**NETSUITE SCRIPT TYPES ASSIGNMENT**

**1. Differentiate between Client side and Server-Side scripting. Mention at least 3 points for each with an example.**

**Client-Side Script:**

1. It runs in the user’s browser or in the client side or in client’s machine.
2. It includes functionalities like field validation, form customization and field manipulation.
3. It appears in the user’s UI and the user can directly interact with each and every field.
4. It has entry points like -
   * + fileChanged ■ validateInsert
     + pageInit ■ validateLine
     + saveRecord ■ validateDelete
     + lineInit ■ sublistChanged
     + validateField ■ postSourcing
5. It doesn’t manipulate anything in the database.

**Scenario:**

In a sales order form, the generation of country name field will be automatically generated according to the customer’s current city name, without the user having to do it manually which improves the user experience.

**Server-Side Script:**

1. It runs or executes on the NetSuite server.
2. It has functionalities such as create, read, delete.
3. Here user can’t directly access it as it processes data and logic in the server side in background.
4. It has entry points like -
   * + beforeLoad
     + beforeSubmit
     + afterSubmit
5. It manipulates the database.

**Scenario:**

Lets suppose a sales order is fulfilled, then a user event script will automatically trigger and create an invoice corresponding to that particular order without any user interaction in the background.



**2. Give one use case for each of the following script types.**

Belows are the simple use case for each NetSuite script type: -

**i. Client Script:**

***Use case:*** Automatically filling the shipping address during creation of a sales order.

***Scenario:*** Let’s suppose a sales representative is creating a Sales Order for all the repeat customers. Instead of manually entering the shipping address for each individual customer every time, a client Script runs in the user's browser.

***How it Works:*** When the representative selects the customer on the form, the script will fetch the customer’s default shipping address from the database and automatically fills in the shipping address field. It reduces human error during feeling a form and reduces the time as well.

**ii. User Event Script:**

***Use case:*** Automatically generating an Invoice after an order fulfillment

***Scenario:*** After fulfilled a sales order, the invoice should automatically generate.

***How it Works:*** When the status of the Sales Order changes to "Fulfilled", a User Event Script triggers on the server. The script generates an Invoice just after the order fulfillment which links to that particular sales order.

**iii. Suitelet Script:**

***Use Case:*** Creating a service request form for the customers

***Scenario:*** Let a company offer services and maintenance for its products. So, customers need an easy way to submit service requests from the company’s website whenever they face any issue with any product of that company.

***How it Works:*** The company created a Suitelet Script that helps to create custom forms for customers to fill out when they have an issue with a product. This form collects necessary details such as product id, purchase date, product invoice, product issues and some basic customer info to proceed further.

**iv. Scheduled Script:**

***Use Case:*** Sending automated payment reminders to the customers

***Scenario:*** The finance team of an e-commerce company wants to automatically remind their customers about overdue invoices.

***How it Works:*** A Scheduled Script is set to run every night which does not hamper the peak time of the finance team. It checks all outstanding invoices in the system, identifies which ones are overdue, and sends automated email reminders to those customers.

**v. Map/Reduce Script:**

***Use Case:*** Bulk updation of all the items after certain period of time.

***Scenario:*** At the start of every quarter, the business needs to update the pricing for thousands of items in its inventory based on market conditions or inflation rate.

***How it Works:*** A Map/Reduce Script is created to handle this large-scale data processing. The script retrieves all items that need to be updated, breaks them down into smaller batches to do execute parallelly, which helps to operate this situation smoothly with less time.

**vi. RESTlet Script:**

***Use Case***: Integrating an E-Commerce Platform with NetSuite

***Scenario:*** Let a company run an online store on Amazon, and whenever a customer places an order, they want the order to automatically appear in NetSuite as a sales order.

***How it Works:*** A RESTlet Script triggers whenever an order is placed by any customer on Amazon, it automatically triggers an API call to this RESTlet, which receives the order details such as customer info, product id, prices, shipping address and creates a Sales Order in NetSuite.

**vii. Portlet Script:**

***Use Case:*** Displaying important information on the dashboard

***Scenario:*** Let a person having Administrator role want to see a live update of all the Sales Order generated in a particular year.

***How it Works:*** A Portlet Script is used to display a List Portlet on the dashboard. The script pulls live data from NetSuite Sales Order list, showing the showing the details of all orders for the selected year. This helps the administrator to prioritize the interactions with their most valuable clients.

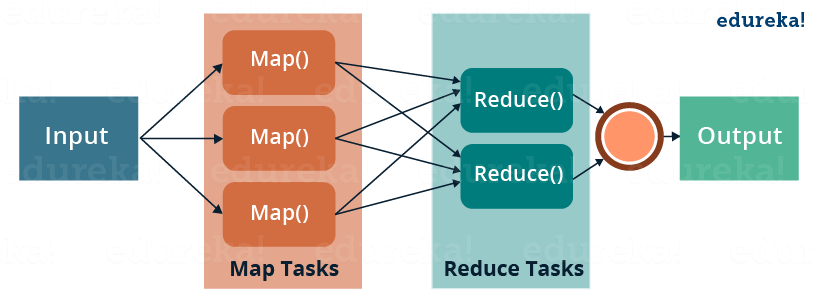
**3.Differentiate Map-Reduce and Scheduled script. Mention at least 5 points.**

**Scheduled Script:**

1. It is used for basic time-based tasks such as running a particular script in some time intervals.
2. It processes all the data at once without breaking it into chunks.
3. It is not ideal for a bulk amount of dataset or a huge dataset.
4. It’s a single-phase execution.
5. It is less efficient than Map reduce script.

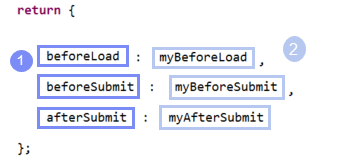
**Map-Reduce Script:**

1. It is used for handling large datasets or a bulk number of datasets.
2. It processes all the data by breaking it further into smaller parts or multiple batches using map.
3. It is perfect for a bulk amount of dataset or a huge dataset to run efficiently.
4. It’s a multi-phase execution that includes map, reduce, summarize.
5. It is more scalable and optimized.



**4. Name the entry points of Userevent Script.**

* **beforeLoad** - This script runs before the record is loaded either in the UI of NetSuite or even when calling record.load in another SuiteScript file.
* **beforeSubmit** - This script runs before the record is submitted. As soon as the user presses the save button, the script runs. If during beforeSubmit any error occur, then it won't be saved.
* **afterSubmit** - This entry point runs after the record is submitted. If any error occurs during the deploying time, unlike beforeSubmit it saves the record.



**5. Name the entry points of Client Script.**

1. ***fileChanged*** – It triggered when a user changes any file or record.
2. ***pageInit*** – It runs when a page is loaded for the first time.
3. ***saveRecord*** – It executes when the user saves a record.
4. ***lineInit*** – It runs when the user selects a line in a sublist.
5. ***validateField*** – It runs when the user changes a field value.
6. ***validateInsert*** – It executes before a new line is added to a sublist. It checks whether correct data is added or not.
7. ***validateLine*** – It runs when we save a line in a sublist.
8. ***validateDelete*** – It triggers before the user deletes a line.
9. ***sublistChanged*** – It runs when the user changes a sublist.
10. ***postSourcing*** – It executes when a field that sources information from another field is accessed.

**6. What are the different types of portlet scripts?**

In NetSuite, portlet scripts are used to create custom dashboards or to display dynamic content in portlet. There are four types of portlet script. i.e. -

1. ***Simple form*** – It displays a basic form with some fields, where users can input data.

Example – A simple form generated by a service provider, which provides service to their customers.In that form the customer can add data such as product id, issue facing, customer details etc.

1. ***Inline HTML*** – It displays custom HTML content such as images, flash and custom HTML.
2. ***Links and Indents –*** It helps to format and organize the contents within the portlet's user interface.
3. ***Simple Lists*** – It displays data in a simple lists or in tabular format. It also includes rows and headers.

Example – A list of top sales orders and some recent transactions.

**7. Write down the governance for map reduce script and stages as well.**

* A map reduce script are designed to handle large datasets efficiently.
* A governance refers to the number of units used by a script when it runs.
* In case of map reduce script, it consumes **10000 units.**

There are four stages in map reduce script.

i.e. -

1. **Get Input Data Stage** – This stage is responsible for providing the input data to be processed further. It will gradually proceed into map stage.
2. **Map Stage** – This stage processes each data independently. Each input data is processed to a map function. If the process takes much time to process, the more governance unit it will consume.
3. **Reduce Stage** – In case where multiple results from the map need to be combined then reduce stage help to do so as it reduces the results into a bundled structure. It groups the output from the map stage by key and process together.
4. **Shuffle Stage** – This is an optional stage that allows NetSuite to reorder the data which we get as an output from the reduce stage. It is automatically managed by NetSuite.

**8.Name of the scripts that contains a single-entry point in it.**

There are a few scripts that have single entry points, such as:

1. ***SuiteLet Script*** – It has a single-entry point. i.e. onRequest and it has two functions such as
   1. **GET request**
   2. **POST request**
2. ***Scheduled Script*** - It has a single-entry point. i.e. onExecute.
3. ***Portlet Script*** – It has only one entry point. I.e. render.