

# T estpassport Q&A



---

*La meilleure qualité le meilleur service*

<http://www.testpassport.fr>

Service de mise à jour gratuit pendant un an

**Exam** : **GCX-SCR**

**Title** : Genesys Cloud CX:  
Scripting Certification

**Version** : DEMO

1.The \_\_\_\_\_ tab in the right pane lists the errors in the script and helps you to locate them.

- A. Validation
- B. Actions
- C. Debug
- D. Container

**Answer: A**

**Explanation:**

In Genesys Cloud CX Scripting, the Validation tab in the right pane is crucial as it lists all the errors in the script and helps you locate them quickly. This feature is part of the integrated development environment (IDE) within Genesys Cloud CX, which aids developers in identifying and resolving issues in their scripts effectively. The validation process checks for various syntax errors, configuration mismatches, and other potential issues that might cause the script to fail during execution. This helps ensure that scripts are error-free and function as intended before they are deployed.

The Validation tab is an essential tool in the Genesys Cloud CX scripting environment, providing a streamlined way to detect and correct errors, thereby improving the accuracy and reliability of customer experience automation.

Reference: Genesys Cloud CX Scripting Documentation.

2.You can modify the data type of a variable once it has been created.

- A. True
- B. False

**Answer: B**

**Explanation:**

In Genesys Cloud CX Scripting, once a variable has been created and assigned a data type, you cannot modify its data type later. This design is intentional to prevent errors that might occur if a variable's type were changed after it was already in use. Variables in Genesys Cloud CX scripting are strongly typed, meaning the type of data they hold is fixed upon their creation.

To change the type of a variable, you would need to create a new variable with the desired type and then use that instead. This ensures that the script's logic remains consistent and avoids runtime errors that could arise from unexpected data types.

This behavior is consistent with best practices in many scripting environments where strict typing is enforced to maintain stability and predictability in scripts.

For more details on variable handling in Genesys Cloud CX, you can refer to Genesys Cloud CX Scripting documentation that explicitly states these constraints to help developers write more reliable and error-free scripts.

Reference: [Genesys Cloud CX Scripting Documentation]

3.Which of the following statements about templates in scripting are true? (Choose two.)

- A. Script templates are entire scripts that have been packaged for reuse.
- B. Use a component template when you create a new script.
- C. A script template contains a single saved component such as text, or a layout of components such as a horizontal stack container.
- D. Component templates are parts of a page that you can reuse in other scripts.

**Answer: A, D**

**Explanation:**

In Genesys Cloud CX Scripting, templates play a vital role in creating reusable elements for various scripts, streamlining the development process.

**Script Templates:** These are entire scripts that have been packaged for reuse. They serve as a starting point for new scripts, allowing users to replicate existing scripts quickly without having to rebuild them from scratch. This is particularly useful for creating standardized scripts across different campaigns or use cases.

**Component Templates:** These refer to parts of a page that can be reused in other scripts. Component templates are particularly helpful when you need to reuse specific UI elements or layouts across multiple scripts. For instance, if a script has a specific layout or a component, such as a customer information panel, that needs to appear in multiple scripts, you can create a component template for it. This ensures consistency and saves development time.

Reference: Genesys Cloud CX Scripting Documentation.

4. Richard is assigned the task of creating custom Outbound Campaign scripts. He wants to display the current value of the customer's first name on the script page. Help him by choosing the correct syntax.

- A. {{(Outbound.First Name)}}
- B. ((Outbound.First Name))
- C. Outbound.First Name
- D. {{Outbound.First Name}}

**Answer: D**

**Explanation:**

In Genesys Cloud CX, when creating custom scripts, the correct syntax to display dynamic data, such as a customer's first name, involves using double curly braces `{{ }}`. This syntax is used to reference data variables within the script.

`{{Outbound.First Name}}`: This syntax correctly refers to the "First Name" attribute from the Outbound object in the script. When the script is run, this placeholder will dynamically display the customer's first name based on the data provided in the Outbound Campaign.

Using the correct syntax is crucial for ensuring that the data is pulled correctly and displayed as intended in the script.

Reference: Genesys Cloud CX Scripting Documentation.

5. Using the \_\_\_ action in the call flow, you can transfer data to Scripts.

- A. Get Participant data
- B. Call data action
- C. Set Participant data
- D. Update data

**Answer: C**

**Explanation:**

In Genesys Cloud CX, the Set Participant Data action is the correct choice for transferring data to scripts during a call flow. This action allows you to set specific data attributes for the participant (such as a customer or agent) that can then be accessed by scripts running during the interaction.

The Set Participant Data action is typically used to pass information gathered during an interaction, such as caller input or data fetched from external sources, so that it can be used in scripts to customize the

experience further. This is crucial for dynamic and personalized script content based on the interaction context.

For more details, you can refer to the Genesys Cloud CX documentation on call flows and scripting, which explains how to use the Set Participant Data action effectively in various scenarios.

Reference: [Genesys Cloud CX Call Flow Documentation]

[Genesys Cloud CX Scripting Documentation]