difference between UI Policy and Client Script is that UI Policy can access the Server Side data which is not visible to us on the form while Client Script can not fetch the data which is not visible on the form.
- To access data in client script from Server side we have to use Script Include, Display BR or getReference method.

Note: According to Execution Order Client Script always runs first after that UI Policy runs. Therefore, if both clash, we obtain the outcome of the UI Policy as it will override Client Script.



Thank you